

Article

StrawSnake: A Real-Time Strawberry Instance Segmentation Network Based on the Contour Learning Approach

Zhiyang Guo ^{1,*} , Xing Hu ², Baigan Zhao ¹, Huaiwei Wang ¹ and Xueying Ma ¹

¹ School of Traffic Engineering, Jiangsu Shipping College, Nantong 226010, China; zbg@st.usst.edu.cn (B.Z.); wh@jssc.edu.cn (H.W.); mx@jssc.edu.cn (X.M.)

² School of Optical-Electrical and Computer Engineering, University of Shanghai for Science & Technology, Shanghai 200093, China; huxin@usst.edu.cn

* Correspondence: gzy@jssc.edu.cn

Abstract: Automated harvesting systems rely heavily on precise and real-time fruit recognition, which is essential for improving efficiency and reducing labor costs. Strawberries, due to their delicate structure and complex growing environments, present unique challenges for automated recognition systems. Current methods predominantly utilize pixel-level and box-based approaches, which are insufficient for real-time applications due to their inability to accurately pinpoint strawberry locations. To address these limitations, this study proposes StrawSnake, a contour-based detection and segmentation network tailored for strawberries. By designing a strawberry-specific octagonal contour and employing deep snake convolution (DSConv) for boundary feature extraction, StrawSnake significantly enhances recognition accuracy and speed. The Multi-scale Feature Reinforcement Block (MFRB) further strengthens the model by focusing on crucial boundary features and aggregating multi-level contour information, which improves global context comprehension. The newly developed TongStraw_DB database and the public StrawDI_Db1 database, consisting of 1080 and 3100 high-resolution strawberry images with manually segmented ground truth contours, respectively, serves as a robust foundation for training and validation. The results indicate that StrawSnake achieves real-time recognition capabilities with high accuracy, outperforming existing methods in various comparative tests. Ablation studies confirm the effectiveness of the DSConv and MFRB modules in boosting performance. StrawSnake's integration into automated harvesting systems marks a substantial step forward in the field, promising enhanced precision and efficiency in strawberry recognition tasks. This innovation underscores the method's potential to transform automated harvesting technologies, making them more reliable and effective for practical applications.

Keywords: deep learning; snake convolution; transform; contour segmentation



Citation: Guo, Z.; Hu, X.; Zhao, B.; Wang, H.; Ma, X. StrawSnake: A Real-Time Strawberry Instance Segmentation Network Based on the Contour Learning Approach. *Electronics* **2024**, *13*, 3103. <https://doi.org/10.3390/electronics13163103>

Academic Editors: Manohar Das, Haibin Wu, Aili Wang and Yuji Iwahori

Received: 14 July 2024

Revised: 2 August 2024

Accepted: 4 August 2024

Published: 6 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The development of automatic picking machines is crucial for advancing agricultural intelligence, where crop perception plays a central role. These machines face numerous challenges that require sophisticated visual perception technology, such as manipulating the robotic arm [1], the classification of ripeness [2] or the detection of individuals [3] and diseases [4]. Strawberries, which grow in clusters and exhibit significant variability in shape, size, and ripeness, present unique challenges. The dense foliage and stems further complicate the task by obstructing the view and making accurate strawberry localization particularly difficult for the visual systems of harvesting robots. Current mainstream strawberry recognition technologies primarily rely on bounding box and pixel classification methods. While bounding boxes provided by object detection and masks from instance segmentation are commonly used, contour segmentation technology offers superior boundary accuracy. Unlike bounding boxes, which can be affected by complex backgrounds, and masks, which demand high memory usage, contour segmentation

delivers precise boundaries, thereby enhancing the visual system's capability to differentiate and locate individual strawberries amidst cluttered environments. The research into contour segmentation for strawberry recognition is of significant value, as it promises to overcome the limitations of existing methods. By improving the accuracy and efficiency of crop detection, this technology can lead to more effective automation in harvesting, ultimately contributing to increased agricultural productivity and reduced labor costs.

In current strawberry recognition work based on deep learning, there is little exploration of strawberry contours. However, recently, in the identification of other crops, Wang et al. [5] combined saliency detection and traditional color difference with a real-time deep-snake [6] deep learning contour segmentation model to achieve fast detection and recognition of apple fruits. In addition, in terms of dataset work, Borrero et al. released a large-scale high-resolution dataset of strawberry images, along with corresponding manually labeled instance segmentation masked images. They used Mask R-CNN to achieve strawberry instance segmentation. However, these methods still struggle to meet the real-time and high-precision operational requirements of the visual system in harvesting robots, as the picking robots are typically equipped with energy-limited power supplies and low-computing devices.

As shown in Figure 1, the dense growing area of strawberries produce a lot of overlapping areas. The mainstream detection-based method (Figure 1b) cannot be able to make an accurate judgment of the strawberry position. In addition, the segmentation-based method (Figure 1c) requires pixel-level judgment and a large amount of computation. Inspired by classic strawberry identification approaches, we think that strawberry contour can provide a more efficient presentation. The strawberry contour consists of a sequence of strawberry boundary points along the contour. In contrast to the detection-based and segmentation-based methods, the strawberry contour is not limited to a bounding box and has fewer computational parameters. Therefore, the contour-based representation is well suited for strawberry identification.

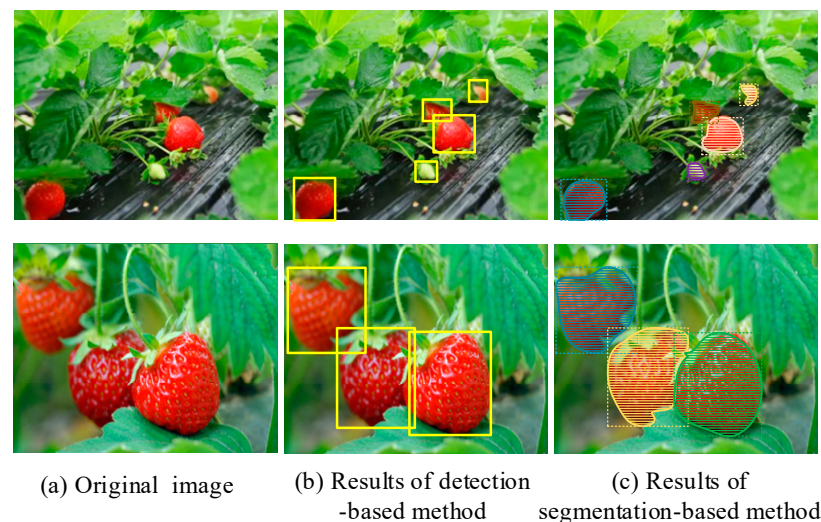


Figure 1. The existing issues of strawberry detection. The yellow boxes represent the strawberry detection area.

Based on the exploration of apple contours in Ref. [5], we also use the Deep Snake framework for strawberry contour segmentation. Due to the extensive occlusion of strawberries by leaves and stems, we need a more powerful contour boundary segmentation capability. As shown in Figure 2, standard convolutional kernels are designed to extract local features (Figure 2a), deformable convolutional kernels can enrich their applications and adapt to geometric deformations of different targets (Figure 2b), and dynamic snake convolutions (DSConv) can effectively focus attention on fine and curved boundary structures, enhancing perception of geometric structures adaptively (Figure 2c). Therefore,

we propose using dynamic snake convolutions to extract strawberry boundary features combined with a Transform feature fusion module, to learn and aggregate multi-level strawberry contour features, forming the “StrawSnake”.

