ISSN 2616-5910 Vol. 6, Issue 1: 130-136, DOI: 10.25236/AJALS.2025.060118

BS-Yolov8n: An Improved Yolov8n Network for Tomato Detection at Different Ripeness Degrees in Complex Greenhouse Environments

Zhengdong Li[1], Yu Wang[1,a,*], Minghua Han[2], Zhou Zheng[1]

[1]*College of Information Engineering, Nanjing University of Finance and Economics, Nanjing, China*
[2]*Changshu Binjiang Agricultural Technology Co., Ltd., Suzhou, China*
[a]*yuwang@nufe.edu.cn*
**Corresponding author*

Abstract: *With the advancement of artificial intelligence, computer vision has become a widely adopted method to replace human visual observation. However, the complexity of the greenhouse tomato-growing environment poses significant challenges in using computer vision to quickly and accurately assess the ripeness of tomatoes. In order to solve these problems, we incorporate SPD-Conv and BoTNet to enhance the YOLOv8n network's performance in feature extraction and target recognition capabilities in greenhouse tomato-growing environments. In simulations, we compare the performance of YOLOv8n with that of BS-YOLOv8n. Empirical findings demonstrate that the proposed BS-YOLOv8n performs better than YOLOv8n in both the accuracy and the response speed of tomato recognition in complex greenhouse environments.*

Keywords: *Fruit Detection; Tomato; Deep Learning; Production Forecasts*

1. Introduction

Tomato, as an easy-to-grow and common vegetable, holds a prominent position in global agricultural activities. For tomato cultivation, real-time monitoring of the growth and development process, and ultimately prediction of its yield is crucial. This not only provides a reliable production projection to growers, but also helps them to make effective adjustments to their planting methods and marketing strategies[1].

Traditional production forecasting methods are mainly based on biological models of tomato growth. To simulate photosynthesis and respiration accurately, the modeler must gather extensive environmental data and also the precise planting metrics. Validation of the model requires the destructive weighing of the dry weight of the fruit[2]. Meanwhile, machine learning breakthroughs have enabled the widespread application of computer vision-based artificial intelligence across various areas, including precise agriculture and smart farming. This provides growers with new ways to predict crop yields[3]. Image detection, as one important learning technology, has become a popular way of collecting crop growth information. The convenience brought by integrating computer vision technology into the field of tomato yield prediction is mainly reflected in several levels: first, it can accurately detect the ripening state of fruits; second, it can accurately determine the growing stage of plants; In addition, it can effectively identify the lack of water and pigmentation of plants, as well as timely detect the pests and diseases and other problems. Among the tasks related to computer vision, fruit detection is closely associated with yield prediction, which has become the top priority of research in this field[4].

In the context of rapid development of machine learning and deep penetration into the field of visual image processing, target detection algorithms represented by deep learning are becoming the core technical support for smart agriculture research and application. The most popular methods include R-CNN[5], Faster R-CNN[6], Mask R-CNN[7], SDD[8] and YOLO[9] series. YOLO series are in the rapid iteration, while maintaining the fast speed characteristics, and its accuracy is also gradually improved. Wang et al. [10] introduced an improved version of Mask R-CNN for the recognition and segmentation of apples at three distinct stages of ripeness within an orchard setting. Wang Z et al.[11] presented an enhanced version of Faster R-CNN for the recognition and detection of tomatoes in greenhouse environments. Although they both work to effectively detect fruit, both networks are structured as two-layer networks, which leads to the problem of training speed for accuracy. The SSD and YOLO

ISSN 2616-5910 Vol. 6, Issue 1: 130-136, DOI: 10.25236/AJALS.2025.060118

architectures, as representative single-stage frameworks, exhibit superior processing speed in comparison to two-stage networks. Liu ell al.[12] used an improved YOLOv3 for tomato recognition model to enable the model to correctly recognize yellow tomatoes in a shaded environment. Yan et al.[13] improved YOLOv5 for detecting apple fruits based on the problem that the fruits are easily occluded and the fruits directly overlap each other. Xue et al.[14] proposed an enhancement of YOLOv2 for unripe mango identification, and the model was mainly used to overcome the difficulty of detecting mangoes that are occluded or overlapped. Tang et al.[15] improved YOLOv7 to include an attention mechanism for detecting plum fruits in complex environments. The training images in some studies are usually close-up images, which is very difficult to get under greenhouse environment. So, it not only needs to handle the issue of overlapping occlusion at close range, but also the challenge of detecting small targets at larger distances.

