RT-DETR-Tomato: Tomato Target Detection Algorithm Based on Improved RT-DETR for Agricultural Safety Production

Zhimin Zhao 1, Shuo Chen 1, Yuheng Ge 1, Penghao Yang 1, Yunkun Wang 1 and Yunsheng Song 1,2,*

1 School of Information Science and Engineering, Shandong Agricultural University, Taian 271018, China; zzm_030114@163.com (Z.Z.); cywscs@outlook.com (S.C.); geyuheng1219@gmail.com (Y.G.); 15969828931@163.com (P.Y.); wyk020831@126.com (Y.W.)
2 Key Laboratory of Huang-Huai-Hai Smart Agricultural Technology of Ministry of Agriculture and Rural Affairs, Shandong Agricultural University, Taian 271018, China
* Correspondence: sys_sd@126.com

Abstract: The detection of tomatoes is of vital importance for enhancing production efficiency, with image recognition-based tomato detection methods being the primary approach. However, these methods face challenges such as the difficulty in extracting small targets, low detection accuracy, and slow processing speeds. Therefore, this paper proposes an improved RT-DETR-Tomato model for efficient tomato detection under complex environmental conditions. The model mainly consists of a Swin Transformer block, a BiFormer module, path merging, multi-scale convolutional layers, and fully connected layers. In this proposed model, Swin Transformer is chosen as the new backbone network to replace ResNet50 because of its superior ability to capture broader global dependency relationships and contextual information. Meanwhile, a lightweight BiFormer block is adopted in Swin Transformer to reduce computational complexity through content-aware flexible computation allocation. Experimental results show that the average accuracy of the final RT-DETR-Tomato model is greatly improved compared to the original model, and the model training time is greatly reduced, demonstrating better environmental adaptability. In the future, the RT-DETR-Tomato model can be integrated with intelligent patrol and picking robots, enabling precise identification of crops and ensuring the safety of crops and the smooth progress of agricultural production.

Keywords: tomato target detection; RT-DETR; Swin Transformer; attention mechanism; feature fusion; small target detection; computer vision

1. Introduction

Tomatoes not only meet the daily nutritional needs of humans but also hold substantial economic value, making significant contributions to the economies of many regions and countries. As one of the most important vegetables globally, accurate yield detection of tomatoes is crucial for farmers and food processing enterprises. Tomato target detection technology can precisely monitor the growth status and yield of tomatoes. This technology not only enhances production efficiency and fruit quality but also effectively protects the agricultural ecological environment, playing a critical role in ensuring crop safety and the smooth progress of agricultural production.

Traditional tomato target detection methods typically rely on manual sampling and visual judgment. Workers usually estimate tomato yields based on characteristics such as size and density. However, this approach is not only time-consuming and labor-intensive but also prone to subjective biases, potentially leading to erroneous decisions [1,2]. These issues have become bottlenecks restricting the sustainable development of the tomato industry. Currently, machine vision has matured and is applied in both traditional and modern industries, such as express sorting [3] and road vehicle detection [4]. To address the inefficiencies of manual observation, many researchers have applied machine vision to tomato target detection. For instance, Yamamoto et al. [5] combined pixel-based segmentation...
and blob-based segmentation strategies for tomato detection, employing decision tree and random forest classifiers, achieving a recall rate of 80% and a precision rate of 88%. Zhao et al. [6] used an AdaBoost classifier combined with color analysis for tomato detection, employing Haar-like features to train the classifier. Although this method yields reasonable results, its speed is relatively low, failing to meet the real-time requirements of harvesting robots. Luo et al. [7] also proposed a grape clustering detection framework based on AdaBoost and color features. Their experimental results indicated that this method could partially mitigate the effects of weather conditions, leaf occlusion, and lighting changes. Liu et al. [8] proposed a coarse-to-fine framework for ripe tomato detection, using support vector machines and pseudo-color removal methods, achieving recall and precision rates of 90.00% and 94.41%, respectively. However, this method is unsatisfactory for overlapping and occluded tomatoes. Traditional tomato detection methods suffer from long detection times and poor robustness.

In recent years, deep learning technology has achieved significant breakthroughs in the field of computer vision, particularly in object detection. Deep learning-based object detection offers the advantages of short detection times and high accuracy, greatly meeting the requirements for real-time detection in complex environments. Currently, mainstream deep learning models include a series of YOLO models such as YOLOv6 [9], YOLOv7 [10], and YOLOv8. However, there are still some shortcomings in the specific field of tomato detection: due to design trade-offs, their performance in detecting small objects is not as good as models specifically optimized for small object detection; in tomato fields, where tomatoes grow densely and occlude each other, YOLO models may experience issues with missed detections or inaccurate detection of overlapping targets; tomatoes often grow in complex natural environments with leaves, stems, and other plants, which can interfere with target detection; finally, in resource-constrained environments, models need further optimization to meet computational and storage limitations. RE-DETR offers a balance between speed and accuracy, allowing direct object prediction from images without the need for predefined anchor boxes or candidate boxes. This approach reduces computational costs and avoids misdetections and missed detections caused by incorrect box selection. Additionally, DETR can handle objects of different sizes and quantities and can directly output object feature vectors, which can be used for subsequent tasks such as target tracking. To address its limitations in feature extraction, we combine it with Swin Transformer [11] and BiFormer Attention [12].

