2023 3rd International Conference on Electronic Information Engineering and Computer Science (EIECS) | 979-8-3503-1667-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/EIECS59936.2023.10435392

Real Time Object Recognition Based on YOLO Model

Chuhang Zong*
North China Electric Power University (Baoding)
Baoding, China
1911581215@mail.sit.edu.cn

Kehan Meng
High School Attached to Northeast Normal University
Changchun, China

Jingtang Sun
Suzhou Science Technology Town Foreign Language School
Suzhou, China

Qirui Zhou
U-Link college of Suzhou
Suzhou, China

Abstract—**Real-time object recognition is a fundamental task in computer vision with many applications. This paper presents a comprehensive survey of the evolution of the You Only Look Once (YOLO) object detection models, spanning from YOLOv1 to YOLOv8. Each iteration of the YOLO model is examined in detail, elucidating architectural advancements and innovations that have propelled real-time object recognition performance. The survey delves into the strengths and limitations of each YOLO version, highlighting their respective contributions to the field. Additionally, two prominent practical implementations of YOLO models are elucidated, exemplifying the models' efficacy in complex scenarios. The first case study explores YOLO's role in enabling real-time object detection for autonomous driving systems, enhancing safety and situational awareness. The second case study investigates the integration of YOLO models in unmanned aerial vehicles, showcasing their utility in aerial surveillance and reconnaissance. By providing an in-depth exploration of YOLO models and their evolution, this paper equips researchers and practitioners with a comprehensive understanding of real-time object recognition techniques. Furthermore, the analysis of practical applications underscores the tangible impact of YOLO models on cutting-edge technologies. Looking forward, this survey sets the stage for future advancements in real-time object recognition, addressing challenges and opportunities for refining performance in dynamic and complex environments.**

Keywords-real-time object recognition; YOLO models; computer vision; autonomous driving; unmanned aerial vehicles; survey

I. Introduction

“Real-time Object Recognition Based on the YOLO Model” is a significant research topic in the field of computer vision, focusing on the real-time detection and identification of objects in images or video streams. The YOLO (You Only Look Once) model is a deep learning algorithm that efficiently and accurately detects multiple objects in a single image frame. It accomplishes this by segmenting the image into a grid, subsequently making predictions pertaining to bounding boxes and class probabilities within individual grid cells.

In the early stages of computer vision, object recognition methods were inefficient and struggled to handle multiple objects simultaneously [1]. With the advancement of deep learning, researchers began applying Convolutional Neural Networks (CNNs) to object detection tasks, leading to substantial progress. However, some methods still suffered from slow detection speeds [2].

In 2016, Joseph Redmon et al. introduced the YOLO model, which facilitated the conversion of the object detection challenge into a regression-centric task, achieving high-speed object detection. Subsequently, researchers continually improved and optimized the YOLO model, enhancing both detection accuracy and efficiency. Versions such as YOLOv2, YOLOv3, YOLOv4, and others were introduced, and real-time object recognition based on the YOLO model found widespread applications in various fields, including video surveillance, autonomous driving, and robotics. Its efficient detection speed and accuracy make it an ideal choice for real-time applications.

This article provides an overview of the development process and improvements of different versions of the YOLO model, starting with the foundational YOLOv1. Each version's characteristics are introduced, and the advantages of model updates are summarized, with a glimpse into future development directions. Additionally, it briefly highlights the practical applications of the YOLO model in real-time object recognition.

II. Development and Optimization of Different Versions of YOLO Models

The evolution and refinement of various iterations of YOLO models have significantly contributed to the advancement of real-time object recognition in computer vision. This section outlines the progression of different YOLO versions, highlighting their enhancements and improvements over time.

A. YOLOv1

The sequence of algorithms known as R-CNN (comprising R-CNN, SPPNet, Fast R-CNN, and Faster R-CNN) uniformly embrace a dual-stage methodology: firstly, the extraction of region proposals, followed by second-stage tasks encompassing classification and box regression [3]. The central emphasis lies in discerning and pinpointing region proposals. While these techniques yield commendable detection precision, the requirement for an independent network to extract region proposals eventually constrains their speed performance, thereby becoming a velocity bottleneck.

