Occluded Object Detection in Real Time

Dhairya Jain[a], Mannat Arora[b], Nehal Garg[c]
[a,b,c]Dr. Akhilesh Das Gupta Institute of Professional Studies

ARTICLE INFORMATION:

Keywords - Occluded object detection, BDD100K , YOLOv1 , Faster R-CNN, Deep learning, Computer vision , YOLOv5 , BiFPN.

ABSTRACT: This paper investigates occluded object detection by comparing the performance of two deep learning models, Faster R-CNN and YOLO. Despite their success in generic object detection, their efficacy in handling occlusion is underexplored. We utilize a tailored dataset emphasizing occluded instances and fine-tune both models accordingly. Evaluation parameters comprising precision, recall, and F1 score assess their performance under varying occlusion levels. Results unveil strengths and weaknesses, providing practical insights for selecting the appropriate model in occlusion-rich applications like surveillance and autonomous driving.

1. INTRODUCTION

Recent years have seen an unheard-of growth in the field of computer vision due to the convergence of sophisticated algorithms, rising computing power, and the accessibility of enormous datasets. One of the central challenges within this domain is the accurate and efficient classification of objects within visual data, a task that forms the cornerstone of many practical applications like autonomous systems, surveillance, and human-computer interaction.

One of the core functions of computer vision is object classification, which is the process of recognizing and classifying things in images or video streams. This process is pivotal for systems to comprehend and interpret their visual surroundings, enabling them to make informed decisions in response to dynamic and complex environments. As technology continues to advance, the demand for robust and reliable object classification methods becomes increasingly paramount.

The advent of deep learning has brought about a paradigm shift in the way computers perceive and understand visual information. CNN's, in particular, are deep learning models that have shown unmatched performance in image processing tasks like object detection.The intricate hierarchical feature learning enabled by deep architectures allows these models to automatically extract relevant patterns and representations from raw visual data.

YOLO and Faster R-CNN are two well-known deep learning models that have advanced object detection to unprecedented levels.

2. LITERATURE SURVEY

2.1 RELATED WORK FOR YOLO

Scholars have achieved noteworthy progress in object detection in recent literature by putting out creative models that draw from both RCNN and YOLO architectures. One noteworthy contribution comes from Wang et al., who introduced RSI-YOLO as an enhanced version of YOLOv5 tailored for efficient object detection in remote sensing images [7]. To enhance feature fusion, they used complex channel attention and spatial attention processes in their study. The integration of these attention mechanisms led to superior detection performance compared to traditional object detection algorithms.

Furthermore, Sridhar performed research in which a machine learning model was trained using the YOLO real-time object identification technique. Notable for its simplicity and effectiveness, YOLO distinguishes itself by accurately anticipating bounding boxes and class probabilities for the full image, leading to quicker detection. Sridhar's research emphasizes YOLO's benefits in terms of speed and ease of use, as well as how effective it is in real-time applications..

In a different domain, Zhou et al. proposed YOLO-CIR, an object detection algorithm specifically designed for infrared images using

transfer learning techniques [8]. YOLO-CIR surpassed the performance of both YOLOv5 and Faster R-CNN, particularly on the basis of mean average precision (map50). This achievement emphasizes the adaptability of YOLO-based models to different domains, showcasing its versatility in handling specialized data types like infrared imagery.

These studies collectively contribute valuable insights into the versatility and effectiveness of RCNN and YOLO architectures for object detection across diverse domains, ranging from remote sensing to real-time applications and infrared imaging. The incorporation of attention mechanisms, transfer learning, and domain-specific optimizations demonstrates the continuous evolution and refinement of object detection methodologies.

2.2 RELATED WORK FOR FASTER R-CNN

This model has emerged as a powerful and widely utilized framework for object detection in diverse domains. Researchers have actively contributed to the continuous improvement of the Faster R-CNN model, proposing modifications and enhancements to address specific challenges and optimize its performance in various applications.

The Fast Guided Anchored Stereo RCNN (FGAS RCNN), developed especially for 3-Dimensional object identification in autonomous driving scenarios, is one noteworthy development by Chongben Tao et al. [12]. The key innovation of FGAS RCNN lies in its ability to reduce computational costs while maintaining a high regression rate. This is crucial in autonomous driving, where real-time processing is essential. By integrating guided anchoring techniques, the model achieves efficient 3D object detection, demonstrating its suitability for resource-constrained environments.

