Convolutional Neural Network (CNN)-based transfer learning framework for cherry tomato production

Hyeongjun Lim[1], Youngjin Kim[1], Sumin Kim[2], Sojung Kim[1*]

(1. *Department of Industrial and Systems Engineering, Dongguk University-Seoul, Seoul, 04620, Republic of Korea*;
2. *Department of Environmental Horticulture & Landscape Architecture, Dankook University, Chungnam, Republic of Korea*)

Abstract: As crop harvesting becomes more difficult in environments affected by climate change, the application of artificial intelligence technology to crop management through accurate yield prediction is receiving worldwide attention. This study proposes a convolutional neural network (CNN)-based transfer learning framework to increase the productivity and improve the economic feasibility of cherry tomatoes (*solanum lycopersicum*) in South Korea. You-Only-Look-Once 10 Nano (YOLOv10n) is adopted as a CNN-based algorithm. The source model for transfer learning is trained using cherry tomato imagery from the Tomato Plantfactory Dataset, while the target model is trained based on field survey data collected by the National Institute of Horticultural & Herbal Science, Rural Development Administration, Korea. In that process, an image segmentation technique is developed to improve the prediction accuracy, which reduces the root-mean-square deviation of the existing YOLOv10n from 32.3 to 19.8, a 38.7% reduction. Also, the devised economic feasibility analysis method finds the cost of producing cherry tomatoes in South Korea to be 11.12 USD/m^2, while the maximum revenue can reach 22.44 USD/m^2. As a result, the proposed transfer learning framework helps general farms where it is difficult to collect big data to use machine learning techniques to predict crop or vegetable production.

Keywords: transfer learning, smart farming, cherry tomatoes, yield estimation, convolutional neural network, computer vision.
DOI: 10.25165/j.ijabe.20251805.9827

Citation: Lim H, Kim Y, Kim S, Kim S. Convolutional Neural Network (CNN)-based transfer learning framework for cherry tomato production. Int J Agric & Biol Eng, 2025; 18(5): 90–101.

1 Introduction

The tomato is one of the most popular fruits worldwide. The total world tomato production in 2021 was approximately 189.1 million metric tons[1]. Production quantities from China, India, Turkey, and the United States were 67.5, 21.1, 13.1, and 10.5 Mt, respectively. These four countries produce 59.37% of the total world production. Tomato is considered to be one of the primary dietary sources in many countries, due to its nutrients, like vitamins, carotenoids, and phenolic compounds[2]. South Korea produced 0.4 Mt of tomatoes in 2021[3], valued at 896.59 million USD. Therefore, as in other countries, tomato in South Korea is considered an essential fruit.

Numerous studies have been conducted in different countries to improve tomato production; e.g., Türkten and Ceyhan[4] proposed an environmentally efficient way to produce tomatoes using soilless farming technology in a greenhouse farm in Turkey; Rivard et al.[5] investigated the impact of grafting techniques on tomato (*Solanum lycopersicum*) production in Ivanhoe, North Carolina and Strasburg, Pennsylvania in the U.S.; Guo et al.[6] attempted to reduce fertilizer and pesticide use in cherry tomato production using a life cycle assessment (LCA) approach; while Lee et al.[7] used four different rootstocks (i.e., 'Powerguard', 'T1', 'L1', and 'B.blocking') for cherry tomato cultivation to improve its growth and yield in South Korea.

There have also been efforts to use machine learning (ML) techniques to improve the production yield of cherry tomatoes and tomatoes. Due to climate conditions, tomatoes in South Korea are cultivated in greenhouses. The ML techniques are integrated with the concept of smart farming, which refers to advanced technology use (e.g., the IoT) to improve the productivity of crops by weather data collection, crop growth monitoring, preventive maintenance of crop diseases, and prevention of inefficient activities in crop harvesting[8]. For example, Liu et al.[9] proposed a tomato detection algorithm using the YOLO version 3 algorithm to enhance fruit detection accuracy under a complex monitoring mechanism involving illumination variation and branch, leaf, and fruit overlap; Yang and Ju[10] presented a deep learning-based approach to distinguish the ripeness of cherry tomatoes in real time by leveraging YOLOv5 and YOLOv8 (with ResNet50 backbone) models; Nyalala et al.[11] estimated the volume and mass of tomatoes via computer vision technology based on a cherry tomato model with support vector machine (SVM) and radial basis function (RBF); while Kabas et al.[12] used an artificial neural network (ANN), logistic regression, and decision tree to estimate the deformation energy of cherry tomatoes from twelve independent variables of length, thickness, width, geometric diameter, sphericity, surface area, rupture force, firmness, Poisson's ratio, and modulus of elasticity. These studies have shown that while ML or artificial intelligence (AI) methodologies can contribute to improving crop yields, they have the disadvantage of first requiring big data for model learning, while they have not suggested how to establish production plans that take economic feasibility into account based on prediction results obtained through ML. In particular,

Received date: 2025-03-31 **Accepted date:** 2025-08-18
Biographies: **Hyeongjun Lim**, MS Student, research interest: artificial intelligence, Email: 2019112520@dgu.ac.kr; **Youngjin Kim**, PhD Candidate, research interest: artificial intelligence and simulation, Email: yjin@dgu.ac.kr; **Sumin Kim**, Assistant Professor, research interest: smart farming and crop modeling, Email: sumin.kim@dankook.ac.kr.
***Corresponding author: Sojung Kim**, Associate Professor, research interest: artificial intelligence and simulation. Department of Industrial and Systems Engineering, Dongguk University-Seoul, Seoul, 04620, Republic of Korea, Tel: +82-2-2260－3375, Email: sojungkim@dgu.ac.kr.

considering that it is difficult for general farmers to develop separate ML (or AI) models and collect big data, a new approach should be considered in countries like South Korea, where the goal is to spread smart agricultural technology to farmers in general[13].

This study proposes a convolutional neural network (CNN)-based transfer learning framework to increase the productivity and improve the economic feasibility of cherry tomatoes (*solanum lycopersicum*) in South Korea. The You-Only-Look-Once 10 (YOLOv10) algorithm, one of the popular convolution neural network algorithms used to detect an object[14], is selected as the CNN-based cherry tomato detection algorithm. The proposed framework consists of four modules: (1) cherry tomato monitoring, (2) cherry tomato detection, (3) harvest yield estimation, and (4) economic feasibility analysis. After the camera collects the tomato images in the cherry tomato tracking module, the cherry tomato detection module uses the devised CNN-based transfer learning algorithm to detect cherry tomatoes. The harvest yield estimation module analyzes the sizes of the detected cherry tomatoes, and computes the total yield. The economic feasibility analysis module considers the sales revenue, production cost, and transportation cost of the supply chain to compute the total profit of cherry tomatoes. Two data sources, namely the Tomato Plantfactory Dataset[15], a publicly available cherry tomato dataset, and the field study data collected from the National Institute of Horticultural and Herbal Science in South Korea, are used to develop the cherry tomato detection module via CNN-based transfer learning. This study also conducts experiments to determine the detection accuracy of the devised algorithm and the estimated profitability of cherry tomatoes sold in major wholesale markets in South Korea. This study makes the following contributions: first, the concept of transfer learning is applied to the existing CNN methodology including YOLO, which requires a big dataset; i.e., a source model is created through a big dataset, and a target model is created for individual farms. This has the advantage of helping farms, which have relative difficulty in collecting big data, use ML techniques such as CNN. Second, the proposed framework addresses the detection accuracy of crops by applying ML techniques, while also predicting quantity prediction and performing economic analysis at the time of sale, showing that the use of ML techniques makes it possible to manage the economic production of cherry tomatoes. Third, the proposed framework uses You-Only-Look-Once 10 Nano (YOLOv10n), and exploits the developed image segmentation technique in the image preprocessing step to improve the detection ability for cherry tomatoes. In consequence, the proposed transfer learning framework helps small-scale farms—where it is difficult to collect big data—use ML techniques to predict crop or vegetable production, showing that this contributes to the expansion of smart agricultural technology, while increasing farm income.

The remaining sections are organized as follows: Section 2 describes the proposed method, and summarizes the cherry tomato data from the Tomato Plantfactory Dataset[15] and field study data in South Korea[16]. Section 3 evaluates the performance of the proposed framework concerning the detection accuracy of cherry tomatoes, and estimates the profit of cherry tomatoes sold to wholesale markets in South Korea. Section 4 provides the discussion, while Section 5 concludes the study.