YOLO Base is alternatively referred to as YOLO Version 1 (YOLOv1). This iteration approaches detection through a regression paradigm (Figure 1) [4]. Employing a singular convolutional network, it concurrently forecasts numerous bounding boxes alongside the respective class probabilities tied to these boxes.

Main components： the YOLOv1 takes an input image of fixed size (e.g., 448x448 pixels) as its input. The input image is divided into a grid of cells. The grid size can be adjusted depending on the desired trade-off between speed and accuracy. Common grid sizes are 7x7 or 13x13 [5]. For each cell in the grid, YOLOv1 predicts multiple bounding boxes. These bounding boxes are characterized by their coordinates (x, y) within the cell, width (w), and height (h). Each bounding box also has an associated confidence score, which represents the likelihood of containing an object and is a value between 0 and 1. YOLOv1 also predicts a probability distribution over a predefined set of object classes for each bounding box. This is done using softmax activation, and the number of classes depends on the dataset being used. After making predictions for all cells and bounding boxes, YOLOv1 applies non-maximum suppression to filter out overlapping and low-confidence predictions, resulting in a final list of detected objects.

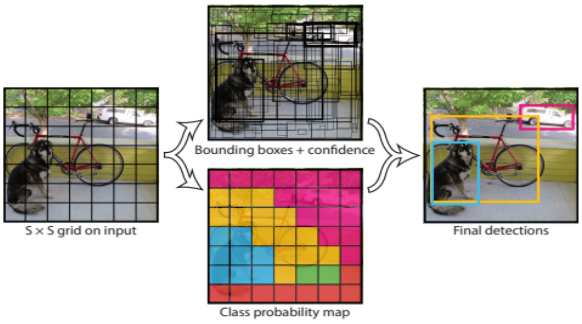

Figure 1. The Model. *(It divides the image into an S × S grid. These predictions are encoded as an S×S×(B 5+C) tensor [4].)*

Detailed theoretical derivation process: YOLOv1 uses a combination of regression loss and classification loss to train the model. The loss function is defined as the sum of two components. First, localization loss can measure how well the predicted bounding box coordinates (x, y, w, h) match the ground truth bounding box coordinates. Second, confidence loss can measures how well the predicted confidence scores match the ground truth, penalizing false positives and false negatives. Third, class loss measures how well the predicted class probabilities match the ground truth class labels.

Loss calculation: The localization loss is typically computed using the mean squared error (MSE) between the predicted bounding box coordinates and the ground truth coordinates. The confidence loss is computed as the MSE between the predicted confidence scores and a binary indicator function that equals 1 if an object is present in the cell or 0 otherwise. The class loss is calculated using cross-entropy loss between the predicted class probabilities and the ground truth class labels.

Total Loss: The total loss is the weighted sum of the localization loss, confidence loss, and class loss. The weights for these components are typically adjusted empirically.

Backpropagation: The gradient of the total loss with respect to the model parameters (neural network weights) is computed using backpropagation.

Optimization and training: The model is updated using an optimization algorithm like stochastic gradient descent (SGD) to minimize the loss. And YOLOv1 is trained on a labeled dataset where each image has annotations for object bounding boxes and class labels.

During inference, the trained YOLOv1 model makes predictions for input images by passing them through the neural network, and then applying non-maximum suppression to the predicted bounding boxes to obtain the final object detections.

YOLO presents two shortcomings: the first pertains to imprecise positioning, while the second concerns a relatively reduced recall rate when contrasted with approaches founded on area suggestions. To illustrate, irrespective of the number of bounding boxes foreseen by a grid cell, said cell exclusively anticipates a singular array of class probability values.