Wenshun Sheng et al. presented CF-RCNN, an improved Faster R-CNN variation, in a different contribution. CBAM and FPN, two essential modules, are integrated by CF-RCNN. These enhancements seek to increase the detection and recognition accuracy, particularly for hard circumstances including small-sized, obscured, or truncated items in complicated settings. The use of feature pyramid networks and attention processes improves the model's capacity to handle objects with different sizes and degrees of occlusion.

Furthermore, Hao Wang and Nanfeng Xiao focused on optimizing Faster R-CNN for underwater object detection [14]. They replaced the original VGG16 structure with the Res2Net101 network and introduced optimization techniques such as Online Hard Example Mining (OHEM), Generalized Intersection over Union (GIOU), and Soft Non-Maximum Suppression (Soft-NMS). These modifications were tailored to the unique challenges of underwater environments, resulting in improved performance.

Collectively, these papers showcase the adaptability and effectiveness of the Faster R-CNN model in diverse applications. The proposed enhancements address specific challenges in domains such as autonomous driving, complex scenes, and underwater environments, demonstrating the model's versatility and potential for advancing object detection capabilities in various fields. The continuous refinement of Faster R-CNN through innovative modifications underscores its significance as a foundational framework in the domain of Computer Vision.

3. OBJECTIVE

With the advent of deep learning techniques, the field of computer vision has advanced significantly in recent years. Among these developments, models such as Faster R-CNN and (YOLO) have distinguished themselves with their remarkable capacity to identify and pinpoint objects in pictures and videos for a diverse range of applications .However, despite their prowess in conventional object detection tasks, these models face significant challenges when confronted with real-world scenarios where objects are occluded, or partially obscured by other elements in the scene.

Occlusion, a prevalent phenomenon in natural environments, presents a critical obstacle to the accurate detection of objects in images and videos. Whether encountered in surveillance footage, autonomous driving scenarios, or robotics applications, the ability to detect occluded objects remains paramount for ensuring the effectiveness and reliability of computer vision systems. The capability of a model to accurately identify and locate objects even when partially hidden is pivotal for its practical utility in these contexts.

This paper is dedicated to addressing the specific issue of occluded object detection, with the aim of contributing to our understanding of how YOLO and Faster R-CNN perform in scenarios involving partial visibility. While both models have garnered acclaim for their distinctive architectures - YOLO characterized by its one-shot detection process, and

Faster R-CNN employing a two-shot approach - their adaptability to occlusion remains a subject of inquiry, necessitating a systematic evaluation and comparison.

In order to conduct a thorough analysis of these models' performance in occluded object recognition, we have carefully selected a specialized dataset that highlights partial obstructions in a variety of object types. We rigorously explore and refine both YOLO and Faster R-CNN on this dataset, enabling the models to manage the complexities caused by occlusion. The assessment criteria that are utilized include precision, recall, and F1 score. These metrics offer a quantitative foundation for evaluating the models' performance at different levels of occlusion.

Beyond shedding light on the specific challenges of occluded object detection, this research endeavors to offer practical insights for choosing an appropriate model based on the unique requirements of applications where occlusion is a prevalent concern. The outcomes of this study are anticipated to enrich the ongoing discourse surrounding the optimization of deep learning models for real-world scenarios, thereby advancing the field of computer vision toward more robust and reliable solutions.

Autonomous driving is a pivotal technology of the 21st century, offering navigation without human intervention through responsive adaptation to the vehicle's surroundings [1]. Object detection, a key aspect of autonomous navigation, poses challenges in Computer Vision and Machine Learning. It involves localizing and classifying objects, often requiring algorithms to handle multiple potential object positions, or "proposals," with trade-offs between speed, accuracy, and simplicity [2].

Deep learning has spurred the development of sophisticated object detection models such as R-CNN introduced in 2013, and the real-time optimized (YOLO) unveiled in 2015 [3]. The transformation of transportation towards safer, more efficient, and low-emission autonomous vehicles hinges on real-time, precise target detection for informed decision-making and control signals. However, achieving accurate target recognition with minimal computational power remains a substantial challenge, especially in complex real-world settings.

This review focuses on identifying occluded objects in intricate driving scenarios, examining various occluded object detection methods utilizing available sensors. Covering the last five years, it provides insights into detecting occluded pedestrians, cars, and traffic signs. A comprehensive overview of target detection techniques, including diverse methods and theories, is presented, followed by a comparative analysis of their advantages and disadvantages. The discussion concludes by highlighting current limitations and suggesting future research directions in this evolving field.