In response to this limitation, an enhanced yolov8n network, referred to as bs-yolov8n, is proposed to address the challenges of occlusion, small-scale target detection, and low-resolution image resolution commonly encountered in complex greenhouse environments. We also apply the method to detect tomato fruits in complex greenhouse environments, and categorize the images into three ripeness levels, aiming to determine the developmental stage of tomato fruits with the help of webcams. The key contributions of this study can be summarized as follows:

(1) SPD-Conv is employed to transfer spatial information from the feature map to the depth dimension, thereby minimizing information loss and enhancing detection performance, particularly for low-quality images and small targets.

(2) BoTNet is incorporated into the backbone network to integrate a multi-head self-attention mechanism, thereby enhancing the model's overall performance.

2. Methods

2.1 YOLOv8

YOLOv8[16] is a prominent version in the YOLO series, and since it ensures that the accuracy is still good while being fast, this network is widely utilized for multiple tasks, including object detection, image categorization, and instance segmentation.

The YOLOv8 architecture incorporates three fundamental modules: a backbone network, a feature fusion neck, and a detection head. The backbone part, CSP-Darknet53[17], serves as the primary feature extractor that processes input images to generate hierarchical feature representations. CSP-Darknet53 incorporates several crucial modules, including convolutional layers (Conv), enhanced CSP bottleneck implementation utilizing two convolutional layers(C2f) and efficient spatial pyramid pooling-fast variant (SPPF). The C2f module is an important innovation that differentiates YOLOv8 from other networks, and its function is to learn residual features, aiming to achieve lightweighting while maintaining the gradient flow information. SPPF, the Spatial Pyramid Pooling-Fast module, performs multi-scale pooling and transforms variably-sized feature maps into dimensional-fixed feature representations, improving spatial information aggregation.

The Neck of YOlov8 is the Path Aggregation Network (PAN)[18], which is conceived with the innovative approach of incorporating a bottom-up pyramid structure alongside the traditional top-down architecture of a Feature Pyramid Network (FPN).

Head is responsible for the final object detection prediction. YOLOv8 employs an"Anchor-Free" approach, removes the reliance on predefined anchor boxes, and incorporates a decoupled head structure to isolate the processing of classification and regression tasks. This separation enhances both efficiency and accuracy, allowing for more precise predictions of object categories and bounding box coordinates. The decoupling of these tasks streamlines the process, reducing computational complexity while improving performance. Additionally, the head processes feature maps at multiple resolutions, enabling YOLOv8 to effectively detect objects of varying sizes and complex spatial configurations with high precision.

In this study, we focus on leveraging YOLOv8's object detection capabilities and select the n-scale variant for training. This model architecture is designed to be lightweight while maintaining high detection accuracy, making it suitable for a wide range of operational scenarios.

ISSN 2616-5910 Vol. 6, Issue 1: 130-136, DOI: 10.25236/AJALS.2025.060118

2.2 SPD-Conv

SPD-Conv (Spatial Depth Transformation Convolution)[19] is an innovative convolutional neural network (CNN) building block designed to mitigate the performance degradation of traditional CNNs when dealing with low-resolution images and small objects. The principal source of these issues stems from the loss of fine-grained information attributed to the application of stride-based convolutions and pooling layers in traditional CNN architectures. And SPD-Conv is designed to replace the stride-based convolutional and pooling layers in traditional CNN architectures. The most important layer, space-to-depth (SPD) layer, is introduced which is responsible for down sampling the channel dimensions of the feature maps while retaining critical information.

Specifically, the SPD layer expands the incoming image or feature map tensor of the previous layer by expanding it in depth according to a set multiple, assuming that the incoming image or feature map tensor scale is S × S × C1, and we can thus slice and dice the sub-elements of X as follows:

$f_{0,0}$ = X[0: S: scale,0: S: scale], $f_{1,0}$ = X[1: S: scale,0: S: scale], ..., $f_{scale-1,0}$ = X[scale−1: S: scale,0 :S: scale];

$f_{0,1}$ = X[0: S: scale,1: S: scale], $f_{1,1}$, ..., $f_{scale-1,1}$ = X[scale − 1: S: scale,1: S: scale];

...

