Pre-Print Version

An Evaluation of Advancements in YOLO Algorithm

Nasreen Jawaid[1*], Najma Imtiaz Ali[2], Kamran Taj Pathan[3], Imtiaz Ali Brohi[4]

***Abstract*- The development and refinement of object detection models have significantly advanced computer vision, with the YOLO (You Only Look Once) framework emerging as a leading method due to its efficiency and real-time processing capabilities. This paper provides a detailed review of YOLO's evolution, from its inception to its most recent iterations. Key improvements in accuracy, speed, and model architecture across different YOLO versions are discussed. The paper also explores YOLO's diverse applications, including autonomous vehicles, surveillance systems, and healthcare, showcasing its adaptability and broad impact. Despite its success, YOLO faces challenges, particularly in balancing speed and accuracy. This review highlights these challenges and identifies potential areas for future research aimed at further optimizing YOLO models.**

***Keywords*— Data analysis; Object detection; YOLO; review**

INTRODUCTION

The You Only Look Once (YOLO) algorithm stands out as a widely recognized and extensively utilized system for its exceptional object identification capabilities [1]. Initially introduced by Redmon et al. in 2015 [2], YOLO has since seen significant contributions from various academics, resulting in subsequent iterations such as YOLO V2, V3, V4, and V5 [3–10], as well as adaptations like YOLO-LITE [11–12]. This study specifically focuses on analyzing the first five iterations of YOLO.

By scrutinizing the primary differences among these five versions, this research considers both their theoretical foundations and practical applications. Understanding the unique objectives, advancements in features, limitations, and interdependencies among these iterations becomes pivotal as they progress. The conclusions drawn from this review hold particular significance for researchers in the field of object detection, especially those who are newly entering the discipline.

[1-2] University of Sindh, Jamshoro
[3-4]Government College University, Hyderabad
Country: Pakistan
Email: nasreenjawaid07@gmail.com

Evolution of Yolo Algorithm

Key Variances (Characteristics)

The YOLO target detection algorithm is characterized by its compact size and rapid computation. Due to its straightforward architecture, this neural network can directly predict both the bounding box coordinates and object categories.

Its speed is ascribed to the fact that YOLO enables real-time video detection by processing images directly through the network to produce the final detection results. YOLO reduces errors in identifying backgrounds as objects by encoding comprehensive information through the use of the global image for detection. YOLO learns generalized traits that are applicable to a variety of fields, demonstrating strong generalization abilities. Although target detection is formulated as a regression problem, the accuracy of detection can be improved. Specifically, YOLO has trouble identifying objects that are grouped together or in close proximity, which results in less-than-ideal performance.

Careful loss function design is necessary to address the need for increased detection efficiency, especially with regard to handling objects of different sizes and positioning errors. Multiple lower sampling layers are used by YOLO to improve target feature learning and detection results.

The original YOLO architecture, featuring 24 convolutional layers and two fully connected layers, was capable of predicting multiple bounding boxes for each grid cell. To select the most accurate bounding box, the algorithm uses a technique known as non-maxima suppression, which identifies the box with the highest Intersection over Union (IoU) with the ground truth [13].

Though it has advantages, YOLO has two main flaws: it is less accurate in positioning and has a lower recall rate than area-based recommendation systems. In order to overcome these drawbacks, YOLO V2 concentrates on enhancing accuracy and recall rate rather than expanding or deepening the network and instead chooses to simplify. Two major improvements are included in YOLO V2: faster processing and better accuracy.

Normalization

This means that each layer's input should be standardized, the convergence rate should be accelerated, losses should be minimized, and mAP should be improved by 2%.

Classifier with High Precision

The original YOLO network uses 224 x 224 pixel images for training and 448 x 448 pixel images for detecting attacks. During this transition, the network is modified from an image classification model to a detection model. The pre-training procedure of YOLO V2 is divided into two phases. First, the network is trained using 224 × 224 pixel pictures for 160 epochs. Subsequently, the network undergoes 10 further epochs of fine-tuning using images of 448 × 448 pixels.