4. METHODOLOGY

The objective of this research is to conduct a comprehensive evaluation and comparison of the YOLOv1, Faster R-CNN and YOLOv5 algorithms for object detection tasks.

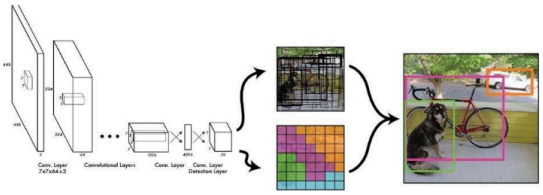

Figure 1: Work Flow of YOLO's algorithm

For YOLOv1, we aim to delve into its unique single-pass approach, which analyzes the entire image once for predictions instead of using multiple passes or region proposals like other algorithms. Specifically, we seek to understand its grid-based prediction mechanism, where each grid cell predicts bounding boxes and class probabilities, leading to an end-to-end processing architecture.

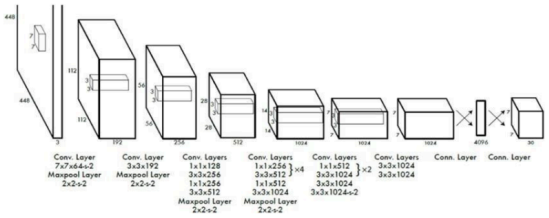

Figure 2: Visualised View of YOLO's

Furthermore, our objective involves studying the model architecture of YOLOv1, including its convolutional layers, pooling layers, and fully connected layers. We aim to comprehend how it incorporates GoogLeNet's Inception Modules and Leaky ReLu activation functions, as well as the implementation of dropout to mitigate overfitting

concerns.

Figure 3: Implementing YOLO with a grid size of 14

Furthermore, we want to study the customised loss function of YOLOv1, which incorporates a squared classification loss and manages differences in centre coordinates, bounding box size, and confidence scores.

Our goal is to investigate the core elements of the region proposal network (RPN), the fast R-CNN module, and the network backbone of the faster R-CNN. We want to comprehend the production of high-resolution feature maps by the network backbone, which is usually a classification network such as ResNet50 pretrained on ImageNet, as well as the process by which region suggestions from the RPN are fed into the rapid R-CNN module.

Moreover, we study the fast R-CNN module, which uses fully connected layers and ROI pooling for classification and bounding box refinement, and the region proposal network of Faster R-CNN, which generates region proposals using specified anchor boxes.

Our overall goal is to contribute to the improvement of computer vision research and applications by offering insights on the advantages, disadvantages, and performance characteristics of both YOLO and Faster R-CNN for object detection tasks.

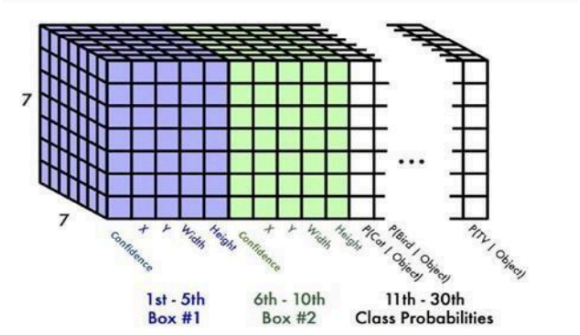

Figure 4: Tensor Output of YOLO.

4.1. YOLO(V1) Objectives:

4.1.1 Single-Pass Approach: With its method of analyzing a complete image in a single neural network pass, YOLO is different from other approaches. The conventional approaches, which make use of several passes or region recommendations, are different from this.

4.1.2 Grid-Based Prediction Mechanism: YOLO splits the input image into a grid and then projects bounding boxes and class probabilities for each grid cell. Understanding this technique is crucial to understanding the end-to-end processing architecture of YOLO.

4.1.3 Model Architecture: Understanding YOLO's convolutional, pooling, and fully linked layers is necessary to investigate its architecture. This includes investigating the ways in which dropout for regularization, leaky ReLu activation functions, and Inception Modules are incorporated.

4.1.4 Tailored Loss Function: To deal with differences in different parts of the projected bounding boxes and class probabilities, YOLO employs a particular loss function. To understand how YOLO is trained and optimized, one must have a thorough understanding of the elements and implications of this loss function..

