

Object Detectors in Autonomous Vehicles: Analysis of Deep Learning Techniques

Lei Du*

Henan Polytechnic Institute Nanyang Henan 473000, China

Abstract—Autonomous vehicles have emerged as a transformative technology with wide-ranging implications for smart cities, revolutionizing transportation systems and optimizing urban mobility. Object detection plays a crucial role in autonomous vehicles, accurately identifying and localizing pedestrians, vehicles, and traffic signs for safe navigation. Deep learning-based approaches have revolutionized object detection, leveraging deep neural networks to extract intricate features from visual data, enabling superior performance in various domains. Two-stage algorithms like R-FCN and Mask R-CNN focus on precise object localization and instance-level segmentation, while one-stage algorithms like SSD, RetinaNet, and YOLO offer real-time performance through single-pass processing. To advance object detection for autonomous vehicles, comprehensive studies are needed, particularly on two-stage and one-stage algorithms. This study aims to conduct an in-depth analysis, evaluating the strengths, limitations, and performance of R-FCN, Mask R-CNN, SSD, RetinaNet, and YOLO algorithms in the context of autonomous vehicles and smart cities. The research contributions include a thorough analysis of two-stage algorithms, a comprehensive examination of one-stage algorithms, and a comparison of different YOLO variants to highlight their advantages and drawbacks in object detection tasks.

Keywords—Autonomous vehicles; object detection; deep learning; two-stage object detectors; one-stage object detectors; comprehensive analysis

I. INTRODUCTION

Autonomous vehicles have emerged as a groundbreaking technology with wide-ranging implications for smart cities [1, 2]. These vehicles, equipped with advanced sensors and intelligent systems, have the potential to revolutionize transportation systems, enhance road safety, and optimize urban mobility [3]. As autonomous vehicles continue to evolve, their applications in smart cities are becoming increasingly significant, paving the way for efficient and sustainable transportation networks.

Object detection plays a crucial role in the operation of autonomous vehicles [4, 5]. It involves the accurate identification and localization of various objects, such as pedestrians, vehicles, and traffic signs, from sensor data [5]. Reliable object detection is essential for autonomous vehicles to perceive their surroundings, make informed decisions, and navigate complex environments safely [6, 7]. By detecting and tracking objects in real-time, autonomous vehicles can anticipate potential hazards, react accordingly, and ensure the safety of both passengers and pedestrians.

Deep learning-based approaches have emerged as a dominant paradigm in object detection [8, 9]. Leveraging the capabilities of deep neural networks, these approaches have revolutionized the field by automatically learning and extracting intricate features from visual data [10, 11]. They have demonstrated superior performance in a variety of domains, including autonomous vehicles [12], surveillance systems, and robotics [9]. Deep learning-based object detection methods have paved the way for significant advancements in accuracy and real-time processing, enabling more robust and efficient autonomous driving systems.

Two-stage and one-stage object detection algorithms are two popular categories in the field of deep learning-based object detection [13]. In the two-stage category, algorithms such as R-FCN and Mask R-CNN have gained prominence [14]. R-FCN focuses on precise object localization using position-sensitive score maps, while Mask R-CNN introduces instance-level segmentation alongside object detection. In the one-stage category, widely recognized algorithms include SSD, RetinaNet, and YOLO. These models operate with a single pass over the input data, offering real-time performance [15]. SSD utilizes a multi-scale approach with default anchor boxes, RetinaNet addresses class imbalance with the Focal Loss, and YOLO achieves efficient object detection by simultaneously predicting object locations and class probabilities.

To further advance object detection algorithms in autonomous vehicles, there is a need for comprehensive studies that focus on both two-stage and one-stage approaches. Additionally, a comparison of YOLO algorithms, given their popularity, would provide valuable insights into their strengths and weaknesses.

The existing literature on object detection in the context of autonomous vehicles and smart cities tends to prioritize algorithmic performance without delving deeply into real-world implementation, which is a significant gap. To address this, the proposed study should give prominence to how these algorithms function in the dynamic and complex environments of urban areas, considering variables such as weather changes, diverse road users, and intricate traffic scenarios. Additionally, the research should establish a comprehensive and tailored set of evaluation metrics, as the conventional ones might not fully capture the distinct challenges presented by autonomous vehicles in smart cities. Novel metrics, especially those accounting for safety and real-time performance, may be required to provide a more accurate assessment of algorithm effectiveness in this context.

In this study, our aim is to conduct an in-depth analysis of deep learning methods for object detection, with a particular focus on two-stage and one-stage algorithms. This analysis will involve reviewing previous studies, identifying their research contributions, and examining these algorithms' performance, efficiency, and suitability in the context of autonomous vehicles and smart city applications.

The research contributions of this study are as follows:

- Providing a thorough analysis of two-stage object detection algorithms, including R-FCN and Mask R-CNN, and evaluating their strengths and limitations in the domain of autonomous vehicles and smart cities.
- Conducting a comprehensive examination of one-stage object detection algorithms, namely SSD, RetinaNet, and YOLO, and assessing their performance, efficiency, and suitability for real-time applications.
- Comparing and contrasting different YOLO algorithms to highlight their respective advantages, drawbacks, and performance in object detection tasks.