Fine features

The key point is the inclusion of a layer that connects the 26 x 26 feature map from the previous layer with the 13 x 13 feature map of the current layer, as the latter excels at predicting large objects. However, predicting smaller objects can be challenging since they may be lost after passing through multiple convolution and merging layers. Therefore, it is crucial to integrate the operations of the preceding layer, particularly for detecting larger objects.

Training at Multiple Scales

This training method enables a single network to detect images at different resolutions. Although training may slow down with larger input sizes, it speeds up with smaller ones. Multi-scale training enhances accuracy, achieving an optimal balance between precision and speed.

Darknet-19

YOLO's training network is built on GoogleNet. In a straightforward comparison of GoogleNet with VGG16, the author demonstrates that GoogleNet has a higher computational efficiency (8.25 billion operations compared to 30.69 billion operations). On ImageNet, GoogleNet performs marginally less accurately (88% vs. 90%). The author uses Darknet-19, a novel categorization model, as the main network for YOLO V2.

Table 6 presents the completed network architecture. It takes just 5.58 billion operations to run Darknet-19. Darknet-19 includes 19 convolutional layers and five max pooling layers, in contrast to GoogleNet in YOLO V1, which has 24 convolutional layers and two fully connected layers. Darknet-19 hence employs fewer convolutional layers and operations, greatly increasing the computing efficiency of YOLO. Lastly, the fully linked layer for prediction is replaced by an average pooling layer.

Classification and Training

Pre-training on ImageNet, which consists of two primary steps, is part of the classification training procedure. With 160 training epochs, a 224 x 224 input image size, and an initial learning rate of 0.1, the ImageNet dataset is used. During training, common data augmentation methods like rotation, random cropping, and chroma and brightness adjustments are used.

The network is then fine-tuned: with an input size of 448 x 448, all parameters stay the same, with the exception of the epoch and learning rate. In this stage, the network goes through ten more training cycles with the learning rate set to 0.001. The findings show that the top-1 and top-5 accuracy values are 76.5% and 93.3%, respectively, following fine-tuning.

By contrast, Darknet-19 attains a top-1 accuracy of 72.9% and a top-5 accuracy of 91.2% when using the original training methodology.

Detection Training

First, a modification is made to the YOLO V3 network that involves removing the final convolution layer. Three convolution layers, each with 1024 filters, are added in its stead. There are 11 convolution layers that come before each of these convolution layers. YOLO V3 assigns category probabilities to each box, with each box corresponding to a specific category probability rather than a grid, in contrast to YOLO V2, where category probabilities for two boxes in a cell are the same.

YOLO V3 introduces two major improvements over YOLO V2: using multi-scale features for object detection and changing the basic network architecture. YOLO V3 uses three prior boxes for each position and feature graphs with three scales (416 x 416 input). Nine previous boxes are acquired and dispersed throughout the three scale feature maps using K-means. Smaller prior boxes are used in larger-scale feature maps.

Moreover, YOLO V3 uses a residual model for feature extraction, as opposed to YOLO V2's Darknet-19 model. There are notable improvements and a greater emphasis on data comparison in the YOLO V4 style. To achieve optimal performance, it combines multiple components, which can be summed up as follows: YOLO V4 = CSP Darknet53 + SPP + Pan + YOLO V3. The development of an effective and potent target detection model, the confirmation of the impact of cutting-edge bag-of-freebies and bag-of-specials techniques during detector training, and the improvement of these techniques for effectiveness and suitability in single GPU training are among the major contributions.

Changes to anchor point responsibilities are introduced in YOLO V4, where multiple anchor points are now accountable for a single ground truth instead of just one anchor point. This modification reduces the imbalance between positive and negative samples by raising the selection ratio of positive samples. YOLO V4 solves bounding box containment issues by using the CIOU (Complete Intersection over Union) loss function, which removes grid sensitivity and quickly converges.