2 Materials and methods

2.1 Data collection

Two datasets are needed for transfer learning: (1) a big dataset to develop a source model, which is a generalized machine learning model that can be commonly applied to multiple cases, and (2) a case-specific dataset to develop a target model, which can be applied to a specific farm. First, this study used a publicly available cherry tomato dataset, the Tomato Plantfactory Dataset[15], to develop a source model. This dataset includes 520 images collected in a fully artificially illuminated plant factory laboratory at the Henan Institute of Science and Technology (HIST), Xinxiang, China. This dataset, focusing on the micro tomato variety, includes a total of 9112 tomato objects (5996 green and 3116 red tomatoes), and was collected from the flowering stage in December 2021 through to the maturation stage in February 2022. The dataset was captured by Canon 80D DSLR camera at (6000×4000) pixels resolution and an iPhone 11 wide-angle camera at (4032×3024) pixels resolution under diverse artificial lighting conditions, including variations in tomato fruit development, complex lighting environments, distance changes, occlusion, and blurring.

Second, to develop a target model, field study data on cherry tomatoes (*solanum lycopersicum*) were collected at the National Institute of Horticultural and Herbal Science in Wanju, South Korea (35°830′N, 127°030′E) (Figure 11 shows the location of the subject farm, near Jeonju)[16]. Seeds were sown on March 16, 2022, in plastic trays ((54 cm×28 cm) in size, (5 cm×10 cm) cells with pot volume 3.7 L) with commercial bedding soil labeled 'Bio Sangto'. The soil contains (67.5%, 17.0%, 5.0%, 10.0%, 0.3%, 0.014%, and 0.185%), cocopeat, peat moss, zeolite, perlite, pH adjuster, humectant, and fertilizers containing 270 mg/kg each of N, P, and K, respectively. Seedlings were grown to fully expanded mature leaf stages of 25-30 cm height in a glasshouse at the subject facility. Forty-eight days after sowing, cherry tomato seedlings were transferred to a greenhouse with black plastic mulch film on May 3, 2022. Plants were watered and fertigated weekly with nutrient solution A (Nitrogen, Potassium, Calcium, Boron, Iron, Zinc, and Molybdenum (N, K, Ca, B, Fe, Zn, and Mo) at 5.5%, 4.5%, 4.5%, 0.000 14%, 0.05%, 0.0001%, and 0.0002%, respectively, and solution B of (N, P, K, Mg, B, Mn, Zn, and Cu) at 6%, 2%, 4%, 1%, 0.05%, 0.01%, 0.005%, and 0.0015%, respectively (Mulpure, Daeyu, Seoul, Republic of Korea). The average air temperature in the greenhouse was maintained between 25°C-35°C, while the relative humidity ranged from 50%-85%. The sub-plot in the greenhouse was laid out in a randomized complete design with five transplants (30 cm apart) and three replicates of single-row plots of 1.5 m length. The distance between single-row plots was 140 cm. The weights of harvested fruits ranged (15 to 25) g (Park et al., 2023). Imagery data of the cherry tomato fruits were collected by a Red–Green–Blue (RGB) camera sensor from May 17, 2022 to July 6, 2022.

2.2 Tomato and cherry tomato detection using YOLO

The YOLO algorithm[17] is one of the most widely used state-of-the-art algorithms in the field of object detection[18]. In contrast to conventional two-stage detection methods, like deformable parts models (DPM)[19] and Region-based Convolutional Neural Networks (R-CNN)[20], which initially identify potential object locations within an image and subsequently examine the identified regions individually, YOLO integrates object classification and localization into a unified regression problem, focusing on class probability. This allows a single neural network to analyze an entire image, and simultaneously predict bounding boxes and class probabilities.

YOLO divides the input image into an S×S grid, with each grid cell tasked with detecting an object if the center of the object falls within that cell. Each grid cell predicts bounding boxes, wherein each box is defined as a 5-tuple (a, b, w, h, Confidence score),

corresponding to the center coordinates, width, height, and confidence score of the box. The confidence score is expressed as the product of the probability that an object exists in a cell ($\Pr(obj)$) and the Intersection over Union (IoU) between the predicted box and the ground truth box ($\mathrm{IoU}_{\mathrm{pred}}^{\mathrm{truth}}$).

$$\text{Confidencescore} = \Pr(obj) \times \mathrm{IoU}_{\mathrm{pred}}^{\mathrm{truth}} \quad (1)$$

Although the early versions of the YOLO algorithm focused on real-time processing, it has, through continuous improvements, demonstrated a high level of accuracy[21]. Liu and Nouaze[9] utilized YOLO version 3 to detect tomatoes even under complex environmental conditions, such as lighting changes, occlusion of branches and leaves, and overlapping of fruits. To this end, the existing rectangular bounding box (R−Bbox) was replaced with a circular bounding box (C−Bbox) that is close to the tomato model, thereby improving the IoU calculation for the Non-Maximum Suppression (NMS), hence enhancing tomato detection performance.

YOLOv5 and YOLOv8, developed in 2020 and 2023, respectively, have been widely adopted in various object detection research fields[21]. The YOLOv5 model is particularly useful in complex agricultural environments with irregular lighting conditions, because it improves the basic recognition accuracy by applying various data augmentation techniques that include image rotation, saturation adjustment, and exposure control. It has the particular advantage of automatically optimizing anchor selection by dynamically calculating anchor boxes that fit the training dataset during training to maximize performance. However, when dealing with complex backgrounds, it faces limitations due to reduced detection capabilities, which can lead to errors in distinguishing object boundaries, and inaccurate recognition of crop health, diseases, and pests. YOLOv8 replaces the Cross Stage Partial (CSP) layer used in YOLOv5 with a more efficient and streamlined C2f module to reduce structural complexity and improve computational efficiency, making it a more suitable algorithm for real-time processing. When the Spatial Pyramid Pooling Fast (SPPF) layer is included, it has the advantage of pooling image features of various sizes into a fixed-size feature map, which can further accelerate the processing speed. Considering the characteristics of YOLOv5 and YOLOv8 mentioned above, Yang and Ju[10] applied both algorithms to a real-time cherry tomato ripening classification robot, and showed that the performance of the cherry tomato ripening classification task could be improved.

YOLOv10[22], released in 2024, represents a significant innovation, in that it is free of NMS. Previous versions of YOLOv10 have addressed duplicate detection by removing bounding boxes with IoU below a certain threshold via a post-processing mechanism termed NMS, whereas YOLOv10 adopts a dual-label assignment strategy that combines one-*vs.*-one and one-*vs.*-many assignment strategies to minimize duplicate predictions, and eliminate the dependency on NMS.

Figure 1 shows that YOLOv10 consists of Backbone, Neck, and Head structures[23], which are the same as the previous YOLO versions, while the Backbone layer consists of a Cross-stage partial network (CSPNet)[22] structure, which repeatedly performs convolution operations to extract low-level to high-level features, and transfer them to the Neck, respectively. The Neck layer adopts the improved Path aggregation network (PANet)[24] structure, and integrates the multi-scale features extracted from the YOLO backbone through the asymmetric path aggregation method that adds a bottom–up augmentation path to the top–down path of the Feature pyramid network (FPN)[25]. The head layer is the layer that performs the final prediction, and performs simultaneous bounding box regression and object classification. YOLOv10 has, through extensive experiments, shown better detection accuracy and latency improvement, compared to previous versions. Because of its lightweight design, it is a more suitable algorithm for individual farmers who, due to budget restrictions, must use low-spec computers. YOLOv10 is classified into five versions, *n*, *s*, *m*, *l*, and *x*, according to the number of parameters. The YOLOv10n version has the minimum number of parameters, so it is particularly suitable for lightweight computing environments.

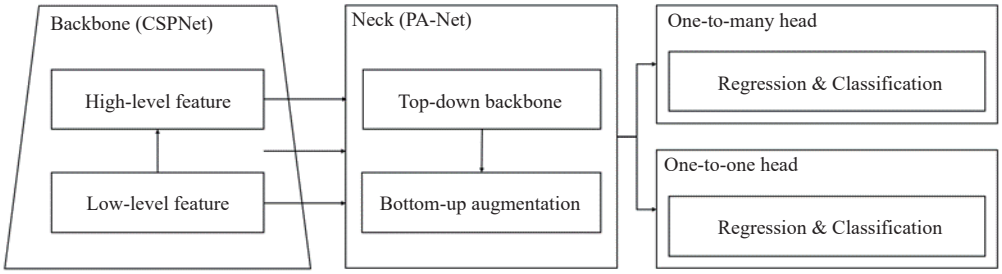


Figure 1 Architecture of You-Only-Look-Once version 10

2.3 CNN-based transfer learning framework for cherry tomato production

This study proposes a transfer learning framework for economical cherry tomato production using the YOLOv10n algorithm. Figure 2 represents the proposed framework that consists of four modules: (1) cherry tomato monitoring, (2) cherry tomato detection, (3) harvest yield estimation, and (4) economic feasibility analysis. In the framework, cherry tomato cultivation conditions are monitored through the cherry tomato monitoring module involving camera sensors, microclimate sensors, and soil sensors. The image data of cherry tomatoes are mainly used in the detection module to train and test the devised YOLOv10n transfer learning algorithm. The status of the detected cherry tomatoes is analyzed, while the harvest yield estimation module estimates the total yield. The economic feasibility analysis module computes the production cost of the cherry tomatoes, and the sale to the best market that maximizes the total profit.

