An Improved YOLOv8 Tomato Leaf Disease Detector Based on the Efficient-Net backbone

Runqian Zhang[1], Yifei Ge[1], Qi Li[1] and Lin Meng[2,*]

[1]Graduate School of Science and Engineering, Ritsumeikan University, 1-1-1 Noji-higashi, Kusatsu, Shiga, 525-8577, Japan
[2]College of Science and Engineering, Ritsumeikan University, 1-1-1 Noji-higashi, Kusatsu, Shiga, 525-8577, Japan

Abstract

Tomato, as a main cultivated vegetable with a worldwide production of over 170 million tons annually, draws huge attention on its disease prevention. Numerous types of tomato diseases that target the tomato's leaf at an alarming rate have been discovered. Though traditional method requires high human involvement and people are seeking autonomous method, the early symptoms of tomato disease which can be observed on leaves are too tiny for existing lightweight detectors. By improving the backbone, we have proposed a lightweight object detection model that improves feature extraction, especially for tiny objects, compare to the original YOLOv8. We have benchmarked the original YOLOv8 and our improved model at different input image sizes and model scales. Experiments on the tomato leaf disease dataset have shown that the best accuracy at mAP50 is 1.8% higher than YOLOv8l. Additionally, our improved model has exhibited a 74% reduction in parameter size.

Keywords

YOLOv8, Efficient-Net, Agriculture detection, Deep learning

1. Introduction

In the field of agriculture, crop diseases are critical problems. Especially with an worldwide production of over 170 million tons annually, tomatoes have been one of the most commonly cultivated vegetables worldwide [1]. Traditionally, experienced farmers have been required to identify and address these issues based on their experience and skills as quickly as possible to avoid damage. With the development of deep learning, high-quality object detection models can now be deployed on mobile devices, which means they are no longer confined to the laboratory. This advancement allows for real-time application in agricultural production directly on the farm, thereby enhancing efficiency and effectiveness.

At the moment, the traditional industry and the field of computer vision have become increasingly interconnected. The demand for object detection has grown significantly, making it

The 6th International Symposium on Advanced Technologies and Applications in the Internet of Things (ATAIT 2024), August 19-22, 2024, Kusatsu, Shiga, Japan

*Corresponding author.
gr0684ef@ed.ritsumei.ac.jp (R. Zhang); gr0607pe@ed.ritsumei.ac.jp (Y. Ge); gr0517rs@ed.ritsumei.ac.jp (Q. Li); menglin@fc.ritsumei.ac.jp (L. Meng)
https://github.com/radiuson/Effi-YOLOv8 (R. Zhang)
0009-0005-1961-4306 (R. Zhang); 0000-0003-3479-9802 (Y. Ge); 0000-0002-1963-5263 (Q. Li); 0000-0003-4351-6923 (L. Meng)

one of the most popular domains within computer-vision science. Currently, depending on the method of implementation detection algorithms can be categorized into two groups. Two-stage models, including SPPNet[2], R-CNN[3], Fast R-CNN[4], Faster R-CNN[5], etc., generate a set of candidate frames, which are then classified and predicted with convolutional neural networks. One-stage models, including the YOLO (You Only Look Once) series[6, 7, 8], SSD[9], CenterNet[10], etc., directly turn the problem of target boundary prediction into a regression task. These algorithms each have their own merits; the former excels in terms of accuracy and precision, while the latter performs better in terms of detection speed[11].

In this research, the aim was to improve the accuracy of detection on the tiny symptoms of tomato disease which can be observed on leaves. To address this challenge, we have adapted the channel attention mechanism and compound scaling method to construct our backbone network for better feature extraction ability. To achieve mobility as well as high performance of our proposed model, we have searched our whole network with multiple combinations for the best model structure. Finally, our proposed model has outperformed the original YOLOv8 on datasets with classes that have indistinct appearance without significantly increasing its number of parameters or inference time.

Overall, our main contributions can be summarized below:

- An improved model for tiny objects has been proposed by combining the MBConv module and YOLOv8 head.
- For usability on mobile devices, a series of models with different sizes have been searched and eventually a lightweight network has been proposed.

The remainder of this article is organized as follows. Section 2 reviews the previous agricultural detection and related YOLO models. Details of our proposed method are introduced in Section 3. Section 4 presents the experiments and the dataset. Finally, Section 5 concludes the paper.