B. YOLOv2

YOLOv2 primarily revolves around enhancing the original YOLOv1 architecture [6]. Additionally, YOLOv2 doesn't adopt an approach of increasing network depth or breadth; instead, it streamlines the network structure. YOLOv2 introduces two notable enhancements: heightened effectiveness and expedited processing. By employing an innovative, multi-scale training technique, the identical YOLOv2 model becomes adaptable to diverse dimensions, providing a convenient balance between processing speed and precision.

YOLOv2 introduced VB layers after each convolutional layer and removed fully connected dropout. By implementing the BN strategy, the mean Average Precision (mAP) was improved by 2%. YOLOv2 fine-tunes the network on ImageNet at a resolution of 448x448 for 10 epochs to adapt it to high-resolution inputs. By using higher resolution inputs, YOLOv2 improves the mean Average Precision by approximately 4%.

To recapitulate, YOLOv2 integrates a multitude of methodologies sourced from alternate object detection approaches, including the adoption of anchor boxes from Faster R-CNN and the incorporation of multi-scale detection akin to SSD. Additionally, YOLOv2 employs numerous network design tricks that allow it to enhance detection accuracy while maintaining speed. The integration of Multi-Scale Training empowers a singular model to accommodate inputs of varying dimensions, thereby facilitating a dynamic equilibrium between velocity and precision.

Firstly, YOLOv2's Darknet19 architecture continues to rely on a VGG-style sequential stacking, which can lead to challenges like gradient vanishing during backpropagation and potential loss of information in feature extraction. Despite the inclusion of a passthrough layer to merge features from two scales, a more comprehensive contextual description is lacking. As a response, later models frequently incorporate residual modules and attention mechanisms within feature extraction, and adopt approaches like FPN or PAN in the intermediate stages to amalgamate features across different scales.

Secondly, YOLOv2's loss function still treats center point and dimensions independently, while treating positional data holistically could be more beneficial. Consequently, subsequent models often replace this with IoU and its derivatives. For category and confidence losses, employing cross-entropy loss is recommended over mean squared error.

C. *YOLOv3*

Significant disparities exist between YOLOv3 and preceding iterations, manifesting in the domains of velocity, accuracy, and class particularity (Table 1) [7]. YOLOv2 and YOLOv3 diverge considerably with regard to precision, swiftness, and structural design.

TABLE I. COMPARISON OF BACKBONES [7]

Backbone	Top-1	Top-5	Bn Ops	BFLOP/s	FPS
Darknet-19	74.1	91.8	7.29	1246	171
ResNet-101	77.1	93.7	19.7	1039	53
ResNet-152	77.6	93.8	29.4	1090	37
Darknet-53	77.2	93.8	18.7	1457	78

Accuracys, billion floating point operations per second, and FPS for various networks.

In the evolution from YOLOv2 to YOLOv3, a significant transition is observed in the backbone feature extractor. While YOLOv2 relied on Darknet-19 for this purpose, YOLOv3 adopts Darknet-53 as its new backbone. Darknet-53, another creation by the YOLO developers Joseph Redmon and Ali Farhadi, boasts a noteworthy advancement. With its 53 convolutional layers, Darknet-53 surpasses the previous Darknet-19 in terms of potency. Moreover, it achieves this heightened capability while also retaining a notable edge in efficiency compared to competing backbone architectures such as ResNet-101 or ResNet-152.

The latest iteration, YOLOv3, incorporates autonomous logistic classifiers and employs the binary cross-entropy loss technique for its class predictions during the training phase. These modifications render the utilization of intricate datasets feasible for the training of the YOLOv3 model.

The disadvantages of YOLOv3 is that it can struggle with detecting small objects in images or videos and training YOLOv3 requires a large and diverse dataset, along with substantial computational resources. Fine-tuning the model and optimizing hyperparameters can be complex and time-consuming. It might prove suboptimal for the utilization of specialized models in scenarios where acquiring extensive datasets could pose a challenge.

D. *YOLOv4*

In April 2020, YOLOv4 was released, sparking widespread discussions in the field of object detection [8]. Following the announcement by Joseph Redmon, the original author of the YOLO series, that he was stepping away from the computer vision domain and that official updates to YOLOv3 would cease, AlexeyAB took up the mantle and continued to improve and develop the YOLO series. Building upon YOLOv3, AlexeyAB released YOLOv4, which gained recognition from Joseph Redmon himself (Figure 2).