2.3.1 Cherry tomato detection via transfer learning

Datasets collected from individual farms reflect the actual cultivation environment, so they incur several inherent difficulties, such as low image quality, high density and overlap of tomatoes, variation in tomato size depending on the perspective, and difficulty in distinguishing green immature tomatoes from leaves[26]. In addition, the small size of the dataset makes it difficult to sufficiently learn various patterns, which makes it highly likely to overfit to the training sample, making it difficult to develop a

generalized and reliable model[27].

To solve the problems that can occur in learning ML algorithms that can be brought about by such small-scale field datasets, this study applies transfer learning to develop a tomato detection model. Transfer learning is developed for farms (i.e., target farms) that wish to apply an optimized machine learning model with fast and reliable performance by using the weights of a model learned through a large-scale source dataset (i.e., an online tomato dataset). In general, the tomato dataset collected from a target farm has a small number of samples, making it difficult to sufficiently teach the ML model. However, by performing additional learning for the target farm based on the weights of the model learned through the big dataset in advance, it helps develop an efficient machine learning model. Figure 3 shows the YOLO transfer learning process used for tomato detection in this study.

In the transfer learning, the Convolution (Conv) block of the YOLO backbone is the basic component that performs down-sampling. For the given input feature map (i.e., multi-dimensional tensor representing spatially extracted features from the input image), a 3×3 kernel (k=3) is strided by 2 (s=2), and the feature map size is significantly reduced. Faster cross-stage partial bottleneck with 2 convolutions (C2f) and spatial-channel decoupled down-sampling (SCDown) are also convolution-based blocks that process the feature map. The C2f block splits the input feature map into two paths: one path bypasses transformation and retains the original features, while the other passes through a bottleneck block consisting of two convolution layers with k=3 and s=1, and then finally both paths are concatenated to integrate the extracted features. Note that s=1 preserves the feature map size, while performing feature extraction and refinement. This allows the gradient flow to be distributed across different network paths, preventing redundant computations and improving computational efficiency[22]. Compact inverted block (CIB) replaces the bottleneck block in C2f and maintains the overall structure of C2f, which is termed C2fCIB. CIB performs three depth-wise convolutions of k=3, s=1, and two alternating convolutions of k=1, s=1 to maintain the feature extraction performance of the bottleneck block and enhance computational efficiency.

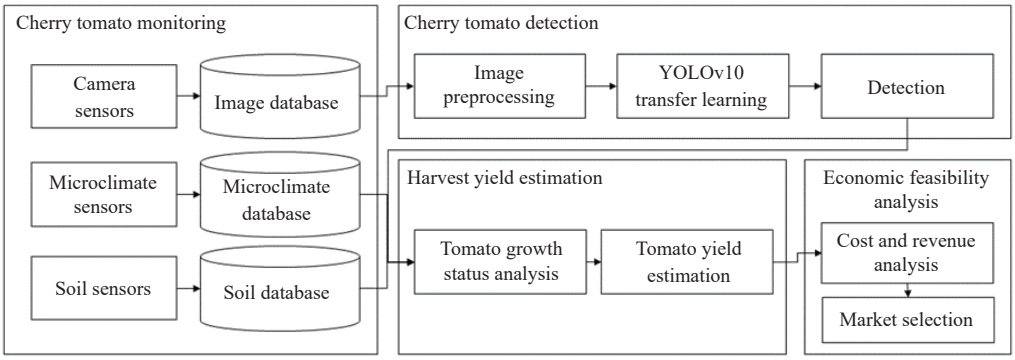


Figure 2 Overview of the proposed framework

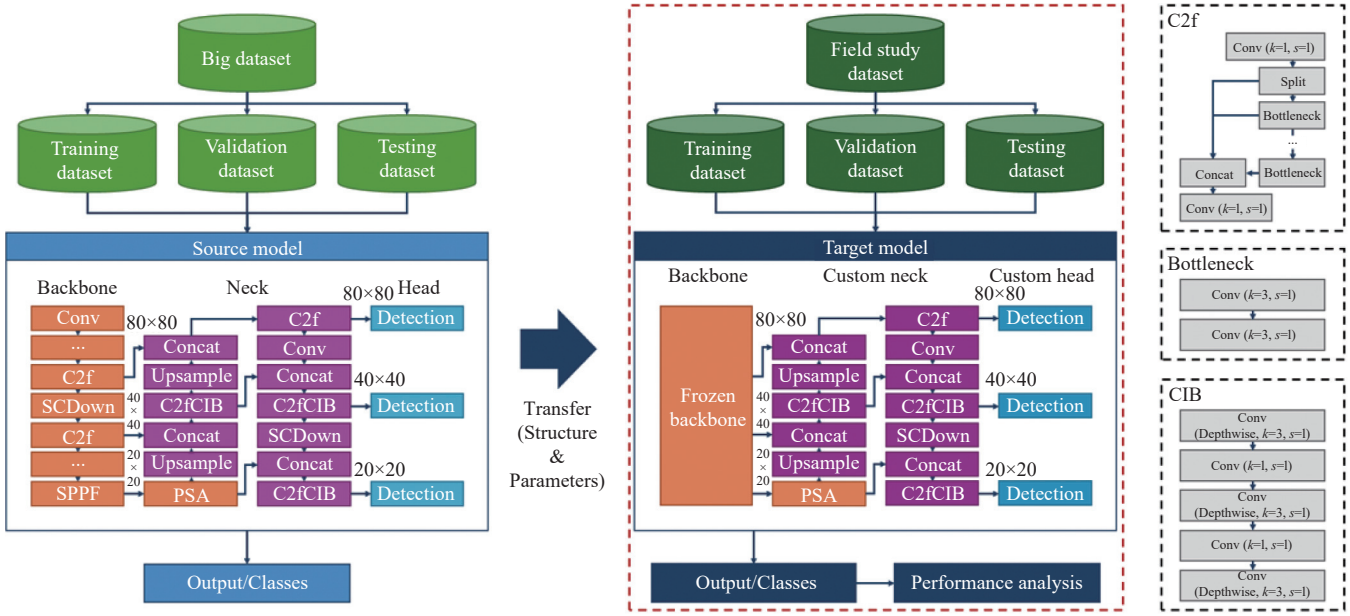


Figure 3 Transfer learning for tomato detection with You-Only-Look-Once version 10

The SCDown block down-samples by applying a computationally efficient depth-wise convolution after adjusting the channel-level features by a convolution operation of k=1 and s=1. The YOLO backbone repeatedly connects the above blocks to perform gradual downsampling, extracting low-level features, such as the color, texture, and edge of the tomato in the early stage, and learning more abstract and high-level features, like the round shape and size of the tomato as the network deepens.

The spatial pyramid pooling-fast (SPPF) block successively applies the max pooling operation, which down-samples by selecting the maximum value in the kernel in three stages, and then concatenates the operation results of each layer to generate scale-

invariant feature representations of cherry tomatoes. The partial self-attention (PSA) block incorporates a feedforward network to the attention block to enhance global feature modeling. While the existing convolutional neural networks (CNNs) learn local information (i.e., shape, texture, and edges of individual tomatoes), self-attention learns the relationships between pixels and distant features from cherry tomatoes and their surroundings, improving detection accuracy in occluded or clustered scenarios. Equation (2) represents the mechanism of self-attention; this calculates the similarity by calculating the inner product between the feature query ($\mathcal{Q}$) of the current position and the feature key ($\mathcal{K}$) of other positions, and then weights the feature value $\mathcal{V}$ to emphasize features that are highly correlated with the other elements. It is presumed that d_k denotes the dimension of both $\mathcal{Q}$ and $\mathcal{K}$ ($\dim(\mathcal{Q}) = \dim(\mathcal{K}) = d_k$).

$$Attention(\mathcal{Q},\mathcal{K},\mathcal{V}) = softmax\left(\frac{\mathcal{Q}\mathcal{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\mathcal{V} \tag{2}$$

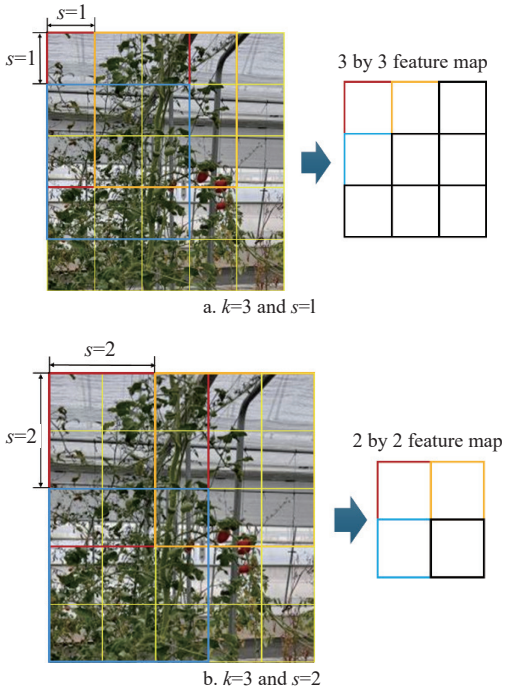


Figure 4 Examples of down-sampling

YOLO neck aims to effectively combine high-level and low-level features to accurately detect cherry tomatoes in various environments (e.g., lighting, distance, size). The Up-sampling (Up-sample) block is repeatedly used to integrate multi-scale features extracted from the backbone. The nearest neighbor up-sampling method, which replicates and uses adjacent pixel values, is adopted to increase the feature map resolution, and the expanded feature map is concatenated with the feature maps of ((40×40) and (80×80)) pixels resolution generated from each C2f block of the backbone. This integration from high-level to low-level is termed a top–down method, while a bottom–up augmentation path is added to supplement the information of the up-sampling process and transfer low-level information back to the high-level[24]. This consists of C2f, C2fCIB, Conv, and SCDown blocks to repeatedly perform feature refinement and down-sampling, and the concatenation between the two paths enables information exchange (see Figure 4).

