ITEJ December-2023, Volume 8 Nomor 2 Page 96 - 106

ITEJ
Information Technology Engineering Journals
eISSN : 2548-2157

Url : https://syekhnurjati.ac.id/journal/index.php/itej
Email : itej@syekhnurjati.ac.id

A Comparative Analysis of Modern Object Detection Algorithms: YOLO vs. SSD vs. Faster R-CNN

Dalmar Dakari Aboyomi
Computer Science
University of Lagos, Nigeria
dalmard@unilag.edu.ng

Cleo Daniel
Computer Science
University of Lagos, Nigeria
daniel@unilag.edu.ng

Abstract–In recent years, object detection has become a crucial component in various computer vision applications, including autonomous driving, surveillance, and image recognition. This study provides a comprehensive comparative analysis of three prominent object detection algorithms: You Only Look Once (YOLO), Single Shot MultiBox Detector (SSD), and Faster Region-Based Convolutional Neural Networks (Faster R-CNN). The background of this research lies in the growing need for efficient and accurate object detection methods that can operate in real-time. YOLO is known for its speed, SSD for its balance between speed and accuracy, and Faster R-CNN for its high detection accuracy, albeit at a slower pace. The methodology involves implementing these algorithms on a standardized dataset and evaluating their performance based on various metrics, including detection accuracy, processing speed, and computational resource requirements. Each algorithm is tested under similar conditions to ensure a fair comparison. The results indicate that while YOLO excels in real-time applications due to its high speed, SSD offers a middle ground with respectable accuracy and speed, making it suitable for applications requiring a balance of both. Faster R-CNN demonstrates superior accuracy, making it ideal for scenarios where detection precision is paramount, despite its slower performance. This comparative analysis highlights the strengths and weaknesses of each algorithm, providing valuable insights for researchers and practitioners in selecting the appropriate object detection method for their specific needs.

Keywords: Object Detection, YOLO, SSD, Faster R-CNN

INTRODUCTION

In the era of rapidly advancing technology, the field of computer vision has seen significant growth and development, particularly in the realm of object detection[1][2]. Object detection, a critical aspect of computer vision, involves the identification and localization of objects within an image or video[3]. This capability has far-reaching implications across various industries, from autonomous vehicles and surveillance systems to medical imaging and augmented reality[4].

The evolution of deep learning techniques[5][6] has propelled object detection to new heights, enabling more accurate and efficient detection mechanisms. Among the myriad of object detection algorithms

developed, three have emerged as frontrunners due to their widespread adoption and performance: You Only Look Once (YOLO)[7][8][9][10], Single Shot MultiBox Detector (SSD)[9], and Faster Region-Based Convolutional Neural Networks (Faster R-CNN)[7][11][12][13]. Each of these algorithms presents a unique approach to object detection, with distinct advantages and trade-offs.

YOLO is renowned for its impressive speed and real-time detection capabilities, making it a popular choice for applications that require instantaneous response. SSD offers a compelling balance between speed and accuracy, providing reliable performance across a variety of tasks. Faster R-CNN, on the other hand, is acclaimed for its high detection accuracy, making it suitable for applications where precision is critical, despite being slower compared to YOLO and SSD.

This paper aims to conduct a comprehensive comparative analysis of YOLO, SSD, and Faster R-CNN, evaluating their performance based on a range of metrics. By doing so, we seek to provide a nuanced understanding of each algorithm's strengths and weaknesses, guiding researchers and practitioners in selecting the most appropriate object detection method for their specific needs.

The following sections will delve into the methodology employed for this comparative study, the experimental setup, and the detailed results of our evaluation. Through this analysis, we aim to shed light on the practical implications of choosing one algorithm over another, contributing to the ongoing discourse in the field of computer vision.

RELATED WORKS

The field of object detection has witnessed significant advancements over the past decade, driven largely by the development of deep learning techniques. Early approaches, such as the Viola-Jones detector, laid the groundwork for subsequent innovations by introducing the concept of detecting objects using Haar-like features and a cascade of classifiers. However, these methods were limited by their reliance on handcrafted features and inability to handle the variability in object appearance and pose effectively.