II. RELATED WORKS

In study [16], a performance analysis of object detection algorithms is presented for traffic surveillance applications, specifically focusing on the use of neural networks. The study evaluates the effectiveness and efficiency of various neural network-based algorithms in detecting and tracking objects in traffic scenes. By analyzing the performance metrics of these algorithms, the paper provides insights into their strengths, limitations, and applicability in traffic surveillance. The aim is to enhance the understanding of object detection and tracking techniques using neural networks, enabling the development of more effective and efficient solutions for traffic surveillance applications.

In study [17], the implementation of a real-time system for detecting traffic signs and road objects was investigated using mobile GPU platforms. The study focuses on developing an efficient and robust algorithm that can accurately identify and classify traffic signs and other objects in real-time. By utilizing mobile GPU platforms, the system achieves high-performance processing and responsiveness. The paper discusses the implementation details, performance evaluation, and practical implications of the proposed approach. The aim of the study is to provide a practical solution for real-time traffic signs and road object detection on mobile devices, contributing to the advancement of intelligent transportation systems.

The study in [18] presented a comparative study of deep learning-based algorithms for road object detection. The study focuses on two-stage and one-stage object detection methods, analyzing their strengths, limitations, and performance in various applications. The algorithms examined include R-FCN, Mask R-CNN, SSD, RetinaNet, and YOLO. The paper reviews previous studies, identifies research contributions, and highlights the need for further analysis and exploration in this

field. The objective is to contribute to the advancement of object detection techniques, particularly in the context of road and traffic scenarios.

Finally, in [19], small-object detection in autonomous driving systems is explored using the YOLOv5 algorithm. The study addresses the challenge of accurately detecting small objects, such as pedestrians or traffic signs, which are crucial for safe autonomous driving. By utilizing YOLOv5, the paper proposes an approach that improves the detection performance for small objects in real-time. The study evaluates the effectiveness of the YOLOv5 algorithm and its suitability for autonomous driving applications. The aim is to enhance the object detection capabilities in autonomous driving systems, particularly for small objects, contributing to safer and more reliable autonomous vehicles.

III. METHODOLOGY

This study conducts a comprehensive analysis of deep learning methods for applications in road object detection, with a specific emphasis on both two-stage and one-stage approaches. Our objective is to address the most superior algorithms in the field. Additionally, it conducts a thorough comparison of the YOLO algorithms, which are widely recognized as the most popular object detection algorithms. By focusing on two-stage and one-stage methodologies and conducting this comparative analysis, it aims to provide valuable insights into the strengths, weaknesses, and performance of these algorithms. The study will contribute to a better understanding of deep learning methods in object detection and help identify the most effective approaches for various applications.

A. Two-Stages Object Detectors

Two-stage object detectors are a type of deep learning architecture used for object detection tasks. They typically consist of two main stages: region proposal generation and object classification/refinement. These detectors have been widely adopted due to their ability to accurately localize and classify objects in images with complex backgrounds. Inspired from [18], two popular two-stage object detectors are R-FCN [20] and Mask R-CNN [21].

The R-FCN is an object detection model that operates in a fully convolutional manner. In R-FCN, the first stage involves generating a set of region proposals using an external algorithm such as Selective Search. These region proposals serve as potential object locations. In the second stage, R-FCN performs region-based classification and refinement. Instead of using fully connected layers, R-FCN utilizes position-sensitive score maps, which are computed using convolutions. These score maps encode the class probabilities at different spatial locations within each region proposal. Finally, a position-sensitive pooling operation is applied to obtain a fixed-length feature vector for each class. R-FCN achieves state-of-the-art object detection accuracy while being more computationally efficient compared to other two-stage detectors. Fig. 1 shows R-FCN architecture.

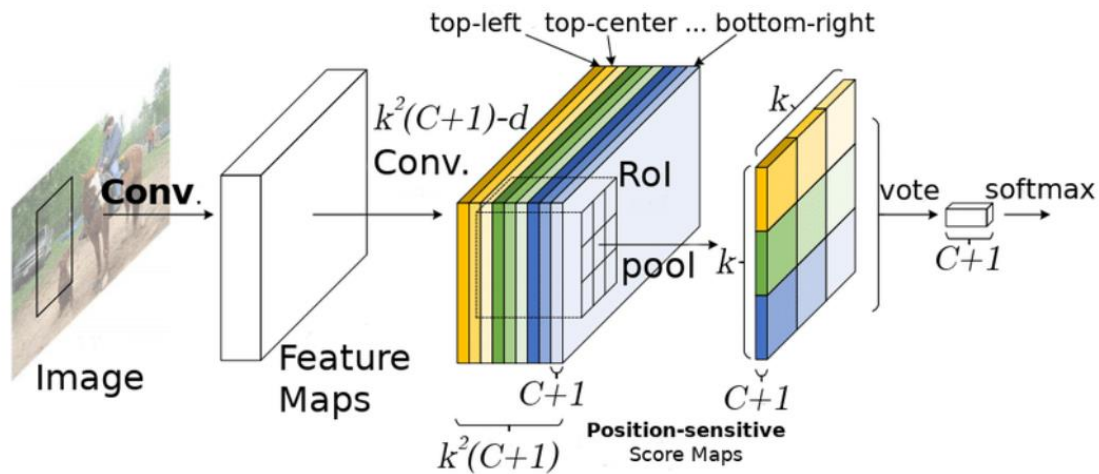


Fig. 1. R-FCN architecture [22].

