Improved Tomato Detector Supporting for Automatic Harvest Systems

Duy-Linh Nguyen[0000−0001−6184−4133], Xuan-Thuy Vo[0000−0002−7411−0697],
Adri Priadana[0000−0002−1553−7631], Jehwan Choi[0009−0005−8494−2170], and
Kang-Hyun Jo[0000−0002−4937−7082]

Department of Electrical, Electronic and Computer Engineering, University of Ulsan,
Ulsan 44610, South Korea
ndlinh301@mail.ulsan.ac.kr, xthuy@islab.ulsan.ac.kr,
priadana@mail.ulsan.ac.kr, cjh1897@ulsan.ac.kr, acejo@ulsan.ac.kr

Abstract. Currently, Artificial Intelligence has penetrated every corner of social life. Agriculture is one of the most important fields that attracts a lot of attention from researchers to develop serving tools. This paper focuses on developing a vision-based tomato detector to support robotics and automatic harvesting systems. The main technique is to improve the YOLOv8n network architecture with the entire replacement of the original convolution module with a new convolution module, named the Receptive Field Attention Convolution. The experiment was trained and evaluated on the Laboro Tomato dataset. As a result, the proposed network achieved 88.2% of mAP@0.5 and 45.8% of mAP@0.5:0.95. These results show that the proposed network has better performance than other networks under the same experimental conditions.

Keywords: Convolutional neural network (CNN) · Tomato detection · Receptive Field Attention Convolution · YOLOv8.

1 Introduction

Since ancient times, planting, tending, and harvesting have been the main activities in agriculture that are carried out by manual methods. In particular, harvesting is the last stage and requires the most labor to ensure product quality. Tomato is an agricultural product with high nutritional and economic value. According to a report from the Food and Agriculture Organization of the United Nations (FAO), the world produces 190 million tons of tomatoes every year which is concentrated largely in countries such as China, India, Turkey, USA, Italy, Egypt, Spain, Mexico, Brazil, and Nigeria [14]. Tomatoes are a watery fruit and are easily damaged. Therefore, it requires care and precision in harvesting and storage. With the development of robotics and artificial intelligence (AI), agricultural activities are gradually automating harvesting [2], pruning [12], and spraying [11]. Since then, many smart farms have appeared and machines have gradually replaced farmers. Also to automate tomato harvesting, this paper proposes an improved computer vision-based detector from the YOLOv8n network with a perfect replacement of the original convolution operations by the Receptive Field Attention Convolutions (RFAConv) [15]. Using a combination of lightweight architectures and attention mechanisms, the detector can be applied in low-computation mobile devices used in robotics or automated harvesting systems.

The new contributions of this paper are shown as follows:
1 - Proposes an improved tomato detector based on YOLOv8n architecture that can be applied to robots and automated harvesting systems.
2 - The proposed tomato detector performs better than other detectors on the Laboro Tomato dataset.
The remaining parts of this paper are distributed like this: Section 2 introduces the tomato detection methods used in smart agriculture. Section 3 explains the details of improved architecture. Section 4 analyzes the experimental setup and results. Section 5 concludes the issue and future work orientation.

2 Related works

2.1 Traditional machine learning-based methods

Traditional machine-learning techniques have long been applied to fruit detection and classification in agriculture. The study by [8] applied the Support Vector Machine (SVM) method in RGB color space to identify fruits and branches in natural environments. The work [10] combined the HSV space method with an advanced segmentation algorithm to find mature tomatoes placed in complex backgrounds. The authors in [6] implemented the Hough transform and SVM based on the color and texture properties of fruits to distinguish them from tomatoes. The research [13] proposed a pomegranate recognition method combining multi-feature fusion and support vector machine (SVM) using the 3D point cloud. In general, these traditional methods achieved quite good accuracy but had high computational complexity, making it difficult to deploy in real-time applications.

