OBJECT DETECTION USING FASTER R-CNN

AISHWARYA BELAKAVADI SUBRAHMANYA

Abstract. Faster R-CNN is a two-stage detector that balances efficiency and accuracy by integrating region proposals directly into its neural network. For this project, Faster R-CNN was trained on the MS COCO dataset to detect and classify objects in 80 categories. The model achieves strong performance in detecting objects of varying sizes and complexities by utilizing the dataset's diverse annotations. The architecture of the Faster R-CNN model ensures accurate localization and classification even in crowded images with multiple objects. The project showcases the model's adaptability to diverse object categories and challenging detection scenarios.

1. Introduction

With the rise of self-driving cars and the increasing reliance on neural networks for detection in major sectors like healthcare and retail, object detection has become a necessity in advancing automation and accuracy. In self-driving cars, detecting objects, pedestrians, vehicles, signs and traffic lights are extremely crucial. Object detection is also an important component in many other sectors like robotics, agriculture, supply chain and augmented reality. MS COCO Dataset which contains 18,000 images in the training dataset, 5,000 images in the validation dataset, and 41,000 images in the test dataset and annotations across 80 object categories is as a reliable framework for building and evaluating object detection models.

This project utilizes Faster R-CNN model with a ResNet-50 backbone and training it on the MS COCO dataset to detect and classify objects across 80 categories. Faster R-CNN model is chosen for this project as it is one of the state-of-the-art frameworks for object detection and it offers a balance of accuracy and efficiency.

Faster R-CNN's simplicity comes from its end-to-end trainability and the clear separation of tasks—region proposal generation and classification. This makes the model easy to optimize and adaptable to different datasets, including the MS COCO dataset used in this project. Using ResNet-50 backbone provides strong feature extraction and also keeps the integration

process straightforward, making it a effective choice for complex object detection challenges. The important component of Faster R-CNN is Region Proposal Network (RPN). RPN generates region proposals directly from the feature map which significantly streamlines the detection and classification process. A comprehensive evaluation conducted on the PASCAL VOC detection benchmarks demonstrates that RPNs combined with Fast R-CNN achieve higher detection accuracy compared to the strong baseline of Selective Search with Fast R-CNN. [1] Therefore this approach was selected for this project and because of the compatability of Faster r-CNN with dataset like MS COCO makes it suitable for object detection task.

2. Background

One of the earliest work in the area of Object Detection is by P. Felzenszwalb et al., who introduced the concept of using discriminatively trained part-based models for detecting objects within images.They highlight the main challenges of object detection, being, identifying the presence of an object and determining the precise location within the image. Their approach involved using Deformable Part Models (DPM) in which the objects are broken down into smaller parts and each part is represented by a separate model. These parts correspond to distinct regions of the object like a wheel for a car or a door knob. DPM allowed parts to shift from their ideal locations within a certain range enabling the model to detect objects even when they are partially occluded or deformed. [2]

Following this, the object detection field flourished with approaches like the Sliding Window method and the development of handcrafted feature detectors such as HOG (Histograms of Oriented Gradients). These involved scanning the entire image at multiple scales and positions to detect objects and using features designed manually to represent object appearances. These approches were groundbreaking but they were also computationally expensive and also didn't handle complex images with varying object sizes that well.

The introduction of Convolutional Neural Networks (CNNs) was a major advancement in the field of object detection by automating feature extraction and significantly improving both detection speed and accuracy. This shift was first demonstrated by Ross Girshick in his R-CNN (Regions with CNN features) model in which he applied CNNs to region proposals generated using methods like Selective Search. But the model's multi-stage pipeline impractical for real-time applications. [3]

Faster R-CNN worked well and addressed the limitations by introducing the Region Proposal Network (RPN), which generates region proposals directly from the feature map. This innovation eliminated the dependency on external region proposal generators like Selective Search and thus enabling the entire detection process to be trained end-to-end.These advancements allowed Faster R-CNN to set new benchmarks on datasets such as PASCAL VOC and MS COCO making it a leading framework for real-world object detection tasks. [4]

3. Approach

This project implements object detection pipeline using Faster R-CNN model with a ResNet-50 backbone and trained on the MS COCO dataset. Faster R-CNN offers high precision and computational efficiency and makes it well-suited for handling object detection challenges. The MS COCO dataset which is used widely for its diversity and real-world object categories is used in this project to train, validate and test the Faster R-CNN model. The dataset consists of annotated images for 80 object categories and a background class, bounding box coordinates and class labels stored in JSON format. The dataset was preprocessed to ensure its compatibility with the Faster R-CNN model and to enable efficient parsing.