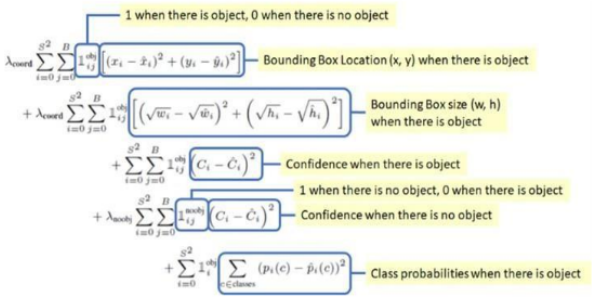

Figure 5: Loss Function of YOLO.

4.2. Faster R-CNN Objectives:

4.2.1 Fundamental Components: Understanding the components of Faster R-CNN, including the network backbone, the region proposal network (RPN), and the fast R-CNN module, is essential. This involves comprehending how each component contributes to overall architecture and functionality.

4.2.2 Region Proposal Network (RPN): Exploring the RPN involves understanding how it generates region proposals using predefined anchor boxes and predicts bounding boxes and classification scores for these proposals.

4.2.3 Fast R-CNN Module: Investigating the fast R-CNN module entails understanding how it integrates high-resolution feature maps and proposed regions using ROI pooling and fully connected layers. This includes understanding the loss functions used for classification and bounding box regression.

4.2.4 Performance Characteristics: Finally, the research aims to provide insights into the performance characteristics of both YOLO and Faster R-CNN. This involves evaluating their strengths, weaknesses, and trade-offs in terms of accuracy, speed, and robustness for object detection tasks.

4.3 You only look once (YOLOv5) ‘

The YOLOv5 network consists primarily of four components: the input layer, the main backbone layer, the feature fusion layer in the neck, and the output layer in the head. This framework is illustrated in Figure 6.

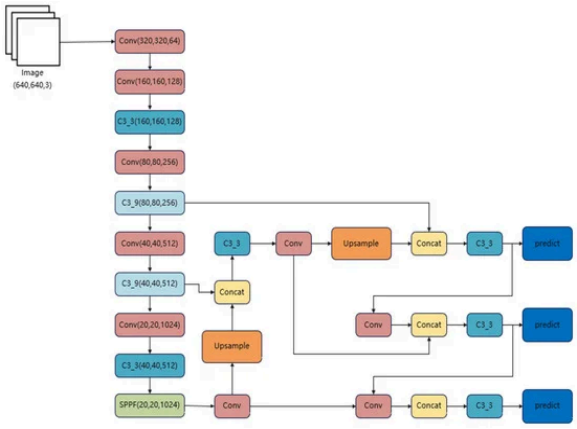

Fig 6 Structural diagram of the YOLOv5 network.

Three modules make up the YOLOv5 main backbone layer: the Conv, C3, and SPPF modules. With constant feature map size reduction, its goal is to extract features from the input picture. The backbone network of YOLOv5 is called CSPDarknet, and it extracts features from input images by applying numerous CBS convolutional layers. After convolution, features are extracted further by the C3 module, and pooling operations are carried out by the SPPF module to create feature layers at three different scales: 80 × 80, 40 × 40, and 20 × 20.

SiLu activation function, BatchNorm2d, and Conv2d are all included in the Conv module. It is mostly used for organizing the feature map and extracting features. BatchNorm2d normalizes batch data, while the SiLu function enhances the model's non-linear fitting capability for detection tasks, as depicted in Figure 7.

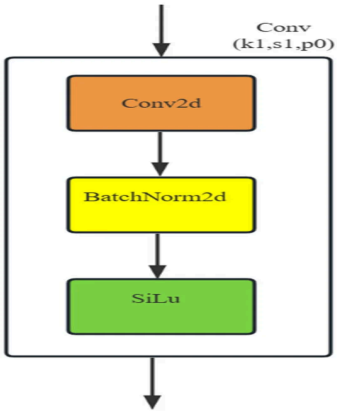

Figure 7. CBS convolutional layer.

As a feature extraction unit, the C3 module improves the feature representation by combining the picture features that the CBS convolutional layer extracted. As seen in Figure 8, the feature map travels through two stages of processing when it enters the C3 module. First, the C3's Conv module lowers the feature map's dimensionality, which helps the convolution kernel understand the feature data better. This improves the dimensionality in order to extract more detailed feature data. In order to remove superfluous gradient information, input and output are combined and a residual structure is utilised for feature extraction.