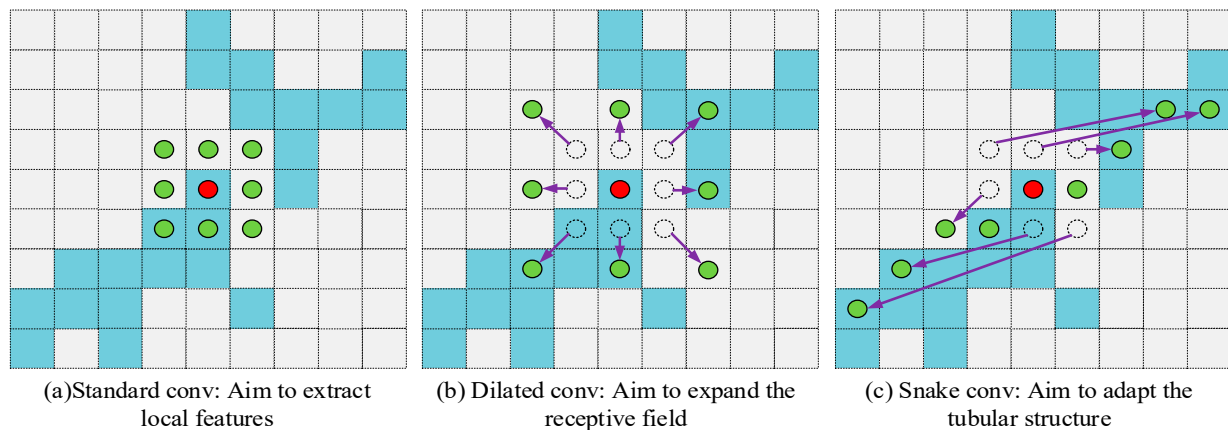


Figure 2. Diagram of the action of different convolutions. These arrows represent the calculation ideas when the convolution kernel is computed. The red dots represent the initial image values and the green represents the convolution kernel.

Building on the Deep Snake framework, we also adopted a two-stage pipeline approach for strawberry instance segmentation, treating strawberry recognition as a contour segmentation problem. Additionally, based on the characteristics of strawberry posture, we specifically designed an octagon contour composed of strawberry extremal points as the initial strawberry contour. Finally, we used StrawSnake to deform the initial contour into the strawberry boundary. Our main contributions can be summarized as follows:

- (1) We propose a real-time contour segmentation network (StrawSnake) for strawberry detection in virtue of the structure of the deep snake model and design an octagon contour specifically for strawberry contour segmentation, which can effectively enclose the strawberries tightly, and a contour feature aggregation module is used to aggregate the multi-level strawberry contour features.
- (2) We propose using dynamic snake convolutions to extract strawberry boundary features combined with a Transform feature fusion module, to learn and aggregate multi-level strawberry contour features.
- (3) We use an edge detection algorithm to annotate the ground truth of the strawberry contour on StrawDI_Db1 [7] and our TongStraw_DB as the training data. We conduct extensive experiments to verify StrawSnake is boosting the performance of strawberry segmentation. The experimental results show that our method achieves state-of-the-art performance in both accuracy and speed.

2. Related Work

Currently, most of the strawberry recognition work is carried out by mature object detection networks such as FCN [8], Mask RCNN [9], YOLO series [10], etc. Here, we introduce the latest segmentation methods for strawberry detection and other fruits.

2.1. Classical vs. Deep Learning-Based Strawberry Detection Approaches

Most classic strawberry detection approaches are based on hand-crafted shape features and employ an active contour model such as a Snake model. However, it is difficult to extract very deep and complex features using these traditional methods [11–13]. In contrast, the deep learning-based approach achieves higher performance on classification and detection problems compared to traditional computer vision. Deep learning introduces the concept of end-to-end learning, where algorithms are presented with a large number of images annotated with object classes. For example, Bai et al. [14] proposed to build the

Swin Transformer [15] prediction head on the high-resolution feature map of YOLOv7 [10] to better use spatial position information to enhance the detection of small target flowers and fruits, and improve the spatial interaction and feature extraction capability of the model in similar color and overlapping occlusion scenes. Similar to this, Pang et al. [16] proposed an improvement method based on YOLO for strawberry detection. By building a CSP2 module, they created a double cooperative attention mechanism to improve feature representation in complex environments.

2.2. Strawberry Instance Segmentation

Instance segmentation algorithms based on deep learning have recently been applied to different studies in the agricultural field. At first, Pérez et al. [7] proposed a method for instance segmentation of strawberries using an improved Mask RCNN technique. They designed a new backbone network and mask network architecture; eliminated the target classifier and bounding box regressor; and, without increasing the complexity order, replaced the non-maximum suppression algorithm with a new region grouping filtering algorithm. Similarly, Usman et al. [17] also used the Mask R-CNN architecture to segment these seven diseases. They used the ResNet backbone and followed a systematic data enhancement approach that allowed for segmentation of target diseases under complex environmental conditions. Cao et al. proposed a lightweight StrawSeg segmentation framework for strawberry instances [18]. They directly segmented each strawberry with a single lens, independent of object detection. They designed a new feature aggregation network to combine features of different scales and improve the resolution and reduce the channel of features. In addition, there are also some works that used contour instance segmentation. To solve the problem of inaccurate image segmentation of strawberries with different maturity due to fruit adhesion, accumulation and other reasons, Wang et al. [19] proposed a strawberry image segmentation method based on improved DeepLabV3+ model [20]. The technology introduces the attention mechanism to the backbone of the DeepLabV3+ network.

2.3. Feature Fusion Strategy

Feature fusion is the process of combining feature information from different levels or different networks to obtain a richer and more comprehensive representation. The purpose of feature fusion is to improve the performance of the model, enabling it to better understand and process complex data. Some work has also been carried out on fruit detection models, e.g., Swin Transformer [15], CBAM [21], and GAM [22]. Zhao et al. [23] addressed the complex background and small lesion size issues in strawberry disease images by proposing a new, faster R-CNN architecture for detecting 7 types of strawberry diseases. The multi-scale feature fusion network composed of ResNet, FPN, and CBAM blocks can effectively extract rich features of strawberry diseases. The mAP value is 92.18%, with an average detection time of only 229 ms. In our work, based on the significant differences between contour boundary features and surrounding features, we prioritize these important feature channels and enhance the model's global context understanding capability, proposing a novel feature fusion strategy.

3. Our Methods

Figure 3 shows the pipeline of our framework, which adopts a two-stage pipeline, including the initial strawberry contour proposal (Figure 3A) and the strawberry contour deformation (Figure 3B). We take advantage of DeepSnake's structure to construct the initial strawberry contour and contour deformation. The difference is that we use the YOLO V8 [24] to improve the detection effect of small-sized strawberries. Specifically, we designed a suitable octagon for strawberries (Figure 3A(f)) in the initial contour proposal. In addition, our StrawSnake consists of four parts: a feature encoding block, a feature fusion block, a feature aggregation block and an offset prediction layer. We will introduce these blocks in detail in Section 3.2.