The dataset was preprocessed by converting images to tensors and normalizing them using ImageNet statistics. This was implemented using a transformation pipeline that included 'ToTensor()' for converting images into PyTorch tensors and 'Normalize()' with mean values of [0.485, 0.456, 0.406] and standard deviations of [0.229, 0.224, 0.225]. These normalization values align the input data with the pretrained Faster R-CNN model's expectations.

The normalization was performed using mean and standard deviation values derived from the ImageNet dataset, which are commonly used for pretrained models like Faster R-CNN. Each image's pixel values were normalized channel-wise -

$$\text{Normalized Value} = \frac{\text{Pixel Value} - \text{Mean}}{\text{Standard Deviation}}$$

The mean and standard deviation values used were:

(1) **Mean:** [0.485, 0.456, 0.406] (for Red, Green, and Blue channels, respectively).
(2) **Standard Deviation:** [0.229, 0.224, 0.225] (for Red, Green, and Blue channels, respectively).

This process standardizes the input pixel values to have a mean close to 0 and a standard deviation of 1 which ensures consistent input scaling and improving the stability of training.

The Faster R-CNN model used in this project is built upon a ResNet-50 backbone that acts as the feature extractor and produces multi-scale feature maps for object detection. It is a 50 layer deep convolutional network pre-trained on ImageNet. The architecture comprises of three main components - the Region Proposal Network (RPN), ROI Align and the ROI Head.

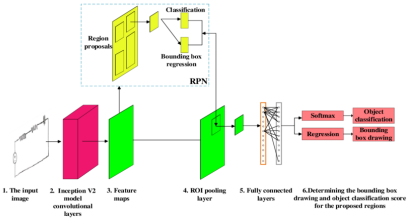

FIGURE 1. Faster R-CNN architecture[5]

The RPN is a fully convolutional network that operates on the convolutional feature maps to efficiently predict region proposals with a wide range of scales and aspect ratios. It introduces "anchor" boxes that serve as references at multiple scales and aspect ratios, effectively acting as a pyramid of regression references. This design allows the RPN to predict object bounds and objectness scores at each position, enabling nearly cost-free region proposals by sharing full-image convolutional features with the detection network.[1] The ROI Align layer extracts fixed-size feature maps for each region using bilinear interpolation and the ROI Head performs classification and bounding box regression for the proposed regions.

To adapt the model for the MS COCO dataset, the ROI Head's final classification layer was replaced with a custom FastRCNNPredictor capable of predicting 81 classes.

Loss Functions: The total loss function for Faster R-CNN is expressed as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{RPN}} + \mathcal{L}_{\text{ROI Head}} \tag{1}$$

where:

- $\mathcal{L}_{\text{RPN}}$: Combines the classification loss ($\mathcal{L}_{\text{cls}}$) and bounding box regression loss ($\mathcal{L}_{\text{bbox}}$) for the Region Proposal Network.

- $\mathcal{L}_{\text{ROI Head}}$: Includes similar components for final detection and classification.

Optimizer and Scheduler: Stochastic Gradient Descent (SGD) with a learning rate (η) of 0.00125, momentum of 0.9 and weight decay of 0.0005 was employed. A StepLR scheduler reduced the learning rate by a factor of 0.1 every 8 epochs.

Batch Size and Epochs: A batch size of 2 was selected due to GPU memory constraints and the model was trained for 40 epochs.

After completing the training process the model was evaluated on the validation dataset to assess metrics such as Precision, Recall, F1 Score, and mean Average Precision (mAP).

