A Lightweight ST-YOLO Based Model for Detection of Tea Bud in Unstructured Natural Environments

Xin Wen, Yi Yao, Ying Cai, Zixing Zhao, Tianjiao Chen, Ziyu Zeng, Zhen Tang, and Fangzheng Gao

Abstract—To address the difficulty of tea shoot recognition in natural environments, an enhanced YOLOX-Nano model (denoted as ST-YOLO) is introduced in this paper. In the proposed algorithm, the Depthwise Separable Convolution is incorporated into the backbone network to alleviate the feature extraction workload. And the Swin Transformer structure is also integrated into the neck network to augment feature extraction capabilities and to enhance the recognition of tea leaf shoots across diverse environmental conditions. Furthermore, to expedite model convergence, the SIoU loss function is employed in localization loss computation. Experimental results show the efficacy of the proposed ST-YOLO algorithm, with achieved Average Precision (AP) and $F_{1}$ scores of 86.2% and 88%, respectively. Comparing ST-YOLO with the baseline YOLOX-Nano in the same setting, it reveals notable improvements of 3.3% in AP and 5% in $F_{1}$. Additionally, compared with the widely adopted target detection algorithms like YOLO v5-s, YOLO v4, and Faster R-CNN, ST-YOLO exhibits the superiority of a remarkable reduction in model size by half while simultaneously enhancing AP values by 6.0%, 6.1%, and 6.88%, respectively. As a result, the proposed algorithm not only excels in terms of model lightweights but also significantly improves detection accuracy, which offers a promising solution for the deployment of intelligent tea-picking systems on embedded hardware platforms.

Index Terms—ST-YOLO, identification of tea buds, Swin-Transformer Structure.

I. INTRODUCTION

THE culture of tea consumption which has originated in the era of Shennong, dating back to approximately 3000 BC, boasts a rich and enduring history. Tea has traditionally been a significant export commodity for China [1]. With the continuous improvement of people’s living standards, there is a substantial surge in people’s demand for tea, particularly premium tea. However, tea picking remains a labor-intensive task characterized by a brief harvesting period and high levels of physical exertion. The traditional manual picking method faces some constraints such as labor shortages and inefficiencies, rendering it insufficient to meet the contemporary demands of tea production [2]. Consequently, the exploration of methods to automate the identification and detection of tea shoots for the realization of intelligent tea harvesting [3].

In recent years, driven by the advancement and maturation of deep learning algorithms, object detection methodologies based on deep learning have found extensive applications in agricultural product detection [4]. For instance, in the field of apple harvesting, Ref. [5] introduced an embedded platform-oriented, lightweight detection approach denoted as YOLO v4-CA, which achieved an average detection accuracy of 92.23% for apple detection. In Ref. [6], a method was proposed for apple blossom detection based on YOLO v5s, which exhibited notable stability and robustness under varying weather and lighting conditions. Ref. [7] presented an apple localization method based on YOLO v3, which efficiently and accurately detected apples within complex environments through a single convolutional neural network. These research results provide a foundational framework for the application of apple-picking robots.

In the research realm of agricultural product detection, Ref. [8] proposed an improved YOLO v3-based strawberry recognition method, which achieved an average accuracy of 87.51% on the test dataset and demonstrating excellent performance even in cases of occlusion and complex scenes. Ref. [9] extended the capabilities of YOLO v3 by introducing an improved tomato detection model named YOLO-Tomato. This model utilized circular bounding boxes to enhance tomato localization accuracy and when compared to several advanced detection methods, including Mask R-CNN, exhibited superior detection performance in complex environments. Additionally, Ref. [10] presented an enhanced approach for tomato ripeness recognition employing the improved YOLO v4-tiny model by integrating techniques, which achieved an average precision of 97.9%.

However, it remains relatively uncommon in the field of tea shoot target detection. Ref. [11] employed YOLO v4 as the foundational detection algorithm and introduced enhancements to propose a mobile-end recognition method for tea leaf shoots based on the Compact-YOLO v4 algorithm. This approach reduces the hardware performance demands.
of model reasoning computation by utilizing only 1/5 of the original model’s memory occupancy while maintaining comparable detection accuracy and speed. In a different study, Ref. [12] explored the application of the Faster R-CNN-based target detection algorithm for identifying tea shoots. The average accuracy achieved by this method was 54%, without distinguishing between different types of tea shoots, and the root mean square error was measured at 3.32. However, it is worth noting that these studies on tea shoot detection do not take account of the influence of lighting conditions within complex and ever-changing natural environments. As a result, these algorithms exhibit high sensitivity to environmental and lighting factors, and thus limits their robustness and practical applicability.