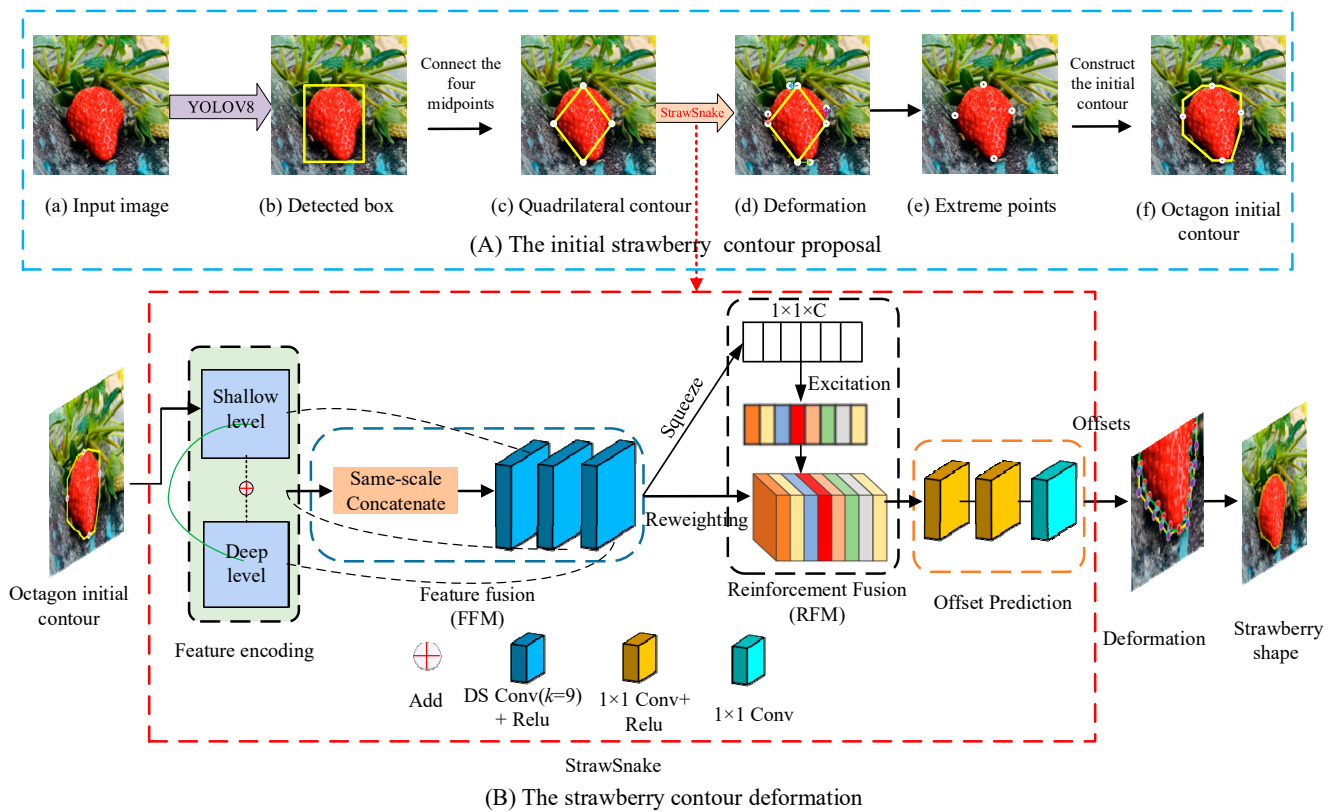


Figure 3. The pipeline of our framework. The white point represents the extreme value point, and the red point represents the deformed contour point.

3.1. Contour Feature Representation

Given an initial contour with N vertices $\{x_i | i = 1, \dots, N\}$ the program learns the features and vertex coordinates to represent each vertex $\{x_i | i = 1, \dots, N\}$ as a feature vector f_i . The input feature f_i for a vertex x_i is a concatenation of learning-based features and the vertex coordinate. We use $[F(x_i); x_i]$ to represent the input feature, including the feature and position information. In addition, we use the bilinear interpolation of features at the vertex coordinate x_i to compute the contour feature $F(x_i)$, as shown in Figure 4.

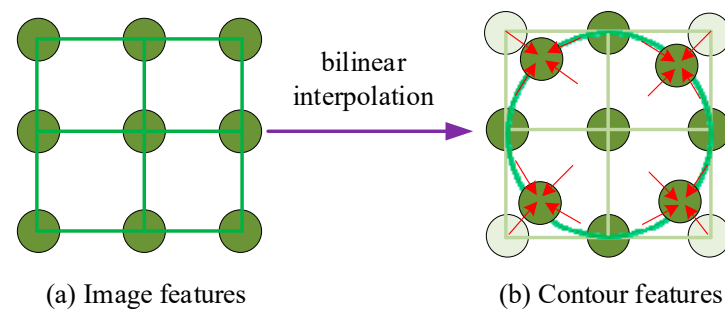


Figure 4. The schematic for the contour features. The red arrow represents the interpolation method of conventional convolution to circular convolution.

3.2. Initial Strawberry Contour Proposal

Strawberry Contour Design

Refs. [14,25] have proved that an octagon whose edges are centered on the extreme points can provide precise initial contours. Thus, we also choose the octagon as the initial strawberry contour. Firstly, we used the YOLO V8 detector to obtain the bounding box for the strawberry and denote $xBox | i = 1, 2, 3, 4$ as four midpoints at the four box borders. Then we connected four midpoints to construct a quadrilateral contour (Figure 3b). In

addition, we denoted pixels at the top, leftmost, bottom and rightmost as four extreme points in a strawberry. StrawSnake will take the quadrilateral contour as the input, and output the offsets that point from each xBox (Figure 3c). In order to obtain more contour vertex offsets, we uniformly upsample 60 points on the quadrilateral contour in our experiment (Figure 3d). At the same time, StrawSnake will output 60 offsets to deform the strawberry extreme points. After that, we generated four lines based on the extreme points to construct the octagon contour (Figure 3e). In general, the aspect ratio of strawberries is approximately 1:2. Thus, we designed an octagon contour specifically for strawberries, which can effectively enclose the strawberries tightly. For the top and bottom extreme points, a line will extend from the extreme point as the midpoint in both directions to $1/4$ of the border length. For the left and right extreme points, the extension length is $1/3$ of the border length. In particular, the line will be truncated if it meets the box corner. Finally, the initial strawberry contour (Figure 3f).

Figure 5 shows the schematic of contour vertex deformation. Through contour deformation, the model can adjust the vertex position iteratively, make the final contour more consistent with the target boundary, correct the error, and improve the segmentation accuracy. It can also flexibly adjust the contour shape to adapt to various complex object boundaries.

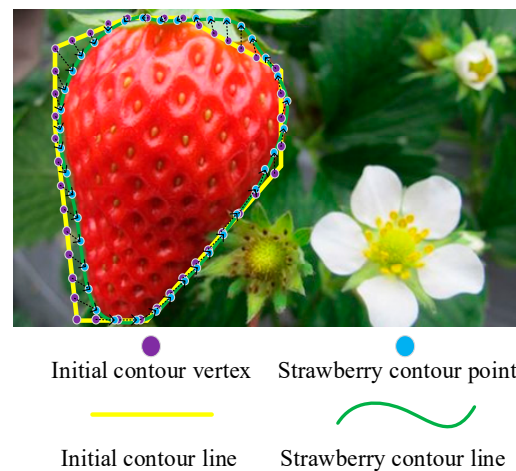


Figure 5. The schematic of contour vertex deformation. Purple points represent the initial contour vertexes. Blue points represent the strawberry contour points. Yellow lines represent the initial contour. Green lines represent the strawberry contour line.

3.3. Dynamic Snake Convolution (DSConv)

In this section we will discuss how to perform Dynamic Snake Convolution (DSConv) to extract local features of tubular structures. Given a standard 2D convolution coordinate K , the central coordinate is $K_i = (x_i, y_i)$. A 3×3 convolution kernel K is represented by:

$$K = \{(x-1, y-1), (x-1, y), \dots, (x+1, y+1)\} \quad (1)$$

In order to give more flexibility to the convolution kernel so that it can focus on the complex geometric features of the target, we introduce deformation deflection Δ , inspired by [26]. However, if the model is completely free to learn deformation and migration, the perception field will often deviate from the target, especially in the case of thin tubular structures. Therefore, we adopted an iterative strategy (Figure 3) to select the next position of each target to be processed for observation in order to ensure continuity of attention and not to spread the perception too far due to large deformation shifts.