$f_{0,\ scale-1}$ = X[0: S: scale, scale−1: S: scale], $f_{1,\ scale-1}$, ..., $f_{scale-1,\ scale-1}$ = X[scale−1: S: scale, scale−1: S: scale].

Where scale is an adjustable tangent scale parameter and $f_{x,\ y}$ are the tangent sub-feature maps of X'. Next, we connect these sub feature maps along the channel-wise, yielding composite feature representation X' with the desired spatial dimensions that decreases by a scaling factor and a channel dimension that increases by a scaling factor, i.e., $\frac{S}{scale} \times \frac{S}{scale} \times scale^2 C_1$.

After the SPD layer, SPD-Conv uses a non-step-size (step-size of 1) convolutional layer. This helps in extracting important features by utilizing the information in the increased channels followed by reducing the channel cardinality.

2.3 BoTNet

BoTNet[20], or Bottleneck Transformers for Visual Recognition, a collaborative innovation between UC Berkeley and Google Research teams, designed to combine the strengths of Convolutional Neural Networks (CNNs) and Transformer[21] models for visual recognition tasks. Through synergistic integration of CNN's hierarchical feature learning and transformer-style global context-awareness enabled by the self-attention mechanism in Transformers, BoTNet leverages the complementary strengths of both architectures. This hybrid approach allows BoTNet to outperform traditional CNNs and self-attention models individually, delivering 84.7% accuracy on ImageNet and demonstrating the synergistic potential of CNN-Transformer fusion.

The BoTNet architecture strategically substitutes the conventional 3×3 convolutional layers in ResNet's final three residual blocks with global multi-head self-attention (MHSA) modules. While incorporating self-attention contributes to higher computational complexity and increased memory usage, BoTNet addresses these challenges by strategically placing self-attention modules in the final bottleneck layers. Each bottleneck originally contains a 3×3 convolution, which is substituted by MHSA, enabling multi-scale feature learning through hierarchical receptive field adaptation. In the first bottleneck, where the 3×3 convolution uses a stride of 2 and MHSA lacks stride support, BoTNet utilizes 2×2 average pooling for down sampling. The architecture of BoTNet is shown in Figure 1.

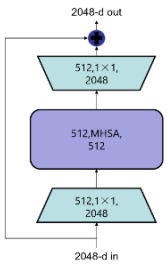

Figure 1: Structure of the bottleneck transformer.

ISSN 2616-5910 Vol. 6, Issue 1: 130-136, DOI: 10.25236/AJALS.2025.060118

2.4 BS-YOLOv8n

As deep learning has rapidly advanced, the YOLO network family have seen widespread adoption across various detection tasks in agriculture. In this study, we select YOLOv8n developed by Ultralytics as the baseline network in order to balance the relatively fast detection speed with high target detection accuracy for complex environments. To enhance the model's efficacy in handling the intricate dynamics of the greenhouse environment, the improved YOLOv8n network, BS-YOLOv8n, is presented in Figure 2.

In terms of model improvement, in order to reduce the information loss when acquiring features for the backbone network, we include several SPD-Conv modules, which are primarily used to convert the input feature maps from spatial scale to depth, thereby reducing information loss. In addition to this, BoTNet is integrated into the terminal layer of the backbone network to extract salient feature representations in the feature maps through its self-attention mechanism, which further improves model's performance in detecting tomatoes.

3. Experimentation and discussion

The experiment is carried out on a Windows 11 x64 platform with a 14th generation Intel i7-14700 KF CPU and an NVIDIA GTX4080 SUPER GPU. The programming environment is created using a virtual environment created by Ana conda3 with python3.12, Cuda12.1, and the network is built under a pytorch 2.2.2 framework.

The training parameters are configured as follows. The training parameters are as follows: the input image size is 640×640 pixels, with a batch size of 25. The model is trained for 400 epochs, using an Intersection over Union (iou) threshold of 0.5 for both training and testing. To prevent overfitting, early stopping is implemented, halting training if the mean average precision (mAP) fails to improve significantly over 100 consecutive iterations. Comparative ablation studies demonstrate the incremental performance gains attributable to each architectural modification in BS-YOLOv8n. Finally, comparing BS-YOLOv8n against established detectors including YOLOv7, YOLOv6, YOLOv5s, YOLOv5n, and YOLOv3 to validate its superior detection capability.