Based on the above considerations, this paper proposes an RT-DETR-Tomato target detection algorithm to improve the efficiency and speed of RE-DETR in detecting tomatoes. In tomato detection, the objects often lack feature expression due to issues such as occlusion, small target size, and few effective pixels [13–15]. Consequently, some tomatoes may go undetected in practical scenarios. Based on the aforementioned analysis, we innovatively propose the RT-DETR-Tomato model for tomato target detection to achieve high efficiency and reliability in natural environment tomato target detection. The RT-DETR-Tomato detection model is designed for tomato detection under complex environmental conditions, with the main innovations as follows to improve detection performance:

1. By replacing the original ResNet50 backbone of the RE-DETR model with the Swin Transformer, the model’s detection accuracy was improved.
2. By integrating the BiFormer Vision Transformer with dual-level routing attention, the small object feature extraction capability was enhanced, leading to improved model performance and high computational efficiency.

The remainder of this paper is structured as follows: Section 2 focuses on related work in tomato target detection. Section 3 describes the model framework and implementation details. Section 4 verifies the effectiveness of the algorithm through experiments. Section 5 provides a summary.
2. Related Work

In natural environments, tomato growth conditions are complex, and the accuracy of tomato detection is affected by factors such as leaf and branch occlusion, varying ripeness, overlapping clusters [16], different densities, and loss of tomato characteristics due to natural lighting changes [17]. Therefore, improving the accuracy and robustness of tomato detection in natural environments is a critical technical challenge.

Over the years, the development of computer vision systems for intelligent fruit and vegetable detection, akin to human capabilities, has undergone several stages [18,19]. Bulanon et al. [20] utilized brightness and RGB color difference models for color-based segmentation to identify apples, while Mao et al. [21] experimented with color indices for segmenting apples from their surroundings. Yin et al. [22] utilized the L*a*b* color space to extract features of ripe tomatoes, and Wei et al. [23] employed color-based segmentation methods to extract fruits from their backgrounds. However, selecting the optimal color model for real-life scenarios remains challenging due to the reliance of fruit detection on the effectiveness of the color space used. Additionally, Kurtulmus et al. [24] achieved a detection accuracy of 75% for green oranges in natural outdoor conditions by combining circular Gabor texture features and several fixed threshold features. Changes in lighting conditions significantly affect the classification of green apples and the classifier based on color and texture information between detected circles and Linker’s heuristic model [25]. Payne et al.’s improved algorithm [26] for mango crop yield estimation based on color and texture was restricted by artificial lighting, and Kelman et al. [27] pointed out the influence of lighting and leaves on the obtained results regarding ripe apple localization. Zhao et al. [28] segmented ripe tomatoes from the background using the optimal threshold with fused image features, achieving an accuracy of 93%, albeit susceptible to lighting conditions. Looking ahead, researchers have attempted to use various sensors for fruit detection to overcome lighting variations and occlusion issues [29–31].

Machine learning has been incorporated into computer vision tasks in agriculture to promote growth and development. Early fruit recognition relied on machine learning [32], necessitating the design of human-customized features, a highly complex process. Some methods utilized combined techniques such as [16] limited contrast adaptive histogram equalization (CLAHE), red–blue mapping, OTSU thresholding, and morphological operations for image segmentation. These methods were applied to fruits of different ripeness levels but needed more detection accuracy and real-time performance. Traditional digital image detection methods were based on color extraction and analysis [33,34], shape [5,35], and texture [36,37]. For instance, Qiang et al. [38] achieved an accuracy of 92.4% in identifying fruits and branches in natural scenes using a support vector machine trained solely on the RGB color space but were susceptible to lighting effects. Various classifiers such as AdaBoost and Haar-like features were used for tomato detection in greenhouse scenarios [6], albeit with relatively poor real-time performance and low speed. Support vector machine (SVM) [39,40] and K-means clustering algorithms [41,42] were used to remove image backgrounds based on RGB color channel components and detect red tomatoes using region-growing methods. Kurtulmus et al. [43] employed support vector machine neural networks with different classifiers to detect immature peaches.

Thus far, improving traditional machine learning in computer vision has significantly succeeded. In recent years, the limitations of machine learning have been addressed with the introduction of deep learning into computer vision tasks. Deep learning based on convolutional neural networks (CNNs) has made significant advances, offering advantages in efficiency and accuracy over traditional machine learning. For instance, based on improving the MobileNetV2 model [44], the application of transfer learning achieved a fruit image classification accuracy of 99%, highlighting the significant improvement in the proficiency of deep learning-based object detection algorithms [45–48]. Therefore, the latest object detection algorithms in deep learning have also been applied to tomato yield detection. Zheng et al. [49] constructed the tomato detection model RC-YOLOv4, improving the detection accuracy of tomatoes in natural environments. Rong et al. [50]
proposed an improved tomato cluster counting method that combines object detection, multi-object tracking, and specific tracking area counting to reduce the misidentification of background tomatoes, proposing YOLOv5-4D with fused RGB and depth images as inputs.

Although the aforementioned studies using color or shape features and machine learning have made some progress in fruit detection, there are still some issues with tomato detection: (1) changes in natural lighting conditions have a significant impact on the color features of tomatoes, leading to unstable performance of color-based detection methods; (2) leaf and branch occlusion, overlapping between tomatoes, and cluster growth make it difficult for detection algorithms to identify tomatoes accurately; (3) improvement in small target detection has not been achieved; (4) although some methods have improved detection accuracy, the detection speed cannot meet real-time requirements.