The Mask R-CNN is an extension of the Faster R-CNN object detection framework. Similar to Faster R-CNN, Mask R-CNN has two main stages. The first stage involves generating region proposals using a region proposal network (RPN). These proposals are refined and classified in the second stage, similar to Faster R-CNN. However, Mask R-CNN introduces an additional branch that performs instance mask prediction for each region of interest (RoI). This branch produces a binary mask indicating the object's precise boundary. This allows Mask R-CNN to simultaneously handle object detection and segmentation tasks, making it a powerful framework for a wide range of applications, including instance segmentation and object tracking. Fig. 2 shows Mask-RCNN architecture.

B. One-Stage Object Detectors

One-stage object detectors are a type of deep learning architecture used for object detection tasks. Unlike two-stage detectors, they directly predict object bounding boxes and class probabilities in a single pass without the need for explicit region proposal generation. This makes one-stage detectors

faster and more efficient, making them suitable for real-time applications. Similar to two-stage and inspiring from [18], this study selected three popular examples of one-stage object detectors Single Shot MultiBox Detector (SSD) [24], RetinaNet [25], and You Only Look Once (YOLO) [26].

SSD, also known as the Single Shot MultiBox Detector, is an efficient one-stage object detection model that strikes a balance between high accuracy and real-time processing. The approach employed by SSD involves dividing the input image into a grid of different sizes. Each grid cell takes on the responsibility of predicting bounding boxes and class probabilities for objects within its designated region. This multi-scale strategy enables SSD to effectively detect objects of varying sizes. Additionally, SSD incorporates default anchor boxes with diverse aspect ratios and scales, enhancing the precision of object localization. Through a sequence of convolutional layers that progressively reduce spatial dimensions, SSD efficiently predicts object bounding boxes and class probabilities across multiple scales in a single pass. The architecture of SSD can be visualized in Fig. 3.

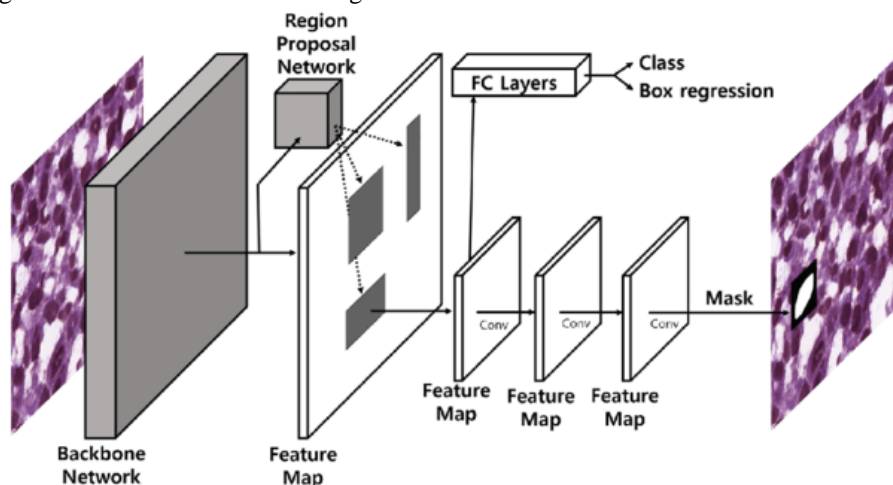


Fig. 2. Mask-RCNN architecture [23].