2.2 CNN-based methods

The rapid development of CNN networks in the Computer Vision domain has brought many improvements in performance and accuracy beyond traditional machine learning-based methods. In particular, the advent of the YOLO network series has accelerated the deployment of computer vision-based applications in agriculture. The work [9] evaluated the Single-Shot MultiBox Detector (SSD) and YOLO networks to detect green and reddish tomatoes. The authors in [7] replaced circular boundary boxes with traditional rectangular boundary boxes in the YOLOv3 network to improve tomato detection. The study [3] enhanced mAP cherry detection by modifying the labeled boxes using the DenseNet in YOLOv4. The experiments in [5] optimized the YOLOv5 network using the Focus, Cross-stage network, and EIOU loss to detect tomatoes with small sizes. The CAM-YOLO detector [1] incorporated attention mechanisms to enhance the small-size tomato detection in the YOLOv5 network. The research [16] introduced

RepGhost and ECA attention to YOLOv7 to build a dragon fruit detector. CNN networks have achieved high accuracy and performance in real-time systems but still contain a lot of potential for improvement and development.

3 Methodology

Fig. 1 shows the overall proposed tomato detection network. This network is an improvement from YOLOv8 architecture [4] which consists of three modules: backbone, neck, and detection head.

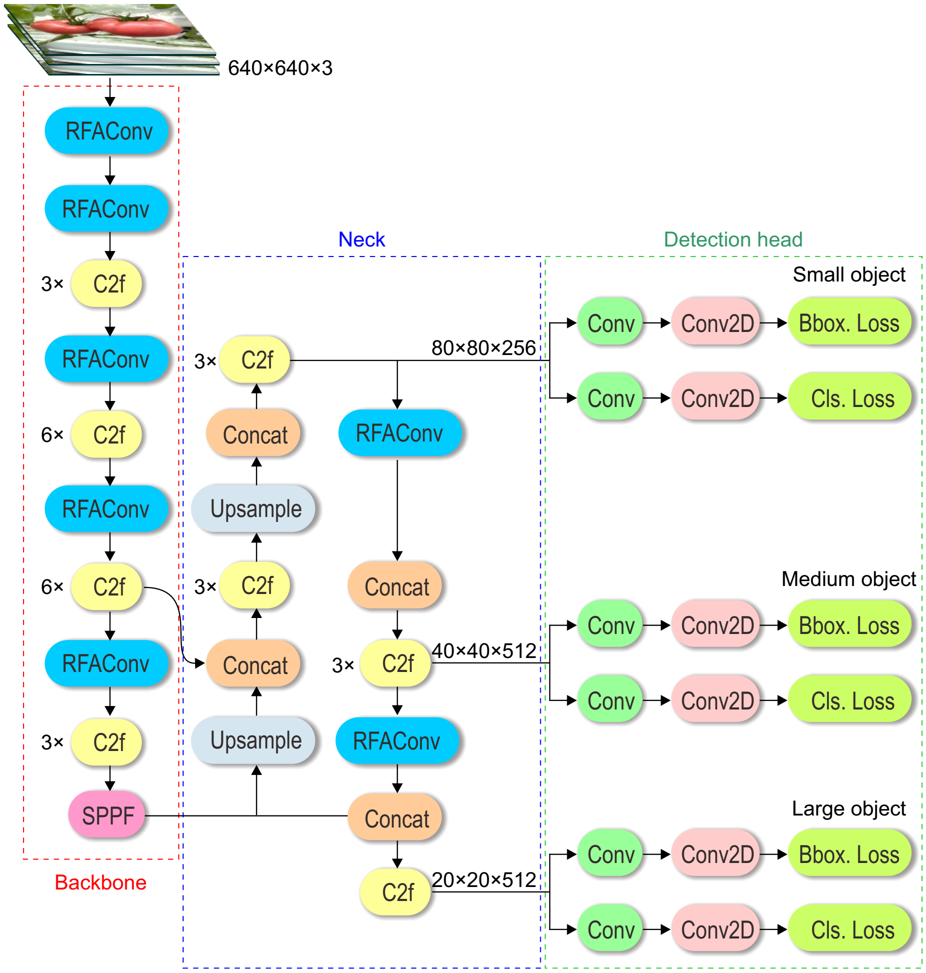

Fig. 1. The architecture of proposed tomato detector.

3.1 Proposed network architecture

Inspired by the original YOLOv8 network architecture [4], this work focuses on testing and evaluating the blocks in use. From that observation, the research modified the backbone and neck module and reused the original architecture of the detection head. Specifically, in the backbone and neck modules, the Cross Stage Partial Bottleneck with two convolutions (C2f) and the Spatial Pyramid Pooling Fast (SPPF) are reused and the Conv block is replaced with a new convolutional architecture called Receptive Field Attention Convolution (RFA-Conv) [15].