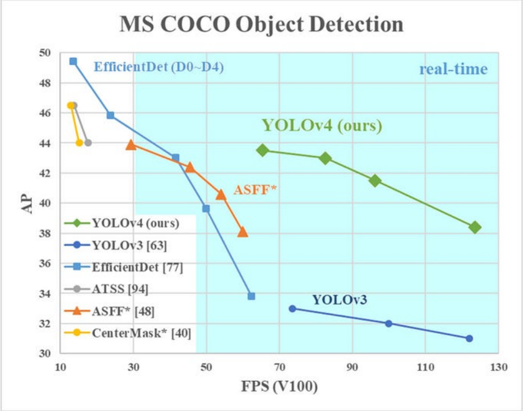

Figure 2. Comparison of the proposed YOLOv4 and other state-of-the-art object detectors [8].

YOLOv4 can be trained and tested using conventional GPUs and provides real-time, high-precision detection results. It maintains performance comparable to EfficientDet while being twice as fast in inference speed. Compared to YOLOv3, YOLOv4 demonstrates a 10% improvement in Average Precision (AP) and a 12% increase in Frames Per Second (FPS).

The optimization contributions made by YOLOv4 can be summarized as follows: It introduces a real-time, high-precision object detection model. YOLOv4 is capable of training a fast and accurate object detector using common GPUs such as the 1080Ti or 2080Ti. During the detector training phase, the model validates the effectiveness of cutting-edge Bag-of-Freebies and Bag-of-Specials techniques.

Additionally, YOLOv4 improves upon state-of-the-art (SOTA) methods to enhance efficiency, making it better suited for single GPU training. This includes enhancements like CBN (Cross-Stage Batch Normalization), PAN (Path Aggregation Network), SAM (Spatial Attention Module), among others.

E. *YOLOv5*

YOLOv5 is a single-stage object detection algorithm released by the company UltraLytics LLC [9]. Compared to YOLOv4, YOLOv5 offers a smaller mean weight file, shorter training time, and faster inference speed while only slightly reducing the average precision of detection. The network architecture of YOLOv5 consists of four main components: the Input End, Backbone, Neck, and Head.

The utilization of the PyTorch framework in YOLOv5 results in a significantly lightweight model size, with the potential for a reduction in model parameters of nearly 90% compared to YOLOv4. In comparison to YOLOv4 implemented using the Darknet framework, YOLOv5 demonstrates a comparable level of accuracy. The computational speed of models in YOLOv3 and YOLOv4 is influenced by their network architectures, with deeper models incurring higher computational costs.

In terms of network architecture, YOLOv5 offers four different scales of network structures: YOLOv5-s, YOLOv5-m, YOLOv5-l, and YOLOv5-x. Among these, YOLOv5-s is the shallowest in terms of network depth and boasts the fastest training speed, while the depth of the other structures increases incrementally. The design of YOLOv5's architecture draws inspiration from the concept of CSPNet (Cross-Stage Partial Network), incorporating CSP structures within the backbone network. Furthermore, distinct variations of CSP structures are strategically employed at different segments of the network, thereby mitigating computational and memory costs.

F. *YOLOv6*

YOLOv6 is a detection framework developed by the Meituan Tech Team, which amalgamates various recent concepts including network architectures, training strategies, testing techniques, quantization, and model optimization [10]. YOLOv6-nano achieves an impressive performance of 35.0% Average Precision (AP) and 1242 Frames Per Second (FPS) on the COCO dataset. Moreover, models of other dimensions also exhibit enhanced precision and efficiency, thus marking significant advancements in both accuracy and computational effectiveness.

YOLOv6 has made the following key contributions: 1. Introducing distinct model architectures tailored for various industrial deployment scenarios, striking a balance between precision and speed. 2. Employing a self-distillation strategy for both classification and regression tasks. 3. Analyzing different strategies for label assignment, loss functions, and data augmentation techniques. 4. Improvements in the quantization approach by leveraging the RepOptimizer optimizer and implementing channel distillation techniques.