As shown in Figure 6, we linearize the standard convolution kernel in both the X-axis and Y-axis directions in the dynamic snake convolution. We consider a convolution kernel of size 9, taking the X-axis direction as an example, and the specific position of each grid

in K is expressed as: $K_{i\pm c} = (x_{i\pm c}, y_{i\pm c})$, Where $c = 0, 1, 2, 3, 4$ represents the horizontal distance from the central grid. Starting at the center position K_i , the position away from the center grid depends on the position of the previous grid: K_{i+1} increases the offset $\Delta = \{\delta \mid \delta \in [-1, 1]\}$ with respect to K_i . Therefore, the offsets need to be accumulated Σ to ensure that the convolution kernel conforms to the linear morphology. The change of X-axis direction in Figure 4 is as follows:

$$K_{i\pm c} = \begin{cases} (x_{i+c}, y_{i+c}) = (x_i + c, y_i + \sum_{i=1}^{i+c} \Delta y) \\ (x_{i-c}, y_{i-c}) = (x_i - c, y_i + \sum_{i=1}^c \Delta y) \end{cases} \quad (2)$$

The change in the Y-axis direction is:

$$K_{j\pm c} = \begin{cases} (x_{j+c}, y_{j+c}) = (x_j + \sum_{j=1}^{j+c} \Delta x, y_j + c) \\ (x_{j-c}, y_{j-c}) = (x_j + \sum_{j=1}^c \Delta x, y_j - c) \end{cases} \quad (3)$$

Due to two-dimensional (X-axis, Y-axis) variations, our dynamic serpentine convolution kernel covers a selective range of 9×9 receptive fields during deformation. The dynamic serpentine convolution kernel is designed to better accommodate elongated boundary overlapping regions based on dynamic structures to better perceive key features.

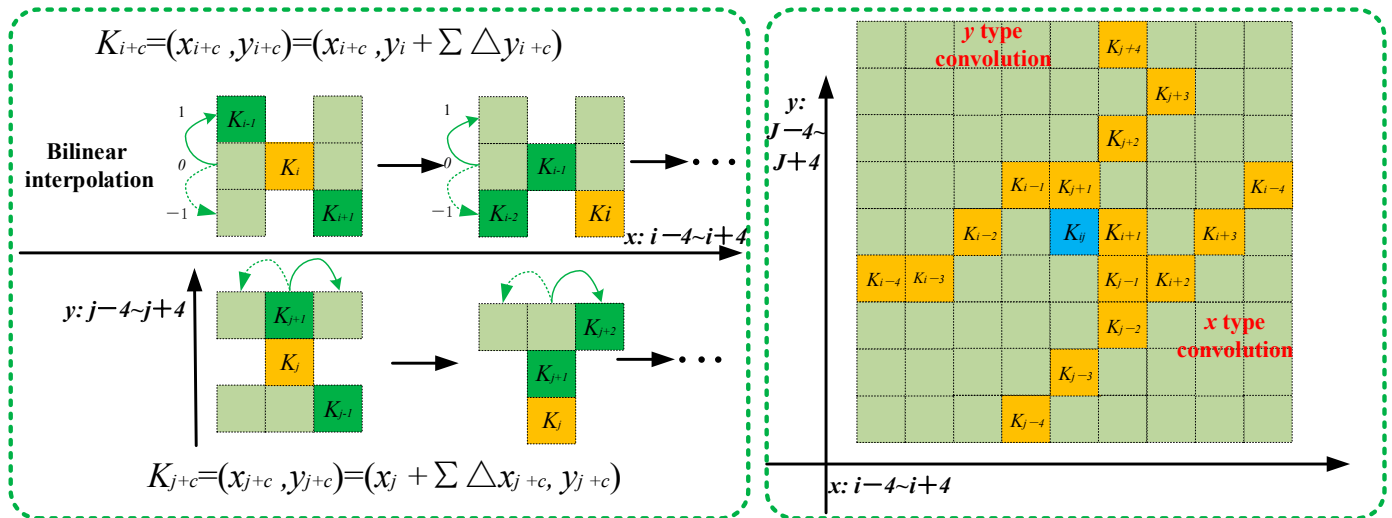


Figure 6. Schematic diagram of coordinate calculation (Left) and optional receptive field (Right).

3.4. Multi-Scale Feature Reinforcement Block (MFRB)

The role of the Multi-Scale Feature Reinforcement Block is to enhance our model's ability to recognize and segment various targets by integrating features from different scales. It helps the model capture both detailed and global information about the target at different scales. Conventional networks commonly employ basic element-wise addition or feature concatenation operations to fuse multi-scale features. However, we believe that such a simple feature fusion strategy may not fully capitalize on the inherent potential of the diverse features. With the recent rise in popularity of Transformer architectures for multi-modal visual-linguistic tasks, we argue that more advanced fusion techniques are warranted. Drawing inspiration from potent attention mechanisms, we have integrated attention mechanisms to optimize the fusion of multi-scale features extracted by the encoder. In light of this, we introduce FFM (depicted in Figure 7a), which leverages the self-attention

mechanisms of Transformers to achieve effective fusion. Our fusion module can be defined as follows:

$$F_i^H = \text{Reshape}(\text{Norm}(\text{Softmax}(Q_i K_i^T) k_i V_i + F_i) \quad (4)$$

where F_i^I and F_i^D are concatenated and reshaped to form F_i , which is then identically mapped to query Q_i , K_i and value V_i embeddings, and F_i denotes the output of our fusion module. In addition, we introduce a learnable coefficient k_i to adaptively adjust the attention significance, enabling a more flexible fusion of multi-scale features.

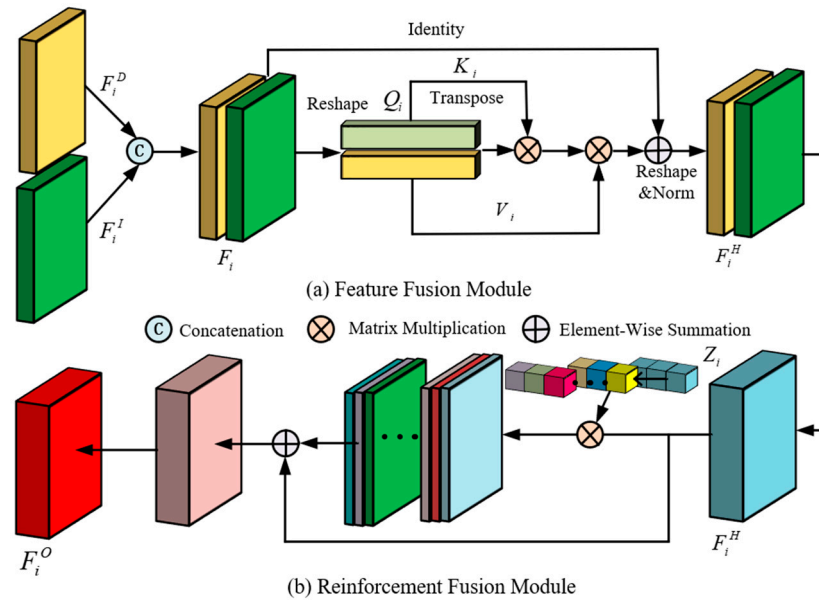


Figure 7. Multi-Scale Feature Reinforcement Block.

In conventional single-encoder architecture design, it is observed that not all multi-channel features contribute positively to semantic prediction. Furthermore, some uncorrelated feature mappings may diminish the model's performance. Consequently, we have developed RFM (illustrated in Figure 7b), derived from SEB [27], to strengthen the fused multi-scale features. We incorporate a residual connection to SEB to bolster the training, and employ point-wise convolution for adaptable computation on the features after Feature Fusion Module (FFM). The formulation of our reinforcement fusion module is as follows:

$$F_i^O = \text{Conv}_{1 \times 1}(F_i + (\text{Osigmoid}(\text{Conv}_{1 \times 1}(Z_i))) * F_i) \quad (5)$$

where O represents a matrix of ones and $*$ is the Hadamard product operation. Z_i stores the average pooling results of each feature map in F_i .