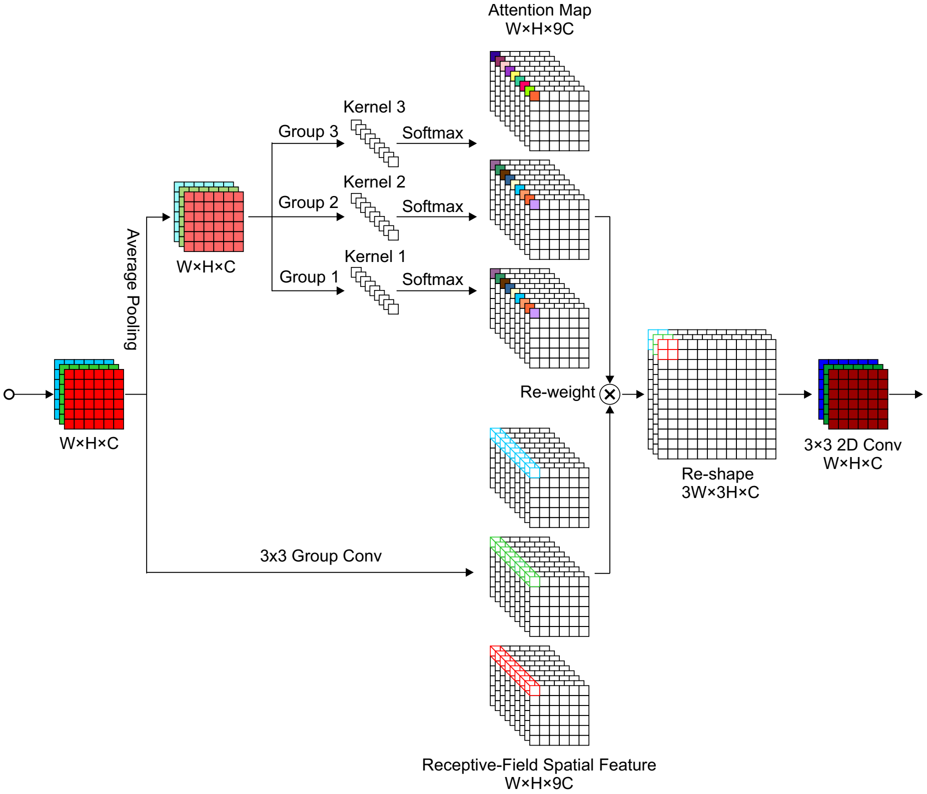

Fig. 2. The architecture of RFAConv module.

The Backbone module starts with an RFAConv block, followed by four identical aggregation blocks (each consisting of an RFAConv block and C2f blocks in a ratio of 3, 6, 6, and 3 times) and an SPPF block. Fig. 2 describes the architecture of RFAConv which is a combination of the Recptive Field Attention (RFA) mechanism and standard convolution (2D Conv). The RFA is proposed to solve the problem of convolution kernel parameters sharing and improve the feature extraction ability of standard convolution. This block implements the lightweight convolution layers (group convolution) that can save a lot of network parameters. Besides, the generated attention mechanism controls the network to focus on learning important information on each feature map level. Suppose, $F \in R^{H\times H\times W}$ and $F' \in R^{H\times H\times W}$ are input and output feature maps, respectively. The operating principle of RFAConv can be expressed as follows:

$$F' = Conv2D^{3\times 3}(Reshape(A_{RF} \times F_{RF})), \tag{1}$$

where A_{RF} is the Receptive Field Attention map, F_{RF} is the Receptive Field Spatial Feature, $Conv2D^{3\times 3}$ is the 3×3 standard convolution, and $Reshape$ is the reshape operation to change the dimension of tensor.

A_{RF} and F_{RF} are calculated based on the following equations:

$$A_{RF} = Softmax(g^{1\times 1}(AvgPool(F))), \tag{2}$$

$$F_{RF} = ReLU(BN(g^{3\times 3}(F))), \tag{3}$$

in which, $g^{i\times i}$ presents the group convolution operation with kernel size $i \times i$, $AvgPool$ is average pooling layer, BN stands for batch normalization (BN). $Softmax$ and $ReLU$ are activation functions.