This study proposes an RT-DETR-Tomato model for tomato object detection, aiming to balance detection speed and accuracy in complex environments. The main improvements of this model are as follows: (1) reconstruction of the RT-DETR model backbone: By replacing the original ResNet50 backbone of the RT-DETR model with the Swin Transformer, the model can capture richer global dependencies and contextual information due to its hierarchical Transformer structure. This capability allows it to process complex images and scenes with a stronger feature representation compared to traditional convolutional networks, and to handle features at different resolutions, thereby enhancing the model’s detection accuracy. (2) Integration of the BiFormer Vision Transformer with dual-level routing attention: The BiFormer enhances the ability to extract features of small objects. The BiFormer technology achieves more flexible, content-aware computation allocation. Additionally, since BiFormer focuses on a small subset of relevant tokens in a query-adaptive manner without dispersing attention to irrelevant ones, it ensures good performance and high computational efficiency. This work not only provides an effective solution for tomato detection but also offers a reference for developing detection technologies for other crops, promoting the advancement of agricultural automation and intelligence.

3. Methods
3.1. RT-DETR-Tomato Model

RT-DETR-Tomato-S model: We replaced the original ResNet50 backbone of the RT-DETR model with the Swin Transformer. Due to its hierarchical Transformer structure, the Swin Transformer can capture richer global dependencies and contextual information. This capability allows it to handle complex images and scenes with a stronger feature representation compared to traditional convolutional networks, and to process features at different resolutions, thereby improving the model’s detection accuracy.

RT-DETR-Tomato-B model: We introduced the BiFormer Vision Transformer with dual-level routing attention to enhance the feature extraction capability for small objects. BiFormer technology achieves more flexible, content-aware computation allocation. Moreover, since BiFormer focuses on a small subset of relevant tokens in a query-adaptive manner without dispersing attention to irrelevant ones, it improves the model’s performance and computational efficiency.

Both of these modified models demonstrate significant improvements over the original RT-DETR model. To further validate this, we simultaneously incorporated the above two modification strategies into the original model and compared it with RT-DETR, RT-DETR-Tomato-S, and RT-DETR-Tomato-B. We found that the model performance with both modification strategies was superior to that of the models with only one modification strategy. Through comparative validation, we ultimately obtained the most effective model, the RT-DETR-Tomato-BS model.

The detailed pseudo-code for RT-DETR-Tomato-BS is provided to describe the implementation process more clearly, as follows.
RT-DETR-Tomato-BS

Input: original image $I$.
Output: Final detection and tracking results.

1. Swin Transformer feature extraction representation: $f_{Swin}(I)$.
2. BiFormer block processing feature representation: $BB(I) = f_{BiFormer}(f_{Swin}(I))$.
3. Transformer Encoder generates embedding vector: $TE(I) = f_{Transformer}(BB(I))$.
4. The target matching result $M$ obtained by the Hungarian algorithm:
   $M = \text{HungarianMatching}(E_{\text{current}}, E_{\text{previous}})$, where $E_{\text{current}}$: The embedding vector of the current frame, $E_{\text{previous}}$: Embedding vector of the previous frame.
5. Output final detection and tracking results Return $\text{GenerateDetectionTracking}(TE(I), M)$.

This paper presents the RT-DETR tomato object detection model (RT-DETR-Tomato-BS) by integrating the Swin Transformer structure with the BiFormer module, as shown in Figure 1. The model primarily comprises a Swin Transformer block, BiFormer module, path merging, multi-scale convolution layers, and fully connected layers. Images are first input into the backbone network for feature extraction. Addressing the limited extraction capability of ResNet50, the Swin Transformer is chosen as the new backbone network due to its superior ability to capture extensive global dependencies and contextual information. However, the Swin Transformer block within the backbone network has redundant parameters, so the lightweight BiFormer block is utilized. This module effectively reduces computational complexity through content-aware flexible computation allocation. Meanwhile, the BiFormer block retains a token-to-token attention mechanism within its routing area, significantly enhancing the model’s sensitivity to small object detection. This optimization is particularly effective for detecting small targets in the dataset used in this paper, thereby achieving optimal results during the feature extraction process. The architecture of the RT-DETR-BS model is illustrated in Figure 1:

![Figure 1. Architecture of the RT-DETR-Tomato-BS Model.](image)

### 3.2. Swin Transformer

In object detection networks, the backbone network plays several critical roles. Its primary function is to extract useful features from raw image data. This is typically achieved through multiple layers of convolutional networks, with each layer capturing different levels of detail and abstract features in the image, ranging from bare edges and textures
to more complex object parts. The original RT-DETR selects different sizes of backbone networks based on the model size to extract image feature information. In this paper, ResNet50 is used as the comparison backbone network model. ResNet50 was initially proposed by Kaiming He et al. [51]. This network design addresses the training difficulties that arise with increasing network depth, particularly the problems of vanishing and exploding gradients. The core of ResNet is the residual learning block (Residual Block). Additionally, skip connections allow inputs to connect to the next layer, skip one or more layers, and connect directly to deeper layers. This connection strategy facilitates more effective gradient flow during training, thereby resolving the vanishing gradient problem in deep network training. The architecture of ResNet50 is introduced in detail next, with its architectural diagram shown in Figure 2:

![Figure 2. Architecture of ResNet50.](image)