These contributions collectively contribute to YOLOv6's advancements in terms of model performance, adaptability to various real-world scenarios, and optimization techniques. And YOLOv6 demonstrates improved accuracy compared to other detectors while maintaining similar latencies.

G. *YOLOv7*

YOLOv7 showcases an impressive balance between speed and accuracy across a wide range, spanning from 5 FPS to 160 FPS [11]. The YOLOv7-E6 object detector (operating at 56 FPS on V100 hardware and achieving a 55.9% Average Precision) demonstrates superior performance compared to both transformer-based detector models like SWIN-L Cascade-Mask R-CNN (9.2 FPS on A100 with a 53.9% AP), surpassing them by a remarkable 509% in speed and 2% in accuracy. Similarly, when pitted against the convolutional-based ConvNeXt-XL Cascade-Mask R-CNN (running at 8.6 FPS on A100 and delivering a 55.2% AP), YOLOv7-E6 excels with a notable 551% speed increase while maintaining a competitive 0.7% edge in accuracy.

Notably, YOLOv7 also outperforms other prominent object detectors such as YOLOR, YOLOX, Scaled-YOLOv4, YOLOv5, DETR, Deformable DETR, DINO-5scale-R50, ViT-Adapter-B, along with numerous others, in terms of both speed and accuracy." (Figure 3)

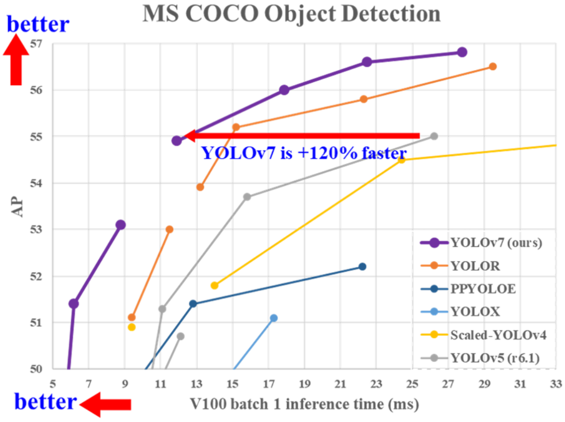

Figure 3. Comparison with other real-time object detectors. *(YOLOv7 achieve state-of-the-arts performance [11].)*

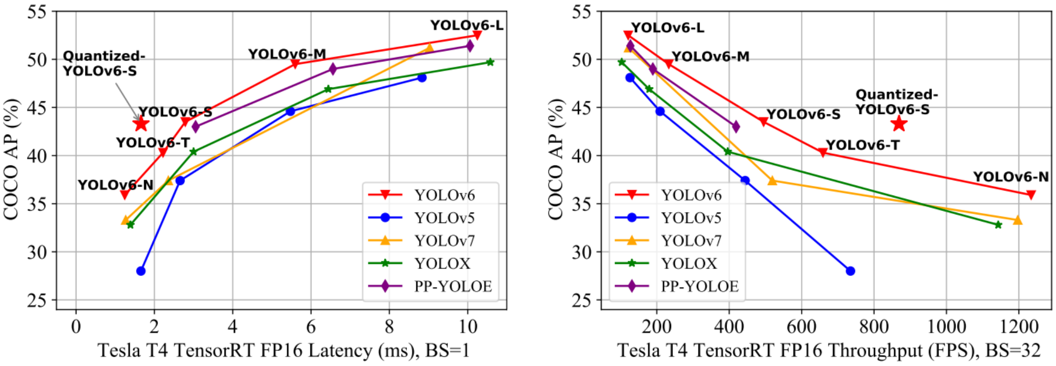

Figure 4. Comparison of state-of-the-art efficient object detectors [10].

H. *YOLOv8*

YOLOv8 is the next major update to YOLOv5, released by Ultralytics on January 10, 2023. At present, it extends its support to an array of tasks encompassing image classification, object detection, and instance segmentation (Figure 4).