Based on the analysis above, to achieve accurate and fast recognition of tea buds in natural environments, this paper proposes a tea bud recognition algorithm based on the YOLOX-Nano model. First, the Swin Transformer structure is incorporated into the neck network to enhance the focus on target features. Then, deep separable convolution is introduced into the CSPDarkNet network to reduce the workload of feature extraction, effectively overcoming the challenges posed by complex backgrounds and lighting variations in recognition. Finally, the SIoU loss function is employed as the bounding box regression loss function to ensure fast convergence of the loss function. This algorithm simultaneously combines high detection accuracy and model lightweight, providing an effective solution for the embedded hardware deployment of mobile robots.

II. IMAGE ACQUISITION AND DATASET CREATION

A. Experimental Data Acquisition

Due to the lack of available tea bud datasets, this study conducted tea bud image capture work at the Jiangsu Tea Expo Park. The captured images were screened after the shooting to reduce the interference of duplicate and unqualified images on model training. As a result, 3641 images that met the requirements were obtained, some samples of which are shown in Fig. 1.

B. Dataset Preparation

In this study, the open-source software LabelImg was used to annotate the tea bud images. The annotation method was box labeling, and the labeled label was named “Teabud”. The specific annotation principles are as follows:

1. Annotation was performed on tea buds that were unobstructed on tea branches.
2. For instances where tea buds on tea branches were partially obstructed, manual estimation was used for annotation. Clear and visible tea bud images were annotated, while obstructed tea bud images were excluded from annotation.
3. Tea buds in images with distant views were not annotated if their pixel area was too small.

C. Dataset Augmentation

Data augmentation can effectively increase sample diversity and improve the robustness of the model [13]. The following image processing methods were employed in this research:

1. Image brightness was adjusted to simulate variations in actual lighting conditions.
2. Affine transformations were applied to simulate changes in camera angles in real-world scenarios.
3. Random occlusions were introduced to simulate situations where tea buds can be obscured in complex and changing environments.
4. Random cropping was implemented to augment the dataset size.

As depicted in Fig. 2, these techniques collectively expanded the dataset to a final count of 5312 images.

III. NETWORK MODEL AND IMPROVEMENTS

YOLOX-Nano[14] is a single-stage object detection algorithm that introduces several innovative techniques, such as the Fcous structure, spatial pyramid pooling, YOLOHead, and SimOTA dynamic positive sample matching [15], [16]. These improvements enable YOLOX-Nano to achieve faster convergence and higher accuracy, making it suitable for high-precision and fast object detection and recognition tasks.

A. Swin Transformer Attention Mechanism

The attention mechanism is a mechanism in neural networks that autonomously learns a set of weighting coefficients and emphasizes the regions of interest while suppressing irrelevant background regions through “dynamic weighting”. Attention mechanisms can be divided into two main categories: hard attention mechanisms and soft attention mechanisms. Hard attention mechanisms are random predictions that emphasize dynamic changes, but their application is limited due to their non-differentiable nature. Soft attention mechanisms are differentiable everywhere and can be trained using neural networks with gradient descent, making them more widely applicable. Soft attention can be categorized into channel attention, spatial attention, and self-attention based on different dimensions.

Channel attention mechanisms are primarily used to focus on meaningful information in the input image [17]. They obtain the importance of each channel through global pooling and use weighting coefficients to enhance or suppress features based on their significance. However, these weighting...
coefficients have a global receptive field, which may result in poor performance for detecting small objects in complex environments. Spatial attention mechanisms help the model focus attention on the regions of interest, improving the model’s perception performance and accuracy [18]. However, spatial attention mechanisms only focus on task-relevant regions and may be affected by occlusion or poor lighting conditions when detecting tea buds. Self-attention mechanisms, on the other hand, are different from the previous two mechanisms [19]. They dynamically determine the related weights between features by utilizing attention mechanisms between the input and output of the same layer in the network, discovering the relationships between features. Self-attention mechanisms are not dependent on convolutional networks, making them more suitable for recognizing small objects.

The Swin Transformer attention mechanism replaces long sequences with windows and hierarchical structures, which can improve detection performance while reducing the impact on runtime speed [20]. The network structure of the Swin Transformer is illustrated in Fig. 3. Firstly, a 3-channel image is inputted, and after undergoing Patch Partition, it is sent to the Linear Embedding layer to obtain embedding vectors. Then, the embedding vectors are fed into several Swin Transformer blocks with self-attention, which are computed layer by layer in four stages (stage 1 to stage 4).