The figure shows that ResNet50 consists of 50 convolutional and fully connected layers. However, this structure has certain drawbacks. Although residual connections help alleviate the vanishing gradient problem and facilitate the training of deeper networks, the complex structure of ResNet50 can lead to overfitting when the data volume is insufficient. This means the model performs well on the training data but poorly on unseen data. In profound networks, information becomes excessively smoothed after passing through many layers, especially with continuous residual connections, potentially causing the loss of some important feature information. This issue can be problematic in object detection tasks that require capturing fine details. To address this shortcoming, this paper replaces the backbone network with one based on the Transformer framework [52]. The Transformer handles sequential data through a self-attention mechanism, supports parallel computation, and can capture long-range dependencies. It was initially introduced in the field of natural language processing. Swin Transformer [11] brings this technology into the computer vision field and achieves remarkable results. The structure of the Swin Transformer is shown in Figure 3:

![Figure 3. Architecture of the Swin Transformer.](image)

In the Swin Transformer, the image undergoes multiple feature extraction and refinement steps, utilizing patch partition: the image is first divided into $4 \times 4$ pixel patches. Each patch is flattened along the channel dimension to form the initial feature representation. Convolutional layer (as linear embedding layer): A convolutional layer is used as the linear embedding layer to transform these features into a higher-dimensional feature space $C$ to enhance the feature representation capability.
Processing through four stages: Each stage includes a patch merging layer and paired Swin Transformer blocks. The structure of the Swin Transformer block is shown in Figure 4:

![Figure 4. Structure of the Swin Transformer block.](image)

The Swin Transformer block employs a multi-head self-attention module based on shifted windows (W-MSA/SW-MSA), followed by a 2-layer MLP with GeLU non-linearity in between. LayerNorm (LN) is applied before each MSA module and each MLP, and a residual connection is applied after each module. W-MSA is more efficient compared to global self-attention MSA, with the computational complexities of the two self-attention mechanisms being as follows:

\[
\Omega(MSA) = 4hwC^2 + 2(hw)^2C \\
\Omega(W-MSA) = 4hwC^2 + 2M^2hwC
\]

MSA has quadratic complexity concerning the number of patch tokens \(h \times w\) (with \(h \times w\) patch tokens, each patch token computes \(h \times w\) times globally). In contrast, when \(M\) is fixed, W-MSA is set to 7 by default and has linear complexity (with \(h \times w\) patch tokens, each patch token computes \(M^2\) times within its respective local window). The immense \(h \times w\) is unbearable for global self-attention computation, while window-based self-attention (W-MSA) has good scalability. These stages progressively process the image to extract and refine features. Patch merging layer: This layer merges each 2 \(\times\) 2 pixel block into a new patch and concatenates in the depth direction to downsample the feature map, halving its height and width while doubling the feature depth. Then, LayerNorm and a fully connected layer are used to adjust the feature depth further, optimizing the feature representation. The Swin Transformer block: This block includes W-MSA (window multi-head self-attention) and SW-MSA (shifted window multi-head self-attention) structures. The W-MSA module divides the feature map into windows of size \(M \times M\) and performs self-attention processing within each window independently. However, W-MSA only processes self-attention within each window, meaning there is no information exchange between windows. To address this limitation, the SW-MSA module is introduced, allowing information to be exchanged between windows and enhancing the overall information flow in the model. The model can effectively reduce the computational load through this structural design while ensuring effective information transfer between different stages, enhancing feature expression capabilities and model performance. The patch merging layer, W-MSA, and SW-MSA modules are shown in Figure 5:
By replacing the traditional ResNet50 with Swin Transformer as the network backbone, the model’s ability to handle complex image scenes has been significantly enhanced, particularly demonstrating outstanding performance in processing dense and stacked tomato images within the dataset. Swin Transformer adopts a hierarchical Transformer structure, enabling it to capture richer global dependencies and contextual information, which is crucial for understanding and analyzing densely stacked object scenes. Furthermore, the structure of the Swin Transformer allows for feature processing at multiple resolutions, achieved through intra-window self-attention mechanisms and inter-window connections. This flexibility supports multi-scale feature extraction, essential for handling objects of different sizes in object detection tasks. Compared to traditional convolutional networks based on fixed convolutional kernel structures, Swin Transformer is not constrained by convolution operations, thus allowing for more flexible learning of underlying patterns and complex geometric structures within the data. This replacement of the backbone network significantly improves the model’s feature extraction capability, leading to a substantial enhancement in the model’s accuracy and efficiency in practical applications. This improvement strengthens the model’s ability to handle complex scenes and enhances its adaptability and accuracy across various application scenarios, particularly excelling in challenging object detection and image recognition tasks.

3.3. Integrating the BiFormer Module to Enhance Small Object Feature Extraction Capability

Although traditional attention mechanisms excel at capturing contextual semantic information from feature maps, they impose a significant computational burden and heavy memory usage due to the need to compute pairwise token interactions across all spatial positions. The BiFormer technology [12] introduced in this paper, which employs a dual-level routing attention mechanism, achieves more flexible computation allocation with content-awareness. Due to its ability to adaptively focus on a small subset of relevant tokens without dispersing attention to other irrelevant tokens, BiFormer exhibits good performance and high computational efficiency. Therefore, it provides a viable solution to the problem of the complex redundancy of Swin Transformer modules, resulting in substantial parameter and computational overhead in the overall model. A lightweight effect is achieved by replacing Swin Transformer blocks with BiFormer blocks. The specific structure of the BiFormer block is illustrated in Figure 6;
Figure 6. Framework of the BiFormer block.