The advent of convolutional neural networks (CNNs) marked a paradigm shift in object detection. R-CNN (Regions with Convolutional Neural Networks) was one of the pioneering works that leveraged CNNs for object detection by combining region proposals with deep feature extraction. Despite its high accuracy, R-CNN was computationally expensive due to its multi-stage pipeline. To address this, Fast R-CNN and later Faster R-CNN were introduced, significantly reducing detection time by integrating the region proposal network (RPN) directly into the CNN architecture. Faster R-CNN became a cornerstone for high-accuracy object detection, setting a new standard in the field.

Parallel to these developments, Single Shot MultiBox Detector (SSD) emerged as a groundbreaking algorithm that performed object detection in a single forward pass of the network, eliminating the need for a separate region proposal stage. SSD achieved a remarkable balance between speed and accuracy, making it suitable for real-time applications. Similarly, You Only Look Once (YOLO) revolutionized object detection by framing it as a single regression problem, predicting bounding boxes and class probabilities directly from full images. YOLO's impressive speed and simplicity have made it a popular choice for various applications requiring real-time performance.

Subsequent versions of these algorithms, such as YOLOv3, YOLOv4, and YOLOv5, as well as SSD with ResNet and MobileNet backbones, have continued to push the boundaries of performance. Numerous comparative studies have evaluated these algorithms under different conditions, revealing insights into their respective strengths and trade-offs. For instance, Huang et al. (2017) conducted a comprehensive

evaluation of several object detection models, highlighting the trade-offs between speed and accuracy across different algorithms.

In this study, we build upon these foundational works by conducting a detailed comparative analysis of YOLO, SSD, and Faster R-CNN. Our objective is to provide a nuanced understanding of their performance in various scenarios, contributing to the broader discourse on the applicability of object detection algorithms in diverse contexts. Through rigorous experimentation and analysis, we aim to offer valuable insights that can inform the selection of appropriate algorithms for specific application needs.

METHOD

In this study, we employ a systematic approach to compare the performance of three prominent object detection algorithms: You Only Look Once (YOLO), Single Shot MultiBox Detector (SSD), and Faster Region-Based Convolutional Neural Networks (Faster R-CNN). Our methodology is designed to ensure a fair and comprehensive evaluation, focusing on key performance metrics such as accuracy, speed, and computational efficiency.

1. Dataset Selection

   We use the Common Objects in Context (COCO) dataset for our experiments[10][14]. The COCO dataset is widely recognized for its diverse range of object categories and challenging scenarios, making it an ideal choice for evaluating object detection algorithms. The dataset is split into training, validation, and test sets, ensuring that each algorithm is evaluated on the same set of images.

2. Algorithm Implementation

For each algorithm, we use well-established implementations with pre-trained weights. Specifically, we use YOLOv4, SSD with MobileNet backbone, and Faster R-CNN with ResNet-50 backbone[15]. These implementations are selected based on their popularity and performance in the object detection community.