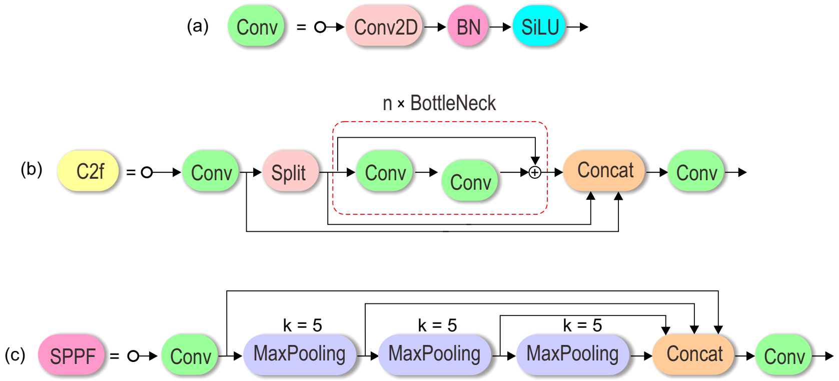

Fig. 3. The architecture of Conv (a), Cross Stage Partial Fast BottleNeck (b), and Spatial Pyramid Pooling Fast (c) blocks.

The final part of the backbone module is the Spatial Pyramid Pooling Fast (SPPF) block. The architecture of the SPPF in the YOLOv8 is reused as shown in Fig. 3 (c). This experiment only applies the kernel size of 5×5 for the whole of max pooling layers.

The Path Aggregation Network (PAN) architecture is reutilized in the neck module of the proposed network following the original YOLOv8 and also replaces the whole of the Conv blocks with the RFAConv blocks. This module upsamples and aggregates the current feature maps with previous feature maps from the backbone module by concatenation operations. The neck module generates the three scale output feature maps corresponding to three scales of object (small, medium, and large). Those feature maps were enriched the information and serve as three inputs for the detection heads.

The detection head module also leverages the architecture of three detection heads from the original YOLOv8. To predict the object, this method applies a decouple head and free-anchor technique. Three feature maps from the neck module transfer to two siblings of the Conv block and standard convolution for bounding box regression (four coordinates: x, y, h, w) and classification (number of classes: c) on three object scales: small, medium, and large. Fig. 3 (a) describes the Conv block. This block is built by a 1×1 standard convolution layer (Con2D), a batch normalization (BN), and a ReLU activation function. In the proposed network, the Conv blocks are only used in the detection head module.

Table 1. The details of the detection head.

Heads	**Input**	**Anchor**	**Ouput**	**Object**
1	$80 \times 80 \times 256$	Free	$80 \times 80 \times 4/80 \times 80 \times 2$	Small
2	$40 \times 40 \times 512$	Free	$40 \times 40 \times 4/40 \times 40 \times 2$	Medium
3	$20 \times 20 \times 512$	Free	$20 \times 20 \times 4/20 \times 20 \times 2$	Large

3.2 Loss function

The loss function is defined as follows:

$$\mathcal{L} = \lambda_{box}\mathcal{L}_{box} + \lambda_{dfl}\mathcal{L}_{dfl} + \lambda_{cls}\mathcal{L}_{cls}, \tag{4}$$

where $\mathcal{L}_{box}$ and $\mathcal{L}_{dfl}$ use the CIoU loss and Distribution Focal Loss (DFL) respectively to calculate the bounding box regression. The classification loss $\mathcal{L}_{cls}$ applies the Binary Cross Entropy loss to compute. The λ_{box}, λ_{cls}, and λ_{dfl} are balancing parameters.

4 Experiments

4.1 Datasets

The Laboro Tomato is an image dataset of growing tomatoes at different stages of ripeness, created for object detection and segmentation tasks. Following the work in [1], this experiment uses 989 images with 903 images for the training phase and 86 images for the evaluation phase. The annotations of each object in the image are converted to YOLO format with two main categories: ripe and unripe.