2. Related Work

Object detection aims to classify certain objects from defined categories contained in the dataset and then predict the position of the object. The challenges in object detection can be basically categorized into increasing the inference speed and enhancing the accuracy of classification, as well as predicting object location. Benefiting from the development of deep learning, improved accuracy and detection speed can now be achieved with state-of-the-art deep learning algorithms. YOLO models have been used for many purposes due to their superior performance, especially for detection tasks in videos[12].

2.1. Agricultural detection

To achieve maturity detection, positioning, and harvesting in crop production, as well as disease, pest, and climate hazard prevention during the growth process, object detection technology is being introduced into the fields to enhance agricultural productivity. Taking tomato cultivation as an example, classification networks can be used to categorize potential diseases on tomatoes.

TM et al.[13] have employed a very classic convolutional neural network model "LeNet" to address the challenge of tomato leaf disease detection, demonstrating that comparable results to state-of-the-art techniques can be achieved with minimal computational resources. This indicates that neural networks can successfully classify tomato leaf diseases even under unfavorable conditions. Agarwal et al.[14] have proposed a deep learning-based disease detection method, utilizing a convolutional neural network with 3 convolutional layers and 3 max pooling layers. Experimental results demonstrated that it outperformed some popular pre-trained models. Yang et al.[15] have proposed a tomato object detection method based on an improved YOLOv8 model to address the problem of low automation level in tomato harvesting in agriculture. The model utilizes depth-wise separable convolution to reduce computational complexity, and incorporates a dual-path attention gate module to enhance the network's ability to distinguish between tomatoes and the background. Additionally, a feature enhancement module is introduced to highlight target details and prevent the loss of effective features.

2.2. The improvement of YOLO series algorithms

To achieve real-time object detection and automation in industrial production or services, the YOLO series continues to play a vital role due to its high speed, high accuracy, lightweight and locally deploy-able features.

Egi et al. [16] have utilized deep learning algorithms such as YOLOv5 and Deep-Sort to perform object detection on tomato greenhouse videos captured by drone platform. This approach enabled systematic counting of red tomatoes, green tomatoes, and flowers in greenhouse environments based on computer vision and drone systems. To address the labor shortage in food service, Ge et al. [17] have designed a fast and accurate empty-dish detection model for robots. A GPU-efficient Ghost module is utilized to replace the original YOLOv4 backbone network, achieving high-speed computation and high accuracy while reducing the number of parameters. To address the challenge of object detection for single-class multi-deformation objects, Yue et al. [18] have designed a Densely Connected Multi-scale (DCM) module to enhance semantic extraction of deformed objects and constructed a lightweight Neck structure for feature fusion, ultimately achieving a good balance between speed and accuracy for single-class multi-deformation objects. To address the negative impact of top-down connections between adjacent layers in FPN on tiny object detection, Gong et al. [19] have proposed a novel concept called the "fusion factor" to control the information delivered from deep layers to shallow layers, thereby adapting FPN for tiny object detection. By using statistical methods to estimate the number of objects within specific size ranges distributed in the dataset, they derived an appropriate fusion factor, which significantly improved performance on tiny object detection datasets.

3. Methodology

In this study, we have utilized network structure extracted from Efficient-Net[20] to replace the backbone of original YOLOv8[6] which is called CSPDraknet-53 and we have conducted experiments on tomato leaf disease dataset at different scales.

3.1. YOLOv8

Released in January 2023, YOLOv8[6] is a mainly used real-time object detection and segmentation model. YOLOv8 has been built on the basis of previous versions and it supports a wide range of tasks including detection, segmentation, pose estimation, tracking and classification.

YOLOv8 consists of a backbone network and a head network. With scaling factors N/S/M/L/X, YOLOv8 provides five models of different sizes, each possessing unique characteristics in terms of the number of parameters and required computational resources. These differences result in significantly different detection performance and inference times.

The backbone network utilizes convolution operations to extract features. In this process, the feature map gradually downgrades its size while increasing its number of channels. As a result, the backbone network can retain and extract important features passed to the head network for further processing. The head network is mainly designed to fuse features from shallow layers representing geometry and features from deep layers representing semantics. Compared to its predecessor, YOLOv5[21], YOLOv8 has two significant improvements. Firstly, it has been changed from the Anchor-Based method to the Anchor-Free method. Secondly, it has also been replaced with the Decoupled-Head. Due to these changes in the head network, YOLOv8 introduces Task Aligned Assigner[22] and Distribution Focal Loss (DFL)[23] to enhance the network's ability to focus on the distribution of positive and negative samples.