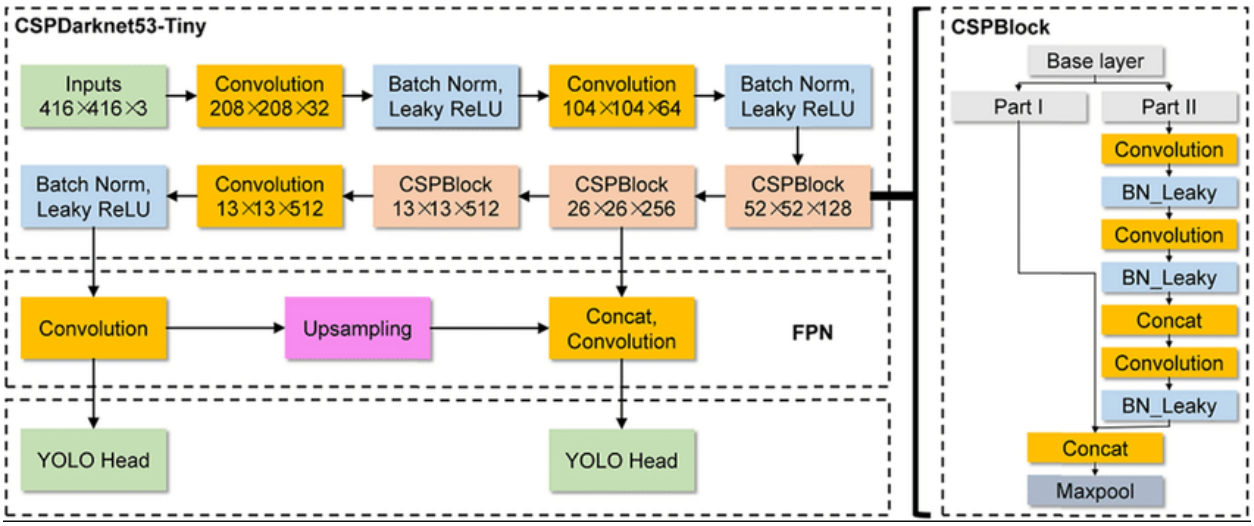

Figure 1: YOLOv4 Architechture

YOLOv4, short for "You Only Look Once version 4," represents the latest iteration of the YOLO series of object detection models. YOLOv4 is renowned for its efficiency and accuracy in real-time object detection tasks. The architecture of YOLOv4 builds upon the principles of its predecessors while incorporating several enhancements to achieve better performance.

1. Backbone Network: YOLOv4 utilizes a powerful backbone network for feature extraction. Typically, this backbone network is based on variants of the Darknet architecture, such as CSPDarknet53[16]. These networks are known for their ability to efficiently extract hierarchical features from input images, crucial for subsequent object detection tasks.

2. Neck and Feature Pyramid: YOLOv4 incorporates a neck module, specifically a Spatial Pyramid Pooling (SPP)[17] module or a Path Aggregation Network (PANet)[17], to enhance feature representation. These modules help in capturing multi-scale features from different levels of the feature hierarchy, enabling the model to detect objects of varying sizes effectively.

3. Detection Head: The detection head of YOLOv4 consists of multiple detection layers responsible for predicting bounding boxes, objectness scores, and class probabilities. This head is designed to output detections directly from the final feature maps, optimizing the model for speed and efficiency[18].

4. Improvements and Optimization: YOLOv4 introduces several optimizations to enhance performance:
   - Data Augmentation[9]: Enhanced data augmentation techniques during training help in improving model robustness and generalization.
   - Regularization Techniques: Incorporation of techniques like DropBlock regularization to prevent overfitting and improve model generalization.
   - Advanced Training Strategies: YOLOv4 benefits from advanced training strategies, such as cosine annealing scheduler for learning rate adjustment and focal loss to handle class imbalance, ensuring better convergence and stability during training.

5. Post-Processing: After predictions are made, YOLOv4 employs Non-Maximum Suppression (NMS)[19] to filter out redundant bounding boxes and retain only the most confident detections.

YOLOv4 represents a state-of-the-art object detection architecture that balances between accuracy and speed. Its efficient design and optimization strategies make it suitable for a wide range of applications, from real-time surveillance and autonomous driving to industrial automation and beyond.