4.2 Experimental setup

The proposed model is implemented on the Pytorch framework and the Python programming language. The experiment trains and evaluates on a GeForce GTX 1080Ti 11GB GPU. The optimizer is Stochastic Gradient Descent (SGD) optimization. The learning rate is initialized at 10^{-2} and ends at 10^{-4}. The momentum was set at 0.937. The training process uses 200 epochs with a batch size of 64. The balance parameters are set as follows: λ_{box}=7.5, λ_{cls}=0.5, and λ_{dfl}=1.5. This work applies data augmentation methods such as mosaic, translate, scale, and flip to enrich the training dataset and avoid over-fitting problems. In the inference process, several arguments are set based on an image size of 640×64, a batch size of 64, a confidence threshold = 0.5, and an IoU threshold = 0.5. The speed testing is reported in milliseconds (ms).

4.3 Experimental results

Table 2. Comparison result of proposed tomatoes detection network with other networks on the Laboro Tomato validation set. The symbol "†" denotes the re-trained networks. N/A means not-available values.

Models	**Parameter**	**GFLOPs**	**Weight (MB)**	**mAP@0.5**	**mAP@0.5:0.95**	**Inf. time (ms)**
YOLOv5n†	1,766,623	4.2	3.8	87.1	37.5	4.9
YOLOv8n†	3,006,038	8.1	6.2	88.1	42.2	3.5
YOLOv8s†	11,126,358	28.4	22.5	88.1	44.0	5.2
YOLOv8m†	25,840,918	78.7	52.0	86.7	45.3	10.5
YOLOv8l†	43,608,150	164.8	87.6	86.3	41.4	18.1
YOLOv8x†	68,125,494	257.4	136.7	88.0	46.5	25.5
YOLOv5 [1]	N/A	N/A	N/A	85.9	N/A	N/A
YOLOv5+CSP [1]	N/A	N/A	N/A	86.9	N/A	N/A
CAM-YOLO [1]	N/A	N/A	N/A	88.1	N/A	N/A
Our	**3,054,394**	**8.4**	**6.4**	**88.2**	**45.8**	**10.3**

The proposed network's performance is evaluated based on the comparison results with the re-trained networks from scratch and the recent research on the Laboro Tomato dataset. More specifically, this work trains and evaluates the proposed network, one version of YOLOv5 architectures (YOLOv5n), and five of YOLOv8 architectures (x, l, m, s, n). After that, its results are compared to the results in [1] on the Laboro Tomato dataset. As a result, the proposed network achieves 88.2% of mean Average Precision with an IoU threshold of 0.5 (mAP@0.5) and 45.8% of mAP with ten IoU thresholds from 0.5 to 0.95 (mAP@0.5:0.95). From these experimental results, for the mAP@0.5 measure, the object detection ability of the proposed network is superior to other networks even when compared with larger versions of the YOLOv8 network (0.1 % ↑ compared to the best competitor). For mAP@0.5:0.95, the proposed network is still better than other networks except for the YOLOv8x network architecture (0.7% ↓). Speed testing (Inference time) also presents that the proposed network has