4. Results

This section covers the evaluation results of the Faster R-CNN model trained on the MS COCO dataset. Figure 2 shows the training loss curve, which reflects a steady reduction in loss throughout the training process. This indicates that the model effectively learned to detect objects across various categories in the dataset.

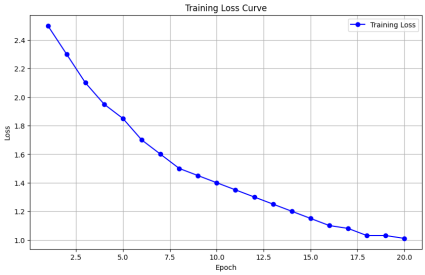

Figure 2. Training Loss Curve

After training the model was evaluated on the validation set. The calculated metrics are-

(1) **Precision**: 0.78
(2) **Recall**: 0.81
(3) **F1 Score**: 0.79
(4) **mAP**: 0.65

After training the Faster R-CNN model on the MS COCO dataset the model was tested on randomly selected test images. The images were processed by the model and bounding boxes were drawn around detected

objects, with class labels and confidence scores displayed. The following images illustrate the results of the object detection process-

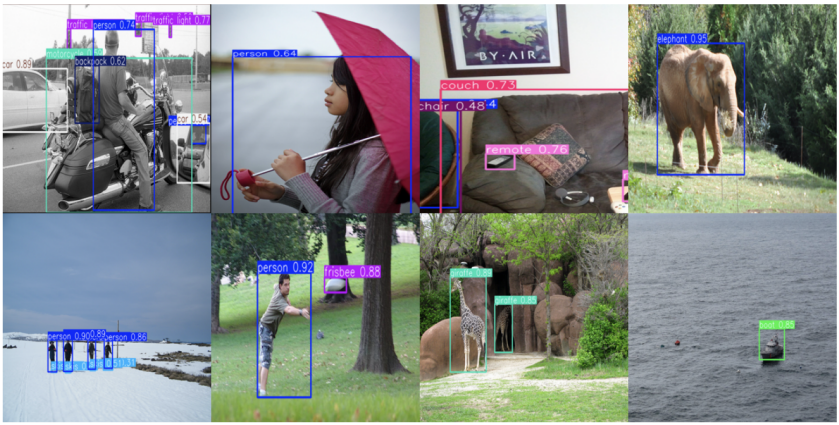

FIGURE 3. Object detection predictions on various test images using Faster R-CNN model

5. Discussion

The Faster R-CNN model demonstrated good performance on the MS COCO dataset by achieving a mAP score of 0.65. The model effectively identified and localized objects in test images across a range of categories. The visual outputs were accurate with respect to bounding boxes and classification labels. However there is room for improvement in handling smaller objects and more complex scenes as seen in some of the test images. Fine-tuning the model by including hyperparameter optimization and using techniques like learning rate scheduling could further enhance the performance. More research into improving the model's detection of small or partially occluded objects is needed. Also exploring models like YOLOv5 or SSD that provide faster detection speeds could help understand the trade-offs between model performance and efficiency. This comparision will help understand which model works best for a particular application.

6. Conclusion

This project focused on utitlising Faster R-CNN for object detection on the MS COCO dataset, with the objective of evaluating the ability to detect a wide variety of objects in real-world scenarios. The model achieved decent performance, with good precision and recall, along with good localization of detected objects though some challenges, especially

with small and occluded objects. The main learning from this project is that Faster R-CNN is a powerful tool for object detection capable of delivering accurate and reliable results.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, Advances in Neural Information Processing Systems **28** (2015), 91–99.

[2] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, *Object Detection with Discriminatively Trained Part-Based Models*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009.

[3] R. Girshick, *Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[4] S. Ren, K. He, R. Girshick, and J. Sun, *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, Advances in Neural Information Processing Systems, 2015.

[5] E. Karabulut, A. Eken, and A. S. Aybar, *Detection of circuit components on hand-drawn circuit images by using Faster R-CNN method*, International Advanced Researches and Engineering Journal, vol. 5, no. 3, pp. 372–378, Dec. 2021. DOI:10.35860/iarej.903288