Though there are some concerns regarding its lack of innovation when compared to YOLO V4, YOLO V5 offers advantages in terms of user-friendliness, code readability, and ease of configuration, especially when used with the PyTorch framework. All things considered, YOLO V4 is acknowledged for offering a faster and more accurate advanced detector than current alternatives, making it a promising benchmark for future research and development.

Training data can now be enhanced in batches with YOLO V5 thanks to the data loader. Scaling, adjusting color space, and enhancing the mosaic are all part of this process. Mosaic enhancement works well for handling small object problems during model training. Although there is disagreement regarding the nomenclature and application of YOLO V5, which is still under development, there have been advancements from YOLO V1 to V5, with each version bringing new features like anchor addition, multi-scale detection, SPP, MISH activation function, data enhancements like mosaic/mixup, and the GIOU loss function. YOLO V5 adds more data enhancements, includes the Hardswish activation function, and permits even more flexible control over model size.

Connection

Multi-scale detection is introduced by YOLO V3, as a result of YOLO and YOLO V2's inability to detect small targets. YOLO V3 distinguishes itself as a proficient heir to its forerunners. Throughout the entire process, YOLO V4 methodically arranged and tested every possible optimization, determining the most efficient configurations. Notably, YOLOv4 improves YOLOv3's Average Precision (AP) and Frames Per Second (FPS) by 10% and 12%, respectively, while operating at twice the speed of EfficientNet with similar performance [15]. From 10+M to 200+M models, YOLO V5 provides controllability and impressive performance, even in its smaller model. Although the general network structures of YOLO V3 and YOLO V5 are similar, their main objective is to detect objects in three distinct sizes.

Insights from Publicly Available Data

Using data that is readily available to the public, this section offers a succinct summary of the YOLO versions. The YOLO algorithm, which debuted in 2015, brought a revolutionary method to object detection. YOLO demonstrated competitive performance but also revealed room for improvement, in contrast to previous approaches that modified classifiers for detection [14]. Subsections one and two provide an overview of algorithmic trends and community insights, respectively. Both sections show how YOLO is a dynamic framework that is updated continuously using both text and numerical data. The open dataset on GOOGLE (www.google.com) is the source of all the data presented; noise concerns prevented the inclusion of the YOLO V1 results in this section.

Patterns

To highlight the patterns, the publication data has been compiled in this section. The quantity of scholarly research papers for each version is shown in Table 1. The split indicates that 2019 and 2020 had a significant increase in the number of research articles published.

Notably, researchers have focused a great deal of attention on the YOLO V3 and V2 versions, though this observation may be influenced by the temporal dimension. The V4 and V5 versions, on the other hand, have relatively smaller numbers, which reflects their recent arrival in the field.

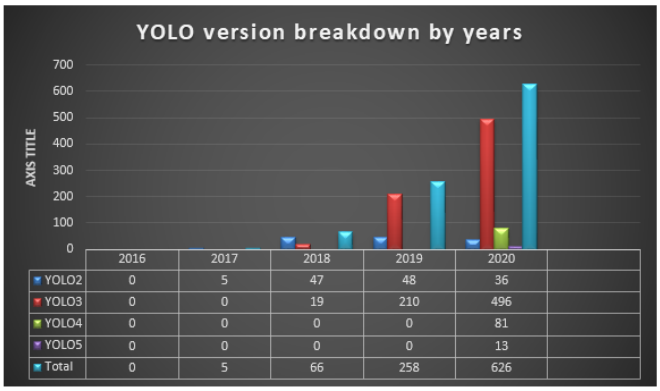


Fig.1. Overview of YOLO Versions by Year

Figure 2 depicts the trends over time in web search performance, covering news search, image search, and YouTube search. The scale is relative, where 100 signifies peak popularity and zero represents the lowest level of interest. For example, a value of 50 indicates that the term is half as popular as its peak. A value of zero means there is either insufficient data or no interest in the topic.