Finally, the detection block of the head is concatenated with feature maps of different resolutions of ((80×80), (40×40), and (20×20)) pixels, respectively, generated from the bottom–up augmentation path of the neck to detect tomato objects of multiple sizes (i.e., small, medium, and large, respectively). This detection head consists of two convolutional layer branches, which perform bounding box coordinate regression and object classification, respectively.

The learning of the YOLO model is the process of developing a model that calculates the loss between the predicted value and the observed value, and minimizes the loss function based on gradient descent (e.g., SGD, Adam optimizer). For this purpose, the weights of the YOLO network are updated in the opposite direction to the gradient. Equations (3)–(5) represent the YOLO loss function. Equation (3) computes the cross-entropy loss between the predicted class probability distribution p_i and the true class distribution q_i, while Equation (4) computes the bounding box coordinate loss. The a_{pred} and b_{pred} represent the predicted bounding box center coordinates, a_{truth} and b_{truth} represent the true center coordinates, w_{pred} and h_{pred} represent the predicted bounding box width and height, and w_{truth} and h_{truth} represent the true width and height, respectively. Equation (5) represents the confidence loss as the sum of the confidence that the predicted bounding box will contain the true object, and the background confidence that it will not (see Equation (1)). The p_{obj} and q_{obj} represent the predicted value and the true confidence, respectively. Equation (6) represents the process of integrating the loss of each task (i.e., class prediction, bounding box prediction, confidence estimation) through weight λ_i. Equation (7) represents the gradient descent update formula to minimize the objective function $L(\theta)$, while t represents the iteration step in the optimization process, and η represents the learning rate.

$$L_{\text{cls}} = -\sum p_i \log q_i \tag{3}$$

$$L_{\text{coord}} = \sum\left(\left(a_{\text{pred}} - a_{\text{truth}}\right)^2 + \left(b_{\text{pred}} - b_{\text{truth}}\right)^2\right) + \sum\left(\left(w_{\text{pred}} - w_{\text{truth}}\right)^2 + \left(h_{\text{pred}} - h_{\text{truth}}\right)^2\right) \tag{4}$$

$$L_{\text{conf}} = -\sum p_{\text{obj}} \log q_{\text{obj}} - \sum\left(1 - p_{\text{obj}}\right)\log\left(1 - q_{\text{obj}}\right) \tag{5}$$

$$L_{\text{total}} = \lambda_1 L_{\text{cls}} + \lambda_2 L_{\text{coord}} + \lambda_3 L_{\text{conf}} \tag{6}$$

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L(\theta) \tag{7}$$

First, the weights of the YOLO network are learned and updated from the source dataset (i.e., a big dataset) from the initial value. In the step of additional learning with the target dataset, the learned weights of the source model are used as the initial values for the target model. This process is conducted through parameter-based transfer learning[21]. At this time, to maintain the visual features of typical tomatoes that have been pre-learned, the weights from the initial layer in the input direction to the SPPF block (i.e., the YOLO backbone) are frozen, without being updated. On the other hand, the PSA block in the backbone plays a role in controlling spatial attention, so it is adjusted to suit the individual-visual farm dataset, where the arrangement of tomatoes varies.

Equations (8) and (9) show the mathematical logic of YOLO learning and transfer learning. During the learning process, for a set of N input images, the goal is to find the YOLO network parameters θ^* that minimize the loss function $L(\theta)$ between the YOLO network output $f(x_i)$ for the input image x_i and the actual value y_i, which is performed using Equation (7). In the transfer learning process, learning is performed using a new input image set M, the YOLO

backbone parameter is fixed to $\theta^*_{\text{backbone}}$, and the goal is to find the optimal parameters θ'_{neck} and θ'_{head} that minimize the same loss function $L(\theta)$. At this time, the initial values of the parameters $\theta^{(0)}_{\text{neck}}$ and $\theta^{(0)}_{\text{head}}$ of the YOLO neck and head are set to θ^*_{neck} and θ^*_{head}, respectively.

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} L(f(x_i;\theta), y_i) \tag{8}$$

$$\theta'_{\text{neck,head}} = \underset{\theta_{\text{neck}},\theta_{\text{head}}}{\arg\min} \sum_{j=1}^{M} L\left(f\left(x_j;\theta^*_{\text{backbone}},\theta_{\text{neck}},\theta_{\text{head}}\right), y_j\right) \tag{9}$$

$$\theta_{t+1} = \theta_t - \eta\nabla_\theta L(\theta) \tag{10}$$

where, $\theta^{(0)}_{\text{neck}} = \theta^*_{\text{neck}}$ and $\theta^{(0)}_{\text{head}} = \theta^*_{\text{head}}$. In Figure 5, the YOLOv10n transfer learning model is pre-trained using source image I_s and bounding box B_s, and next additionally trained using target image I_t and bounding box B_t, and the accuracy is then evaluated using the test code. Specifically, the YOLO model was trained using the Tomato Plantfactory Dataset collected in a controlled environment. This dataset contains numerous tomato images under optimized lighting and cultivation conditions, helping to comprehensively learn key features of the tomatoes, such as their basic shape, color, and texture.

1: READ I_s, I_t which are source, target tomato image set
2: READ B_s, B_t which are text file of bounding boxes
3: SPLIT I_s, B_t, into train and validation sets (8:2)
4: SPLIT I_s, B_t into train, validation, and test sets (6:2:2)
5: TRAIN model with I_s, B_s
6: FREEZE 10 layers
7: TRAIN model with I_t, B_t
8: TEST the trained model
9: CALCULATE the detection accuracy from the detected images

Figure 5 Pseudocode of transfer learning with You-Only-Look-Once 10 Nano

Next, the pre-trained model was used to additionally learn images collected from our farm. In this fine-tuning process, the Backbone layer (i.e., the initial convolutional layers) of the YOLO model was frozen to maintain the general feature extraction ability. By fixing the layer that learned low-level features (e.g. Shape, Edge, Contour, etc.) and adjusting the weights only for the upper layer according to our farm data, the model is specialized for our farm data, without destroying the general feature representation of tomatoes that it has already learned.

The main parameters are batch size and the number of epochs. The batch size affects computational efficiency and learning stability, while the number of epochs can cause overfitting with a small amount of data in additional learning, so it is important to adjust it to an appropriate level.

In Figure 5, the YOLOv10n transfer learning model is pre-trained using source image I_s and bounding box B_s, and next additionally trained using target image I_t and bounding box B_t, and the accuracy is then evaluated using the test code. Specifically, the YOLO model was trained using the Tomato Plantfactory Dataset collected in a controlled environment. This dataset contains numerous tomato images under optimized lighting and cultivation conditions, helping to comprehensively learn key features of the tomatoes, such as their basic shape, color, and texture. Leave-three-out cross-validation (LO3O-CV), a cross-validation (CV) technique that divides the dataset into three parts (training, validation, and test sets), is adopted in Figure 5.

Next, the pre-trained model was used to additionally learn images collected from our farm. In this fine-tuning process, the Backbone layer (i.e., the initial convolutional layers) of the YOLO model was frozen to maintain the general feature extraction ability. By fixing the layer that learned low-level features (e.g. Shape, Edge, Contour, etc.) and adjusting the weights only for the upper layer according to our farm data, the model is specialized for our farm data, without destroying the general feature representation of tomatoes that it has already learned.

The main parameters are batch size and the number of epochs. The batch size affects computational efficiency and learning stability, while the number of epochs can cause overfitting with a small amount of data in additional learning, so it is important to adjust it to an appropriate level.