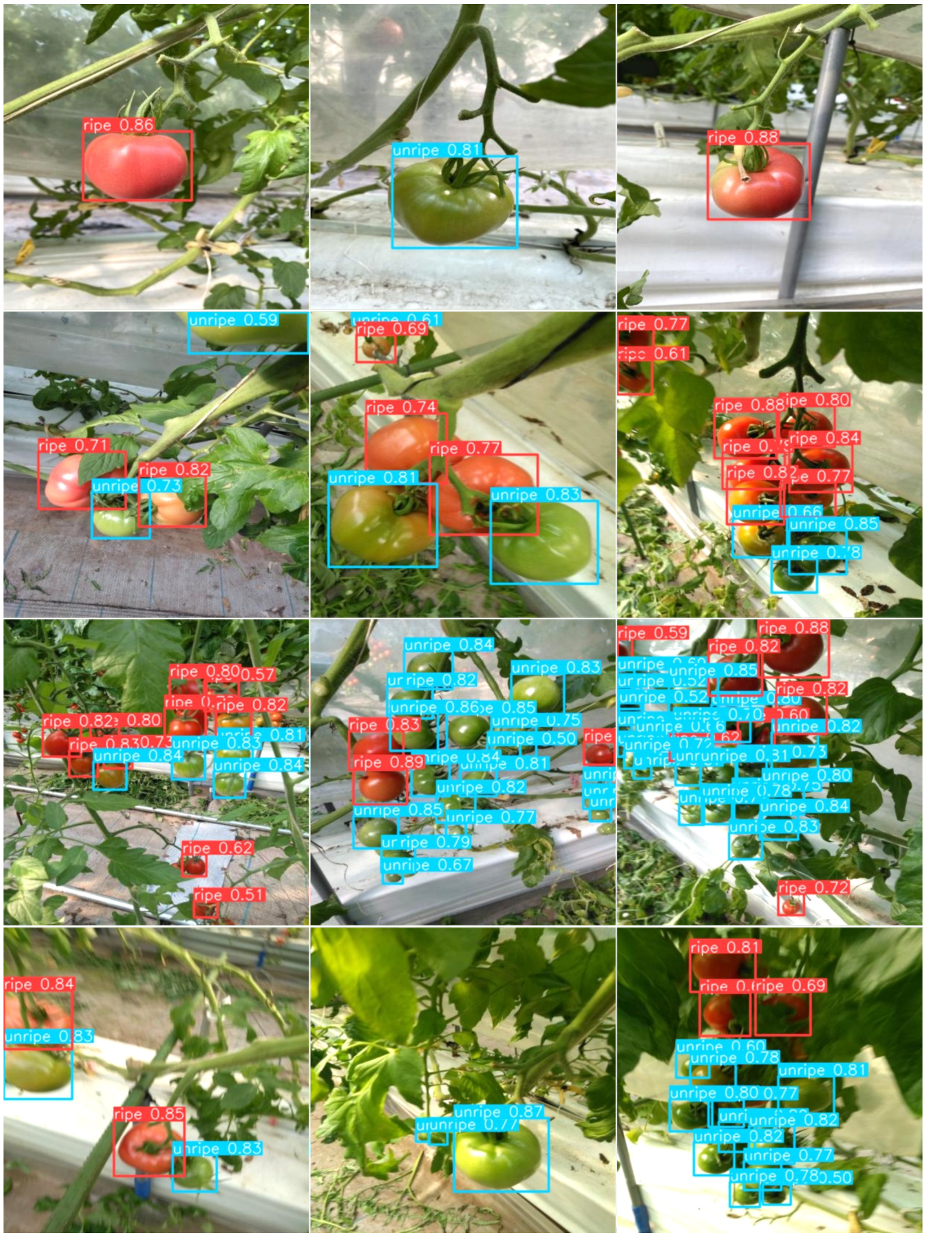

Fig. 4. The qualitative results of the proposed network on the validation set of the Laboro Tomato dataset with IoU threshold = 0.5 and confidence threshold = 0.5. The numbers are predicted confidence scores.