3.2. Efficient-Net

In practice, we have observed that the performance of classification models can be improved with appropriate increments in depth, width (number of channels), or resolution, but the positive gain decreases as the model becomes larger. To address this problem, Network Architecture Search (NAS)[24] has been chosen by the authors of Efficient-Net[20] to search for better sets of depths, widths, and resolutions on their backbone, which is improved from Mobile-Net V3. In the paper, the authors formulated the target of maximizing the model accuracy for any given resource constraints as an optimization problem:

$$\underset{d,w,r}{max} \qquad Accuracy(\nu(d,w,r)) \tag{1}$$

$$s.t. \quad \nu(d,w,r) = \bigodot_{i=1...s} \hat{F}_i^{d\cdot\hat{L}_i}(X_{<r\cdot\hat{H}_i, r\cdot\hat{W}_i, w\cdot\hat{C}_i>}) \tag{2}$$

$$Memory(\nu) \leq target\_memory \tag{3}$$

$$FLOPs(\nu) \leq target\_flops \tag{4}$$

where r,w,d are coefficients for resolution, width and scaling network depth: $\hat{F}_i$, $\hat{L}_i$, $\hat{H}_i$, $\hat{W}_i$, $\hat{C}_i$ are predefined parameters in the baseline network Efficient-Net B0. To avoid tedious manual tuning on balancing network width and depth, a new compound scaling method has been proposed which use a compound coefficient ϕ to uniformly scales network depth, width and resolution in a principled way:

$$depth: \qquad d = \alpha^{\phi} \tag{5}$$

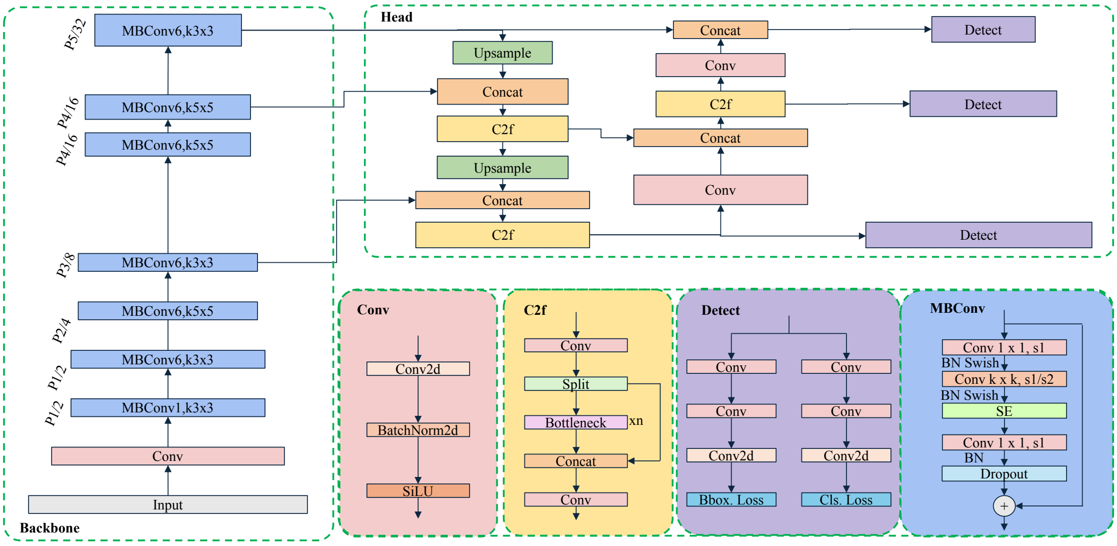


Figure 1: Overall framework of our Improved model with Efficient-Net backbone. The focus is to enhance the ability of feature extraction in order to improve detection performance on tiny objects.

$$\textit{width}: \quad w = \beta^{\phi} \tag{6}$$

$$\textit{resolution}: \quad r = \gamma^{\phi} \tag{7}$$

$$s.t. \quad \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \tag{8}$$

$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1 \tag{9}$$

The most basic and important module in Efficient-Net is MBConv [25]. MBConv is featured with its Inverted Residual and linear bottleneck. It also adapts the Squeeze and Expansion modules from SENet [26] to bring in a channel attention mechanism, which can adaptively study the dependent relationship among different channels.