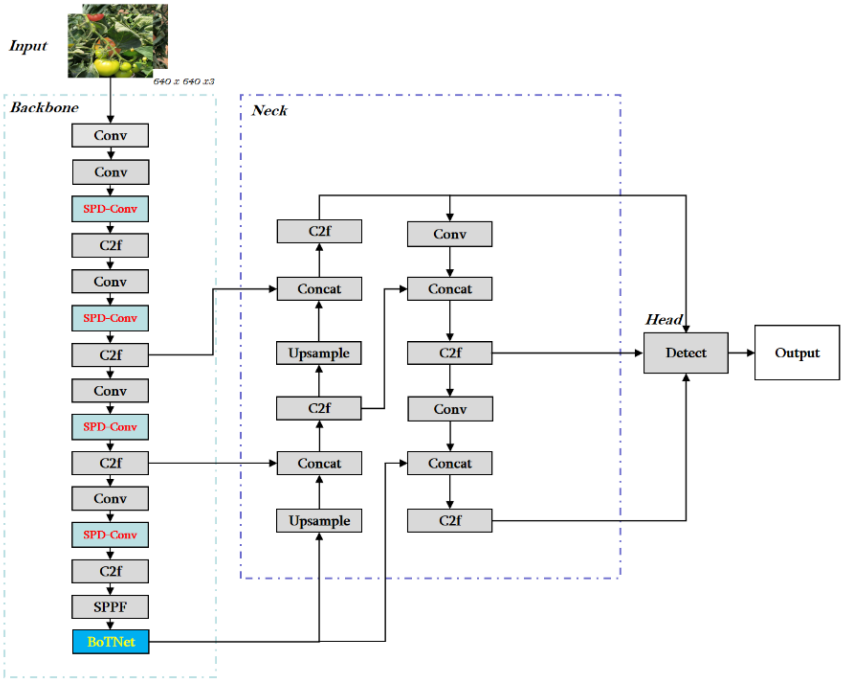

Figure 2: Structure of the BS-YOLOv8n network.

3.1 Ablation experiments

To systematically assess the performance of each individual module in BS-YOLOv8n, ablation experiments are con ducted. The findings from each experiment are presented in Table 1.

ISSN 2616-5910 Vol. 6, Issue 1: 130-136, DOI: 10.25236/AJALS.2025.060118

Table 1: Ablation experiments.

Model	SPD-Conv	BoTNet	AP			mAP@0.5	F1
			ripe	semi	unripe		
YOLOv8n	×	×	84.2%	80%	84.3%	82.8%	0.78
	×	√	84.9%	82.9%	85.2%	84.3%	0.79
	√	×	83.8%	82.8%	83%	83.2%	0.78
	√	√	88.3%	85.8%	85.3%	86.4%	0.80

Initially, only the SPD-Conv module is introduced into the YOLOv8n baseline network, which reduces information loss by transferring the spatial information of the feature maps to the depth dimension. Experimental results show that this improvement enhances tomato detection accuracy across three different maturity levels, with map@0.5 increasing by 1.5% and the F1 score reaching 0.79.

Next, BoTNet's attention mechanism is embedded within yolov8n's backbone, augmenting its spatial feature encoding capacity through parallel attention-convolution fusion. The experimental results show a further improvement in map@0.5 by 0.4%, with the F1 score increasing to 0.78.

Finally, both SPD-Conv and BoTNet are incorporated into the yolov8n baseline network simultaneously. The experimental results indicate a significant improvement in multi-stage tomato detection performance, achieving +3.6% map@0.5 enhancement over the yolov8n baseline and the F1 score rising to 0.8.

In conclusion, ablation studies confirm that SPD-Conv and BoTNet each improve model performance. But when working together, they provide a substantial improvement in overall performance.

3.2 Comparison with other networks

The BS-YOLOv8 nisthoroughl ycompared to othe rYOLO seriesnet works in cluding YOLOv3-tiny[22], YOLOv5n[23], YOLOv5s[23], YOLOv6[24], and YOLOv7[25]. The evaluation uses six key metrics: Recall, Precision, F1 Score, mAP@0.5, FPS, and GFLOPs (see Table2). Comparison results demonstrate that the proposed BS-yolov8n performs very fast with a lighter weight, achieving 11.7 GFLOPS and 218 FPS, while maintaining higher accuracy compared to other models. The training set used in these comparison experiments consisted of augmented but unfiltered balanced datasets. In summary, the BS-YOLOv8n model outperform other models of the same type. The performance of BS-YOLOv8n trained on the augmented balanced dataset using the filtering method is further improved.