3.5. Detector Selection

We adopt YOLO V8 as the detector for all experiments. YOLO V8 is a one-stage detector and achieves impressive accuracy and speed. It fused the low-level and high-level features at multiple scales and has a good effect on detecting small objects. Thus, it is very suitable for detecting strawberries in traffic scenes, and can guarantee the accuracy of the strawberry boundary box as much as possible.

3.6. Loss Function

In our StrawSnake, the smooth L_1 loss function [28] is used to learn the vertex deformation for training StrawSnake. It can be defined as follows

$$\text{Smooth } L_1(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \quad (6)$$

The loss function for four strawberry extreme points is defined as

$$L_{\text{ex}} = \frac{1}{4} \sum_{i=1}^4 L_1(x_i^p - x_i^{\text{ex}}) \quad (7)$$

where x^p is the predicted extreme point. x^{ex} is the strawberry extreme point.

This loss function is used for direct regression to predict the vertex position of the contour. Vertex regression loss helps the model accurately predict the coordinates of each vertex, making the predicted profile as close as possible to the true profile. In addition, the loss function of initial contour deformation is defined as

$$L_{\text{it}} = \frac{1}{N} \sum_{i=1}^N L_1(x_i - x_i^{\text{gt}}) \quad (8)$$

where N is the number of sampling points. x_i is the deformed points. x_i^{gt} is the ground truth points at the strawberry contour.

This function ensures that the energy of the contour is minimized, so that the contour can be accurately positioned to the target boundary, while maintaining the smoothness and continuity of the contour. The accuracy and reliability of contour prediction are improved by direct regression of vertex position.

4. Experiment and Result

We implemented our experiment, and trained and tested network performance on an Inter Xeon(R)Sliver 4110 CPU@2.10GHz, 16G NVIDIA GeForce 2080Ti GPU, Ubuntu 22.04 operating system computer. The network is built based on the PyTorch framework and adopts an adaptive gradient optimizer to minimize the loss function. The learning rate is initialized to 0.0008 and the weight decay rate is 0.01. The number of iterations is 250. After every 50 iterations, we test the effect of model training on the validation set and return the index parameters.

4.1. Datasets and Contour Labels

In our experiment, we made a data set using our own collection (TongStraw_DB) and a public dataset StrawDI_Db1 [7]. Firstly, we form our dataset (TongStraw_DB) through network resources and on-site collection of 1048 images in the natural environment in Haimen County (NanTong City, JiangSu Province, China) to train and validate our model. According to the distribution of the strawberries, they can be divided into strawberries unobstructed by branches and leaves (single unobstructed, multiple strawberries and adjacent strawberries), shaded by branches and leaves, and overlapping strawberries.

At the same time, the data set contains other external factors that affect strawberry recognition, such as patterned labels, plastic bags of strawberries, poor lighting conditions (large areas of shadows, highlight areas), and strawberries with water droplets. The specific structure of the dataset is shown in Table 1. This article selects another hundred strawberry pictures that are different from the test set for image enhancement. By changing the brightness of the picture and flipping the picture horizontally or vertically, the number of training sets is increased to 300.

Table 1. Class information for the test dataset.

	Single Strawberry	Overlapped Strawberries	Connected Strawberries	Branch Shade Strawberries	Multiple Strawberries	Total
Poor light conditions	63	225	15	42	52	397
Set of plastic bags	3	6	3	0	0	12
With water droplets	36	86	7	14	14	157
Patterned label	3	6	1	0	0	10
The testset	175	556	60	117	140	1048

To construct strawberry contour labels, we first converted the instance annotation image Figure 8B(c) into binary annotation, and then used an edge detection algorithm to obtain clear coordinates of the strawberry boundaries (Figure 8B(d)). Afterwards, we converted the ground truth into the standard COCO annotation format, including file name, image size, bounding box, pixel mask, and strawberry contour coordinates.

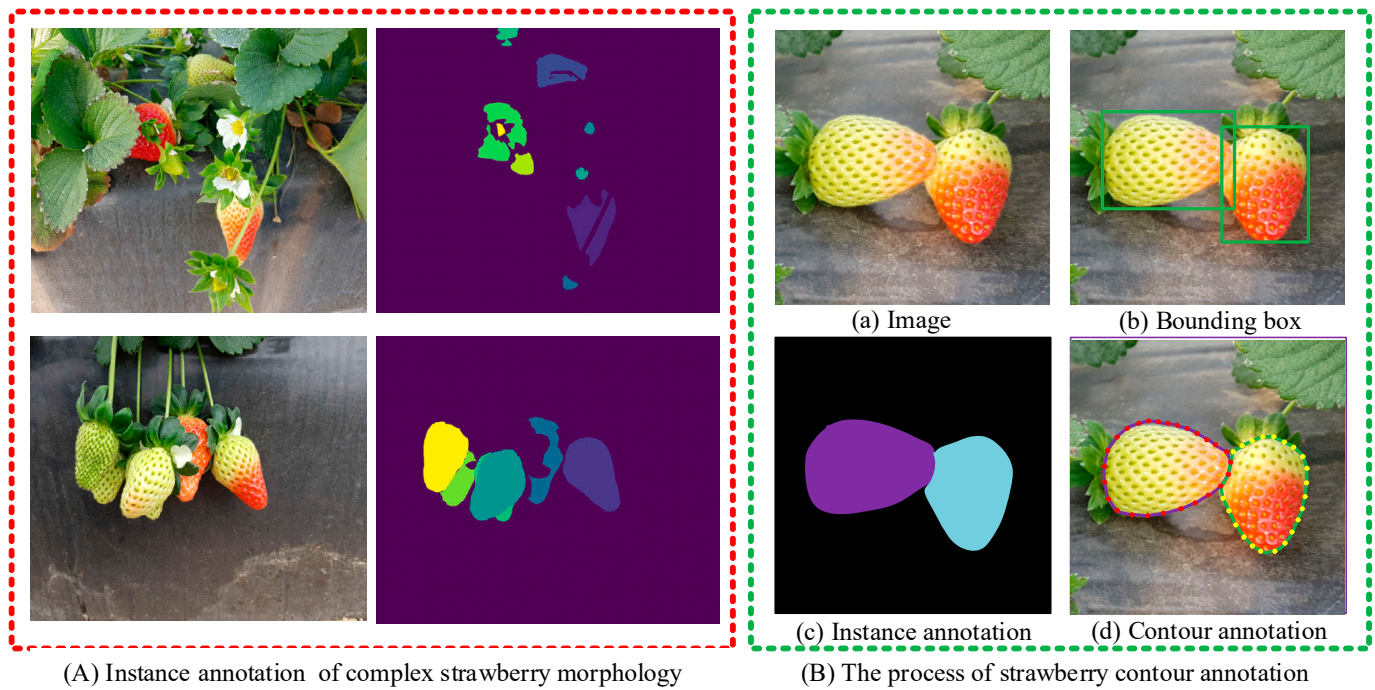


Figure 8. Illustration of the strawberry contour annotation process. (a) Original image. (b) Bounding box image with the strawberry. (c) Extraction of the mask. (d) Polygonal annotation of the strawberry contour. The green boxes represent the marked detection boxes, and the dots represent the marked outline points.

StrawDI_Db1 is a database used for designing and evaluating methods for strawberry instance segmentation. It provides ground-truth segmentation, which generates a mask for each image, where strawberry pixels are identified with labels associated with each strawberry. In this database, the contours of strawberries are accurately marked. The StrawDI_Db1 database contains 3100 photos taken in a strawberry plantation in Spain, captured at various times during the entire harvesting activity. These photos are stored in JPEG format with a resolution of 4032×3024 pixels and 8 bits per color channel. They are released in PNG format, resized to 1008×756 pixels, and divided into training, validation, and test subsets, consisting of 2200, 100, and 800 images respectively. The main challenges faced in implementing the strawberry instance segmentation method with this database are the differences in brightness, perspective, size, and shape of strawberries, as well as possible clustering and occlusion. Figure 8A shows representative images of these difficulties and their corresponding ground-truth segmentation. Difficult images in the strawberry images represent images that contain too many different types of strawberries and have complex overlaps with each other and the leaves.