2.3.2 Harvest yield estimation

Harvest yield is estimated based on the status of the detected cherry tomato images. As each tomato has different morphological characteristics based on its growth status, the detected images are compared to the growth curve of a typical cherry tomato. Equation (10) estimates the yield of cherry tomatoes detected by the proposed framework.

$$Y_{\text{estimated}} = \frac{(D_t - O_t)}{O_t} \times Y_{\text{harvested}} \tag{11}$$

where, D_t is the detected size of a cherry tomato at time t; O_t is the measured size of cherry tomato at time t; $Y_{\text{harvested}}$ is the average weight of harvested cherry tomatoes; and $Y_{\text{estimated}}$ is the estimated weight of detected cherry tomatoes. Note that O_t and $Y_{\text{harvested}}$ should be computed from historical data, because they are used as reference points to estimate $Y_{\text{estimated}}$.

2.3.3 Economic feasibility analysis

Equation (11) computes profit by analyzing cost and revenue under the estimated total yield of cherry tomatoes and the estimated cost and revenue.

$$P_i = R_i - C_i \tag{12}$$

where, P_i, R_i, C_i are the profit, revenue, and cost at market i, respectively. Note that the net present values (NPVs) of cost, revenue, and profit are not computed, because in South Korea, cherry tomatoes are harvested within three months. The market i is selected with the maximum value of P_i.

3 Results

3.1 Experiment scenario

This experiment compares the performance in terms of tomato yield detection accuracy, model learning speed, and overfitting, and predicts yield and evaluates the economics in selected farms. First, the lightest version of the YOLO model, the Nano version, was adopted for the experiment, considering that it would be operated in the selected farm environment. The nano version is designed to maintain detection performance while minimizing memory and computational resources, and can be used even in limited hardware environments. Table 1 describes the specific hardware and software specifications used in this study.

Table 2 describes the different learning settings for the four cases. Although the batch size is set to 16 for all model learning, Cases 1 and 2, which are non-transfer learning, are trained for 700 epochs, while transfer learning is set to 500 epochs for the Source model learning with the Tomato Plantfactory Dataset, and 200 epochs for the Target model learning with the subject Farm dataset. The remaining parameters not described in Table 2 use the default

values provided by YOLOv10n. For reference, the value of "freeze" indicates the number of backbone layers. In YOLOv10n, the backbone is responsible for feature extraction, and an improved version of the Cross-stage Partial Network (CSPNet) is used to improve the gradient flow and reduce computational redundancy.

Table 1 Hardware and software specification used in the experiment

Components	Specification
CPU	Intel® Core™ i7-14700K, 3400Mhz
GPU	NVIDIA GeForce RTX 4060 Ti
RAM	32.0 GB
Operating system	Microsoft Windows 11 Pro
Framework	PyTorch 2.5.1+cu124, YOLOv10n
CUDA	CUDA 12.4
cuDNN	cuDNN 9.1.0

Table 2 Model training parameter settings

Learning type	Number of images	Batch size	Epochs
Case 1	16	16	700
Case 2	540	16	700
Case 3 (freeze=0)	540/16	16	500/200
Case 4 (freeze=10)	540/16	16	500/200

Model performance is evaluated using three standard metrics: (1) mean absolute error (MAE), which measures the average absolute difference between the predicted value and the observed value (see Equation (12)); (2) root mean square error (RMSE), which is used as an indicator to focus on outliers due to model overfitting by calculating the square root of the mean square error (see Equation (13)); and (3) mean absolute percentage error (MAPE), which provides an intuitive sense of accuracy by expressing the absolute error as a percentage of the observed value (see Equation (14)). These metrics allow the error between the observed actual tomato count and the predicted tomato count using the ML algorithm to be quantitatively evaluated.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{13}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{14}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{15}$$

3.2 Cherry tomato detection accuracy

The devised transfer learning is evaluated according to the experiment scenario of Section 3.1. Figure 6 illustrates examples of cherry tomato detection using YOLOv10n.

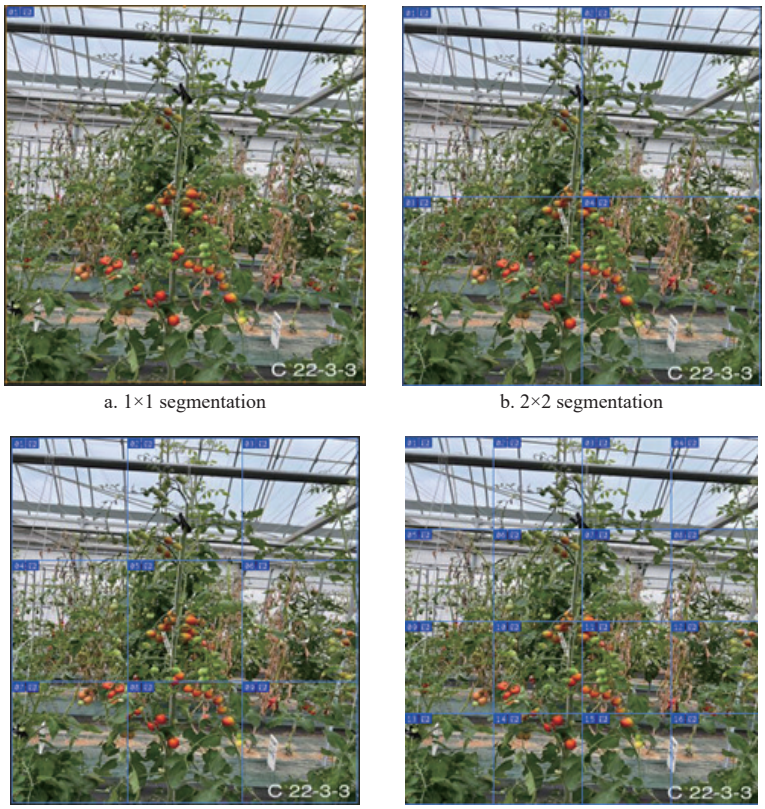
Cherry tomatoes are detected by selecting the most appropriate image segmentation level to improve the detection performance. Figure 7 shows that the image is detected by selecting the segmentation condition with the best detection performance among four segmentation conditions of 1×1, 2×2, 3×3, and 4×4.

a. Original b. Labeled c. Detected image

Figure 6 Examples of cherry tomato detection

Table 3 describes the performance evaluation under four different image segmentation conditions, and shows different performances depending on the three different YOLO versions and experimental cases. In Case 1, which used only a small amount of data from the subject farm, regardless of the segmentation, both no segmentation and auto segmentation types show the same detection performance, because the proposed auto segmentation automatically considers no segmentation (one-by-one segmentation) to four-by-four segmentation. That is, in this case, no segmentation shows the best performance regardless of the YOLO version, and because the amount of data is small, selecting no segmentation is relatively advantageous for the YOLO model learning. In Case 2, which uses a big dataset, auto segmentation showed higher Cherry Tomato detection accuracy than no segmentation in all YOLO versions. This result indicates that as the data becomes more diverse and larger, it is necessary to use an appropriate segmentation technique. On the other hand, similar to Case 1, in Case 3, non-segmentation shows the best Cherry Tomato detection performance. This appears to be because, under the condition freeze=0, model learning can be efficiently conducted even without segmentation, because all layers are adjusted to adapt to the target image. Case 4 shows that non-segmentation is appropriate for YOLOv8, but segmentation shows higher cherry tomato detection performance in YOLOv5 and YOLOv10. Despite the various experimental results, transfer learning models (i.e., Case 3 or Case 4) have the highest detection accuracy in all YOLO versions. In addition, the training time is the shortest in all versions with an average of 314.6 s in Case 1, which has a small number of images, and the longest in Case 2 with an average of 4292.2 s. Case 3 or Case 4 using transfer learning can reduce the learning time by about 26% compared to Case 2. Comparing Case 3 and Case 4, Case 4 takes slightly less training time than Case 3 in all versions, because it did not train with 10 layers fixed, with an average difference of 10.5 s.

Figure 8 shows MAE, RMSE, and MAPE results according to



a. 1×1 segmentation b. 2×2 segmentation

c. 3×3 segmentation d. 4×4 segmentation

Figure 7 Examples of image segmentation

Table 3 Performance evaluation under four different image segmentation cases

Version	Case	Segmentation type	MAE	RMSE	MAPE	Time/s
YOLOv5	Case 1	No segmentation	20.9	25.2	17.7	292.6
		Auto segmentation	20.9	25.2	17.7	
	Case 2	No segmentation	63.0	69.4	52.8	4218.9
		Auto segmentation	20.0	22.5	23.2	
	Case 3	No segmentation	13.9	16.5	13.2	3088.0
		Auto segmentation	13.9	16.5	13.2	
	Case 4	No segmentation	29.9	37.3	23.1	3077.6
		Auto segmentation	24.7	27.0	23.1	
YOLOv8	Case 1	No segmentation	17.9	20.0	19.3	277.3
		Auto segmentation	17.9	20.0	19.3	
	Case 2	No segmentation	55.8	63.9	43.7	4342.5
		Auto segmentation	17.0	23.8	25.8	
	Case 3	No segmentation	9.5	11.8	7.6	3194.6
		Auto segmentation	9.5	11.8	7.6	
	Case 4	No segmentation	21.0	27.1	16.5	3183.4
		Auto segmentation	21.0	27.1	16.5	
YOLOv10	Case 1	No segmentation	25.1	29.3	23.3	373.8
		Auto segmentation	25.1	29.3	23.3	
	Case 2	No segmentation	55.0	63.2	43.4	4315.2
		Auto segmentation	17.8	23.7	26.9	
	Case 3	No segmentation	16.9	18.0	17.0	3186.5
		Auto segmentation	16.9	18.0	17.0	
	Case 4	No segmentation	26.4	32.3	21.4	3176.6
		Auto segmentation	16.4	19.8	21.4	

the Freeze parameter in three YOLO versions (YOLOv5, YOLOv8, and YOLOv10). The *x*-axis of Figure 8 represents four segmentation conditions, while the *y*-axis represents the performance measure.

