A Detailed Comparative Study on Object Detection Using CNN-Based Faster R-CNN and YOLO

Bhushan Zade
Department of Scientific Computing
Savitribai Phule Pune University
Pune, India
bhushanzade02@gmail.com

***Abstract*—Object detection is a fundamental and challenging problem in computer vision that involves identifying object categories and precisely localizing them within images. With the rapid advancement of deep learning, Convolutional Neural Networks (CNNs) have emerged as the dominant approach for object detection tasks. Despite achieving high accuracy, many CNN-based detectors suffer from high computational complexity, making real-time deployment difficult. This paper presents an in-depth comparative study of a two-stage CNN-based detector, Faster R-CNN, and a single-stage detector, YOLOv5. Both models are trained and evaluated on the Pascal VOC 2007 dataset using a unified experimental setup. Performance is analyzed using standard metrics including Precision, Recall, Mean Average Precision (mAP@0.5 and mAP@0.5:0.95), inference time, and Frames Per Second (FPS). Experimental results demonstrate that Faster R-CNN provides strong localization accuracy, while YOLOv5 achieves competitive accuracy with significantly faster inference speed. The study highlights the inherent trade-off between accuracy and real-time performance and provides practical insights for selecting object detection models based on application requirements.**

***Index Terms*—Object Detection, Convolutional Neural Networks, Faster R-CNN, YOLOv5, Deep Learning, Computer Vision**

I. Introduction

Object detection is a core task in computer vision that has gained significant attention due to its importance in real-world applications such as autonomous vehicles, video surveillance, robotics, medical imaging, and intelligent transportation systems. Unlike image classification, which assigns a single label to an entire image, object detection requires identifying multiple objects and accurately localizing each object using bounding boxes. This dual requirement significantly increases the complexity of the problem.

Early object detection methods relied on handcrafted feature extraction techniques such as Haar-like features, Scale-Invariant Feature Transform (SIFT), and Histogram of Oriented Gradients (HOG), combined with classical machine learning classifiers like Support Vector Machines (SVMs). While these methods achieved moderate success in controlled environments, they lacked robustness when dealing with complex backgrounds, varying object scales, occlusions, and illumination changes.

The introduction of deep learning, particularly Convolutional Neural Networks (CNNs), revolutionized object detection by enabling automatic hierarchical feature learning directly from raw image data. CNN-based approaches significantly outperformed traditional methods and became the foundation of modern object detection systems. Region-based CNN architectures such as R-CNN and its variants demonstrated remarkable improvements in detection accuracy by combining region proposals with deep feature extraction.

However, high accuracy often comes at the cost of increased computational complexity. Two-stage detectors such as Faster R-CNN involve multiple processing steps, resulting in higher inference latency. To overcome this limitation, single-stage detectors such as YOLO were proposed, reformulating object detection as a regression problem and enabling real-time performance.

This paper presents a comprehensive experimental comparison between Faster R-CNN and YOLOv5, focusing on both accuracy and computational efficiency.

The main contributions of this paper are as follows:

- A detailed implementation of Faster R-CNN and YOLOv5 using a standard benchmark dataset.
- Extensive evaluation using accuracy-based and speed-based performance metrics.
- Experimental validation of the accuracy–speed trade-off in object detection systems.

II. Related Work

Object detection research has evolved significantly over the past two decades. Early approaches relied on sliding window techniques combined with handcrafted features. The Viola–Jones framework was one of the first real-time object detectors, primarily used for face detection. Despite its efficiency, it was limited to specific object categories.

Dalal and Triggs introduced the HOG descriptor, which improved pedestrian detection by capturing local gradient orientation patterns. Although effective, HOG-based methods required manual feature engineering and struggled with scalability.

The emergence of deep learning introduced CNN-based detectors. Girshick et al. proposed R-CNN, which applied CNNs to region proposals generated using selective search. While R-CNN achieved high accuracy, it required multiple forward passes per image, making it computationally expensive. Fast R-CNN improved efficiency by sharing convolutional computations across region proposals but still relied on selective search.