In the Swin Transformer network structure, the Swin Transformer Block is the core module responsible for computing the attention mechanism. It consists of a concatenation of window-based multi-head self-attention (W-MSA) and sliding window-based multi-head self-attention (SW-MSA). The network structure of this module is shown in Fig. 4. The multi-head attention mechanism of this module allows for the calculation of similarities between all pixels, establishing global relationships between the target and complex background. This enables the model to better extract features of tea buds in complex backgrounds.

B. Depth Separable Convolution

BaseConv is the basic convolutional structure in the YOLOX-Nano backbone network, and its structure diagram is shown in Fig. 5. BaseConv mainly consists of Conv, BN, and SiLU, playing a crucial role in feature extraction within the network. It is one of the most important components in the backbone network of the model.

However, the BaseConv convolution extracts a large number of features and has a high computational cost. Therefore, to reduce the computational burden, this study introduces Depth Separable Convolution (DSConv)[21] into the
By applying Depth Separable Convolution (DSConv), the computational and parameter complexity of feature extraction can be reduced, thus enhancing the detection of images in conjunction with CBAM. In resource-constrained scenarios, the combination of DSConv and CBAM can demonstrate superior performance.

C. Improvement of the Loss Function

The loss function used in the YOLOX-Nano network is as follows:

\[
loss = loss_{cls} + loss_{reg} + loss_{obj}
\]  

In the equation, \(loss\) represents the total loss, \(loss_{cls}\), \(loss_{reg}\), and \(loss_{obj}\) represent the losses for category prediction, localization, and object presence probability, respectively. The losses for category prediction and object presence probability are calculated using binary cross-entropy loss, while the localization loss is calculated using the IoU loss function [22]. The expression for the IoU loss function is as follows:

\[
loss_{IoU} = 1 - IoU = \frac{|A \cap B|}{|A \cup B|}
\]

In the equation, \(loss_{IoU}\) represents the IoU loss function, \(IoU\) represents the intersection over union (IoU) between the predicted bounding box and the ground truth bounding box, \(A\) represents the position of the predicted bounding box, and \(B\) represents the position of the ground truth bounding box.

The IoU loss function does not consider the mismatch in orientation between the desired ground truth box and the predicted box, which can result in slow convergence and low efficiency. To address this issue, this study introduces the SIoU loss function for computing the localization loss[23]. The SIoU loss function takes into account the angle between the regression vectors and redefines the penalty term. The parameters involved in the SIoU loss function are illustrated in Fig.7. The formula for the SIoU loss function is as follows:

\[
loss_{SIoU} = 1 - IoU + \frac{\Delta + \Omega}{2}
\]

In the formula, \(\Delta\) represents the distance cost, and \(\Omega\) represents the shape cost. This loss function consists of three components: angle loss, distance loss, and shape loss. Considering the angle factor, during the regression process of the predicted box, it allows the predicted box to quickly regress to the same horizontal or vertical line as the ground truth box, thereby accelerating the convergence speed of the loss function.

The specific calculation methods for the angle loss, distance loss, and shape loss in the SIoU loss function are as follows.

1) Angle Loss: During the convergence process, it is first determined whether the angle is less than 45°. If it satisfies the condition, it is directly substituted into the following formula for calculation. Otherwise, its complementary angle is used instead. The formula for angle cost calculation is as follows:

\[
\Lambda = 1 - 2 \sin^2(\arcsin x - \frac{\pi}{4})
\]

\[
x = \frac{c_h}{\sigma} = \sin(\alpha)
\]
represents the difference in the y-coordinate of the center points.

2) distance loss: The calculation formula for the distance loss is as follows:
\[ \Delta = \sum_{t=x,y} (1 - e^{\gamma \rho_t}) \]  
(6)

In the formula, \( \Delta \) represents the distance loss, \( \gamma = 2 - \Lambda \), and \( \rho_t \) represents the squared difference between the center point coordinates of the two boxes.

3) shape loss: The formula for shape loss is as follows:
\[ \Omega = \sum_{t=w,h} (1 - e^{-\pi_t})^\theta \]  
(7)

In the formula, \( \Omega \) represents the shape loss, \( \theta \) represents the weight of the shape loss in the localization loss, and in this study, the parameter \( \theta \) is set to the default value of 1. \( \pi_t \) represents the ratio between the difference in width between the ground truth box and the predicted box and the maximum value, while \( \pi_t \) represents the ratio between the difference in height between the ground truth box and the predicted box and the maximum value.