4.2. Evaluation Criteria

Evaluation Criteria for Object Detection and Semantic Segmentation

On the TongStraw_DB dataset, to facilitate further development of the algorithm, we use mean intersection-over-sum (mIoU) to measure the segmentation accuracy, just as DeepSnake does. It includes FG (foreground segmentation), BG (background segmentation) and AVG (average segmentation). We regard the strawberry bounding as contour lines

with widths equal to 8 pixels. The schematic diagram of mIoU evaluation is shown in Figure 9.

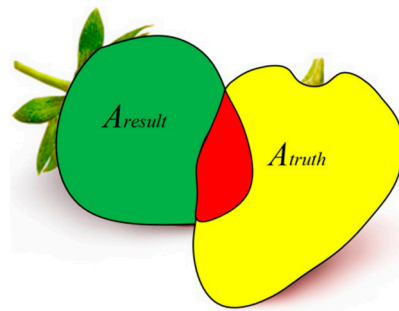


Figure 9. The schematic diagram of mIoU evaluation. The green region is the segmented strawberry. The yellow region is the ground truth. The red region is the rightly segmented strawberry.

The three sizes of strawberries (small, medium and large) were used as a measure to evaluate the segmentation accuracy of the instance on the StrawDI_Db1 dataset. In the region of each instance: small is $\text{area} < 32^2$; medium is $32^2 < \text{area} < 96^2$; and large is $\text{area} > 96^2$. The evaluation of instance segmentation uses average precision (AP). Mask IoU is used to measure the AP metric. The standard AP adopts an IoU of 0.5 and 0.75 ($\text{AP}^{\text{IoU}=0.5}$, $\text{AP}^{\text{IoU}=0.75}$) to distinguish whether a pixel is correctly predicted. Mean average precision (mAP) in general, and for each of the different strawberry sizes that can be identified, is small ($\text{mAP}_{\text{small}}$), medium ($\text{mAP}_{\text{medium}}$) and big (mAP_{big}).

4.3. Results Comparison

4.3.1. Visually Intuitive Evaluation

First of all, we first compared the visual results of StrawSnake with those of Refs. [7,10] in three scenarios on the StrawDI_Db1 dataset. In Figure 10, the first column shows the original images of the three scenarios, while the remaining columns show the visual results of references [7,10], and StrawSnake. Red rectangles indicate areas where the boundary segmentation is not precise, such as at the boundary between leaves and strawberries. Red arrows indicate missed detection or false detection positions. It is evident that in the first scenario, the Ref. [7] missed a green immature strawberry in the top right corner and did not clearly segment mature strawberries with leaf occlusion, as it only used the instance segmentation model Mask-RCNN. Ref. [10], on the other hand, using the contour segmentation model DeepSnake, provided segmentation of leaves occluding the strawberries, although not as refined. In contrast, our StrawSnake provided more refined segmentation results, demonstrating the effects of our DSConv and MFRB in boundary handling and feature fusion. The second scenario is relatively simple, but both Refs. [7,10] encountered the same issues as in the first scenario, whereas StrawSnake still produced perfect results. The third scenario is the most complex, with multiple green immature small strawberries, leading to a missed detection in our method as well. The other two methods missed detecting both small strawberries. In summary, for mature red large strawberries, all three methods could detect them fairly well, but our method showed greater refinement. For immature green strawberries, the detection ability of the other two methods was quite low. This demonstrates that StrawSnake has a more refined and higher-precision visual detection capability.

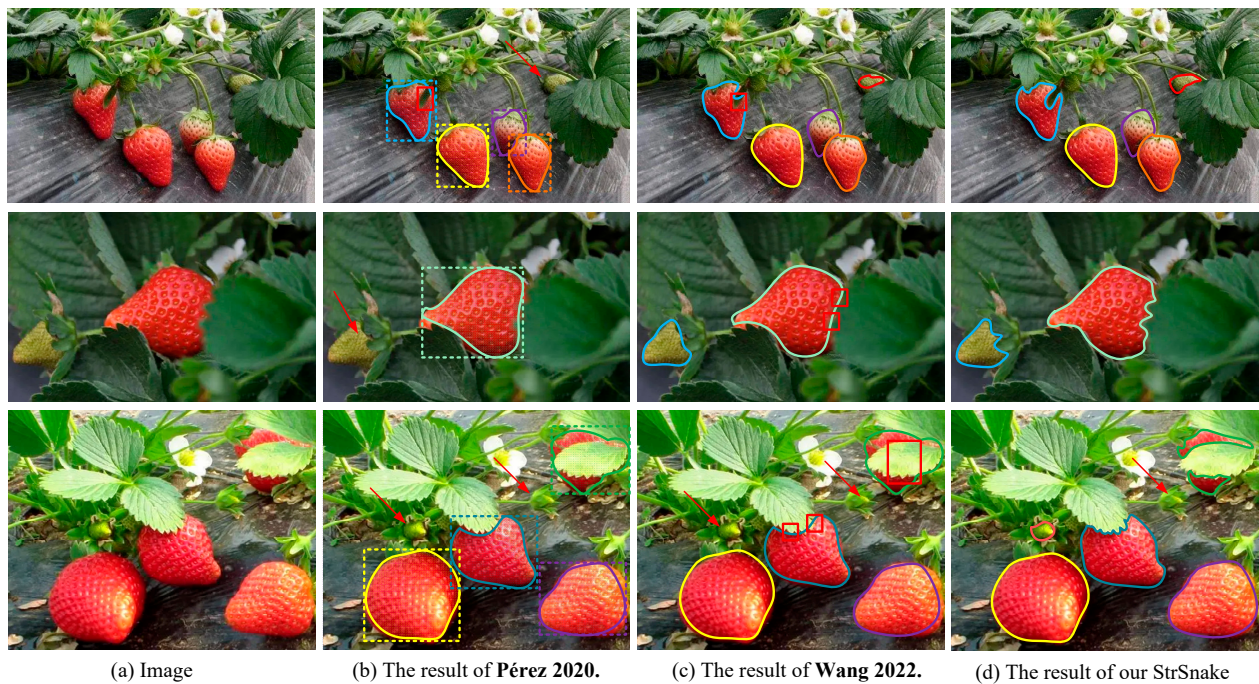


Figure 10. Visual results comparison between [7,10] and StrawSnake.

Additionally, Figure 11 shows the qualitative visual results of our StrawSnake on two datasets. Figure 11A presents the high-quality visual results of our method in some classic scenarios. Figure 11B includes complex situations such as occlusion of immature small strawberries and mature red strawberries, indicating that our method, in complex situations, relies too heavily on the specificity of boundaries, lacking the ability to understand context that led to these unsatisfactory results.