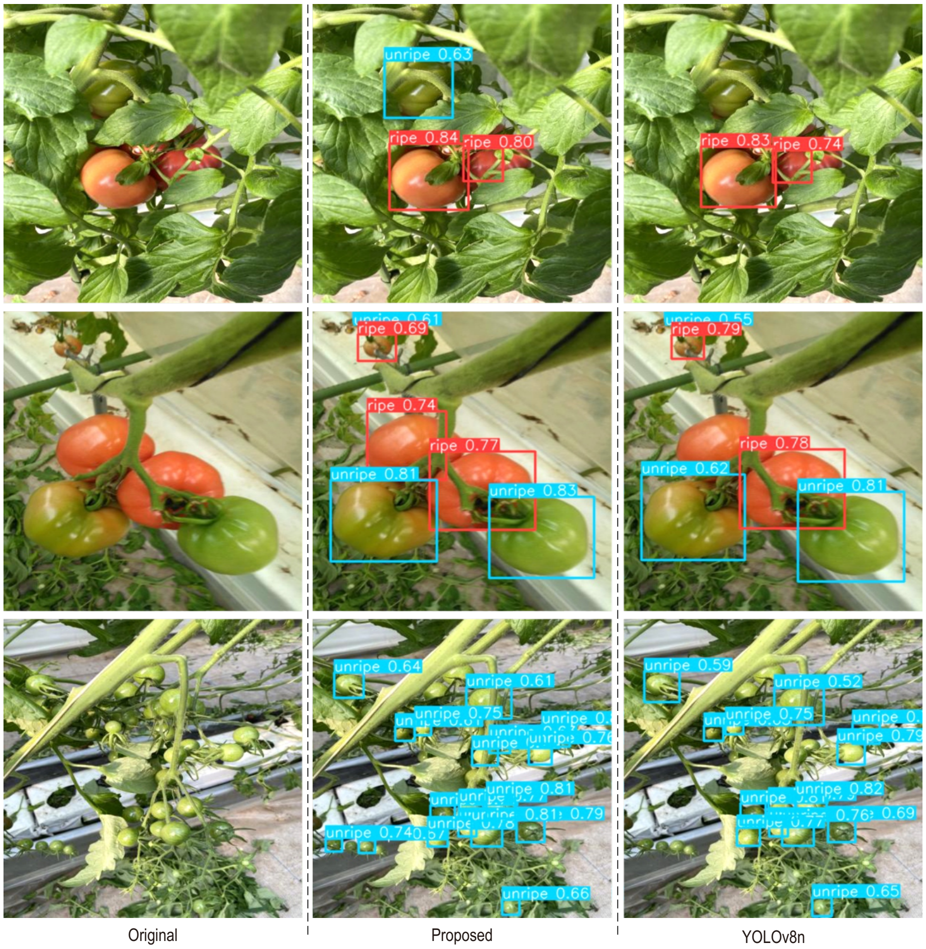

Fig. 5. The comparison results of the proposed network and YOLOv8n on the validation set of the Laboro Tomato dataset with IoU threshold = 0.5 and confidence threshold = 0.5. The numbers are predicted confidence scores.

the same performance as YOLOv8m (0.2% ↓) but the network parameters are nearly 3 times smaller. Additionally, the network parameters are only equivalent to the nano network in the YOLOv8 family (YOLOv8n). These advantages allow the proposed network to be deployed on low-computing devices applied in real-time harvesting systems. Table 2 shows the comparison results and Fig. 4 exhibits several qualitative results on the Laboro Tomato validation set. This work also compares the performance of the proposed network with the YOLOv8n network. The results in Fig. 5 demonstrate that the proposed network can detect well in cases where tomatoes are obscured by leaves (1st row) or branches (2nd row). Besides, the proposed network is also capable of detecting tomatoes that are small in size and quite far away (3rd row). The results mentioned above show better performance of the proposed network when compared with existing networks. However, that performance also depends on several factors such as the color of the tomato and the background, the size of the tomato, the distance, and the moving speed of the camera.

4.4 Ablation study

This work conducted several ablation studies to assess the efficiency of each block in the whole of the proposed network. Each block is replaced in turn, trains and evaluates on the Laboro Tomato set as shown in Table 3. The results show that replacing the first Conv block with the RFAConv block can increase the detection ability of the proposed network (2.0% ↑ of mAP@0.5 and 1.7% ↑ of mAP@0.5:0.95). Adaptation of the SPP with the SPPF block also pushes up the detection accuracy while the network parameters are the same. This is the reason why this research decided to use RFAConv and SPPF block to build the best model.

Table 3. Ablation studies with different types of backbones on the Laboro Tomato validation set.

Blocks	**Proposed backbones**			
First Conv	✓	✓		
RFAConv	✓	✓	✓	✓
SPPF	✓			✓
SPP		✓	✓	
Parameter	3,054,038	3,054,038	3,054,394	3,054,394
GFLOPs	8.3	8.3	8.4	8.4
Weight (MB)	6.4	6.4	6.4	6.4
mAP@0.5	86.7	87.2	89.2	**88.2**
mAP@0.5:0.95	44.6	42.4	44.1	**45.8**

5 Conclusion

This paper improved the YOLOv8 architecture for tomato detection supporting for the robot and automatic harvest systems. The proposed network consists of three modules, including the backbone, neck, and detection head. The backbone and neck modules are redesigned by replacing the whole of the Conv blocks with the RFAConv blocks. While the detection head is reused from the original architecture in YOLOv8. The network achieves 88.2% of mAP@0.5 and 45.8% of mAP@0.5:0.95 which are better performance results when compared to recent methods. The optimization of the model size, inferent time, and detection precision provides the ability to operate on low-computing devices. In the future, the work will be extended with larger tomato datasets and a deeper network with transformer for small-size tomato detection.

Acknowledgement

This result was supported by the "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE)(2021RIS-003).