In this paper, the tomato dataset contains relatively small objects compared to the entire image, constituting a task of small object detection. Therefore, focusing on the study of feature enhancement for small objects is essential. Small object detection has always been a challenging task in object detection because small objects occupy few pixels in the image, making their visual features less distinct and difficult to distinguish and recognize. This also means that the detailed information on small objects is minimal, leading to more difficult classification and localization. Small objects are easily occluded by other objects or backgrounds, especially in dense scenes. Additionally, background noise can resemble the appearance of small objects, increasing the risk of false positives. The size of objects also varies significantly due to factors such as distance and viewpoint. This scale variation poses a particular challenge for small object detection because as the scale decreases, less information is available for detection. The BRA module primarily constructs the BiFormer block here. Its core idea is to filter out the most irrelevant key-value pairs at a coarse region level, retaining only a small number of routing regions. Then, fine-grained token-to-token attention is applied within these routing regions. For example, when inputting an image of a tomato into the backbone network and obtaining feature maps through feature extraction at each layer module, BRA first divides the obtained feature maps into S*S non-overlapping regions and applies the mapping attention mechanism of QKV, as shown in Equation (2):

\[
Q = X^q W^q, K = X^k W^k, V = X^v W^v
\]  

(3)

\(W^q, W^k, \) and \(W^v\) are the query, key, and value projection weights, respectively. Then, attention weights are calculated on coarse-grained tokens, and only the ‘Topen’ regions are selected as relevant regions to participate in fine-grained computations. By transposing \(Q\) and \(K\), \(A^r\) is obtained, as shown in Equation (3):

\[
A^r = Q^r (K^r)^T
\]  

(4)

Finally, the Topk coarse-grained regions most relevant to each token are selected as keys and values to participate in the final computation. To enhance locality, a deep convolution is applied to the values. Figure 7 illustrates the implementation process of BRA \[12\]:
Figure 7. Schematic representation of the specific implementation module of BRA.

Furthermore, integrating this structure into the Swin Transformer module has demonstrated unique advantages in handling small tomato object detection. In this paper, the tomato dataset contains relatively small objects compared to the entire image, constituting a task of small object detection. Therefore, focusing on the study of feature enhancement for small objects is essential. Small object detection has always been challenging in object detection because small objects occupy few pixels in the image, making their visual features less distinct and difficult to distinguish and recognize. This also means that the detailed information on small objects is minimal, leading to more difficult classification and localization. Small objects are easily occluded by other objects or backgrounds, especially in dense scenes. Additionally, background noise can resemble the appearance of small objects, increasing the risk of false positives. The size of objects also varies significantly due to factors such as distance and viewpoint. This scale variation poses a challenge for small object detection because less information is available for detection as the scale decreases.

Firstly, dynamic sparse attention enhances the model’s content-awareness. The BiFormer dynamically selects attention regions through bi-level routing rather than fixed patterns. This dynamic nature allows the model to adaptively select the most relevant regions for in-depth analysis based on input features. For small objects, this means the model can more accurately focus on small regions containing the target rather than uniformly distributing attention across a large area, which is crucial for capturing subtle features of small objects. Region-to-region routing ensures efficient information flow by ensuring that only pre-selected, highly relevant regions exchange information through directed graphs and adjacency matrices. This fine-grained information flow is more suitable for capturing small objects dispersed in the image, as the attention for each region is optimized, focusing on specific areas containing helpful information.

Regarding optimized performance and efficiency, BiFormer reduces memory usage and computational resource consumption by skipping calculations for many irrelevant regions due to its sparse operations while maintaining high performance. This is particularly important for processing large numbers of images and real-time detection scenarios, such as detecting tomato diseases in agricultural applications. The token-to-token attention within the retained routing regions enhances the model’s ability to capture finer details. This fine-grained attention mechanism is particularly suitable for small object detection as it can provide detailed analysis of the target’s details, helping to improve detection accuracy and reduce omissions. Lastly, the adaptive multi-scale processing capability enables flexible handling of targets of different scales. BiFormer can handle multi-scale inputs through its structural design, making it more effective in detecting targets of different sizes, especially in common dense tomato detection scenarios in agricultural images.
4. Results and Discussion

4.1. Dataset Construction

The tomato dataset used in this study was collected from the Big Data Laboratory of Shandong Agricultural University, where tomatoes are cultivated. The images were captured using a digital commercial camera with a resolution of 3968 × 2976 pixels in the RGB color space and saved in JPG format. All images were taken under natural daylight conditions, including the complexity of the growing environment: changes in lighting, occlusions, and overlaps. This significantly increased the difficulty of tomato yield detection. Under natural daylight conditions, 425 tomato images were captured, which were then divided into an 80% training set and a 20% test set to create a dataset representing natural scenes. The ground truth bounding boxes for all objects in each image in the dataset were manually annotated using the graphical image annotation tool LabelImg (https://github.com/tzutalin/labelImg (accessed on 2 March 2024)), and the annotations were saved in YOLO format. The captured images are categorized into single unobstructed targets, single targets occluded by foliage, and multiple targets with or without occlusion. The main purpose of having a diverse dataset is to simulate real-world conditions. When the model is exposed to a broader range of data categories, it can learn richer and more comprehensive features, enabling it to better classify and recognize new, unseen data. This capability enhancement contributes to the model’s stability and reliability in practical applications. Additionally, a diverse set of data categories helps prevent the model from overfitting noise or specific features in the training data, thereby improving its performance on unknown data. Finally, a variety of data categories means the model needs to adapt to more variations and disturbances, which helps improve its robustness against noise and outliers in the input data. In the real world, data are often filled with uncertainties, so training a model that is highly resistant to interference is extremely valuable. Furthermore, to increase the diversity and robustness of the dataset, data augmentation methods are used to expand the training set. Pre-training data augmentation includes operations such as random cropping, rotation, scaling, and flipping. These operations can generate more training samples and enhance the model’s generalization and robustness. To avoid confounding algorithm comparisons due to data augmentation, we incorporated data augmentation uniformly within the training framework. Figure 8 shows some captured images under different environmental conditions. Finally, the training set consisted of 340 images containing 1253 tomatoes, while the remaining 85 images containing 512 tomatoes formed the test set.