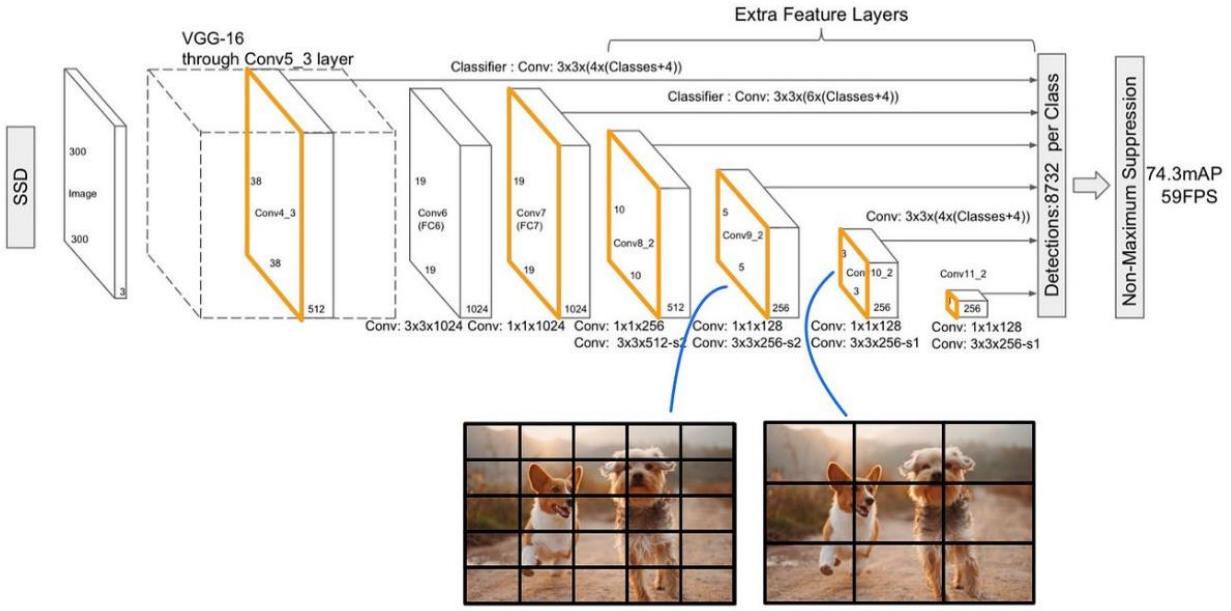

Figure 2: SSD Architechture[20]

SSD, or Single Shot MultiBox Detector, is a popular object detection framework known for its efficiency and effectiveness in real-time applications. The architecture of SSD is designed to perform object detection in a single forward pass of the network, making it particularly suitable for scenarios requiring high speed and responsiveness.

1. Feature Extraction Backbone: SSD begins with a base convolutional neural network (CNN), often using variants of VGGNet[21] or ResNet[19], to extract features from input images. These networks are pretrained on large datasets like ImageNet, enabling them to capture rich hierarchical features that are essential for object detection.

2. Multi-Scale Feature Maps: Unlike traditional approaches that predict objects at a single scale, SSD predicts objects at multiple scales using feature maps from different layers of the CNN. This multi-scale approach allows SSD to detect objects of varying sizes and aspect ratios within the same network architecture.

3. Prediction Head: SSD introduces a series of convolutional layers on top of each feature map to predict bounding boxes and class probabilities. Each convolutional layer is responsible for predicting a set of bounding boxes (multi-boxes) at different spatial locations within its corresponding feature map[11].

4. Default Boxes (Anchor Boxes): SSD uses anchor boxes or default boxes of different aspect ratios and scales at each feature map location. These anchor boxes serve as priors that guide the network to predict accurate bounding boxes regardless of the object's size and position in the image.

5. Loss Function[22]: During training, SSD optimizes its parameters using a combination of localization loss (e.g., Smooth L1 loss) and confidence loss (e.g., softmax loss or focal loss). The localization loss penalizes the network for inaccurate bounding box predictions, while the confidence loss ensures that the network accurately classifies objects and background.

6. Post-Processing: After predictions are made, SSD applies Non-Maximum Suppression (NMS) to filter out redundant bounding boxes and retain only the most confident detections.

SSD strikes a balance between speed and accuracy by leveraging multi-scale feature maps and anchor boxes to efficiently detect objects in real-time. Its ability to handle objects of various sizes and aspect ratios within a unified framework has made it a popular choice for applications such as autonomous driving, robotics, and video surveillance where real-time performance is crucial.