Experiments performed on Image-Net [27] have shown that scaled Efficient-Net models can consistently reduce parameters and FLOPS compared to some other popular models with similar accuracy.

3.3. Improved structure

Original Efficient-Net has been trained and tested on classification tasks. In this study, we aimed to leverage its strong ability on feature extraction and lightweight design so as to achieve better performance on tiny object detection without significantly increasing parameters and FLOPs compared to YOLO series. As shown in Figure 1, the model consists of two parts, Backbone and Head.

Feature extraction is vital for object detection, thus, we aimed to improve the backbone. As shown on the left of Figure 1, the MBConv module has been repeated to form a feature pyramid network. Inside the MBConv module, the squeeze and excitation layer provides channel attention, and depth-wise separable convolution significantly decreases model size. By adjusting the coefficients of depth, width, and resolution of the backbone, models of different sizes for specific applications can be conveniently constructed.

As shown on the right of Figure 1, we have used two components: the Feature Pyramid Network (FPN) [28] and the Path Aggregation Network (PAN) [29]. FPN first extracts feature maps from the backbone to construct a feature pyramid. It merges feature maps from different levels through up-sampling and coarser-grained feature maps, enabling the fusion of different feature levels. The C2f module is used to keep the model light-weighted and enable richer gradient information. Furthermore, the decoupled detect module separates classification and regression bounding boxes with two independent loss functions. For classification loss, Binary Cross Entropy (BCE) Loss is used to calculate the loss for each category, and for bounding box loss, Complete Intersection over Union (CIoU) is used to calculate the difference between the bounding boxes of objects and predictions. Distribution Focal Loss (DFL) is then used to address the extreme imbalance between foreground and background classes.

4. Experiment

In this section, comprehensive experiments have been conducted using models with backbones and heads of different sizes on the tomato leaf disease dataset, demonstrating the effectiveness of the proposed network.

4.1. Dataset

We have used the public Tomato Leaf Disease dataset[30] on roboflow, which consists of 7 categories of tomato leaf diseases: Bacterial Spot, Early Blight, Healthy, Late blight, Leaf Mold, Target Spot, Black Spot. Image augmentation has been applied to each photo to address the deficiency in data variety.

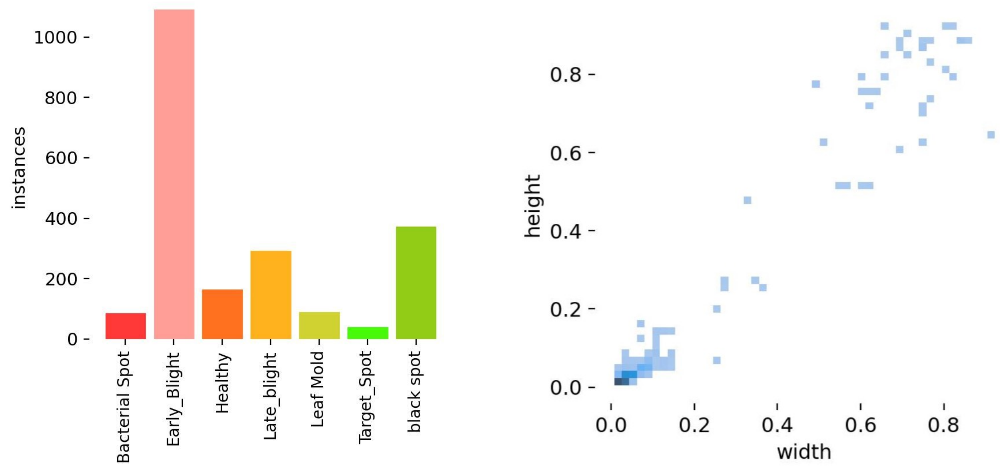
The Tomato Leaf Disease dataset has 737 images in total, 645 images for training set, 61 images for validation set, and 31 images for testing. As shown in the Figure 2, two main challenges can be found:

1. Tiny objects. Symptoms of tomato leaf diseases can be very small, these minuscule symptoms are even difficult for human eyes to capture. As shown in the Figure 2(b), the size of objects are mostly below 10% of the image size.
2. Multiple symptoms. It is not uncommon to find multiple symptoms on the same tomato leaf, and the appearance of different symptoms can be similar. This similarity makes distinguishing between different diseases a significant challenge.