![Figure 8. Tomato image samples under different environments in the natural scene dataset: (a) Single target without occlusion, (b) Multiple targets with occlusion, (c) tomato cluster, (d) enhanced lighting, (e) diminished lighting, and (f) multiple targets with or without occlusion.](image-url)
4.2. Experimental Platform and Evaluation

This study utilized the Ubuntu 20.04.5 operating system, NVIDIA GeForce RTX3060 GPU, CUDA 11.1, Python 3.8.8, and PyTorch 1.8.0. The dataset was randomly divided into training and testing data in an 8:2 ratio. The SGD training optimizer was employed with a batch size of 24, an initial learning rate of 0.01, 200 training epochs, and an input image size of $640 \times 640$. For the batch size, we started with a smaller batch size and then determined the optimal value based on the model’s performance, resource constraints, and multiple experiments. For the initial learning rate, we began testing with 0.1 and gradually decreased the learning rate during training until we achieved the desired result. Regarding the number of epochs, we conducted extensive experiments, allowing the training to stop when the model’s performance on the validation set no longer improved or began to decline, ultimately determining the optimal number of epochs. For the input image size, we chose $640 \times 640$, which provided more details and helped the model learn complex features. Selecting these parameters required extensive experimentation to evaluate the model’s performance under different parameter combinations. Additionally, we referenced several studies, to guide our choices [53].

The Intersection over Union (IoU) [54] is a standard for dividing positive and negative samples in object detection tasks. Precision and recall are two standard metrics for evaluating model performance [55]. These three metrics are calculated by Equations (5)–(7). In Equation (5), the numerator represents the overlap area between the predicted and ground truth boxes. In contrast, the denominator represents the total area of the predicted box and the ground truth box. In Equations (6) and (7), $T$ or $F$ indicates whether the sample is correctly classified, and $P$ or $N$ indicates whether the sample is predicted as positive or negative. Specifically, $TP$ (true positive) means tomatoes are successfully identified and correctly classified, $TN$ (true negative) represents the correct classification of the background, $FP$ (false positive) means the background is incorrectly classified as tomatoes, and $FN$ (false negative) means tomatoes are incorrectly classified as background. However, the choice of confidence threshold limits the evaluation of individual metrics. Fortunately, the mean average precision ($mAP$) has been proven to be a more accurate measure [56,57] and is used in experiments. $AP$ and $mAP$ are calculated by Equations (8) and (9), respectively.

The commonly used metric, $F_1$ score, was employed to measure the accuracy of classification. The $F_1$ score is a comprehensive performance indicator of precision and recall and can be calculated by Equation (10).

\[
\text{IoU} = \frac{\text{area of overlap}}{\text{area of union}} \\
\text{Precision} = \frac{TP}{TP + FP} \\
\text{Recall} = \frac{TP}{TP + FN} \\
AP = \frac{1}{11} \sum_{r=0,0.1,0.2,...,1.0} \text{Precision}(r) \\
mAP = \frac{1}{N_{\text{class}}} \sum AP \\
F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

4.3. Model Performance

To maintain consistency with the resolution of training images, the trained models RT-DETR, RT-DETR-Tomato-S, RT-DETR-Tomato-B, and RT-DETR-Tomato-BS were tested and compared using images with a resolution of $640 \times 640$ pixels. Natural scenes are more reliable because they occur every day. The experimental results of natural scene contrast are
shown in Table 1. The best weights of the models are provided for analysis. The training time for the RT-DETR model is 2.057 h, for the RT-DETR-Tomato-S model is 2.633 h, for the RT-DETR-Tomato-B model is 1.091 h, and for the RT-DETR-Tomato-BS model is 1.305 h. We trained each model for 200 epochs, recording and analyzing the training logs. All algorithms began to converge around 70 epochs, with the RT-DETR-Tomato-BS model converging slightly later, starting at 90 epochs. When the models began to converge, the loss reduction tended to stabilize, and the train and validation loss values decreased in the same trend, without any signs of overfitting. Meanwhile, various performance metrics, such as mAP, increased slowly and steadily, indicating normal model training. Considering both training costs and model performance, we determined that 120 epochs were suitable for achieving an optimal model. The training process is illustrated in Figure 9. All models detected the number of tomatoes in the test dataset, including images affected by lighting and occlusion, and achieved good detection results. However, there are differences in the results obtained from different test models.