Faster R-CNN eliminated selective search by introducing the Region Proposal Network (RPN), enabling end-to-end training and faster region generation. Despite its accuracy, Faster R-CNN remains computationally intensive.

Single-stage detectors such as SSD and YOLO were introduced to address speed limitations. YOLO proposed a unified detection framework that predicts bounding boxes and class probabilities in a single forward pass. YOLOv5 further improved detection performance through architectural optimizations, anchor box refinement, and efficient training strategies.

Existing studies consistently report a trade-off between detection accuracy and inference speed, motivating the detailed comparative analysis conducted in this work.

III. Dataset and Experimental Setup

A. Pascal VOC 2007 Dataset

The Pascal VOC 2007 dataset is a widely used benchmark for object detection research. It consists of 9,963 images annotated with bounding boxes across 20 object categories, including person, vehicle, animal, and household objects. Each image may contain multiple objects of different classes and sizes, making the dataset suitable for evaluating object detection algorithms.

B. Implementation Environment

All experiments were conducted using the PyTorch deep learning framework. Faster R-CNN was implemented using the `torchvision` detection module, while YOLOv5 was implemented using the Ultralytics YOLOv5 repository. Training and evaluation were performed on a GPU-enabled environment using an NVIDIA Tesla T4 GPU to ensure efficient computation.

IV. Methodology

A. Faster R-CNN Architecture

Faster R-CNN is a two-stage object detection framework consisting of a backbone CNN for feature extraction, a Region Proposal Network for generating candidate object regions, and detection heads for classification and bounding box regression. The RPN predicts objectness scores and bounding box offsets using anchor boxes of multiple scales and aspect ratios.

B. YOLOv5 Architecture

YOLOv5 is a single-stage detector that processes the entire image in a single forward pass. The architecture consists of a CSPDarknet backbone, a PANet neck for feature aggregation, and multi-scale detection heads. This design enables efficient detection of objects at different scales while maintaining real-time performance.

C. Data Preprocessing

For Faster R-CNN, the Pascal VOC annotations were used directly in XML format. For YOLOv5, annotations were converted into YOLO format, where bounding box coordinates are normalized relative to image dimensions.

D. Training Strategy

YOLOv5 was trained for 30 epochs using pre-trained weights to accelerate convergence. Faster R-CNN was trained using transfer learning with a ResNet backbone. Both models were trained under similar conditions to ensure fair comparison.

V. Experimental Results and Evaluation

A. Evaluation Metrics

The performance of both models was evaluated using Precision, Recall, Mean Average Precision (mAP@0.5 and mAP@0.5:0.95), inference time, and Frames Per Second (FPS).

B. YOLOv5 Performance

YOLOv5 achieved a precision of 0.768, recall of 0.648, mAP@0.5 of 0.717, and mAP@0.5:0.95 of 0.453. The average inference time was 0.045 seconds per image, corresponding to 22.15 FPS.

C. Faster R-CNN Performance

Faster R-CNN demonstrated strong localization accuracy but slower inference, with an average inference time of 0.136 seconds per image and a speed of 7.35 FPS.

VI. Comparative Analysis

The experimental results clearly demonstrate the trade-off between accuracy and speed. Faster R-CNN prioritizes localization accuracy through its two-stage architecture, while YOLOv5 emphasizes real-time performance with competitive accuracy.

VII. Conclusion and Future Work

This paper presented a detailed comparative study of Faster R-CNN and YOLOv5 for object detection using the Pascal VOC 2007 dataset. Experimental results show that YOLOv5 achieves real-time detection with competitive accuracy, whereas Faster R-CNN offers higher localization accuracy at the cost of inference speed. Future work includes exploring newer YOLO variants, training on larger datasets, and deploying models on edge devices.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *NeurIPS*, 2012.

[2] R. Girshick et al., “Region-Based Convolutional Networks for Accurate Object Detection,” *IEEE TPAMI*, 2016.

[3] J. Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection,” in *CVPR*, 2016.

[4] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” in *CVPR*, 2017.