Table 2: Comparative experimental results between different models.

Model	Recall	Precision	F1	mAP@0.5	FPS	GFLOPs
YOLOv3-tiny	72.1%	84.3%	0.78	82%	449	18.9
YOLOv5n	76%	82%	0.79	83.5%	126	11
YOLOv5s	76%	84.8%	0.80	84.4%	114	16
YOLOv6	77.5%	81.6%	0.79	84.2%	280	11.8
YOLOv7	79.1%	84.9%	0.81	85.7%	166	105.1
BS-YOLOv8n	77.1%	84.3%	0.80	86.4%	218	11.7

Finally, we predict the tomato images using the YOLOv8n baseline network and BS-YOLOv8n network respectively, and the results are shown in Figure 3, where a1, b1, and c1 are the detection effect graphs under yolov8n baseline network. The figure shows that some small objects and occluded targets are not detected. a2, b2, and c2 are the detection effect graphs of BS-YOLOv8n network, relative to the detection effect of the YOLOv8n baseline network detection effect is obviously improved, the leakage rate of misdetection rate is higher than it. a2, b2 and c2 are graphs of the recognition performance of BS-YOLOv8n on greenhouse tomatoes. Compared with the recognition performance of the baseline YOLOv8n, the BS-YOLOv8n network reduces both the false predicted ratio and the false negative ratio.

ISSN 2616-5910 Vol. 6, Issue 1: 130-136, DOI: 10.25236/AJALS.2025.060118

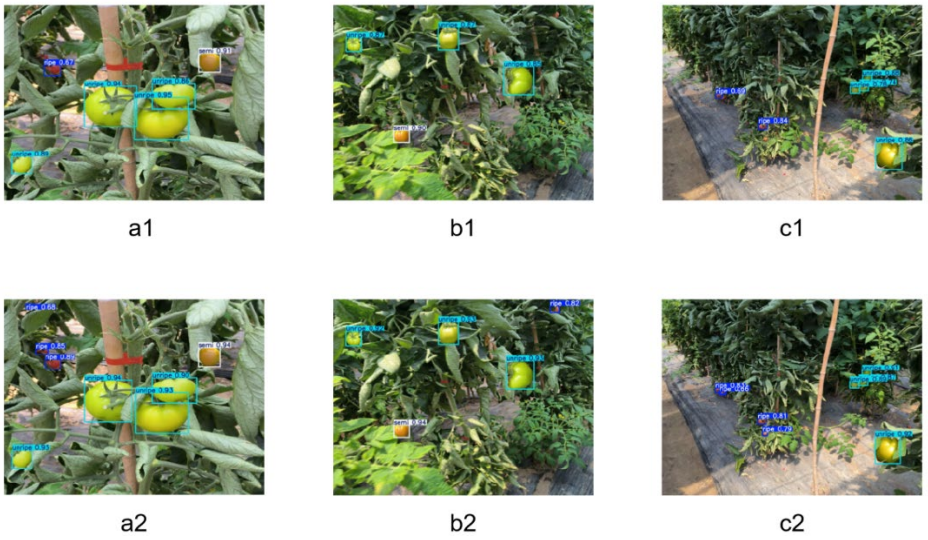

Figure 3: BS-YOLOv8n vs. YOLOv8n detection effect comparison, (a1, b1, c1) is the YOLOv8n detection effect graph; (a2, b2, c2) is the YOLOv8n detection effect graph.

4. Conclusions

Detection of tomatoes with different ripeness levels in complex greenhouse environments through computational vision is of crucial importance for yield estimation and growth status determination in modern smart agriculture. In this paper, we address a series of challenges associated with it, which include more occlusion in complex environments and very small targets in long-distance situations. This paper proposes a BS-YOLOv8n network, compared to the baseline YOLOv8n, it proposed in this paper achieves improvements of 3.6% in mAP@0.5 and 0.02 in F1 score for fruit detection, with a processing speed of 218 FPS, fulfilling the requirements for real-time monitoring. We will build on the existing experimental findings, utilizing binocular cameras for tasks such as segmentation and detection of tomato fruits, with a focus on semi-ripe fruits. The goal is to predict the potential yield of fruits nearing ripeness, enabling short-term yield forecasting in modern agriculture.