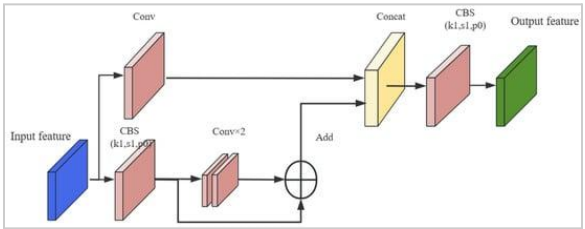

Figure 8. Schematic diagram of the C3 module.

4.4 Enhanced (YOLOv5 + BiFPN)

BiFPN introduces an innovative structure for

multi-scale feature fusion by addressing limitations of traditional FPN architectures, which do not fully exploit features across different scales. The traditional FPN predominantly uses a top-down approach, resulting in a potential loss of detail in lower-resolution features. Conversely, BiFPN integrates a bottom-up path, enhancing the capture and integration of fine-grained details from low-level features with higher-level features.

BiFPN creates bidirectional connections that enable the propagation and fusion of features across many levels in order to optimise feature representation. For computer vision tasks like object recognition and picture segmentation, this bidirectional flow greatly increases the efficiency and accuracy of feature representation.

BiFPN's architecture has been modified to improve feature fusion by removing nodes with a single input and by include more edges. At the same scale, skip connections are also incorporated by BiFPN to improve feature integration without appreciably raising computing requirements. In contrast to PANet, which is limited to a single consecutive top-down and bottom-up path, BiFPN repeats layers to enable more intricate and successful feature fusion by treating each bidirectional link as a distinct layer inside the network. Figure 9 shows the architecture and operation of BiFPN and highlights its sophisticated multi-scale feature information processing capabilities.

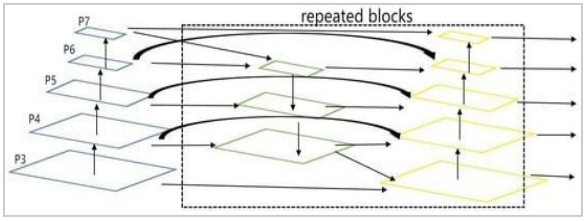

Figure 9. Structural diagram of the BiFPN fusion structure.

5. RESULTS AND DISCUSSION

Our investigative findings disclose that YOLO underwent a training regimen spanning 100 epochs, characterized by a diminishing step size of 10^−5 and a batch size of 25. In contrast, Faster R-CNN underwent training for 75 epochs, utilizing a declining learning rate set at 10^−4 with a batch size of 25. The accompanying picture shows the graphical depiction of the mean Average Precision (mAP) and normalized total loss trajectories during the training phase. We make real-time object identification films available so that viewers may have a thorough grasp of the implementation details and processed data. We demonstrate the effectiveness of both YOLO and Faster R-CNN. We evaluated the Frames Per Second (FPS) and mAP of Yolo and Faster R-CNN in comparison using an Apple M1 Pro processor, which has an 8-core CPU with 6 performance cores, 2 efficiency cores, a 14-core GPU, a 16-core Neural Engine, and a memory bandwidth of 200 GB/s.

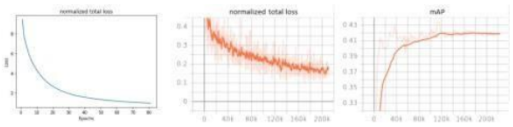

Figure 10: : Collective loss after normalized for YOLO (positioned on the left), collective loss after normalized for Faster R-CNN (positioned in the middle), andMAP for Faster R-CNN (positioned on the right).

Figure 11:Outcomes achieved on BDD100K: Faster R-CNN (displayed on the left)and YOLO (shown on the right).

	mAP
YOLO	18.6
Faster R-CNN	41.8
Hybrid Incremental Net	45.7
YOLOv5s	47.8
YOLOv5s + BiFPN	48.6

Table 1 : Comparison of YOLO, Faster R-CNN, Hybrid Incremental Net, YOLOv5s and YOLOv5s + BiFPN in terms of some standard metrics (mAP) i.e. mean Average Precision (mAP) on the BDD100K dataset.