D. ST-YOLO Network Architecture

Inspired by the aforementioned experiments, this study implements the following main improvements in the ST-YOLO model:

1) Depthwise separable convolutions are introduced into the CSPDarkNet backbone network of the YOLOX-Nano model.

2) The Swin Transformer structure is integrated into the enhanced neck network.

3) The IoU loss function in the original network is replaced with the SIoU loss function.

The improved model is referred to as the ST-YOLO model. The architecture of ST-YOLO is shown in Fig. 8. The colored parts in the figure represent the improvements and optimizations made relative to the original YOLOX-Nano model, while the uncolored parts retain the original network structure without any changes.

E. Model Training and Testing

1) Experimental Platform: The training platform used in this study was a desktop computer equipped with the Windows 10 operating system. It had an Intel Core i7-13700F CPU and an RTX2060 GPU. The training and testing environments were the same.

2) Network Training: The collected dataset of 5312 images was randomly divided into training set, validation set, and test set in an 8:1:1 ratio, as shown in Table I.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of total images</th>
<th>Number of targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Set</td>
<td>4249</td>
<td>6514</td>
</tr>
<tr>
<td>Validation Set</td>
<td>531</td>
<td>789</td>
</tr>
<tr>
<td>Test Set</td>
<td>532</td>
<td>750</td>
</tr>
<tr>
<td>Total</td>
<td>5312</td>
<td>8053</td>
</tr>
</tbody>
</table>

The training parameters were configured as follows: the training mode selected was local training mode using the GPU of the local machine. The number of iterations was set to 300. The base learning rate was set to the default value of 1. A larger patch size was chosen. Data augmentation during training included the use of Mosaic augmentation and Mixup augmentation. However, considering that in the Mosaic augmentation process, if the samples deviate significantly from the actual conditions, it may negatively impact the training effectiveness of the model. Therefore, during the last 30 iterations of training, Mosaic and Mixup data augmentation were disabled[24].

IV. RESULTS AND ANALYSIS

A. Model Lightweight Analysis and Comparison

To validate the superiority of the YOLOX-Nano base network used in this experiment, three lightweight YOLOX models were trained using the same dataset. These models are YOLOX-Nano, YOLOX-Tiny, and YOLOX-S, as shown in Table II. The results show that when tested on the same test set, the performance differences among the three models are small, indicating their relative similarity in object detection tasks. However, compared to YOLOX-Tiny and YOLOX-S, YOLOX-Nano performs better in terms of average precision (AP), with improvements of 4.3 percentage points and 2.6 percentage points, respectively. Furthermore, the model size of YOLOX-Nano is only 3.70MB, significantly smaller than YOLOX-Tiny and YOLOX-S.

<table>
<thead>
<tr>
<th>Model</th>
<th>Num(MB)</th>
<th>( F_1(%) )</th>
<th>AP(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOX-Nano</td>
<td>3.70</td>
<td>83</td>
<td>82.9</td>
</tr>
<tr>
<td>YOLOX-Tiny</td>
<td>19.4</td>
<td>82</td>
<td>78.6</td>
</tr>
<tr>
<td>YOLOX-S</td>
<td>34.3</td>
<td>82</td>
<td>80.3</td>
</tr>
</tbody>
</table>

In the aforementioned results, YOLOX-Nano performs the best with a smaller model size. This is because the same augmentation strategy was applied to all three models during training (i.e., the same period of eliminating Mosaic and MixUp). However, different-sized models may require different augmentation strategies. For larger models, employing stronger augmentation strategies can yield better results.

In conclusion, choosing a small model is crucial when deploying object detection models on embedded and mobile devices, as they can operate within limited computational resources and storage capacity while still maintaining good detection accuracy. Therefore, this study selected YOLOX-Nano as the baseline model, which performs optimally with smaller model size and can efficiently execute object detection tasks under limited resource conditions.

B. Result Analysis of Improved YOLOX Model

To validate the performance of the improved YOLOX-Nano model, a comparative experiment was conducted on the same tea bud dataset. As shown in Fig. 9, the YOLOX-Nano model exhibits false negatives on targets with higher detection difficulty, whereas the ST-YOLO model does not.
Additionally, for correctly detected targets, the YOLOX-Nano model has confidence levels of 0.96, 0.87, 0.97, and 0.98 from left to right, while the ST-YOLO model has confidence levels of 0.99, 0.97, 0.98, and 0.97 for the same tea buds from left to right. This indicates that the ST-YOLO model has significantly higher confidence in recognizing tea buds.