<table>
<thead>
<tr>
<th>Models</th>
<th>IoU (%)</th>
<th>P (%)</th>
<th>R (%)</th>
<th>F (%)</th>
<th>mAP50 (%)</th>
<th>mAP50-95 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT-DETR</td>
<td>78.3</td>
<td>82.5</td>
<td>80.0</td>
<td>81.2</td>
<td>85.4</td>
<td>44.8</td>
</tr>
<tr>
<td>RT-DETR-Tomato-S</td>
<td>78.8</td>
<td>84.4</td>
<td>82.3</td>
<td>83.3</td>
<td>88.0</td>
<td>49.0</td>
</tr>
<tr>
<td>RT-DETR-Tomato-B</td>
<td>79.0</td>
<td>84.1</td>
<td>82.7</td>
<td>83.2</td>
<td>87.4</td>
<td>48.3</td>
</tr>
<tr>
<td>RT-DETR-Tomato-BS</td>
<td>78.8</td>
<td>85.3</td>
<td>82.9</td>
<td>84.1</td>
<td>88.7</td>
<td>50.3</td>
</tr>
</tbody>
</table>

Based on the results of natural scenes in Table 1, the RT-DETR-Tomato detection model significantly improves recall and precision, increasing the F1 score. Compared to the RT-DETR base model, the F1 score of RT-DETR-Tomato-B increased from 81.2% to 84.1%. The improvement is attributed to the self-attention mechanism within the Swin Transformer window and the cross-window connections, which support multi-scale feature extraction. mAP50 is considered more accurate than the F1 score as it displays the global precision–recall relationship. Therefore, the mAP50 values of RT-DETR-Tomato-BS are approximately 3.3%, 0.7%, and 1.3% higher than those of RT-DETR, RT-DETR-Tomato-S, and RT-DETR-Tomato-B, respectively, indicating the need for further research. In this case, the average AP results obtained from Table 1 can be used for comparison. It can be concluded that “RT-DETR-Tomato-BS > RT-DETR-Tomato-S > RT-DETR-Tomato-B > RT-DETR”. Furthermore, all models tested on the natural scene dataset have an IoU of more than 50%, which is considered a good prediction.
4.4. Performance Visualization

The improved RT-DETR model visualization results are presented, showing the detected tomatoes. Compared to RT-DETR-Tomato-S and RT-DETR-Tomato-B, RT-DETR-Tomato-BS identifies missed tomatoes, as shown in Figure 10. Moreover, there is a significant difference between RT-DETR and RT-DETR-Tomato-BS. In natural scene scenarios, the RT-DETR-Tomato model substantially improves over the RT-DETR model. It can be explained that the RT-DETR-Tomato-BS model can be used for both small and large tomato detection.

Figure 10. Comparison of tomato image detection between RT-DETR and RT-DETR-Tomato-BS models based on the natural scene dataset.

4.5. Different Algorithms Comparison

To validate the proposed RT-DETR-Tomato-BS model, we conducted a comparative analysis with other improved models and the baseline model to assess its performance. The evaluation metrics used for comparison were precision, recall, F1 score, and mAP50, as shown in Table 1. The mAP50 values for the RT-DETR-Tomato-S model (88.0%), RT-DETR-Tomato-B model (87.4%), and RT-DETR-Tomato-BS model (88.7%) demonstrated that each of these models achieved the highest recall, precision, and F1 scores in comparison to the baseline RT-DETR, indicating the superiority of the proposed method. Additionally, the P-R curves in Figure 11 show that the AUC for the models follows the order RT-DETR-Tomato-BS > RT-DETR-Tomato-S > RT-DETR-Tomato-B > RT-DETR. At the same time, the precision curves in Figure 12 also show the results as RT-DETR-Tomato-BS > RT-DETR-Tomato-S > RT-DETR-Tomato-B > RT-DETR.

Figure 11. P–R curves of different methods for ablation study.
4.6. Comparison of Computational Costs for Different Models

To compare and analyze the computational costs of the four models, RT-DETR, RT-DETR-Tomato-B, RT-DETR-Tomato-S, and RT-DETR-Tomato-BS, we can consider multiple aspects, including the model training time, resource consumption, and detection accuracy. The following is a detailed cost comparison analysis of each model:

1. Model Training Time
   The training times for RT-DETR, RT-DETR-Tomato-B, RT-DETR-Tomato-S, and RT-DETR-Tomato-BS models are 2.057 h, 2.633 h, 1.091 h, and 1.305 h, respectively. The RT-DETR-Tomato-S model uses the Swin Transformer to replace the original ResNet50 backbone of the RE-DETR model; the RT-DETR-Tomato-B model introduces the BiFormer Vision Transformer with dual-level routing attention to enhance small object feature extraction capability. The RT-DETR-Tomato-BS model replaces the original ResNet50 backbone of the RE-DETR model with the Swin Transformer and simultaneously introduces the BiFormer Vision Transformer to enhance small object feature extraction capability. Shorter training times typically indicate lower computational resource demands and, thus, lower costs. Therefore, it can be seen that the RT-DETR-Tomato-B model has the lowest training time cost, while the RT-DETR-Tomato-S model has the highest.

2. Detection Performance
   The maximum mAP values for the RT-DETR, RT-DETR-Tomato-B, RT-DETR-Tomato-S, and RT-DETR-Tomato-BS models are 85.4%, 87.4%, 88.0%, and 88.7%, respectively. The RT-DETR-Tomato-S model uses the Swin Transformer to replace the original ResNet50 backbone of the RE-DETR model; the RT-DETR-Tomato-B model introduces the BiFormer Vision Transformer with dual-level routing attention to enhance small object feature extraction capability. The RT-DETR-Tomato-BS model replaces the original ResNet50 backbone of the RE-DETR model with the Swin Transformer and simultaneously introduces the BiFormer Vision Transformer to enhance small object feature extraction capability. The RT-DETR-Tomato-BS model shows the best detection performance, while the RT-DETR model shows the worst.