6. CONCLUSION

In the dynamic realm of autonomous driving, characterized by split-second decisions and unparalleled precision, our project embarked on a journey to implement and execute object detection models: the one-stage marvel, YOLO , and the sophisticated two-stage powerhouse, Faster R-CNN. Leveraging the expansive BDD100K dataset, which encapsulates diverse real-world driving scenarios, our experimentation sought to illuminate the distinctive trade-offs between these titans of object detection.

As anticipated, our evaluation unveiled the contrasting strengths of Faster R-CNN and YOLO. Faster R-CNN, with its intricate two-stage architecture, showcased remarkable accuracy, adeptly identifying and localizing objects with exceptional precision. However, this commendable accuracy was accompanied by a reduction in Frames Per Second (FPS), underscoring the computational intensity of its meticulous, two-step detection process.

Conversely, YOLO emerged as the speed demon, boasting significantly higher FPS thanks to its streamlined, one-stage detection mechanism. Yet, this efficiency came at the expense of accuracy, as its simplified architecture may not capture nuanced details as comprehensively as Faster R-CNN.

Looking ahead, our roadmap for future work is guided by a commitment to pushing the boundaries of object detection in the context of autonomous driving. We envision conducting further experiments with the latest iterations of YOLO, harnessing the advancements of subsequent versions to unlock their enhanced capabilities. The decision to employ YOLOv1 in this project serves merely as a stepping stone, laying the groundwork for exploration into the latest and most refined architectures.

In the long term, our aspirations soar towards achieving a harmonious blend of high accuracy and formidable FPS—an elusive goal that epitomizes the holy grail for autonomous driving applications. This pursuit demands a relentless commitment to discovering novel models, implementing ingenious optimizations, and pioneering groundbreaking methodologies that bridge the gap between the precision of Faster R-CNN and the speed of YOLO. Our aspiration is nothing short of setting new benchmarks in performance, ensuring that our models not only meet but exceed the stringent demands of autonomous driving, propelling the industry into a new era of unparalleled efficiency and safety. Our research focused on the implementation and training of two cutting-edge object recognition models, namely YOLO and Faster R-CNN . The context of our study revolves around the critical demands of split-second decisions and precision in the field of autonomous driving. Leveraging the extensive BDD100K dataset, covering diverse real-world driving scenarios, our analysis brought to light distinct trade-offs between these two leading object detection models.

Faster R-CNN exhibited remarkable accuracy in the precise detection and localization of objects, owing to its advanced two-stage architecture. However, this heightened precision came at the expense of a reduction in Frames Per Second (FPS), illustrating the computational complexity of its intricate two-step detection process. On the other hand, YOLO adopted a speed-centric approach, achieving faster FPS due to its simplified one-stage detection process. Nevertheless, this efficiency was accompanied by a compromise in accuracy, as the streamlined design might not capture subtle details as comprehensively as Faster R-CNN.

As we chart our course for future work, we envisage delving deeper into the evolving landscape of YOLO by exploring its latest iterations. The decision to start with YOLOv1 in this project serves as a foundation for harnessing the capabilities introduced in subsequent versions. Our aim is to continually refine and optimize our models, leveraging advancements in deep learning architectures, optimization techniques, and domain adaptation methods.

Looking ahead, our overarching objective is to strike a harmonious balance between high accuracy and formidable FPS—a vital ratio for the demanding applications of autonomous driving. This entails an ongoing pursuit of innovative models, sophisticated optimizations, and groundbreaking methodologies that bridge the accuracy gap between the speed-centric YOLO and the precision-focused Faster R-CNN. Through these endeavors, we aspire to set new benchmarks in performance, ushering in a new era characterized by unparalleled efficiency and safety in the field of autonomous driving.

7. APPLICATIONS

7.1 Autonomous Driving: Enhancing object detection models aids in safer navigation and decision-making for autonomous vehicles, detecting vehicles, pedestrians, and obstacles on the road.

7.2 Traffic Management: Object detection systems optimize traffic flow, detect congestion, and improve

signal control, contributing to efficient traffic management in urban areas.

7.3 Pedestrian Safety: Accurate detection of pedestrians enables the development of warning systems for drivers and enhances pedestrian safety at crosswalks and crowded locations.

7.4 Smart City Infrastructure: Integrating object detection technology improves city management by monitoring vehicle movement, optimizing transportation routes, and enhancing public safety measures.