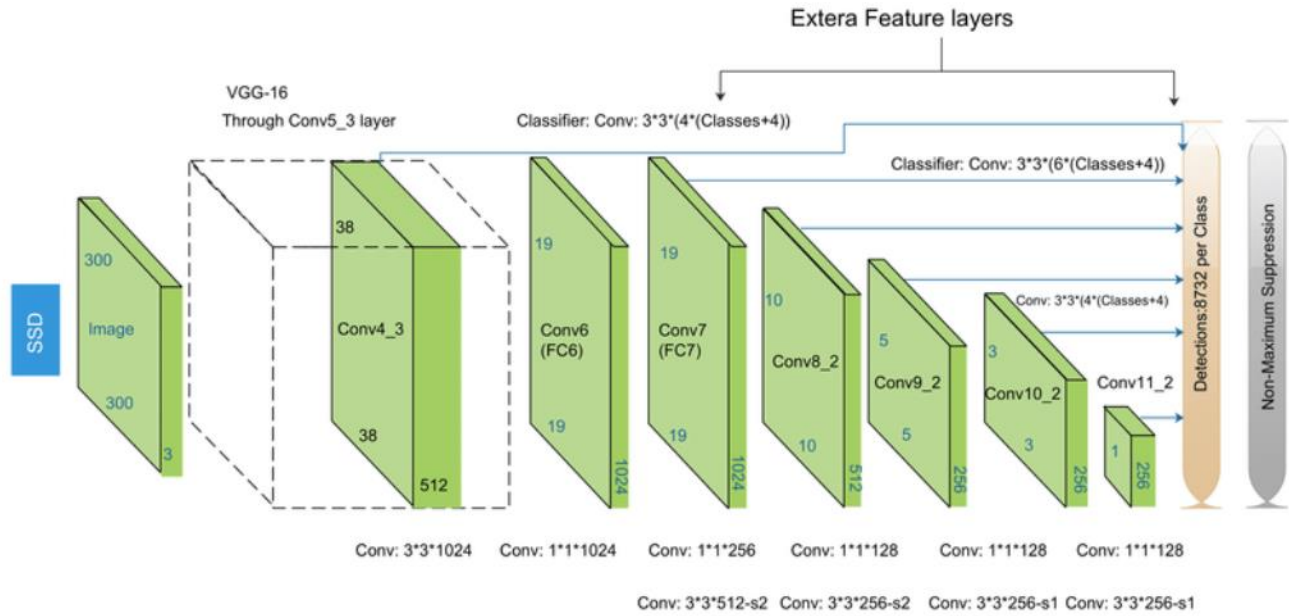


Fig. 3. SSD architecture [27].

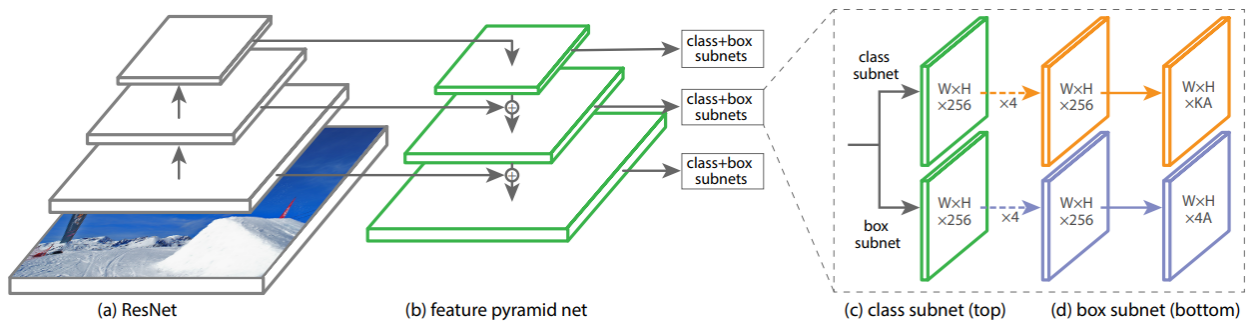


Fig. 4. RetinaNet architecture [25].

RetinaNet is another well-known one-stage object detection system that addresses the issue of imbalanced classes in training. It introduces a unique loss function called Focal Loss, which concentrates on challenging examples that are misclassified or hard to classify. The Focal Loss assigns lower weights to easy examples that are already well-classified, enabling the model to focus more on the difficult examples during the training process. By utilizing this loss function, RetinaNet achieves a better trade-off between accuracy and efficiency. Like SSD, RetinaNet incorporates a feature pyramid network (FPN) that captures multi-scale features and facilitates object detection. The FPN integrates features from different levels of the feature pyramid to effectively handle objects of diverse sizes. The architecture of RetinaNet is depicted in Fig. 4.

YOLO, a groundbreaking one-stage object detection framework, is renowned for its ability to perform in real-time [28]. In contrast to SSD and RetinaNet, YOLO employs a singular neural network that enables simultaneous prediction of object locations and class probabilities, resulting in quicker inference times. Additionally, YOLO incorporates anchor boxes of varying scales and aspect ratios to handle diverse

object characteristics. However, earlier versions of YOLO encountered challenges in accurately detecting small objects. The subsequent releases of YOLOv4 and YOLOv5 have introduced enhancements to overcome this limitation, delivering improved speed and competitive performance.

For the parameter setting of the algorithms, following configuration are used, for the YOLOv4, network architecture, image size, learning rate and batch size are set to Darknet-53, 416x416, 0.001, 64. For the Mask R-CNN, backbone architecture, image size, learning rate, mask resolution are set to ResNet-50, 800x800, 0.001, 28x28. For the RetinaNet, backbone architecture, image size, learning rate and batch size are set to ResNet-50, 800x800, 0.0001 and 4. For the R-FCN, backbone architecture, image size, learning rate and batch size are set to ResNet-50, 800x800, 0.001 and 1. For the SSD, backbone architecture, image size, learning rate and batch size are set to ResNet, 512x512, 0.001 and 32.

IV. RESULTS AND DISCUSSION

This section discusses the performance analysis of two and one-stage object detectors and the performance analysis of Yolo-based object detectors.