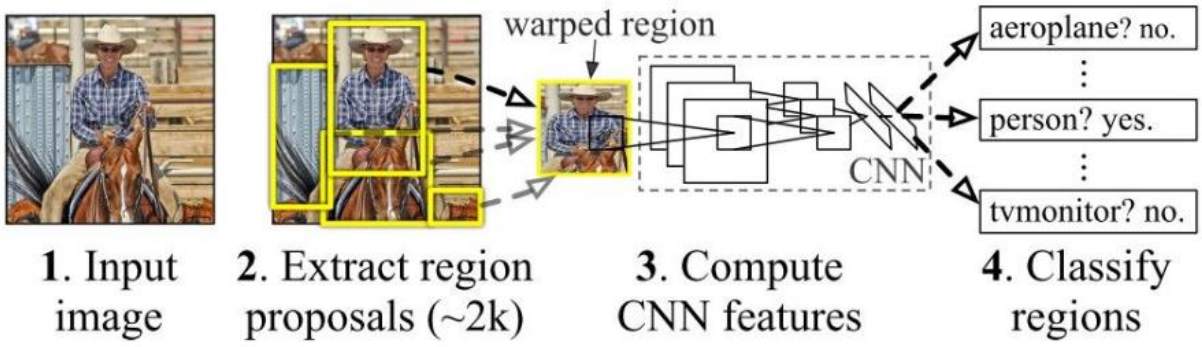

Figure 3: Faster R-CNN Architechture

Faster R-CNN represents a significant advancement in object detection by introducing a region proposal network (RPN) that efficiently generates region proposals within the same deep learning framework. The architecture of Faster R-CNN consists of several interconnected modules designed to enhance both accuracy and speed in object detection tasks.

1. Backbone Network: Faster R-CNN typically starts with a deep convolutional neural network (CNN), such as ResNet or VGGNet, pretrained on large-scale image classification tasks like ImageNet. This backbone network serves as a feature extractor that transforms input images into a series of feature maps with increasingly abstract representations.

2. Region Proposal Network (RPN): One of the key innovations of Faster R-CNN is the integration of an RPN, which operates on the feature maps generated by the backbone network. The RPN is responsible for generating region proposals, which are potential bounding boxes that may contain objects of interest. It achieves this by sliding a small network (typically a set of convolutional layers) over the feature maps to predict objectness scores and bounding box coordinates relative to anchor boxes of different scales and aspect ratios.

3. RoI Pooling and RoI Align: After generating region proposals, Faster R-CNN uses RoI (Region of Interest) pooling or RoI align to extract fixed-size feature maps from the feature maps produced by the backbone network. This step ensures that each region of interest is represented by a uniform-sized feature map, regardless of its scale or aspect ratio.

4. Detection Head: Once the RoI features are extracted, they are fed into a detection head consisting of fully connected layers and softmax classifiers to predict the class probabilities and refine the bounding box coordinates for each region proposal. This stage refines the region proposals generated by the RPN and improves the accuracy of object localization and classification.

5. Loss Function: During training, Faster R-CNN optimizes its parameters using a multi-task loss function. This function includes a combination of classification loss (e.g., softmax cross-entropy) and localization loss (e.g., Smooth L1 loss) to penalize incorrect predictions and encourage accurate localization of objects within the proposed bounding boxes.

6. Post-Processing: After predictions are made, non-maximum suppression (NMS) is applied to filter out redundant bounding boxes and retain only the most confident detections for each object class.

Faster R-CNN's modular architecture and integration of the RPN for efficient region proposal generation have made it a cornerstone in the field of object detection. It achieves state-of-the-art performance in accuracy while maintaining relatively fast inference speeds, making it suitable for a wide range of applications such as autonomous driving, aerial imagery analysis, and medical imaging where precise object localization and classification are critical.

3. Training and Fine-Tuning

Although the algorithms are evaluated using pre-trained models, we perform additional fine-tuning on the COCO training set to ensure optimal performance. Fine-tuning involves adjusting the hyperparameters and learning rates to adapt the models to the specific characteristics of the COCO dataset. We employ standard data augmentation techniques such as random cropping, flipping, and color jittering to enhance the robustness of the models.