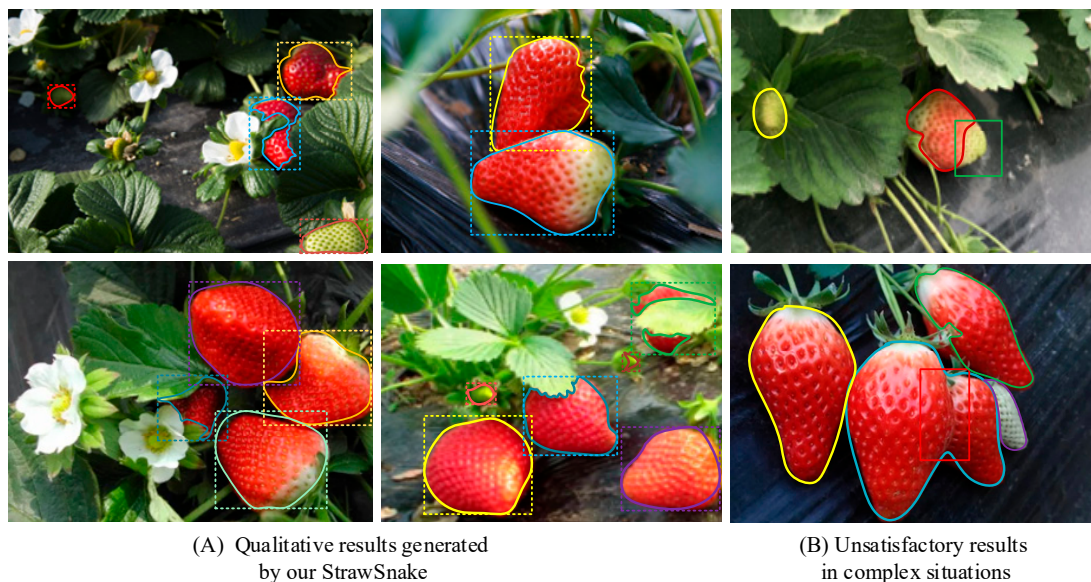


Figure 11. Qualitative results generated by our StrawSnake and unsatisfactory results in complex situations. The dotted lines of different colors represent the detection boxes of different strawberries, and the implementations of different colors represent the Outlines of different strawberries.

4.3.2. Quantitative Evaluation

For quantitative evaluation of our StrawSnake, we use classic contour-based methods for comparison. The experimental images are all from 200 testing images on the

StrawDI_Db1 dataset. It should be noted that DeepSnake is not a segmentation model for strawberries, so we do not compare it to our StrawSnake. In the specific area of strawberry instance segmentation, different versions of Mask R-CNN were used by [7,8,25], with the choice of backbone being the main difference between them. In Table 2 and Figure 12, we summarize the performance comparisons of these methods with our approach. The experiments were conducted on the StrawDI_Db1 database of strawberry crop images provided based on [7]. Firstly, the method proposed by [25] achieved an average precision (mAP) of 45.36 with a processing speed of 5 frames per second (fps). On the other hand, the method proposed by [7] offered a lower precision level (mAP = 43.85) but could operate at a higher rate of 10 fps. The method proposed by [8] achieved an average precision of mAP = 52.61 with a processing speed of 30 fps, showing an overall improvement in performance compared to the previous two. The trade-off between segmentation quality and processing time is particularly crucial for the application of models in real-time automatic harvesting systems. To run these networks, these systems must have high computational capabilities to be executed in a very short time. In contrast, the algorithm applied in this study, specifically designed for this task, effectively reduces the computation time (approximately 10 milliseconds). In our StrawSnake research, the contour-based segmentation architecture aimed to reduce processing time to achieve real-time performance. Experimental results demonstrate high efficiency in terms of accuracy and speed: mAP = 59.23, fps = 39.2. These values are significantly higher than those obtained using other strawberry contour segmentation methods.

Table 2. Performance comparison of the models in terms of AP and FPS on the StrawDI_Db1.

Methods	Perez et al. [8]	Yu et al. [25]	Ref. [7]	StrawSnake
mAP	52.61	45.36	43.85	59.23
mAP _{small}	16.96	7.35	7.54	24.26
mAP _{medium}	65.26	50.03	51.77	71.29
mAP _{big}	53.31	78.30	75.90	82.87
AP ^{IoU=0.5}	69.37	76.57	74.24	81.54
AP ^{IoU=0.75}	57.84	47.09	45.13	66.73
fps	30	10	5	39.2

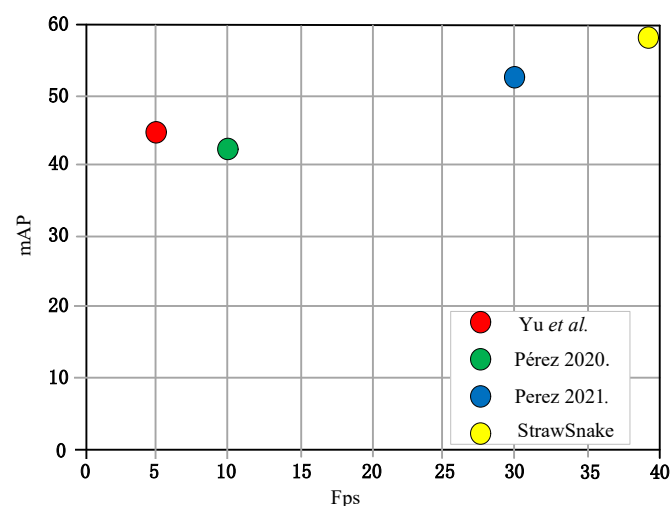


Figure 12. Performance comparison of the models in terms of mean fps and mAP values in the StrawDI_Db1 test set [7,8,25].

Furthermore, we also conducted experimental tests on the TongStraw_DB dataset. Here, StrawSnake (DSConv) indicates the absence of using MFRB. A comparison with other segmentation methods in Table 3 shows that our StrawSnake achieved an FG of 85.6%, which is significantly higher than Yifan et al. [14] by 6.1%, 9.1%, and 4.4% respectively. The

AGV reached 87.4%, reflecting improvements of 4.9%, 6.1%, and 4.4%. The BG reached 88.3%, with enhancements of 5.6%, 4.9%, and 3.8% respectively. It is evident that the StrawSnake designed for strawberry segmentation outperforms other methods. Despite a slight decrease in speed due to the inclusion of DSConv and MFRB, real-time performance of 39.5 fps is still achieved.

Table 3. Comparison of results with the latest methods on TongStraw_DB.

	FG	BG	AVG	fps
Cao et al. [18]	79.5%	82.7%	82.5%	36.8
Wang et al. [19]	76.5%	80.9%	81.3%	22.2
Yifan et al. [14]	80.5%	84.5%	83.0%	18.6
DSsnake [6]	81.2%	80.5%	81.6%	33
StrawSnake (DSConv)	83.1%	86.9%	85.7%	41.1
StrawSnake+ (MFRB)	80.2%	83.6%	82.4%	40.7
StrawSnake	85.6%	88.3%	87.4%	39.5

4.3.3. Ablation Studies

As shown in Table 4, we conducted a total of 8 groups of ablation experiments, analyzing and comparing in detail the impact of each module on our method, as shown in Table 1. The detectors used were based on the CenterNet and YOLO V8 architectures. For example, the second group of experiments indicates that when DSConv and MFRB are not used, the mAP = 39.71, $AP^{IoU=0.5} = 72.52$, $AP^{IoU=0.75} = 45.82$. In summary, through comparisons of experiments in the first group, the third group, and the other three groups, it was found that DSConv increased the mAP by an average of 6.21%, $AP^{IoU=0.5}$ by 5.53%, and $AP^{IoU=0.75}$ by 4.81%, with an average decrease in operating speed of 5fps. Similarly, through comparisons of experiments in the first group, the fourth group, and the other three groups, it was found that MFRB increased the mAP by an average of 2.52%, $AP^{IoU=0.5}$ by 2.77%, and $AP^{IoU=0.75}$ by 2.35%, with an average decrease in operating speed of 5.8fps.

Table 4. Ablation experiment.