A. Performance Analysis of Two and Stages Object Detectors

The PR-curve provides valuable insights into the performance of object detection algorithms [18]. By analyzing this curve, it is possible to assess the precision and recall trade-offs and make informed comparisons between different models.

In Fig. 5, it is evident that YOLOv4 consistently outperforms other models in terms of performance. It achieves the highest precision and recall rates across all levels, demonstrating its effectiveness in accurately detecting objects in various scenarios. The two-stage detection model of Mask R-CNN exhibits commendable precision and recall, surpassing RetinaNet, R-FCN, and SSD, indicating its superior overall detection accuracy and efficiency.

When comparing R-FCN with SSD, it is observed that R-FCN outperforms SSD in terms of precision for target detection at all levels. This indicates that R-FCN provides

more precise detection results, enhancing the reliability of the object detection process. Furthermore, the recall of Mask R-CNN closely matches that of YOLOv4, specifically for target detection with occlusion and truncation, suggesting that Mask R-CNN can accurately detect objects even in challenging scenarios where occlusion and truncation are present.

However, it should be noted that SSD exhibits the lowest recall among all the models, indicating a higher rate of missed detections for targets at all levels. This suggests that SSD may struggle to detect objects accurately compared to the other models. Therefore, based on the PR-curve analysis, YOLOv4 emerges as the best-performing model overall, followed by Mask R-CNN. R-FCN surpasses SSD in terms of precision, while SSD exhibits the lowest recall rate. These insights provide valuable guidance for selecting the most suitable object detection algorithm based on specific requirements and priorities.

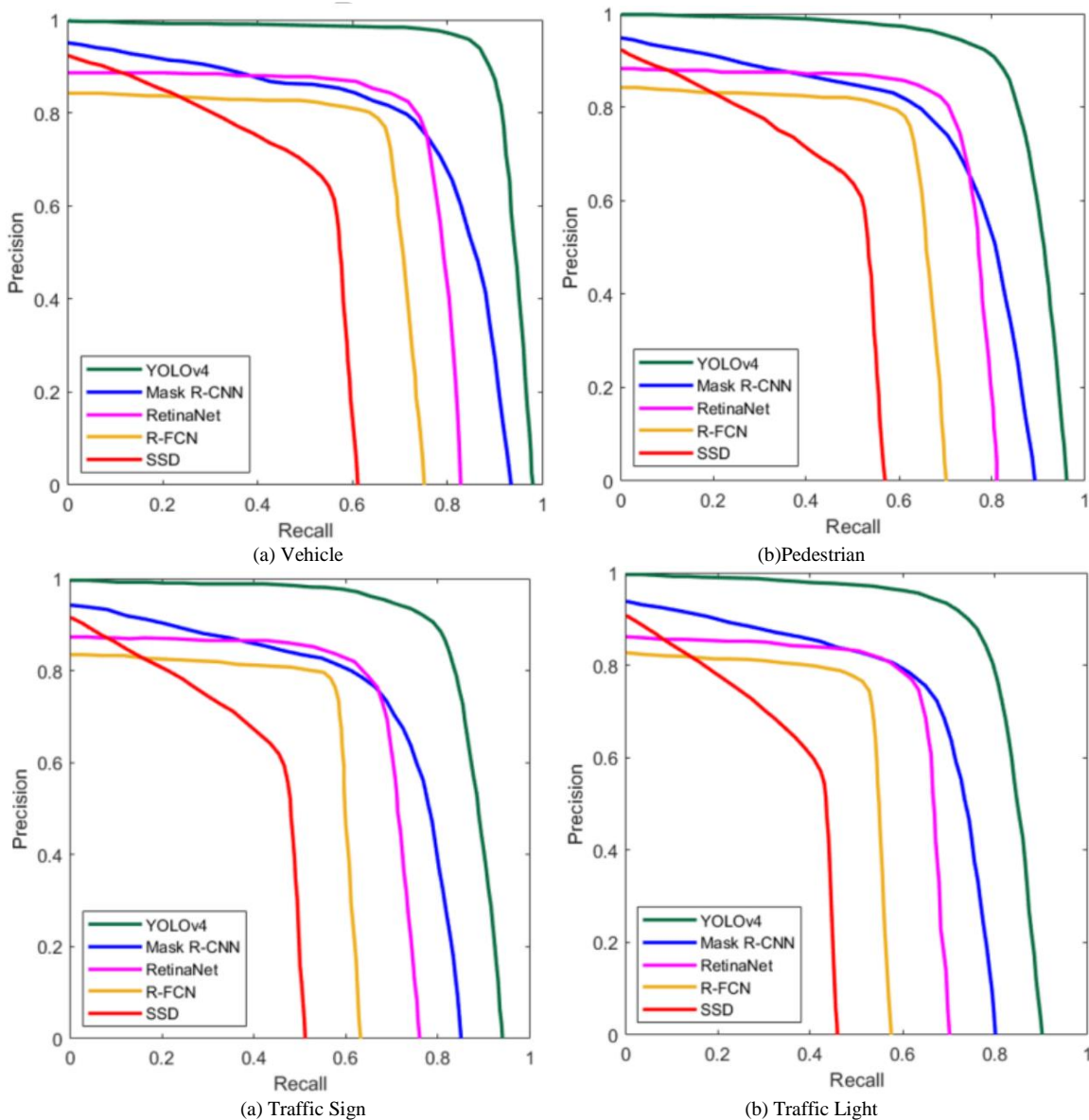


Fig. 5. Precision-Recall (PR) curves of object detection models.