4.2. Implementation details

All experiments have been performed on a platform with an NVIDIA GeForce RTX 4090 graphics card. The deep learning framework used was PyTorch 2.2.2 with CUDA 12.1. The proposed method in this study has been implemented based on Ultralytics 8.1.47, an innovative framework developed by the company that proposed YOLOv5 and YOLOv8.

The epochs and batch size have been set to 400 and 16, respectively, with patience set to 100. The IoU has been set to 0.7 for non-maximum suppression. All models have been trained using



(a) Label distribution (b) Distribution of object sizes

Figure 2: Distribution of training data in tomato leaf disease dataset

the AdamW optimizer, with an initial and final learning rate of 0.01, momentum of 0.937, and weight decay of 0.0005. The learning rate has adopted the warm-up cosine annealing algorithm. The learning rate of the model gradually increases within the first 3 epochs. In the experiment, nearly all models triggered early stopping around epoch 400, which means sufficient training has been conducted on the whole dataset.

4.3. Evaluation metrics

Typical metrics used for object detection tasks have been used to evaluate models in this study, including Precision, Recall, mAP50, and mAP50-95. The definitions of these metrics are as follows:

$$Precision = \frac{TruePositives}{TruePositives+FalsePositives} \tag{10}$$

$$Recall = \frac{TruePositives}{TruePositives+FalseNegatives} \tag{11}$$

$$AP = \sum_{n=1}^{N}(R_{n+1}-R_n)Precision_{max}(R_{n+1}) \tag{12}$$

$$mAP = \frac{1}{C}\sum_{j}^{C}AP_j \tag{13}$$

$$\tag{14}$$

The Average Precision (AP) of all classes is the area of the region below the precision-recall curve. R_n represents the recall of the nth value, and $Precision_{max}(R_{n+1})$ represents the highest precision value in the range R_n to R_{n+1}. The mAP is calculated by averaging the AP of

Figure 3: Visualization results of the proposed method on the tomato leaf disease dataset

Table 1
Performance of original YOLOv8n with different input image size

	YOLOv8n					
Image size	224	240	260	300	380	640
mAP50	0.604	0.623	0.661	0.675	0.726	0.772
mAP50-95	0.399	0.406	0.434	0.446	0.473	0.515

each class in the dataset. mAP50 is obtained by averaging the AP (IoU = 0.5) of all classes, and mAP50-95 is obtained by averaging the mAPs at different IoUs between 0.5 and 0.95.

4.4. Performance Comparison

The comparison model for the original YOLOv8 in this study was reproduced using the Ultralytics framework provided by the author. All models shared the same configuration. To visually demonstrate the effect of our improved method, six images from the tomato leaf disease dataset were used for testing, and the results are shown in Figure 3. The results show that the bounding boxes of all objects have been effectively detected. In the six images, most objects are tiny, occupying an area only 1/100th of the whole image, and the differences in the appearance of each class are indistinct, resulting in classification difficulties.

To address this challenge, our improved network has adopted the MBConv module with a Squeeze and Excitation (SE) layer. The channel attention mechanism helped the backbone focus on effective feature maps during training, improving feature fusion in deeper networks. This has been resulted in better feature extraction.

The results of the original YOLOv8 with scale parameter n on the tomato leaf disease dataset have been depicted in Table 1. To investigate the effectiveness of different backbones with

Table 2
Performance on each class

Class	Precision	Recall	mAP50	mAP50-95
Healthy	1.000	1.000	0.995	0.961
Bacterial spot	1.000	0.684	0.754	0.338
Late blight	0.811	0.889	0.930	0.780
Early blight	0.772	0.792	0.859	0.513
Black spot	0.674	0.486	0.574	0.276
Target spot	0.606	0.833	0.909	0.531
Leaf mold	0.569	0.571	0.509	0.305

different scales, experiments on each set of parameters have been evaluated. Though the depth, width, and resolution should comply with the compound scaling method, the model performance on object detection benefits from higher resolution. In Table 1 and 3, it was evident that the models trained with low resolution result in worse performance than their counterparts trained with higher resolution, thus the comparison has been conducted using an input image size of 640x640.