4. Evaluation Metrics

We evaluate the performance of each algorithm using the following metrics:

- Mean Average Precision (mAP): A measure of the accuracy of the object detection model. We calculate mAP at different Intersection over Union (IoU) thresholds (e.g., 0.5, 0.75) to assess the precision of the detected bounding boxes.
- Inference Time: The average time taken by the algorithm to process a single image. This metric is crucial for applications requiring real-time performance.
- Frames Per Second (FPS): The number of images processed per second. Higher FPS indicates better suitability for real-time applications.
- Computational Resource Utilization: We monitor the GPU memory usage and processing power required by each algorithm, providing insights into their efficiency.

5. Experimental Setup

All experiments are conducted on a machine equipped with an NVIDIA GTX 1080 Ti GPU, 32 GB of RAM, and an Intel Core i7 processor. We ensure that the software environment, including the versions of deep learning frameworks (TensorFlow, PyTorch), remains consistent across all experiments to maintain fairness in comparison.

6. Analysis and Comparison

We analyze the results by comparing the performance metrics of each algorithm. The comparative analysis highlights the trade-offs between accuracy, speed, and computational efficiency, providing a clear understanding of the strengths and weaknesses of YOLO, SSD, and Faster R-CNN. We also perform qualitative analysis by visualizing detection outputs on a subset of images from the COCO test set, showcasing the practical implications of each algorithm's performance.

We aim to provide a comprehensive and unbiased evaluation of the selected object detection algorithms, contributing valuable insights to the field of computer vision and aiding practitioners in selecting the most suitable algorithm for their specific applications.

RESULTS AND DISCUSSION

This section presents the results of our comparative analysis of YOLO, SSD, and Faster R-CNN, followed by a discussion of the implications of these findings. We evaluate the algorithms based on mean Average Precision (mAP), inference time, frames per second (FPS), and computational resource utilization.

1. Mean Average Precision (mAP)

The mAP scores at IoU thresholds of 0.5 (mAP@0.5) and 0.75 (mAP@0.75) are summarized in Table 1. Faster R-CNN consistently achieves the highest mAP scores, followed by SSD and YOLO[23].

Table 1: Comparative Mean Average Precision

Algorithm	mAP@0.5	mAP@0.75
YOLOv4	54.30%	32.10%
SSD (MobileNet)	56.80%	34.50%
Faster R-CNN (ResNet-50)	61.20%	37.80%

Discussion: Faster R-CNN's superior accuracy can be attributed to its two-stage detection process, which refines proposals before classification. SSD, with its balance of speed and accuracy, outperforms YOLO in precision but lags behind Faster R-CNN. YOLO's single-stage architecture, while fast, compromises on precision, particularly at higher IoU thresholds.

2. Inference Time and Frames Per Second (FPS)

The average inference time per image and FPS are presented in Table 2. YOLOv4 demonstrates the fastest inference time and highest FPS, followed by SSD and Faster R-CNN[24].

Table 2: Average inference time per image and FPS

Algorithm	Inference Time (ms)	FPS
YOLOv4	25	40
SSD (MobileNet)	45	22
Faster R-CNN (ResNet-50)	120	8

Discussion: YOLOv4's impressive speed is due to its single-pass detection mechanism, making it suitable for real-time applications such as autonomous driving and live video surveillance. SSD strikes a balance between speed and accuracy, making it versatile for various applications. Faster R-CNN, while offering high accuracy, is significantly slower, limiting its applicability in real-time scenarios.

3. Computational Resource Utilization

We assess the GPU memory usage and processing power required by each algorithm. Table 3 summarizes the GPU memory consumption.

Table 3: Memory consumption

Algorithm	GPU Memory Usage (MB)
YOLOv4	2800
SSD (MobileNet)	3200
Faster R-CNN (ResNet-50)	5400

Discussion: Faster R-CNN's higher memory usage is a result of its complex architecture and two-stage detection process, which require more computational resources. YOLOv4, despite its speed, maintains moderate memory usage, making it efficient for deployment on devices with limited resources. SSD's memory usage falls between YOLO and Faster R-CNN, reflecting its balance between simplicity and performance.

Qualitative Analysis

To provide a qualitative perspective, we visualize the detection outputs of each algorithm on a subset of images from the COCO test set. Figure 1 shows examples where each algorithm excels and where they face challenges.