B. Performance Analysis of Yolo-based Object Detectors

In this section, inspired from [29], the performance of YOLO object detection models on different CPU and GPU architectures is investigated. For this comparison, YOLO base models are first explored on NVIDIA GPUs, including the TESLA P100, TESLA V100, GTX 1080Ti, and RTX 4090. The objective is to determine the fastest YOLO model for each GPU, considering factors such as speed and efficiency. This analysis will provide valuable insights for selecting the most appropriate YOLO model based on specific hardware configurations and real-world application requirements.

The YOLO models are designed for real-time object detection and rely on dividing the input image into a grid, predicting bounding boxes and class probabilities for each grid cell. Different versions of YOLO, such as YOLOv5, YOLOv6, and YOLOv7, offer trade-offs between speed and accuracy. The Nano and Tiny variants prioritize lightweight and faster performance, while larger versions like YOLOv7 provide higher accuracy at the cost of slightly reduced speed. By comparing the performance of these models on the specified NVIDIA GPUs, it is possible to identify the fastest model for each GPU, aiding in the selection of the optimal YOLO model based on the desired balance between speed, accuracy, and specific hardware requirements.

As shown in Fig. 6, the graph provides information on the performance of different YOLO (You Only Look Once) models on various GPU devices. Based on the data presented, we can discuss the better method in terms of speed and throughput.

Firstly, from the graph, it is evident that YOLOv5 Nano performs the best in terms of speed on the RTX 4090 GPU and TESLA P100. This indicates that if the primary concern is achieving the highest frames per second (FPS) for real-time object detection, YOLOv5 Nano would be the preferred choice. Secondly, YOLOv7 Tiny stands out as the model that provides the highest throughput on the GTX 1080 Ti and TESLA V100. Throughput refers to the number of objects detected per unit of time, and YOLOv7 Tiny excels in this aspect on these specific GPU devices. This is particularly

beneficial in scenarios where accurately detecting a larger number of objects is more important than achieving the highest FPS.

On the other hand, the YOLOv6 Nano and Tiny models, while not performing at the same FPS as YOLOv5 and YOLOv7, are still not considered very slow. Although the graph does not provide specific data on their performance, it suggests that these models strike a balance between speed and accuracy. They may be a suitable choice when moderate speed is desired while still achieving satisfactory object detection results.

In conclusion, the better method depends on the specific requirements of the task at hand. YOLOv5 Nano is ideal for real-time applications where achieving the highest FPS is crucial. YOLOv7 Tiny excels in scenarios where high throughput is prioritized over real-time performance. Meanwhile, YOLOv6 Nano and Tiny models offer a compromise between speed and accuracy, making them a viable option in cases where moderate speed is desired without sacrificing too much on detection quality.

As shown in Fig. 7, in the CPUs platform, the YOLOv5 Nano models, specifically the P5, are expected to provide the highest speed. These models can achieve real-time frames per second (FPS) performance, surpassing 30 FPS. This means that they can process and analyze images or video streams in real-time, providing quick object detection results. The YOLOv5 Nano models are optimized for efficiency and speed, making them well-suited for consumer-grade CPUs where real-time performance is a priority.

On the other hand, the YOLOv7 Tiny model, while still capable of real-time object detection, operates at a slightly lower speed compared to the YOLOv5 Nano models. It typically runs at around 20 FPS, which is still quite impressive and suitable for many real-time applications. Although it may not match the speed of the YOLOv5 Nano models, the YOLOv7 Tiny model strikes a balance between speed and accuracy. It provides satisfactory results while ensuring efficient processing on general consumer CPUs.

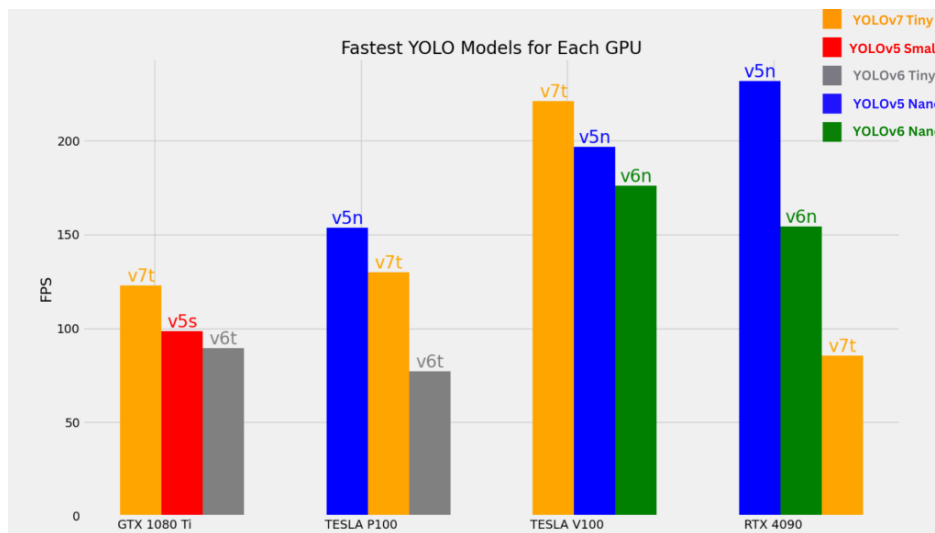


Fig. 6. The fastest YOLO models on each GPU platform [29].