In terms of the number of parameters, from the original YOLO series in Tables3, it was evident that increasing the number of parameters in the model did not necessarily improve the detection ability of the model. Compared to the original YOLOv8 with scale factor l which has the best performance on the task, our improved network with B2 coefficients and YOLOv8 head in m size has achieved the best performance on the evaluation metrics mAP50 with only 26% of the parameters but a 1.8% improvement. Therefore, the improvement in feature extraction ability of the backbone has proven its parameter effectiveness in enhancing object detection performance.

5. Conclusion

In this study, we have proposed our improved network designed for detecting tomato leaf diseases. Our network has improved the feature extraction ability of the backbone by adopting the module and structure from Efficient-Net. The experiments conducted on our improved network and comparison networks have shown that our network, which consists of an Efficient-Net backbone constructed with B2 size coefficients and YOLOv8 head constructed with m size coefficients, achieves the best mAP50 and mAP50-95, outperforming the original YOLOv8 models and proving the effectiveness of our model. In summary, our improved network with Efficient-Net backbone has satisfactory feature extraction capabilities, especially enhancing the object detection performance on tiny objects. Codes are available at https://github.com/radiuson/Effi-YOLOv8.

References

[1] J. M. Costa, E. Heuvelink, et al., The global tomato industry, Tomatoes 27 (2018) 1–26.
[2] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks

Table 3
Results of improved YOLOv8 proposed in this study on tomato leaf disease dataset

Model	scale	Para	Precision	Recall	mAP50	mAP50-95	FLOPs
Original YOLOv8	n	**3.01**	0.759	0.703	0.772	0.515	**8.1**
YOLOv8+EfficientNet B0		4.10	**0.814**	0.738	**0.779**	**0.520**	8.5
YOLOv8+EfficientNet B1		6.60	0.728	0.732	0.749	0.492	12
YOLOv8+EfficientNet B2		8.18	0.798	**0.763**	0.773	0.519	15.8
YOLOv8+EfficientNet B3		10.76	0.803	0.727	0.771	0.529	19.5
YOLOv8+EfficientNet B4		17.49	0.785	0.722	0.756	0.497	29.5
Original YOLOv8	s	11.13	0.755	0.731	0.770	0.518	28.5
YOLOv8+EfficientNet B0		**4.87**	0.757	0.761	0.755	0.498	**10.6**
YOLOv8+EfficientNet B1		7.38	**0.878**	0.72	0.757	0.509	14.2
YOLOv8+EfficientNet B2		9.15	0.792	0.745	0.787	**0.526**	18.7
YOLOv8+EfficientNet B3		11.85	0.715	0.761	0.742	0.495	22.7
YOLOv8+EfficientNet B4		18.87	0.763	**0.762**	**0.790**	0.507	33.6
Original YOLOv8	m	25.84	0.781	0.684	0.772	0.518	78.7
YOLOv8+EfficientNet B0		**6.70**	0.721	**0.784**	0.747	0.499	**15.7**
YOLOv8+EfficientNet B1		9.20	0.819	0.688	0.753	0.515	19.3
YOLOv8+EfficientNet B2		11.35	0.776	0.751	**0.790**	**0.529**	24.8
YOLOv8+EfficientNet B3		14.28	0.781	0.706	0.754	0.503	29.4
YOLOv8+EfficientNet B4		22.16	**0.856**	0.736	0.770	0.511	42.8
Original YOLOv8	l	43.61	0.780	0.764	**0.776**	0.516	164.8
YOLOv8+EfficientNet B0		**9.68**	0.682	0.772	0.729	0.498	**23.4**
YOLOv8+EfficientNet B1		12.18	**0.807**	0.700	0.756	**0.518**	27
YOLOv8+EfficientNet B2		14.97	0.763	0.772	0.762	0.511	34.7
YOLOv8+EfficientNet B3		18.46	0.799	0.680	0.756	0.509	40.6
YOLOv8+EfficientNet B4		27.74	0.727	**0.774**	0.762	0.506	58.1

for visual recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (2015) 1904–1916. doi:10.1109/TPAMI.2015.2389824.

[3] R. Girshick, J. Donahue, T. Darrell, 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.