Figure 4: Detection Outputs

- YOLOv4: Quick detections but occasional misses on smaller objects.
- SSD: Balanced detections, good at identifying both large and medium-sized objects.
- Faster R-CNN: High accuracy in detecting small and occluded objects but slower performance.

Discussion

The visualizations from our experiments highlight that Faster R-CNN excels in complex scenes, particularly those containing small or overlapping objects, due to its robust proposal refinement stage. Faster R-CNN employs a two-stage detection process that first generates region proposals using a Region Proposal Network (RPN) and then refines these proposals for more precise classification. This allows the network to focus accurately on areas likely to contain objects, making it highly effective at detecting small or partially occluded objects. Furthermore, its use of deep convolutional networks for detailed feature extraction contributes to its superior performance in cluttered and complex environments. In contrast, SSD and YOLO, which utilize a single-stage detection process, perform well in less complex scenes but face challenges with smaller objects and high object density. SSD uses multiple feature maps at different scales to improve the detection of objects of various sizes, yet it can still struggle with very small objects due to the resolution limits of these feature maps. YOLO's grid-based system, while fast and effective for larger objects, often misses small objects that do not align well with the grid or are located in

densely packed areas. Both algorithms also rely on Non-Maximum Suppression (NMS) to eliminate redundant bounding boxes, which can inadvertently suppress legitimate detections in high-density scenes, leading to decreased accuracy. These observations indicate that Faster R-CNN is more suitable for scenarios requiring high precision in complex scenes, while SSD and YOLO are better suited for applications where speed is critical and scenes are less complex.

CONCLUSION

Our comparative analysis demonstrates that each object detection algorithm has unique strengths and trade-offs. YOLOv4 is ideal for applications requiring real-time performance with moderate accuracy. SSD provides a balanced approach suitable for various tasks, offering a good compromise between speed and precision. Faster R-CNN excels in scenarios where detection accuracy is paramount, though at the cost of increased inference time and resource consumption. These findings provide valuable insights for selecting the appropriate object detection algorithm based on specific application requirements, contributing to more informed decision-making in the deployment of computer vision systems.

REFERENCES

[1] F. Alsakka, S. Assaf, I. El-Chami, and M. Al-Hussein, "Computer vision applications in offsite construction," *Autom. Constr.*, vol. 154, p. 104980, Oct. 2023, doi: 10.1016/j.autcon.2023.104980.

[2] X. Chen and S. Wang, "An Object Detection Method Based on Topological Structure," no. Iccsnt, pp. 1302–1305, 2015.

[3] Y. Huang, Y. Qian, H. Wei, Y. Lu, B. Ling, and Y. Qin, "A survey of deep learning-based object detection methods in crop counting," *Comput. Electron. Agric.*, vol. 215, p. 108425, Dec. 2023, doi: 10.1016/j.compag.2023.108425.

[4] F. Kistler and E. Andr, "Motion Capturing Empowered Interaction with a Virtual Agent in an Augmented Reality Environment," *2013 IEEE Int. Symp. Mix. Augment. Real.*, no. October, pp. 1–6, 2013.

[5] U. M. Butt, S. Letchmunan, F. H. Hassan, M. Ali, A. Baqir, and H. H. R. Sherazi, "Spatio-Temporal Crime HotSpot Detection and Prediction: A Systematic Literature Review," *IEEE Access*, vol. 8, pp. 166553–166574, 2020, doi: 10.1109/access.2020.3022808.

[6] M. C. S. Santana, L. A. P. Junior, T. P. Moreira, D. Colombo, V. H. C. De Albuquerque, and J. P. Papa, "A Novel Siamese-Based Approach for Scene Change Detection with Applications to Obstructed Routes in Hazardous Environments," *IEEE Intell. Syst.*, vol. 35, no. 1, pp. 44–53, 2020, doi: 10.1109/MIS.2019.2949984.

[7] Y. Zhang, Y. Yin, and Z. Shao, "An Enhanced Target Detection Algorithm for Maritime Search and Rescue Based on Aerial Images," *Remote Sens.*, vol. 15, no. 19, 2023, doi: 10.3390/rs15194818.