The experimental results show that as the number of segments increases, the RMSE tends to increase. In Case 1, for the no segmentation (one-by-one segmentation) scenario, all versions showed higher performance for all metrics, compared to Case 2. However, in Case 1, as the number of segments increased, the RMSE increased, reaching the highest values for YOLOv5, YOLOv8, and YOLOv10 of (237.2, 246.3, and 50.5), respectively, in the four-by-four segmentation case. This behavior occurred more significantly in YOLOv5 and YOLOv8 than in YOLOv10, and the RMSE differences between no segmentation and four-by-four segmentation for YOLOv5, YOLOv8, and YOLOv10 were (212.0, 226.3, and 21.2), respectively.

In Case 2, the RMSE, MAE, and MAPE decreased in all segmented scenarios, while in YOLOv5, the standard deviation of the RMSE for segmentation types was 19.9. Conversely, in Case 1, the RMSE standard deviation was 87.2, indicating a larger performance variation according to the image segmentation and weaker generalization performance, compared to Case 2. This indicates that due to the small number of training images, the model could be overfitted.

In YOLOv10, Case 3 showed the highest performance for no segmentation, with MAE = 16.9, RMSE = 18.0, and MAPE = 17.0. However, in all segmented scenarios, Case 4 recorded lower MAE, RMSE, and MAPE than Case 3. Case 4 showed its best performance in two-by-two segmentation, with MAE = 16.4, RMSE = 19.8, and MAPE = 22.1. The RMSE standard deviations for segmentation types for Case 1, Case 2, Case 3, and Case 4 were (8.1, 15.8, 13.3, and 7.6), respectively, indicating that the proposed Case 4 is more robust to image segmentation than the other cases. These results show that Case 3 follows a similar trend to Case 1, while Case 4 follows a similar trend to Case 2. In conclusion, MAE, RMSE, and MAPE show different performances in all image segmentation cases, which suggests that proper image learning through the proposed auto segmentation helps to improve the accuracy of cherry tomato detection.

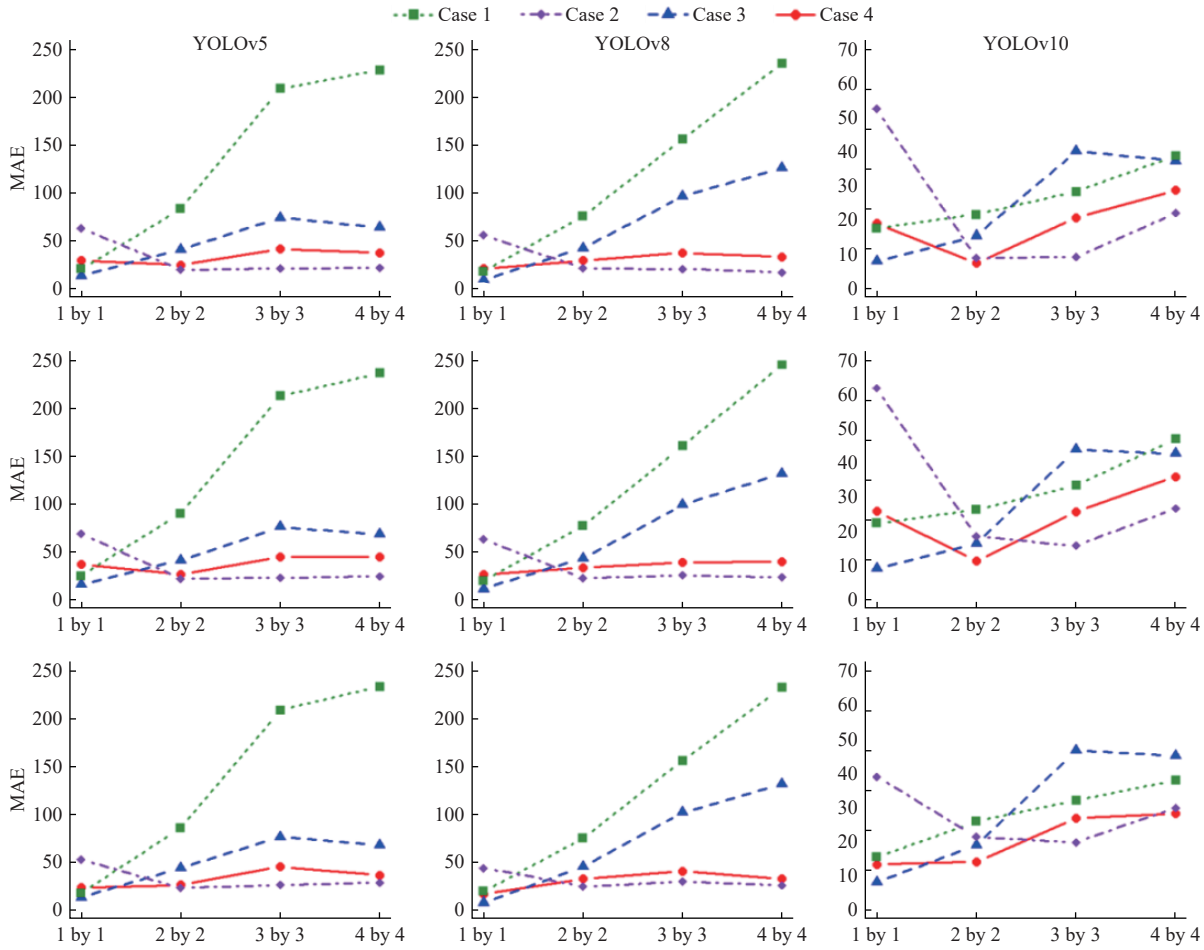


Figure 8 Visualized MAE, RMSE, and MAPE for YOLOv5, YOLOv8, and YOLOv10 across different image segmentation cases (Cases 1−4)

3.3 Harvest yield estimation

As mentioned in Section 2.3.2, Equation (10) estimates the yields of detected cherry tomatoes. To this end, reference points (O_t and $Y_{harvested}$) should be computed from historical data. Figure 9 illustrates the observed plant height and net photosynthesis rate of the cherry tomatoes from May 17, 2022 to July 6, 2022[16]. Note that those values are observed two weeks after the transfer into a greenhouse on May 3, 2022. Cherry tomato plants were fully grown 35 d after the transplant, and their net photosynthesis rates were stable. In addition, morphological characteristics, such as moisture content, leaf area index (LAI), plant height, and stem thickness, of the cherry tomatoes were measured (see Table 4).

Most morphological characteristics in Table 4, except LAI, are stable during the harvesting period from June 21, 2022, to July 6, 2022. Note that LAI decreases during the harvesting season[28]. Table 5 summarizes the information on the harvested cherry tomatoes. Harvest index (HI) was computed by dividing the fresh fruit weight by the total fresh weight. According to previous studies of cherry tomatoes[29,30], the HI values of cherry tomatoes ranged from 0.60-0.68. Therefore, the HI of 0.65 in Table 5 indicates that the field study has been appropriately conducted. The value of $Y_{harvested}$ was computed by dividing the fresh fruit weight by the number of fresh fruit count, resulting in 17.14 g/fruit.

Figure 10 shows the size and weight change of the cherry tomatoes from anthesis to red-ripe. Figure 10a shows that the size of cherry tomatoes in diameter does not significantly change 33 d after anthesis. Joubes et al.[31] showed a similar result. The weight pattern in Figure 10b is quite similar to that of Figure 10a, because of the

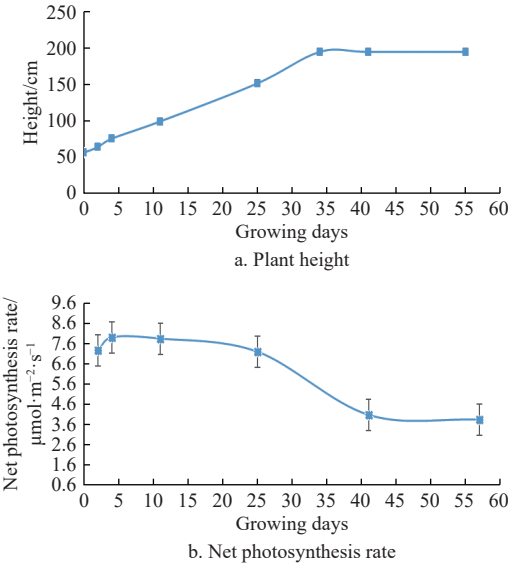


Figure 9 Observed growing status of cherry tomatoes

Table 4 Moisture content, leaf area index (LAI), plant height, and stem thickness of cherry tomatoes

Category	35 d	57 d
Moisture content/%	89±1.1	87±3.6
Leaf area index (LAI)	0.87±0.05	1.82±1.40
Plant height/cm	180±4	186±17
Stem thickness /mm	14.05±0.33	14.89±1.55

high correlation between size and weight[11]. As mentioned in Equation (10), Figure 9a is used to obtain the measured size (O_t) of a cherry tomato at time t.