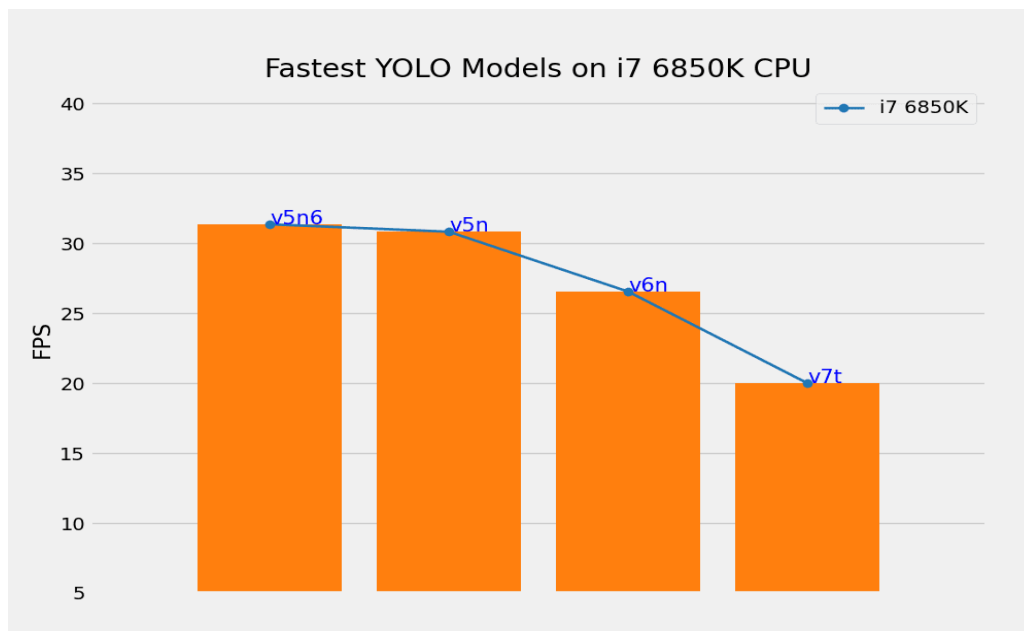


Fig. 7. The four fastest YOLO object detection models on the i7 6850K CPU.

In conclusion, as a result shown, regardless of the specific CPU architecture, it observed that smaller models tend to exhibit faster performance. This is evident in these findings, where the YOLOv5 Nano and Nano P6 models emerged as the fastest options. Remarkably, even on an older generation i7 CPU, these models were able to achieve impressive speeds of over 30 frames per second (FPS). This demonstrates the efficiency and optimization of the YOLOv5 Nano and Nano P6 models for CPU processing, making them excellent choices for real-time object detection on consumer-grade CPUs.

V. CONCLUSION

This study conducted a thorough and comprehensive analysis of deep learning methods for object detection, focusing specifically on both two-stage and one-stage approaches. The main objective of this study is to identify the most superior algorithms in this domain and provide valuable insights into their unique strengths, limitations, and overall performance. It placed particular emphasis on comparing and contrasting the YOLO object detection models, including YOLOv5, YOLOv6, and YOLOv7, with respect to their frames per second (FPS) and accuracy. To ensure the reliability of our results, this study performed experiments using various NVIDIA GPU models such as GTX, RTX, and TESLA. This multi-platform evaluation allowed us to establish a solid foundation for this analysis and draw meaningful comparisons between the different YOLO versions. The findings from the study contribute to a better understanding of deep learning methods in object detection, enabling researchers and practitioners to make informed decisions when selecting the most suitable algorithms for their specific requirements. For future works, one potential future research direction is to investigate the fusion of two-stage and one-stage object detection algorithms to leverage their respective strengths and improve overall performance. Another promising avenue for future work is the adaptation of

object detection algorithms for edge computing, aiming to optimize models for resource-constrained edge devices and enable real-time object detection at the network edge.