The figure reveals that versions V2 and V3 have generally maintained higher popularity over time. However, after April 2020, there is a clear rise in the popularity of versions V4 and V5. This trend aligns with the numerical findings shown in Table 1.

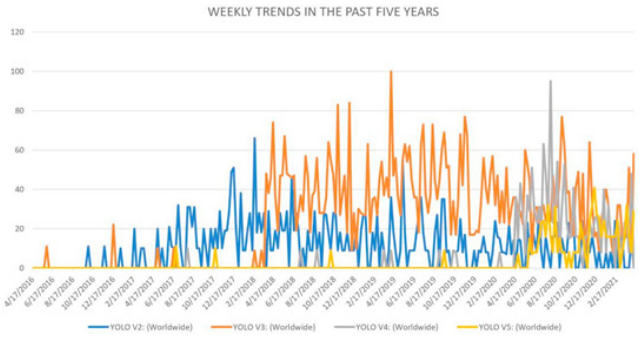


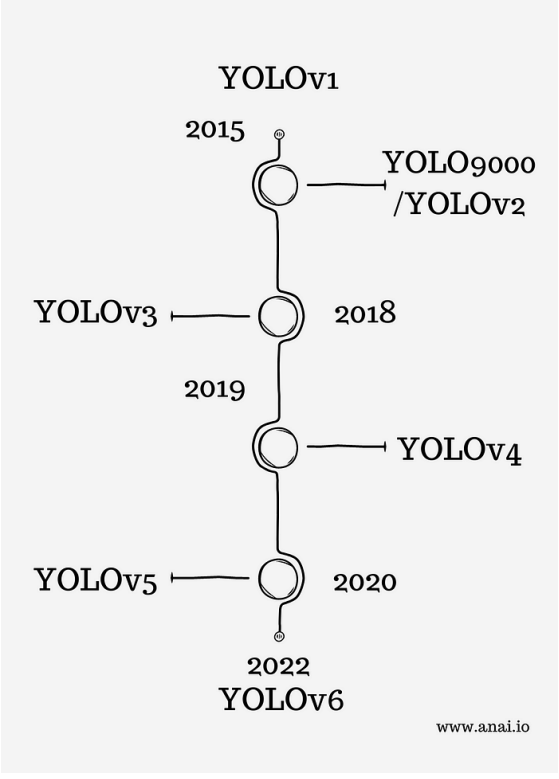
Fig.2. A five-year deep dive into weekly usage trends

Comparison

For comparison between the three versions of YOLO networks, is presented in Table 1 their highlights and disadvantages.

Table 1 – Comparison between three versions of YOLO network: source [16], [17], [18].

Networks	Highlights	Disadvantages
YOLO [2]	First one-stage architecture; Can provide real-time detection;	The accuracy obtained was worst than the time state-of-the-art; Difficulty detecting small objects;
YOLOv2 [3]	Faster and more accurate than its first version;	Difficulty detecting small objects; Low accuracy in comparison to the current state-of-the-art;
YOLOv3 [4]	Better performance than the previous two versions of the YOLO family; Better at small object detection;	Slower than the current state-of-the-art



Yolo 6:

The latest algorithm in the lengthy line of YOLO algorithms is called MT-YOLOv6. It has demonstrated significantly improved and cutting-edge performance on object detection tasks, much like the earlier iterations [19].

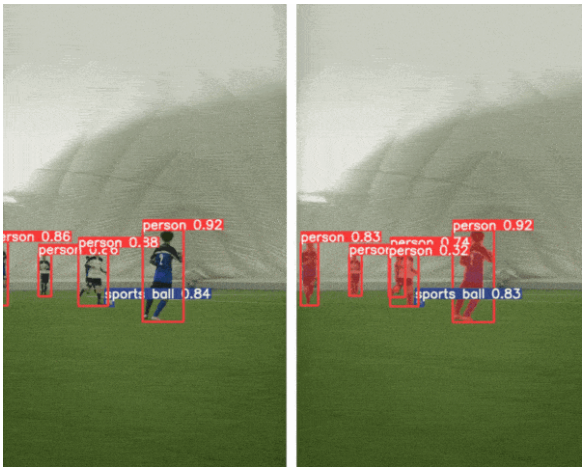
As more people contribute and add new features and stability to the model, the project will continue to be worked on and developed continuously.