Table 5 Fresh weight (g), fresh fruit weight (g), and harvest index of cherry tomatoes

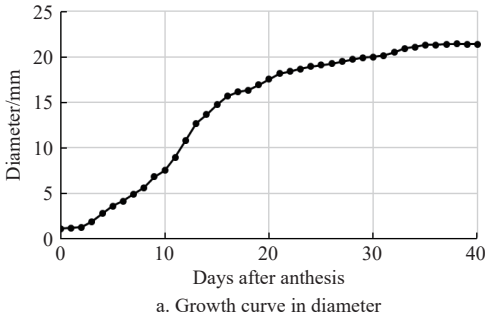
Category	Total fresh weight/g·m^{-2}	Fresh fruit weight/g·m^{-2}	Fresh fruit count/ number·m^{-2}	Harvest index
Observed value	8151±3789	5295±2550	309±150	0.65

3.4 Economic feasibility analysis

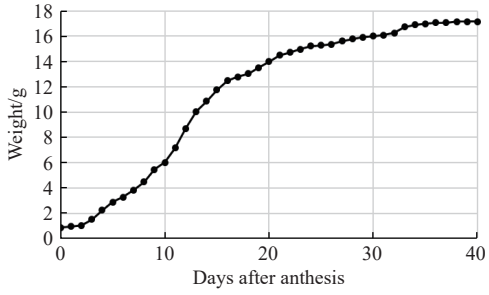
The estimated yield of cherry tomatoes is analyzed under the cost and revenue Equation (11). In particular, five major wholesale markets in South Korea are considered for economic feasibility analysis. Figure 11 illustrates the locations and transportation costs of wholesale markets from the subject farm at the National Institute of Horticultural and Herbal Science in South Korea. The unit transportation cost is assumed to be 1.45 USD/km·t[32].

Table 5 describes the average yield of cherry tomatoes as 5.295 kg/m^2, while Table 6 describes the calculated price and profit of cherry tomatoes based on this result.

Table 6 describes that the production cost of cherry tomatoes is 11.19 USD/m^2, and transportation costs for five different markets are computed for distances from the subject farm to each market (see Figure 11). Due to its location, Daejeon has the minimum transportation cost, while Busan has the highest transportation cost. However, the difference between the five wholesale markets is relatively small, compared with the gaps between the revenues of markets. The market sales prices in Seoul, Busan, Daegu, Daejeon, and Gwangju are 4.79, 6.75, 5.20, 4.34, and 4.01 USD/kg, respectively[33]. Most cherry tomato farms are located in the western region in South Korea. Busan and Daegu have higher sales prices than the other three locations. Therefore, despite their high transportation costs, the Busan and Daegu areas are selected as the most profitable markets. In addition, considering that the profit of cherry tomatoes ranges from 11.12 to 22.44 USD/m^2, the annual operation and maintenance cost of the proposed framework should be less than \$11.12 /m^2.

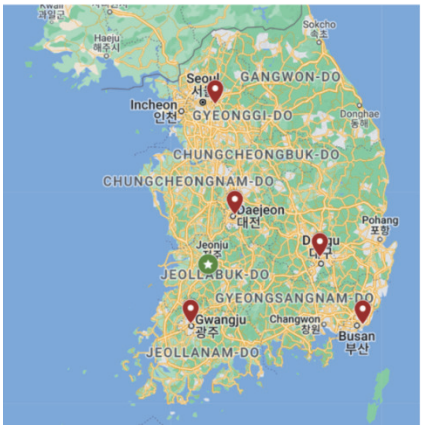


a. Growth curve in diameter



b. Growth curve in weight

Figure 10 Size and weight change of a cherry tomato from anthesis to red-ripe stage



Market	Distance to farm/km	Transportation cost/\$·kg^{-1}
Seoul	212	0.307
Busan	275	0.399
Daegu	224	0.325
Daejeon	83	0.120
Gwangju	90	0.131

Figure 11 Example of subject markets in South Korea

Table 6 Cost, revenue, and profit of cherry tomatoes selling to five subject markets in South Korea

Category		Market				
		Seoul	Busan	Daegu	Daejeon	Gwangju
Cost	Production/ (\$·m^{-2})[1]	11.19	11.19	11.19	11.19	11.19
	Transportation/ (\$·m^{-2})	1.63	2.11	1.72	0.64	0.69
Revenue/(\$·m^{-2})[2]		25.34	35.74	27.52	22.95	21.25
Profit/(\$·m^{-2})		12.52	22.44	14.61	11.12	9.37

[1] Material cost includes costs of seed, fertilizer, pesticides, labor, overhead, tax, and miscellaneous; [2] It considers historical selling price data in South Korea[33].

4 Discussion

The proposed framework is developed using the cherry tomato data from the National Institute of Horticultural and Herbal Science in South Korea[16]. To accurately detect cherry tomatoes from image data, YOLOv10n, one of the most popular image detection algorithms, is applied to the transfer learning framework. Since there are multiple examples of YOLO with different versions in crop and vegetable detection using inexpensive camera sensors[34–36], engineers can easily apply the proposed framework to detecting other crops and vegetables. Eventually, this standard technology will promote the use of smart farming technologies by engineers and farmers. For example, the proposed framework can be used to recognize tomatoes in real time and predict growth by linking with a monitoring system of crop growth through camera sensors in a green house. Moreover, since YOLOv10n can be operated on a small computer equipped with a microprocessor (e.g., raspberry pi), if the proposed framework is implemented on a robot equipped with

a camera sensor, crop growth can be monitored even in unmanned production facilities (e.g., plant factory). If the proposed framework determines whether cherry tomatoes are ripe, it can be installed on a cultivation robot to automatically harvest cherry tomatoes that have reached the ripening time. To this end, additional learning of the model is required by linking with a database that determines whether cherry tomatoes are ripe.

In addition, the transfer learning with the devised image segmentation technique improves the MAE, RMSE, and MAPE by 37.88%, 38.70%, and 3.27%, respectively, over the non-segmented case. Kaur and Kaur[37] advise that image segmentation for a specific object detection could significantly improve detection accuracy, because it divides an original image into small images having similar features and properties. Therefore, small images can have a unified meaning, instead of an original image with multiple meanings. This is not exceptional for either tomato or cherry tomato detection, which involves fruits, plant stems, leaves, and other background noise[38]. This study shows the optimum segmentation size of an image for accurate cherry tomato detection.

In the economic feasibility analysis of cherry tomatoes, profits in the range of 11.12-22.44 USD/m^2 are identified for the production and transportation costs and revenue. Five major wholesale markets are considered to compute the profit ranges. Unlike existing studies that focus only on their methodological aspects for detection accuracy improvement[9,11,12], the proposed framework includes transfer learning with the economic feasibility module, so that farmers can understand how much cost they must invest to operate and maintain the smart farming technology sustainably. According to the identified profit range, the annual operation and maintenance cost of the proposed framework should be less than 11.12 USD/m^2. This implies that if the framework is durable for five years at a discount rate of 5%, the investment cost (or net present value) of the proposed framework should be less than 48.14 USD/m^2. Therefore, to generate a reasonable profit from cherry tomato production through the use of commercial high-end sensors (or a smart farming system) is challenging. This finding supports the claims that Unmanned Aerial Vehicles (UAVs), Internet of Things (IoT), and Artificial Intelligence (AI) are too expensive to deploy on a mechanized farm[8]. To make the smart farm profitable, those technologies should be implemented in inexpensive devices (e.g., Raspberry pi-based image processing or Arduino-based sensors), so that farmers can easily purchase and operate the smart farming devices without incurring economic burden.