	CenterNet	YOLO V8	DSConv	MFRB	mAP	$AP^{IoU=0.5}$	$AP^{IoU=0.75}$	fps
1	✓	-	-	-	36.56	70.29	44.70	41.2
2	-	✓	-	-	39.71	72.52	45.82	42.5
3	✓	-	✓	-	42.88	75.63	49.95	36.6
4	✓	-	-	✓	39.42	72.60	47.57	35.8
5	✓	-	✓	✓	45.72	78.59	52.78	33.9
6	-	✓	✓	-	54.81	78.12	63.37	40.2
7	-	✓	-	✓	51.36	75.89	61.25	40.8
8	-	✓	✓	✓	59.23	81.54	66.73	39.2

5. Conclusions

This paper presents a strawberry fruit contour segmentation method called StrawSnake, characterized by contour representation. It overcomes the issues of poor generality and robustness in traditional computer vision algorithms, showing high precision and real-time capabilities. In strawberry fruit detection, we propose using DSConv to learn boundary information and improve the differentiation between strawberries and leaves. By utilizing MFRB for feature fusion learning, we enhance the accuracy of strawberry feature extraction. We established the TongStraw_DB dataset for strawberry contour and conducted analysis using StrawDI_Db1. Compared with state-of-the-art methods, our approach demonstrates significant advantages in small object detection and boundary extraction based on visually intuitive results. However, challenges remain in detecting multiple overlapping strawberries and immature green strawberries, leading to some false detections. Quantitative

evaluation on StrawDI_Db1 shows our method achieves mAP = 59.23 and fps = 39.2. On TongStraw_DB, the evaluation results in AVG = 87.4 and fps = 39.5. This validates that our method maintains a high advantage in both accuracy and speed. Finally, through ablation experiments, we demonstrate the performance improvement brought by DSConv and MFRB.

Author Contributions: Conceptualization, Z.G.; methodology, Z.G. and X.H.; software, Z.G.; validation, X.H. and X.M.; formal analysis, B.Z.; investigation, Z.G.; writing—original draft preparation, Z.G. and X.H.; writing—review and editing, Z.G., B.Z., X.M. and H.W.; project administration, X.H.; funding acquisition, Z.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the High-level Talents Research Start-up Fund supported by Jiangsu Shipping College (HYRC/202405), the Nantong Social Livelihood Science and Technology Project (MS2023017) and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (24KJB580005).

Data Availability Statement: The datasets generated during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: All authors declare no conflict of interest.

References

- Hilmar, H.Z.; Gesa, B.; Matin, Q. Positive public attitudes towards agricultural robots. *Sci. Rep.* **2024**, *14*, 15607.
- Muñoz-Postigo, J.; Valero, E.; Martínez-Domingo, M.; Lara, F.; Nieves, J.; Romero, J.; Hernández-Andrés, J. Band selection pipeline for maturity stage classification in bell peppers: From full spectrum to simulated camera data. *J. Food Eng.* **2024**, *365*, 111824. [\[CrossRef\]](#)
- Liu, H.; Wang, X.; Zhao, F.; Yu, F.; Lin, P.; Gan, Y.; Ren, X.; Chen, Y.; Tu, J. Upgrading swin-B transformer-based model for accurately identifying ripe strawberries by coupling task-aligned one-stage object detection mechanism. *Comput. Electron. Agric.* **2024**, *218*, 108674. [\[CrossRef\]](#)
- Zhang, B.; Ou, Y.; Yu, S.; Liu, Y.; Qiu, W. Gray mold and anthracnose disease detection on strawberry leaves using hyperspectral imaging. *Plant Methods* **2023**, *19*, 148–158. [\[CrossRef\]](#) [\[PubMed\]](#)
- Wang, J.; Wang, L.; Han, Y.; Zhang, Y.; Zhou, R. On Combining DeepSnake and Global Saliency for Detection of Orchard Apples. *Appl. Sci.* **2021**, *11*, 6269. [\[CrossRef\]](#)
- Peng, S.; Jiang, W.; Pi, H.; Li, X.; Bao, H.; Zhou, X. Deep Snake for Real-Time Instance Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
- Pérez-Borrero, I.; Marín-Santos, D.; Gegúndez-Arias, M.E.; Cortés-Ancos, E. A fast and accurate deep learning method for strawberry instance segmentation. *Comput. Electron. Agric.* **2020**, *178*, 105736. [\[CrossRef\]](#)
- Perez-Borrero, I.; Marin-Santos, D.; Vasallo-Vazquez, M.J.; Gegundez-Arias, M.E. A new deep-learning strawberry instance segmentation methodology based on a fully convolutional neural network. *Neural Comput. Appl.* **2021**, *33*, 15059–15071. [\[CrossRef\]](#)
- He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
- Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
- Lowe, G.D. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [\[CrossRef\]](#)
- Fergus, R.; Ranzato, M.; Salakhutdinov, R.; Taylor, G.; Yu, K. Deep learning methods for vision. In Proceedings of the CVPR 2012 Tutorial, Providence, RI, USA, 16–21 June 2012.
- Shin, J.; Chang, Y.K.; Heung, B.; Nguyen-Quang, T.; Price, G.W.; Al-Mallahi, A. A deep learning approach for RGB image-based powdery mildew disease detection on strawberry leaves. *Comput. Electron. Agric.* **2021**, *183*, 106042. [\[CrossRef\]](#)
- Bai, Y.; Yu, J.; Yang, S.; Ning, J. An improved YOLO algorithm for detecting flowers and fruits on strawberry seedlings. *Biosyst. Eng.* **2024**, *237*, 1–12. [\[CrossRef\]](#)
- Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 10012–10022.
- Pang, F.; Chen, X. MS-YOLOv5: A lightweight algorithm for strawberry ripeness detection based on deep learning. *Syst. Sci. Control Eng.* **2023**, *11*, 2285292. [\[CrossRef\]](#)
- Afzaal, U.; Bhattarai, B.; Pandeya, Y.R.; Lee, J. An Instance Segmentation Model for Strawberry Diseases Based on Mask R-CNN. *Sensors* **2021**, *21*, 6565. [\[CrossRef\]](#)
- Cao, L.; Chen, Y.; Jin, Q. Lightweight Strawberry Instance Segmentation on Low-Power Devices for Picking Robots. *Electronics* **2023**, *12*, 3145. [\[CrossRef\]](#)

19. Cai, C.; Tan, J.; Zhang, P.; Ye, Y.; Zhang, J. Determining Strawberries' Varying Maturity Levels by Utilizing Image Segmentation Methods of Improved DeepLabV3+. *Agronomy* **2022**, *12*, 1875. [[CrossRef](#)]
20. Zhou, E.; Xu, X.; Xu, B.; Wu, H. An enhancement model based on dense atrous and inception convolution for image semantic segmentation. *Appl. Intell.* **2022**, *53*, 5519–5531. [[CrossRef](#)]
21. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
22. Yang, L.; Zhu, Z.; Sun, L.; Zhang, D. Global Attention-Based DEM: A Planet Surface Digital Elevation Model-Generation Method Combined with a Global Attention Mechanism. *Aerospace* **2024**, *11*, 529. [[CrossRef](#)]
23. Zhao, S.; Liu, J.; Wu, S. Multiple disease detection method for greenhouse-cultivated strawberry based on multiscale feature fusion Faster R_CNN. *Comput. Electron. Agric.* **2022**, *199*, 107176. [[CrossRef](#)]
24. Bochkovskiy, A.; Wang, C.-Y.; Liao, H. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
25. Yu, Y.; Zhang, K.; Yang, L.; Zhang, D. Fruit detection for strawberry harvesting robot in non-structural environment based on mask R-CNN. *Comput. Electron. Agric.* **2019**, *163*, 104–846. [[CrossRef](#)]
26. Qi, Y.; He, Y.; Qi, X.; Zhang, Y.; Yang, G. Dynamic Snake Convolution based on Topological Geometric Constraints for Tubular Structure Segmentation. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; Volume 3. [[CrossRef](#)]
27. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
28. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence, Montreal, QC, Canada, 7–12 December 2015; Volume 39, pp. 1137–1149.

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.