The reason MT-YOLOv6 is called YOLOv6 and is not included in the official YOLO series is that its creators describe it as the next generation of YOLO models, drawing inspiration from one-stage YOLO algorithms.

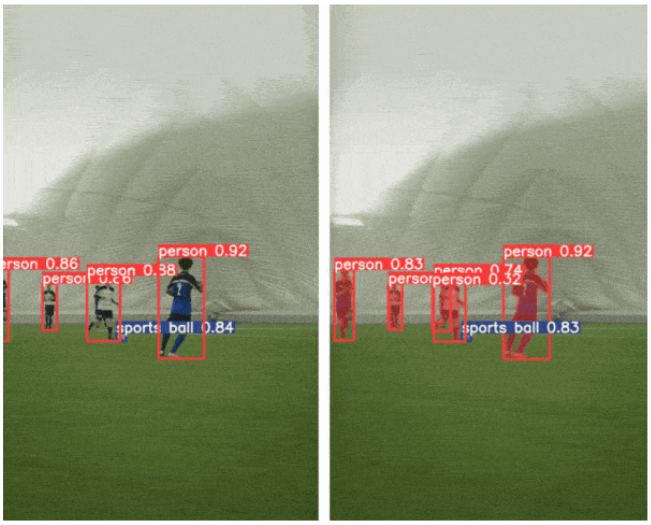
YOLOv7: The Most Powerful Object Detection Algorithm

The most recent official version of YOLO, developed by the original architects of the YOLO architecture, is called YOLOv7. It is anticipated that a significant number of commercial networks will jump straight from YOLOv4 to v7 [20].

YOLOv8: Algorithm for Detecting Small-Size Objects Based on Camera Sensor



Conventional camera sensors observe the world through human eyes. However, human cognition is limited and the human eye becomes fatigued when viewing targets of various sizes for extended periods of time in complex scenes. This greatly reduces efficiency and frequently results in judgment errors. One crucial piece of technology for determining the target category in a camera sensor is target recognition. This paper proposed a small size target detection algorithm for specific



[20]

scenarios to address this issue. Its advantage is that this algorithm can guarantee that the detection accuracy of each size is not less than the current algorithm, in addition to having higher precision for small size target detection. This paper proposed a novel down sampling technique that may better preserve the context feature information. An enhancement was made to the feature fusion network to better integrate deep and shallow data [21].

A novel network architecture was suggested as a practical means of enhancing the model's detection precision. It outperforms YOLOX, YOLOXR, YOLOv3, scaled YOLOv5, YOLOv7-Tiny, and YOLOv8 in terms of accuracy. In this experiment, three reliable public data sets were used: a) DC-YOLOv8 is 2.5% more accurate than YOLOv8 on Visdron data sets (small size targets). b) DC-YOLOv8 is 1% more accurate than YOLOv8 on Tiny person data sets (minimum size targets). On the Normal size target data sets from PASCAL VOC2007, DC-YOLOv8 outperforms YOLOv8 by 0.5%.

Comparison:

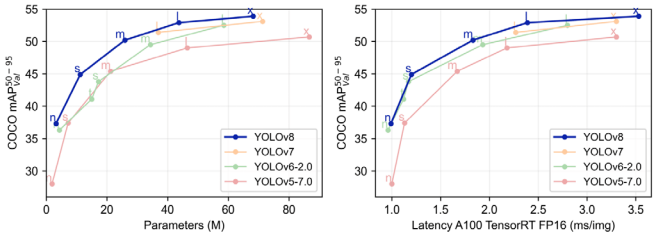
For comparison between the three versions of YOLO networks, is presented in Table 1 their highlights and disadvantages.

Table 1 – Comparison between three versions of YOLO network: source [16], [17], [18].