[4] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

[5] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in neural information processing systems 28 (2015).

[6] G. Jocher, A. Chaurasia, J. Qiu, Ultralytics YOLO, 2023. URL: https://github.com/ultralytics/ultralytics.

[7] C.-Y. Wang, I.-H. Yeh, H.-Y. M. Liao, Yolov9: Learning what you want to learn using programmable gradient information, arXiv preprint arXiv:2402.13616 (2024).

[8] X. Wang, X. Yue, H. Li, L. Meng, A high-efficiency dirty-egg detection system based on yolov4 and tensorrt, in: 2021 International Conference on Advanced Mechatronic Systems

(ICAMechS), IEEE, 2021, pp. 75–80.
[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, SSD: Single Shot MultiBox Detector, Springer International Publishing, 2016, p. 21–37. URL: http://dx.doi.org/10.1007/978-3-319-46448-0_2. doi:10.1007/978-3-319-46448-0_2.
[10] X. Zhou, D. Wang, P. Krähenbühl, Objects as points, arXiv preprint arXiv:1904.07850 (2019).
[11] H. Li, Z. Wang, X. Yue, W. Wang, H. Tomiyama, L. Meng, An architecture-level analysis on deep learning models for low-impact computations, Artificial Intelligence Review 56 (2023) 1971–2010.
[12] X. Wang, H. Li, X. Yue, L. Meng, A comprehensive survey on object detection yolo, Proceedings http://ceur-ws. org ISSN 1613 (2023) 0073.
[13] P. Tm, A. Pranathi, K. SaiAshritha, N. B. Chittaragi, S. G. Koolagudi, Tomato leaf disease detection using convolutional neural networks, in: 2018 eleventh international conference on contemporary computing (IC3), IEEE, 2018, pp. 1–5.
[14] M. Agarwal, A. Singh, S. Arjaria, A. Sinha, S. Gupta, Toled: Tomato leaf disease detection using convolution neural network, Procedia Computer Science 167 (2020) 293–301.
[15] G. Yang, J. Wang, Z. Nie, H. Yang, S. Yu, A lightweight yolov8 tomato detection algorithm combining feature enhancement and attention, Agronomy 13 (2023) 1824.
[16] Y. Egi, M. Hajyzadeh, E. Eyceyurt, Drone-computer communication based tomato generative organ counting model using yolo v5 and deep-sort, Agriculture 12 (2022) 1290.
[17] Y. Ge, X. Yue, L. Meng, Yolo-gg: A slight object detection model for empty-dish recycling robot, in: 2022 International Conference on Advanced Mechatronic Systems (ICAMechS), IEEE, 2022, pp. 59–63.
[18] X. Yue, L. Meng, Yolo-sm: A lightweight single-class multi-deformation object detection network, IEEE Transactions on Emerging Topics in Computational Intelligence (2024).
[19] Y. Gong, X. Yu, Y. Ding, X. Peng, J. Zhao, Z. Han, Effective fusion factor in fpn for tiny object detection, in: Proceedings of the IEEE/CVF winter conference on applications of computer vision, 2021, pp. 1160–1168.
[20] M. Tan, Q. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, PMLR, 2019, pp. 6105–6114. URL: https://proceedings.mlr.press/v97/tan19a.html.
[21] G. Jocher, YOLOv5 by Ultralytics, 2020. URL: https://github.com/ultralytics/yolov5. doi:10.5281/zenodo.3908559.
[22] C. Feng, Y. Zhong, Y. Gao, M. R. Scott, W. Huang, Tood: Task-aligned one-stage object detection, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE Computer Society, 2021, pp. 3490–3499.
[23] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, J. Yang, Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection, Advances in Neural Information Processing Systems 33 (2020) 21002–21012.
[24] B. Zoph, Q. V. Le, Neural architecture search with reinforcement learning, arXiv preprint arXiv:1611.01578 (2016).
[25] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., Searching for mobilenetv3, in: Proceedings of the IEEE/CVF interna-

tional conference on computer vision, 2019, pp. 1314–1324.
[26] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-excitation networks, 2019. arXiv:1709.01507.
[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.
[28] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.
[29] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8759–8768.
[30] S. A. University, Tomato leaf diseases detect dataset, 2024. URL: https://universe.roboflow.com/sylhet-agricultural-university/tomato-leaf-diseases-detect.