The YOLOX-Nano model exhibits noticeable false positives and false negatives in bright sunlight conditions. As shown in Fig. 10(a), the original model missed two relatively subtle targets due to the influence of lighting conditions. Additionally, it mistakenly identified the shadow on the tea leaves as a tea bud. On the other hand, the ST-YOLO model does not encounter these issues, as depicted in Fig. 10(b). A comparison of the detection results before and after the improvement is presented in Table III.

<table>
<thead>
<tr>
<th>Model</th>
<th>Num Average detection time</th>
<th>$F_1$</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOX-Nano</td>
<td>3.70MB</td>
<td>18.46ms</td>
<td>83%</td>
</tr>
<tr>
<td>ST-YOLO</td>
<td>17.3 MB</td>
<td>21.29ms</td>
<td>88%</td>
</tr>
</tbody>
</table>

According to Table III, the proposed ST-YOLO object detection network in this study has achieved an $F_1$ score of 86%, which is a 5 percentage point improvement compared to the original YOLOX-Nano network. The AP value has reached 86.2%, showing a 3.3 percentage point improvement compared to the original YOLOX-Nano network.

C. Comparative Experiment

To objectively evaluate the performance of the ST-YOLO network, this study also trained other lightweight YOLO models under the same conditions and compared them with the ST-YOLO model. The experimental results are shown in Table IV.

It can be observed that the proposed ST-YOLO model achieves the highest $F_1$ score. Compared to YOLO v5-s and YOLO v4, it achieves an improvement of 6.0% and 6.1% in terms of AP, respectively. Additionally, the ST-YOLO model has a smaller model size and faster average detection time compared to YOLO v4. When compared to the two-stage detector Faster R-CNN, ST-YOLO demonstrates a 6.88% increase in AP and also exhibits advantages in terms of average detection time and model size. Furthermore, compared to the lightweight network Compact-YOLO v4, ST-YOLO achieves a significant improvement of 12.67% in AP.

V. CONCLUSION

To deal with the limitations of traditional deep learning classification methods in tea bud recognition, this study has proposed a tea bud detection method based on ST-YOLO. The method builds upon the YOLOX-Nano model and introduces depthwise separable convolutions in the backbone network to reduce feature extraction workload. Additionally, the Swin Transformer structure is incorporated into the neck network to enhance feature extraction capabilities. Moreover, the SIoU loss function is utilized to calculate localization loss, thereby accelerating model convergence. After training, the ST-YOLO network achieves an AP value of 86.2% and an $F_1$ score of 88% in tea bud detection, demonstrating significant improvements over the original YOLOX-Nano model. The proposed ST-YOLO object detection model surpasses commonly used detection models in detection accuracy and holds high practical value. Moreover, its model size and detection speed meet the requirements for deployment on mobile devices. This method holds significant implications for the intelligent harvesting of tea buds and can serve as a reference for the harvesting of other agricultural products.

REFERENCES

Fig. 9. Comparison of confidence levels of tea bud detection before and after algorithm improvement.

![Figures showing comparison of confidence levels before and after improvement.]

Fig. 10. Comparison of tea bud detection results before and after algorithm improvement.

![Figures showing comparison of detection results before and after improvement.]

**TABLE IV**

<table>
<thead>
<tr>
<th>Model</th>
<th>Num(MB)</th>
<th>Frame rate(f/s)</th>
<th>Average detection time(ms)</th>
<th>$F_1$ (%)</th>
<th>AP(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST-YOLO</td>
<td>17.3</td>
<td>47</td>
<td>21.29</td>
<td>88</td>
<td>86.20</td>
</tr>
<tr>
<td>YOLO v5-s</td>
<td>27.10</td>
<td>59</td>
<td>16.87</td>
<td>81</td>
<td>80.20</td>
</tr>
<tr>
<td>Compact-YOLO v4</td>
<td>23.20</td>
<td>43</td>
<td>23.26</td>
<td>74</td>
<td>72.93</td>
</tr>
<tr>
<td>YOLO v4</td>
<td>115.8</td>
<td>32</td>
<td>31.25</td>
<td>80</td>
<td>80.1</td>
</tr>
<tr>
<td>Faster R-CNN</td>
<td>70.8</td>
<td>13</td>
<td>76.82</td>
<td>78</td>
<td>79.32</td>
</tr>
</tbody>
</table>