Networks	Highlights	Disadvantages
YOLO [2]	First one-stage architecture; Can provide real-time detection;	The accuracy obtained was worst than the time state-of-the-art; Difficulty detecting small objects;
YOLOv2 [3]	Faster and more accurate than its first version;	Difficulty detecting small objects; Low accuracy in comparison to the current state-of-the-art;
YOLOv3 [4]	Better performance than the previous two versions of the YOLO family; Better at small object detection;	Slower than the current state-of-the-art

Comparing YOLOv8 to YOLOv7, YOLOv6 to YOLOv5

The YOLOv8 models appear to perform significantly better than the earlier YOLO models right away. In addition to YOLOv5 models, YOLOv8 models outperform YOLOv7 and YOLOv6 models as well.



With an equivalent number of parameters, all YOLOv8 models have better throughput when compared to other YOLO models trained at 640 image resolution.

Overall Comparison:

Performance Comparison of YOLOv8 vs YOLOv5

Model Size	Detection*	Segmentation*	Classification*
Nano	+33.21%	+32.97%	+3.10%
Small	+20.05%	+18.62%	+1.12%
Medium	+10.57%	+10.89%	+0.66%
Large	+7.96%	+6.73%	0.00%
Xtra Large	+6.31%	+5.33%	-0.76%

*Image Size = 640 *Image Size = 224

Object Detection Performance Comparison (YOLOv8 vs YOLOv5)

Model Size	YOLOv5	YOLOv8	Difference
Nano	28	37.3	+33.21%
Small	37.4	44.9	+20.05%
Medium	45.4	50.2	+10.57%
Large	49	52.9	+7.96%
Xtra Large	50.7	53.9	+6.31%

*Image Size = 640

Instance Segmentation Performance Comparison (YOLOv8 vs YOLOv5)

Model Size	YOLOv5	YOLOv8	Difference
Nano	27.6	36.7	+32.97%
Small	37.6	44.6	+18.62%
Medium	45	49.9	+10.89%
Large	49	52.3	+6.73%
Xtra Large	50.7	53.4	+5.33%

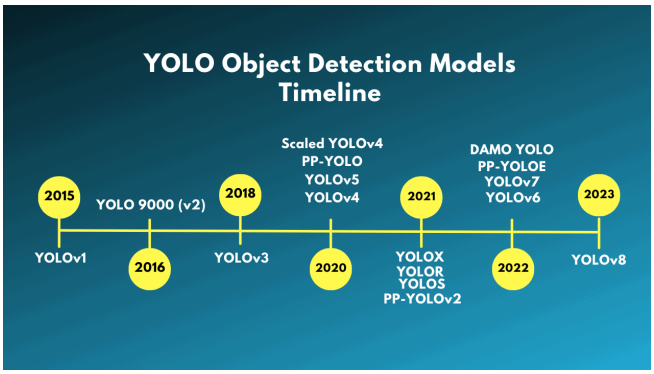
*Image Size = 640

Image Classification Performance Comparisc (YOLOv8 vs YOLOv5)

Model Size	YOLOv5	YOLOv8	Difference
Nano	64.6	66.6	+3.10%
Small	71.5	72.3	+1.12%
Medium	75.9	76.4	+0.66%
Large	78	78	0.00%
Xtra Large	79	78.4	-0.76%

The YOLOv8 Object Detection Model's Evolution

This graphic illustrates the development of YOLOv8 and the timeline of YOLO object detection models.



CONCLUSION

This study conducted a thorough evaluation of the advancements in the YOLO (You Only Look Once) algorithm family, comparing iterations from YOLOv1 to the latest YOLOv8. By meticulously examining performance metrics, model accuracy, and computational efficiency, the research aimed to offer valuable insights into the evolution and effectiveness of YOLO algorithms in object detection [22-24].