Acknowledgements

This research was supported by the Program of Jiangsu Provincial Department of Agriculture and Rural Affairs-Modern Agricultural Machinery Equipment and Demonstration Integrated Promotion Program (Grant no. Nj2023-18).

References

[1] C. Wang, F. Gu, J. Chen, et al., "Assessing the response of yield and comprehensive fruit quality of tomato grown in greenhouse to deficit irrigation and nitrogen application strategies," Agricultural water management, vol. 161, pp. 9–19, 2015.

[2] W.J.Kuijpers, M. J. van de Molengraft, S. van Mourik, A. van't Ooster, S. Hemming, and E. J. van Henten,"Model selection with a common structure: Tomato crop growth models," Biosystems engineering, vol. 187, pp. 247–257, 2019.

[3] A.Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," Computers and elec tronics in agriculture, vol. 147, pp. 70–90, 2018.

[4] A. Cravero, S. Pardo, S. Sepúlveda, and L. Muñoz, "Challenges to use machine learning in agricultural big data: A systematic literature review," Agronomy, vol. 12, no. 3, p. 748, 2022.

[5] 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, 2014, pp. 580–587.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 6, pp. 1137–1149, 2016.

ISSN 2616-5910 Vol. 6, Issue 1: 130-136, DOI: 10.25236/AJALS.2025.060118

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.
[8] W.Liu, D. Anguelov, D. Erhan, et al., "Ssd: Single shot multibox detector," in Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, Springer, 2016, pp. 21–37.
[9] J. Redmon, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.
[10] D.WangandD.He,"Fusion of mask rcnn and attention mechanism for instance segmentation of apples under complex background," Computers and Electronics in Agriculture, vol. 196, p. 106864, 2022.
[11] Z. Wang, Y. Ling, X. Wang, et al., "An improved faster r-cnn model for multi-object tomato maturity detection in complex scenarios," Ecological Informatics, vol. 72, p. 101886, 2022.
[12] F. Liu, Y. Liu, S. Lin, W. Guo, F. XU, and Z. Bai, "Fast recognition method for tomatoes under complex environments based on improved yolo," Transactions of the Chinese society for agricultural machinery, vol. 51, no. 6, pp. 229–237, 2020.
[13] B.Yan,P.Fan, X. Lei, Z. Liu, and F. Yang, "A real-time apple targets detection method for picking robot based on improved yolov5," Remote Sensing, vol. 13, no. 9, p. 1619, 2021.
[14] Xue Yueju, Huang Ning, Tu Shuqin, Mao Liang, Yang Aqing, Zhu Xunmu, Yang Xiaofan, Chen Pengfei. Immature mango detection based on improved YOLOv2. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2018, 34(7): 173-179. DOI: 10.11975/j.issn.1002-6819.2018.07.022
[15] R. Tang, Y. Lei, B. Luo, J. Zhang, and J. Mu, "Yolov7-plum: Advancing plum fruit detection in natural environments with deep learning," Plants, vol. 12, no. 15, p. 2883, 2023.
[16] G. Jocher, A. Chaurasia, and J. Qiu, Ultralytics YOLO, version 8.0.0, Jan. 2023.[Online]. Available: https://github.com/ultralytics/ultralytics.
[17] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "Cspnet: A new backbone that can enhance learning capability of cnn," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 390–391.
[18] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8759–8768.
[19] R. Sunkara and T. Luo, "No more strided convolutions or pooling: A new cnn building block for low resolution images and small objects," in Joint European conference on machine learning and knowledge discovery in databases, Springer, 2022, pp. 443–459.
[20] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, "Bottleneck transformers for visual recognition," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 16519–16529.
[21] A. Vaswani, "Attention is all you need," Advances in Neural Information Processing Systems, 2017.
[22] P. Adarsh, P. Rathi, and M. Kumar, "Yolo v3-tiny: Object detection and recognition using one stage improved model," in 2020 6th international conference on advanced computing and communication systems (ICACCS), IEEE, 2020, pp. 687–694.
[23] G. Jocher, Ultralytics/yolov5: V3.1- bug fixes and performance improvements, https://github.com/ultralytics/yolov5, version v3.1, Oct. 2020. DOI: 10.5281/zenodo.4154370.
[24] C. Li, L. Li, H. Jiang, et al., "Yolov6: A single-stage object detection framework for industrial applications," arXiv preprint arXiv:2209.02976, 2022.
[25] 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," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 7464–7475.