5 Conclusions

This study proposes a CNN-based transfer learning framework for the economic production of cherry tomatoes (*solanum lycopersicum*) in South Korea. As climate change has become a significant issue in harvesting crops and vegetables, it is necessary to use Greenhouse Horticulture to help control the internal environment despite the impact of the external environment. By using smart farming technology in production management at a greenhouse, sustainable crops and vegetables can be realized in practice. The proposed framework comprises four significant modules of cherry tomato monitoring, cherry tomato detection, harvest yield estimation, and economic feasibility analysis. Once cherry tomato growth status is monitored and detected by the framework, yield is estimated based on the typical cherry tomato growth curve. To accurately detect cherry tomatoes using camera sensors, YOLOv10n is applied with the segmentation technique. The experiment result shows that the proposed framework improves the prediction accuracy by 38.7% in terms of the root-mean-square deviation of the existing YOLOv10n. In addition, the harvest yield was estimated by considering the growing curve of cherry tomatoes, and the economic feasibility analysis demonstrated that the profit of cherry tomatoes ranged from 11.12-22.44 USD/m^2 in South Korea. This implies that the annual operation and maintenance cost of the proposed framework should be less than 11.12 USD/m^2. As a result, this study will contribute to the expansion of smart farming technologies, and the provided cost guideline will also be used to improve farmer income.

Although the devised framework accurately estimates the monetary benefits of cherry tomato production using YOLOv10n, more field study cases are needed to develop a reliable cherry tomato detection algorithm. Future studies should consider both the sample size increase of each field study site, and cherry tomato cultivation under various environmental conditions (e.g., humidity, temperature, soil nutrition, and ambient light).

Acknowledgements

This research was supported by a Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (Grant No. RS-2023-00239448). Also, authors gratefully acknowledge the support of the Korea International Cooperation Agency (KOICA). The views expressed in this paper are solely those of the authors and do not represent the opinions of the funding agency.

[References]

[1] Singh S, Singh P, Singh G, Sandhu A S. Crop productivity and energy indices of tomato (*Solanum lycopersicum*) production under naturally-ventilated poly-house structures in north-western India. Energy, 2025; 314: 134239.

[2] Quinet M, Angosto T, Yuste-Lisbona F J, Blanchard-Gros R, Bigot S, Martinez J P, et al. Tomato fruit development and metabolism. Frontiers in Plant Science, 2019; 10: 1554.

[3] Kim D, Shawon, M R A, Lee ., Lee Y, Kim M, Choi K. Effects of drip irrigation volumes on plant growth and yield of tomato grown in perlite. Journal of Bio-Environment Control, 2022; 31(4): 300–310. (in Korean)

[4] Türkten H, Ceyhan V. Environmental efficiency in greenhouse tomato production using soilless farming technology. Journal of Cleaner Production, 2023; 398: 136482.

[5] Rivard C L, Sydorovych O, O'Connell S, Peet M M, Louws F J. An economic analysis of two grafted tomato transplant production systems in the United States. Hort Technology, 2010; 20(4): 794–803.

[6] Guo X X, Zhao D, Zhuang M H, Wang C, Zhang F S. Fertilizer and pesticide reduction in cherry tomato production to achieve multiple environmental benefits in Guangxi, China. Science of the Total Environment, 2021; 793: 148527.

[7] Lee H, Lee J G, Hong K H, Kwon D H, Cho M C, Hwang I, et al. Improving growth and yield in cherry tomato by using rootstocks. Journal of Bio-Environment Control, 2021; 30(3): 196–205. (in Korean)

[8] Idoje G, Dagiuklas T, Iqbal M. Survey for smart farming technologies: Challenges and issues. Computers & Electrical Engineering, 2021; 92: 107104.

[9] Liu G, Nouaze J C, Touko Mbouembe P L, Kim J H. YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3. Sensors, 2020; 20(7): 2145.

[10] Yang D, Ju C. Performance comparison of cherry tomato ripeness detection using multiple YOLO models. Agri Engineering, 2024; 7(1): 8.

[11] Nyalala I, Okinda C, Nyalala L, Makange N, Chao Q, Chao L, et al. Tomato volume and mass estimation using computer vision and machine learning algorithms: Cherry tomato model. Journal of Food Engineering, 2019; 263: 288–298.

[12] Kabas O, Kayakus M, Ünal İ, Moiceanu G. Deformation energy estimation of cherry tomato based on some engineering parameters using machine-

learning algorithms. Applied Sciences, 2023; 13(15): 8906.

[13] Kim S, Kim Y, On Y, So J, Yoon C Y, Kim S. Hybrid performance modeling of an agrophotovoltaic system in South Korea. Energies, 2022; 15(18): 6512.

[14] Jiang P, Ergu D, Liu F, Cai Y, Ma B. A review of Yolo algorithm developments. Procedia Computer Science, 2022; 199: 1066–1073.

[15] Wu Z W, Liu M H, Sun C X, Wang X F. A dataset of tomato fruits images for object detection in the complex lighting environment of plant factories. Data in Brief, 2023; 48: 109291.

[16] Park B M, Jeong H B, Yang E Y, Kim M K, Kim J W, Chae W, et al. Differential responses of cherry tomatoes (*Solanum lycopersicum*) to long-term heat stress. Horticulturae, 2023; 9(3): 343.

[17] Redmon J. You only look once: Unified, real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA: IEEE, 2016; pp.779–788. doi: 10.1109/CVPR.2016.91.

[18] Zhang F, Dong D, Jia X, Guo J, Yu X. Sugarcane-YOLO: An improved YOLOv8 model for accurate identification of sugarcane seed sprouts. Agronomy, 2024; 14(10): 2412.

[19] Felzenszwalb P F, Girshick R B, McAllester D, Ramanan D. Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009; 32(9): 1627–1645.

[20] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, USA: IEEE, 2014; 580–587. doi: 10.1109/CVPR.2014.81.

[21] Ali M L, Zhang Z. The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection. Computers, 2024; 13(12): 336.

[22] Wang A, Chen H, Liu L, Chen K, Lin Z, Han J, Ding G. Yolov10: Real-time end-to-end object detection. In: Thirty-Eighth Annual Conference on Neural Information Processing Systems. Vancouver, Canada: NIPS, 2024; 3429: 107984.

[23] Hussain M, Khanam R. In-depth review of yolov1 to yolov10 variants for enhanced photovoltaic defect detection. Solar, 2024; 4(3): 351–386.

[24] Liu S, Qi L, Qin H, Shi J, Jia J. Path aggregation network for instance segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA: IEEE, 2018; 8759–8768. doi: 10.1109/CVPR.2018.00913.

[25] Lin T Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: IEEE conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA: IEEE, 2017; 2117–2125. doi: 10.1109/CVPR.2017.106.

[26] Rong J, Zhou H, Zhang F, Yuan T, Wang P. Tomato cluster detection and counting using improved YOLOv5 based on RGB-D fusion. Computers and Electronics in Agriculture, 2023; 207: 107741.

[27] Bansal M A, Sharma D R, Kathuria D M. A systematic review on data scarcity problem in deep learning: solution and applications. ACM Computing Surveys (Csur), 2022; 54(10s): 1–29.

[28] Monte J A, Carvalho D F D, Medici L O, da Silva L D, Pimentel C. Growth analysis and yield of tomato crop under different irrigation depths. Soil, Water and Plant Management, 2013; 17: 926–931.

[29] Moccia S, Chiesa A, Oberti A, Tittonell P A. Yield and quality of sequentially grown cherry tomato and lettuce under long-term conventional, low-input and organic soil management systems. European Journal of Horticultural Science, 2006; 71(4): 183–191.

[30] Shabbir A, Mao H, Ullah I, Buttar N A, Ajmal M, Lakhiar I A. Effects of drip irrigation emitter density with various irrigation levels on physiological parameters, root, yield, and quality of cherry tomato. Agronomy, 2020; 10(11): 1685.

[31] Joubes J, Phan T H, Just D, Rothan C, Bergounioux C, Raymond P, et al. Molecular and biochemical characterization of the involvement of cyclin-dependent kinase A during the early development of tomato fruit. Plant Physiology, 1999; 121(3): 857–869.

[32] Kim Y, Kim S, Kim S. An integrated agent-based simulation modeling framework for sustainable production of an Agrophotovoltaic system. Journal of Cleaner Production, 2023; 420: 138307.

[33] Nongnet. Sales price in whole sale markets. https://www.nongnet.or.kr/front/M000000197/content/view.do?pumCd=0806. Accessed on [2023-08-14].

[34] Junos M H, Mohd Khairuddin A S, Thannirmalai S, Dahari M. An optimized YOLO‐based object detection model for crop harvesting system. IET Image Processing, 2021; 15(9): 2112–2125.

[35] Tian Y, Yang G, Wang Z, Wang H, Li E, Liang Z. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Computers and Electronics in Agriculture, 2019; 157: 417–426.

[36] Wang X, Liu J. Tomato anomalies detection in greenhouse scenarios based on YOLO-Dense. Frontiers in Plant Science, 2021; 12: 634103.

[37] Kaur D, Kaur Y. Various image segmentation techniques: A review. International Journal of Computer Science and Mobile Computing, 2014; 3(5): 809–814.

[38] Xiang, R. Image segmentation for whole tomato plant recognition at night. Computers and Electronics in Agriculture, 2018; 154: 434–442.