The comparative analysis revealed significant improvements with each iteration of YOLO. YOLOv1 pioneered real-time object detection but faced challenges with localization accuracy. Subsequent versions like YOLOv2 and YOLOv3 made notable strides in overcoming these issues while enhancing overall performance. YOLOv4 introduced advancements in model architecture and training strategies, resulting in improved detection accuracy and efficiency.

Exploration of Ultralytics YOLOv8 underscored ongoing efforts to advance object detection algorithms. Through optimized model architecture, refined training methodologies, and integration of advanced technologies, YOLOv8 emerges as a promising advancement towards achieving real-time, precise object detection across diverse applications.

However, the selection of the most suitable YOLO version should consider specific application requirements, balancing factors such as accuracy, speed, and resource utilization. Moreover, ongoing research and development in this field are likely to drive further refinements and innovations in YOLO algorithms, addressing current challenges and expanding possibilities for real-world object detection.

In summary, the YOLO algorithm family has undergone significant evolution, with each iteration building upon the strengths of its predecessors. This research contributes to understanding the complexities and trade-offs involved in choosing the optimal YOLO version based on specific application needs, setting the stage for future advancements in the dynamic field of object detection.

REFERENCES

[1] Sultana, F., Sufian, A., & Dutta, P. (2020). A review of object detection models based on convolutional neural network. Intelligent computing: image processing based applications, 1-16.

[2] Zhiqiang, W., & Jun, L. (2017, July). A review of object detection based on convolutional neural network. In 2017 36th Chinese control conference (CCC) (pp. 11104-11109). IEEE.

[3] Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11), 3212-3232.

[4] Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo algorithm developments. Procedia Computer Science, 199, 1066-1073.

[5] Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R., & Menotti, D. (2018, July). A robust real-time automatic license plate recognition based on the YOLO detector. In 2018 international joint conference on neural networks (ijcnn) (pp. 1-10). IEEE.

[6] Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., & Liang, Z. (2019). Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Computers and electronics in agriculture, 157, 417-426.

[7] Jamtsho, Y., Riyamongkol, P., & Waranusast, R. (2021). Real-time license plate detection for non-helmeted motorcyclist using YOLO. Ict Express, 7(1), 104-109.

[8] Han, J., Liao, Y., Zhang, J., Wang, S., & Li, S. (2018). Target fusion detection of LiDAR and camera based on the improved YOLO algorithm. Mathematics, 6(10), 213.

[9] Lin, J. P., & Sun, M. T. (2018, November). A YOLO-based traffic counting system. In 2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI) (pp. 82-85). IEEE.

[10] Lu, J., Ma, C., Li, L., Xing, X., Zhang, Y., Wang, Z., & Xu, J. (2018). A vehicle detection method for aerial image based on YOLO. Journal of Computer and Communications, 6(11), 98-107.

[11] Huang, R., Pedoeem, J., & Chen, C. (2018, December). YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. In 2018 IEEE international conference on big data (big data) (pp. 2503-2510). IEEE.

[12] Gong, B., Ergu, D., Cai, Y., & Ma, B. (2020). A method for wheat head detection based on yolov4.

[13] Jamtsho, Y., Riyamongkol, P., & Waranusast, R. (2020). Real-time Bhutanese license plate localization using YOLO. ICT Express, 6(2), 121-124.

[14] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[15] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

[16] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[17]Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).

[18] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[19] Revca Technologies. (2022, January 15). YOLOv6 Explained in Simple Terms. Medium. https://revca-technologies.medium.com/yolov6-explained-in-simple-terms-c46a0248bddc

[20]Learn OpenCV. (2022, February 8). Ultralytics YOLOv8 Explained. Learn OpenCV. https://learnopencv.com/ultralytics-yolov8/

[21] Hussain, M. (2023). YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. Machines, 11(7), 677.

[22] 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.

[23] M. Hussain, "Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing

and industrial defect detection," Machines, vol. 11, no. 7, p. 677, 2023.

[24]Ultralytics, "YOLOv8—Ultralytics YOLOv8 Documentation." https://docs.ultralytics.com/models/yolov8/, 2023. Accessed: January 7, 2024