YOLOv8 is a state-of-the-art (SOTA) model, building upon the success of previous YOLO versions while introducing new functionalities and enhancements to further elevate performance and flexibility. Notable innovations encompass a novel backbone network, an innovative Anchor-Free detection head, and a new loss function, enabling seamless operation across a range of hardware platforms, spanning from CPUs to GPUs.

The significance of this new technology also lies in the introduction of an entirely new framework. However, at present, this framework is still in its early stages and requires continuous refinement.

III. Examples of Practical Applications of YOLO Models in Real-Time Object Recognition

A. Autonomous Vehicles

YOLO models can be employed in autonomous vehicles to detect pedestrians, vehicles, traffic signs, etc., in real time, aiding the vehicle in making instantaneous decisions.

Numerous contemporary systems that engage in object detection prioritize real-time processing, placing specific demands on computational resources. This becomes particularly crucial when the image capture and processing occur on the same device. This scenario is notably applicable to various autonomous vehicle setups, where the vehicle itself undertakes real-time image capture and analysis to facilitate informed decision-making for its subsequent actions. Within this framework, enhancing the capability of detecting smaller objects carries significant implications. By successfully detecting objects positioned at greater distances from the vehicle, an early identification of such entities becomes feasible. Consequently, this expansion of the detection range significantly enhances the vehicle's situational awareness (Figure 5) [12].

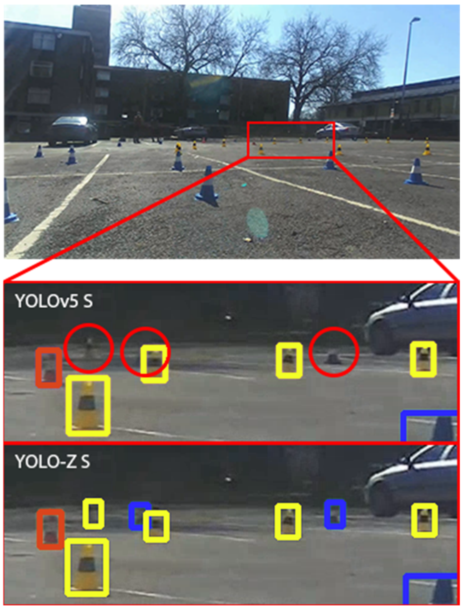

Figure 5. The detection results. *(distant/small-scale object regions in sample images are displayed for YOLOv5S (Top) and YOLO-Z S (bottom). [12])*

The YOLO model's potential to enhance performance in this specific realm holds the promise of further empowering the system. This augmentation equips the system with a more comprehensive and reliable understanding of its surroundings, thereby fostering the ability to execute robust and effective decisions.

B. Drone Applications

Drones can be equipped with YOLO models for real-time detection of forest fires, floods, natural disasters, etc., enabling swift responses and support.

The utilization of YOLO-based target detection technology has yielded notable achievements in the realms of object detection and image recognition. In a study conducted by Khang et al., experiments were carried out on the VisDrone2019 dataset, encompassing 96 videos and 39,988 annotated frames [13]. These trials involved the evaluation of deep learning detectors, and the outcomes underscored YOLO's exceptional prowess in real-time target detection applications. The significance of this lies in the capability to perform object detection and recognition seamlessly on real-time imagery captured by UAV cameras. This convergence effectively transforms the dual steps of UAV data acquisition and computer-based detection into a synchronized process, yielding substantial time savings and elevating operational efficiency.

The advancement in the domain of autonomous target recognition by drones holds the potential to wield a profound impact. This progress stands poised to significantly drive the automation and unmanned operation of drones across diverse industries.

IV. YOLO Model Compared with Other State-of-Art Methods.

YOLO model is known for its high inference speed and has competitive accuracy, especially in detecting objects of various sizes in an image. However, it may have limitations in handling small objects, and it might struggle with crowded scenes.

Next, We will briefly compare the YOLO model with other advanced methods.