7.5 Surveillance and Security: Object detection models are utilized for surveillance purposes, monitoring parking lots, identifying suspicious behavior, and enhancing security in public spaces.

8. FUTURE SCOPE

As we envision the exciting possibilities for our project, we have opted to develop a model using Faster R-CNN, expecting it to reside on the other end of the speed-efficiency tradeoff spectrum. Our decision to transition from YOLO to Faster R-CNN marks a strategic shift towards prioritizing accuracy, adaptability, and efficiency across various applications. Looking ahead, here's a perspective on the future work, considering these factors:

a. Performance:

Faster R-CNN, with its two-stage detection process involving region proposal networks (RPN) and object detection networks, offers superior accuracy. Future work involves exploring advanced backbones such as ResNet, DenseNet, or even more recent architectures like EfficientNet to enhance feature extraction. Experimentation with larger and more diverse datasets, coupled with transfer learning techniques, can further boost performance. Employing attention mechanisms, such as self-attention, could improve the model's ability to focus on relevant parts of the image, enhancing overall accuracy.

b. Adaptability:

Faster R-CNN's versatility lies in its ability to handle a wide spectrum of object detection routines, including object localization and instance segmentation. Future work involves adapting the model to different domains and scenarios. Domain adaptation techniques, such as adversarial training or self-training, can help the model generalize better across diverse datasets. Additionally, investigating few-shot learning methods enables the model to recognize new object classes with minimal labeled data, enhancing its adaptability in dynamic environments.

c. Robustness:

Enhancing the robustness of Faster R-CNN involves addressing challenges such as occlusions, scale variations, and complex backgrounds. Techniques like data augmentation with realistic synthetic data can expose the model to diverse scenarios. Moreover, incorporating contextual information and multi-scale features through advanced network architectures or attention mechanisms can improve the model's understanding of complex scenes, making it more robust in challenging conditions.

d. Interpretability and Explainability:

Understanding the decisions made by the model is crucial, especially in applications like autonomous vehicles and healthcare. Future research should focus on techniques that provide insights into the model's reasoning process. Integrating methods such as attention visualization or saliency maps can offer valuable insights into the regions of interest, enhancing the model's interpretability and making it more trustworthy in critical applications.

9. REFERENCES

1. G. Lewis, "Object detection for autonomous vehicles," 2014.
2. R. Girshick, "Fast R-CNN," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
3. J. Brownlee, "A gentle introduction to object recognition with deep learning," Machine Learning Mastery, vol. 5, 2019.
4. F. Yu et al., "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 2636–2645.
5. P. Bhargava, "On generalizing detection models for unconstrained environments," in Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019, pp. 0–0.
6. L. Jiao et al., "A survey of deep learning-based object detection," IEEE Access, vol. 7, pp. 128837–128868, 2019.

7. S. Ren et al., "Faster R-CNN: Towards real-time object detection with region proposal networks," arXiv preprint arXiv:1506.01497, 2015.
8. J. Redmon et al., "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.
9. A. Geiger et al., "Vision meets robotics: The KITTI dataset," The International Journal of Robotics Research, vol. 32, no. 11, pp. 1231–1237, 2013.
10. P. Sun et al., "Scalability in perception for autonomous driving: Waymo open dataset," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2446–2454.
11. H. Caesar et al., "nuscenes: A multimodal dataset for autonomous driving," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11621–11631.
12. S.-H. Tsang, "Review: Yolov1 — you only look once (object detection)," 2018. [Online]. Available: https://towardsdatascience.com/yolov1-you-only-look-once-object-detection-e1f3ffec8a89. [Accessed: Mar. 15, 2021]. 21 28
13. M. Chablani, "Yolo – you only look once, real time object detection explained," 2017. [Online]. Available: https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006. [Accessed: Mar. 15, 2021].
14. C. Szegedy et al., "Going deeper with convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
15. S. Zhou, J. Zhao, Z. Deng, H. Sun, "Multi-scale object detection in remote sensing imagery with convolutional neural networks," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 145, 2018.
16. Mata, "Faster RCNN in RPN: The Anchor, Sliding Windows, Proposals of Understanding," 2021. [Online]. Available: https://www.programmersought.com/article/31012543832/. [Accessed: Mar. 15, 2021].
17. T. Grel, "Region of Interest Pooling Explained," 2017. [Online]. Available: https://deepsense.ai/region-of-interest-pooling-explained/. [Accessed: Mar. 15, 2017].