REFERENCES

- [1] Y. Bai, B. Zhang, N. Xu, J. Zhou, J. Shi, and Z. Diao, "Vision-based navigation and guidance for agricultural autonomous vehicles and robots: A review," *Computers and Electronics in Agriculture*, vol. 205, p. 107584, 2023.
- [2] P. Shubham and V. Begishev, "Exploration of Autonomous Vehicles in Smart City Environment," *EasyChair*, 2516-2314, 2023.
- [3] M. A. Richter, M. Hagenmaier, O. Bandte, V. Parida, and J. Wincent, "Smart cities, urban mobility and autonomous vehicles: How different cities needs different sustainable investment strategies," *Technological Forecasting and Social Change*, vol. 184, p. 121857, 2022.
- [4] A. Balasubramaniam and S. Pasricha, "Object detection in autonomous vehicles: Status and open challenges," *arXiv preprint arXiv:2201.07706*, 2022.
- [5] N. M. A. A. Dazlee, S. A. Khalil, S. Abdul-Rahman, and S. Mutalib, "Object detection for autonomous vehicles with sensor-based technology using yolo," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 10, no. 1, pp. 129-134, 2022.
- [6] S. Liang et al., "Edge YOLO: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25345-25360, 2022.
- [7] Y. Peng, Y. Qin, X. Tang, Z. Zhang, and L. Deng, "Survey on image and point-cloud fusion-based object detection in autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 22772-22789, 2022.
- [8] V. S. R. Kosuru and A. K. Venkitaraman, "Preventing the False Negatives of Vehicle Object Detection in Autonomous Driving Control Using Clear Object Filter Technique," in *2022 Third International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, 2022: IEEE, pp. 1-6.
- [9] T. Sharma, B. Debaque, N. Duclos, A. Chehri, B. Kinder, and P. Fortier, "Deep learning-based object detection and scene perception under bad weather conditions," *Electronics*, vol. 11, no. 4, p. 563, 2022.
- [10] S. Iftikhar, Z. Zhang, M. Asim, A. Muthanna, A. Koucheryavy, and A. A. Abd El-Latif, "Deep Learning-Based Pedestrian Detection in

- Autonomous Vehicles: Substantial Issues and Challenges," *Electronics*, vol. 11, no. 21, p. 3551, 2022.
- [11] S. Alaba, A. Gurbuz, and J. Ball, "A comprehensive survey of deep learning multisensor fusion-based 3d object detection for autonomous driving: Methods, challenges, open issues, and future directions," 2022.
- [12] X. Ma, W. Ouyang, A. Simonelli, and E. Ricci, "3D object detection from images for autonomous driving: a survey," *arXiv preprint arXiv:2202.02980*, 2022.
- [13] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: Challenges, architectural successors, datasets and applications," *Multimedia Tools and Applications*, vol. 82, no. 6, pp. 9243-9275, 2023.
- [14] L. Du, R. Zhang, and X. Wang, "Overview of two-stage object detection algorithms," in *Journal of Physics: Conference Series*, 2020, vol. 1544, no. 1: IOP Publishing, p. 012033.
- [15] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, "On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data," *Remote Sensing*, vol. 13, no. 1, p. 89, 2020.
- [16] N. Jain, S. Yerragolla, and T. Guha, "Performance Analysis of Object Detection and Tracking Algorithms for Traffic Surveillance Applications using Neural Networks," in *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, 2019: IEEE, pp. 690-696.
- [17] E. Güney, C. Bayilmiş, and B. Çakan, "An implementation of real-time traffic signs and road objects detection based on mobile GPU platforms," *Ieee Access*, vol. 10, pp. 86191-86203, 2022.
- [18] B. Mahaur, N. Singh, and K. Mishra, "Road object detection: a comparative study of deep learning-based algorithms," *Multimedia Tools and Applications*, vol. 81, no. 10, pp. 14247-14282, 2022.
- [19] B. Mahaur and K. Mishra, "Small-object detection based on YOLOv5 in autonomous driving systems," *Pattern Recognition Letters*, vol. 168, pp. 115-122, 2023.
- [20] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," *Advances in neural information processing systems*, vol. 29, 2016.
- [21] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask RCNN. 2017 IEEE International Conference on Computer Vision," 2017.
- [22] M. Peker, "Comparison of tensorflow object detection networks for licence plate localization," in *2019 1st Global Power, Energy and Communication Conference (GPECOM)*, 2019: IEEE, pp. 101-105.
- [23] H. Jung, B. Lodhi, and J. Kang, "An automatic nuclei segmentation method based on deep convolutional neural networks for histopathology images," *BMC Biomedical Engineering*, vol. 1, no. 1, pp. 1-12, 2019.
- [24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 2016: Springer, pp. 21-37.
- [25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980-2988.
- [26] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
- [27] M. Bahaghighat, Q. Xin, S. A. Motamedi, M. M. Zanjireh, and A. Vacavant, "Estimation of wind turbine angular velocity remotely found on video mining and convolutional neural network," *Applied Sciences*, vol. 10, no. 10, p. 3544, 2020.
- [28] C. Liu, Y. Tao, J. Liang, K. Li, and Y. Chen, "Object detection based on YOLO network," in *2018 IEEE 4th information technology and mechatronics engineering conference (ITOEC)*, 2018: IEEE, pp. 799-803.
- [29] V. G. Sovit Rath. Performance Comparison of YOLO Object Detection Models – An Intensive Study [Online] Available: <https://learnopencv.com/performance-comparison-of-yolo-models/>