A. Faster R-CNN

Faster Region Convolutional Neural Network is a two-stage object detection method. It is accurate but slower compared to YOLO [2].

B. SSD (Single Shot MultiBox Detector)

SSD is another single-shot object detection method that balances speed and accuracy. It divides the image into multiple scales and aspect ratios, making it capable of detecting objects of various sizes.

C. EfficientDet and RetinaNet

EfficientDet is an efficient object detection model that optimizes both accuracy and efficiency. It uses a compound scaling method to balance different model characteristics. RetinaNet combines the speed of single-shot detectors with the accuracy of two-stage detectors. It uses a focal loss function to address class imbalance during training.

D. DETR (Data-efficient Object Transformer):

DETR is a transformer-based object detection model that can handle varying numbers of objects in an image without using anchor boxes. It's better for its impressive performance on object detection tasks.

In conclusion, the choice of the best method for real-time object recognition depends on specific circumstances, computational resources, and accuracy requirements. The most suitable method should be selected based on the unique needs of the situation.

V. Conclusion

In conclusion, this article has provided an extensive overview of the evolution and advancements of the You Only Look Once (YOLO) object detection models. The journey from YOLOv1 to YOLOv8 has showcased remarkable progress in real-time object recognition, addressing various challenges and pushing the boundaries of accuracy, speed, and adaptability.

Through our analysis, it becomes evident that each YOLO iteration has contributed significantly to the field of computer vision, introducing innovative architectural modifications and optimization techniques. These advancements have not only propelled the performance of real-time object recognition but have also enabled practical implementations in diverse applications, thereby solidifying YOLO's relevance and impact.

Furthermore, the presented case studies exemplify the real-world applicability of YOLO models. Their seamless integration into these domains has enhanced safety, navigation, and surveillance capabilities, highlighting their potential to revolutionize industries and improve societal well-being.

Looking ahead, the future holds promising opportunities for the continued development of YOLO models. As the demand for real-time object recognition in complex and dynamic environments grows, YOLO is poised to contribute significantly. Future research endeavors could focus on refining detection accuracy for small and occluded objects, exploring multi-modal sensor fusion for enhanced perception.

The YOLO family of models has witnessed an inspiring journey of progress, making real-time object recognition more efficient and effective. The future holds the promise of further advancements and breakthroughs, cementing YOLO's role as a cornerstone in the realm of computer vision and paving the way for smarter, safer, and more interconnected environments.

As the field of computer vision continues to evolve, YOLO models are positioned to be at the forefront of innovation, shaping the next generation of intelligent visual systems.

References

[1] G.-J. Zhang. Machine vision. Beijing Science Press, 2005, pp109-112.

[2] J. Du, “Understanding of Object Detection based on CNN Family and YOLO”, Journal of Physics, Conference Series, vol.1004, issue.1, 2018.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” In Proceedings of the IEEE conference on computer vision and pattern recognition.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[5] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “CNN features off-the-shelf: an astounding baseline for recognition,” In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2014.

[6] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” arXiv preprint, 2016.

[7] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” arXiv:1804.02767, 2018.

[8] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” arXiv:2004.10934, 2020

[9] G. Jocher, A. Stoken, and J. Borovec, et al.Ultralytics/YOLOv5: V3.1-bug fixes and performance improvements [EB/OL]. https://doi.org/10.5281/zenodo.4154370, 2020.

[10] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, and W. Nie, “Yolov6: A single-stage object detection framework for industrial applications,” arXiv preprint. arXiv:2209.02976, 2022.

[11] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” arXiv:2207.02696, 2022.

[12] A. Benjumea, I. Teeti, and F. Cuzzolin, “YOLO-Z: improving small object detection in YOLOv5 for autonomous vehicles”[EB/OL], 2021.

[13] K. Nguyen, NT. Huynh, PC. Nguyen, KD. Nguyen, ND. Vo, AND TV. Nguyen, “Detecting objects from space: An evaluation of deep-learning modern approaches,” Electronics , 2020.