References

1. Appe SN, G A, G.B.: Cam-yolo: tomato detection and classification based on improved yolov5 using combining attention mechanism. PeerJ Comput Sci **v.9**(e1463), 2376–5992 (2023)
2. Bac, C.W., Van Henten, E.J., Hemming, J., Edan, Y.: Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. Journal of Field Robotics **31**(6), 888–911 (2014)
3. Gai, R., Chen, N., Yuan, H.: A detection algorithm for cherry fruits based on the improved yolo-v4 model. Neural Computing and Applications **35**(19), 13895–13906 (2023)
4. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics yolov8 (2023), https://github.com/ultralytics/ultralytics
5. Li, R., Ji, Z., Hu, S., Huang, X., Yang, J., Li, W.: Tomato maturity recognition model based on improved yolov5 in greenhouse. Agronomy **13**(2), 603 (2023)
6. Lin, G., Tang, Y., Zou, X., Cheng, J., jun tao, X.: Fruit detection in natural environment using partial shape matching and probabilistic hough transform. Precision Agriculture **21** (02 2020). https://doi.org/10.1007/s11119-019-09662-w
7. Liu, G., Nouaze, J.C., Touko Mbouembe, P.L., Kim, J.H.: Yolo-tomato: A robust algorithm for tomato detection based on yolov3. Sensors **20**(7), 2145 (2020)
8. Lü, Q., Cai, J., Liu, B., Deng, L., Zhang, Y.: Identification of fruit and branch in natural scenes for citrus harvesting robot using machine vision and support vector machine. International Journal of Agricultural and Biological Engineering **7**, 115–121 (04 2014). https://doi.org/10.3965/j.ijabe.20140702.014
9. Magalhães, S.A., Castro, L., Moreira, G., Dos Santos, F.N., Cunha, M., Dias, J., Moreira, A.P.: Evaluating the single-shot multibox detector and yolo deep learning models for the detection of tomatoes in a greenhouse. Sensors **21**(10), 3569 (2021)
10. Malik, M.H., Zhang, T., Li, H., Zhang, M., Shabbir, S., Saeed, A.: Mature tomato fruit detection algorithm based on improved hsv and watershed algorithm. IFAC-PapersOnLine **51**(17), 431–436 (2018). https://doi.org/https://doi.org/10.1016/j.ifacol.2018.08.183, https://www.

sciencedirect.com/science/article/pii/S2405896318313016, 6th IFAC Conference on Bio-Robotics BIOROBOTICS 2018

11. Oberti, R., Marchi, M., Tirelli, P., Calcante, A., Iriti, M., Tona, E., Hočevar, M., Baur, J., Pfaff, J., Schütz, C., Ulbrich, H.: Selective spraying of grapevines for disease control using a modular agricultural robot. Biosystems Engineering **146**, 203–215 (2016). https://doi.org/https://doi.org/10.1016/j.biosystemseng.2015.12.004, https://www.sciencedirect.com/science/article/pii/S1537511015001865, special Issue: Advances in Robotic Agriculture for Crops
12. Paulin, S., Botterill, T., Lin, J., Chen, X., Green, R.: A comparison of sampling-based path planners for a grape vine pruning robot arm. In: 2015 6th International Conference on Automation, Robotics and Applications (ICARA). pp. 98–103. IEEE (2015)
13. Wang, Y., Zuo, Y., Du, Z., Song, X., Luo, T., Hong, X., Wu, J.: Minet: A novel network model for point cloud processing by integrating multi-modal information. Sensors **23**(14) (2023). https://doi.org/10.3390/s23146327, https://www.mdpi.com/1424-8220/23/14/6327
14. worldostats: Tomato production by country 2023. https://www.worldostats.com/post/tomato-production-by-country-2023, note = Accessed: Feb. 07, 2024. [Online]. Available: https://www.worldostats.com/post/tomato-production-by-country-2023
15. Zhang, X., Liu, C., Yang, D., Song, T., Ye, Y., Li, K., Song, Y.: Rfaconv: Innovating spatial attention and standard convolutional operation (2023)
16. Zhou, J., Zhang, Y., Wang, J.: Rde-yolov7: An improved model based on yolov7 for better performance in detecting dragon fruits. Agronomy **13**(4), 1042 (2023)