[8] D. Lin, Z. Zhou, B. Guo, W. Min, and Q. Han, "YOLO-G Abandoned Object Detection Method Combined with Gaussian Mixture Model and GhostNet | 高斯混合模型与 GhostNet 结合的 YOLO-G 遗留物检测方法," *Jisuanji Fuzhu Sheji Yu Tuxingxue Xuebao/Journal Comput. Des. Comput. Graph.*, vol. 35, no. 1, pp. 99–107, 2023, doi: 10.3724/SP.J.1089.2023.19276.

[9] L. Wang, Y. Zhao, S. Liu, Y. Li, S. Chen, and Y. Lan, "Precision Detection of Dense Plums in Orchards Using the Improved YOLOv4 Model," *Front. Plant Sci.*, vol. 13, 2022, doi: 10.3389/fpls.2022.839269.

[10] C. Wang and H. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *Comput. Vis. Pattern Recognit.*, 2020.

[11] B. I. N. Liu, W. Zhao, and Q. Sun, "Study Of Object Detection Based On Faster R-CNN," pp. 7–10, 2016.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks," pp. 1–14.

[13] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[14] W. Fang, L. Wang, and P. Ren, “Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments,” *IEEE Access*, vol. 8, pp. 1935–1944, 2020, doi: 10.1109/ACCESS.2019.2961959.

[15] J. Ma and O. A. Yakimenko, “The concept of sUAS/DL-based system for detecting and classifying abandoned small firearms,” *Def. Technol.*, vol. 30, pp. 23–31, 2023, doi: 10.1016/j.dt.2023.04.017.

[16] L. Wang, Y. Zhao, S. Liu, Y. Li, S. Chen, and Y. Lan, “Precision Detection of Dense Plums in Orchards Using the Improved YOLOv4 Model,” *Front. Plant Sci.*, vol. 13, Mar. 2022, doi: 10.3389/fpls.2022.839269.

[17] R. Li, X. Zeng, S. Yang, Q. Li, A. Yan, and D. Li, “ABYOLOv4: improved YOLOv4 human object detection based on enhanced multi-scale feature fusion,” *EURASIP J. Adv. Signal Process.*, vol. 2024, no. 1, p. 6, Jan. 2024, doi: 10.1186/s13634-023-01105-z.

[18] D. Ren, T. Sun, C. Yu, and C. Zhou, “Research on Safety Helmet Detection for Construction Site,” in *Proceedings - 2021 International Conference on Computer Information Science and Artificial Intelligence, CISAI 2021*, 2021, pp. 186–189, doi: 10.1109/CISAI54367.2021.00042.

[19] C. Wan, X. Chang, and Q. Zhang, “Improvement of Road Instance Segmentation Algorithm Based on the Modified Mask R-CNN,” *Electronics*, vol. 12, no. 22, p. 4699, Nov. 2023, doi: 10.3390/electronics12224699.

[20] W. Liu *et al.*, “SSD : Single Shot MultiBox Detector.”

[21] D. Saluja, H. Kukreja, A. Saini, D. Tegwal, P. Nagrath, and J. Hemanth, “Analysis and comparison of various deep learning models to implement suspicious activity recognition in CCTV surveillance,” *Intell. Decis. Technol.*, vol. 17, no. 4, pp. 917–942, Nov. 2023, doi: 10.3233/IDT-230469.

[22] Z. Liu and H. Lv, “YOLO_Bolt: a lightweight network model for bolt detection,” *Sci. Rep.*, vol. 14, no. 1, p. 656, Jan. 2024, doi: 10.1038/s41598-023-50527-0.

[23] A. E. Maxwell, T. A. Warner, and L. A. Guillén, “Accuracy assessment in convolutional neural network-based deep learning remote sensing studies—part 1: Literature review,” *Remote Sens.*, vol. 13, no. 13, 2021, doi: 10.3390/rs13132450.

[24] D. Pestana, P. R. Miranda, J. D. Lopes, R. U. I. P. Duarte, and M. P. Véstias, “A Full Featured Configurable Accelerator for Object Detection With YOLO,” *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3081818.