3. Resource Consumption and Complexity
   RT-DETR-Tomato-S model: This model replaces the original ResNet50 backbone of the RE-DETR model with the Swin Transformer. The hierarchical Transformer structure captures richer global dependencies and contextual information, enabling stronger feature representation in complex images and scenes, and allowing processing features at different resolutions, thereby improving detection accuracy. However,
the Swin Transformer blocks in the backbone network have redundant parameters, increasing resource consumption and computational complexity compared to the RT-DETR model. RT-DETR-Tomato-B model: This model introduces the BiFormer Vision Transformer with dual-level routing attention to enhance small object feature extraction capability. BiFormer technology achieves more flexible content-aware computation allocation using lightweight BiFormer blocks. The model effectively reduces computational complexity through flexible computation allocation in the BiFormer block, increasing resource consumption but reducing computational complexity compared to the RT-DETR model. RT-DETR-Tomato-BS model: This model replaces the original ResNet50 backbone of the RE-DETR model with the Swin Transformer and simultaneously introduces the BiFormer Vision Transformer. This improves feature extraction capability while reducing computational complexity but increases resource consumption.

4. Overall Cost Assessment

The RT-DETR model has moderate training time, the lowest performance, and moderate cost. The RT-DETR-Tomato-B model has the shortest training time, good performance, and the lowest cost. The RT-DETR-Tomato-S model has the longest training time, good performance, and the highest cost. The RT-DETR-Tomato-BS model has a relatively short training time, the best performance, and the best overall cost.

Overall, although the RT-DETR-Tomato-BS model shows significant performance improvements, its relatively short training time (1.305 h) achieves a good balance between training cost and performance. Therefore, the RT-DETR-Tomato-BS model might be the optimal choice in terms of overall cost.

5. Conclusions

In conclusion, this study proposes the RT-DETR-Tomato model for tomato object detection, with significant research background and implications. Addressing the issue of poor detection performance for small targets in complex and dense scenes, this study builds upon the RT-DETR model framework. The original model employs a ResNet50 backbone network with limited feature extraction capabilities. Therefore, Swin Transformer is selected as the new backbone network due to its superior ability to capture extensive global dependencies and contextual information. Furthermore, to mitigate the issue of the large number of parameters in the Swin Transformer modules, a lightweight BiFormer block is adopted. This module effectively reduces computational complexity through content-aware flexible computation allocation. The model is trained on the dataset, and comparative experiments reveal that the improved model achieves better detection accuracy. This demonstrates that replacing the backbone network and incorporating attention mechanisms can effectively enhance model performance and accuracy in detecting small targets within scenes. The \( \text{mAP} \) (mean Average Precision) increased by 3.3 percentage points compared to the original model. At the same time, the training time was reduced by half, adequately meeting the requirements for tomato object detection in densely populated plant environments. The main contributions of this improvement to the tomato production process are as follows: In the tomato production process, the harvesting stage is particularly labor-intensive and time-consuming. By using a tomato object detection model, ripe tomatoes can be quickly and accurately identified, providing precise location information for harvesting robots, thereby improving harvesting efficiency. Given the large number of tomatoes, a 3.3% improvement in model efficiency means that more tomatoes can be identified within the same time frame, further increasing harvesting speed and reducing tomato waste for growers. Additionally, tomato sorting is a key step in ensuring tomato quality. Traditional sorting methods rely mainly on manual labor, which is inefficient and prone to errors. Utilizing the tomato object detection model, tomatoes at different stages of ripeness can be automatically identified and classified according to set standards. A 3.3% improvement in model efficiency helps to increase sorting speed, reduce labor costs, and ensure tomato quality. Future work will focus on achieving a lightweight model structure.
that ensures precision while accelerating detection speed, enabling deployment on mobile devices with limited hardware resources. This practical application aims to enhance the monitoring efficiency of tomato yield and quality, promote safe tomato production, and provide more reliable decision support for farmers and food processing enterprises.

Author Contributions: Methodology, Z.Z.; Software, Y.G.; Formal analysis, S.C.; Investigation, P.Y.; Resources, Z.Z.; Writing – original draft, Z.Z.; Writing – review & editing, Y.S.; Supervision, Y.W. and Y.S.; Project administration, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Shandong Provincial Natural Science Foundation, China (no. ZR2020MF146), and the Open Project Foundation of Intelligent Information Processing Key Laboratory of Shansi Province (no. CICIP2021002).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available from the first author upon request.

Acknowledgments: This paper was completed by the Key Laboratory of Huang-Huai-Hai Smart Agricultural Technology of the Ministry of Agriculture and Rural Affairs, Shandong Agricultural University. We thank the school for its support and help.

Conflicts of Interest: The authors declare no conflicts of interest. This paper represents the opinions of the authors and does not mean to represent the position or opinions of the Shandong Agricultural University.

References

1. Zhaoxin, G.; Han, L.; Zhijiang, Z.; Libo, P. Design a robot system for tomato picking based on yolo v5. *IFAC-PapersOnLine* 2022, 55, 166–171. [CrossRef]
2. Lawal, M.O. Tomato detection based on modified YOLOv3 framework. *Sci. Rep.* 2021, 11, 1447. [CrossRef]
3. Xu, X.; Xue, Z.; Zhao, Y. Research on an algorithm of express parcel sorting based on deeper learning and multi-information recognition. *Sensors* 2022, 22, 6705. [CrossRef]


44. Gulzar, Y. Fruit image classification model based on MobileNetV2 with deep transfer learning technique. *Sustainability* **2023**, *15*, 1906. [CrossRef]


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.