

Article

A Method of Fast Segmentation for Banana Stalk Exploited Lightweight Multi-Feature Fusion Deep Neural Network

Tianci Chen , Rihong Zhang, Lixue Zhu * , Shiang Zhang and Xiaomin Li 

College of Mechanical and Electrical Engineering, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China; Chentianci1206@163.com (T.C.); zhangrihong@zhku.edu.cn (R.Z.); GZZSA@163.com (S.Z.); lixiaomin@zhku.edu.cn (X.L.)

* Correspondence: zhulixue@zhku.edu.cn

Abstract: In an orchard environment with a complex background and changing light conditions, the banana stalk, fruit, branches, and leaves are very similar in color. The fast and accurate detection and segmentation of a banana stalk are crucial to realize the automatic picking using a banana picking robot. In this paper, a banana stalk segmentation method based on a lightweight multi-feature fusion deep neural network (MFN) is proposed. The proposed network is mainly composed of encoding and decoding networks, in which the sandglass bottleneck design is adopted to alleviate the information a loss in high dimension. In the decoding network, a different sized dilated convolution kernel is used for convolution operation to make the extracted banana stalk features denser. The proposed network is verified by experiments. In the experiments, the detection precision, segmentation accuracy, number of parameters, operation efficiency, and average execution time are used as evaluation metrics, and the proposed network is compared with Resnet_Segnet, Mobilenet_Segnet, and a few other networks. The experimental results show that compared to other networks, the number of network parameters of the proposed network is significantly reduced, the running frame rate is improved, and the average execution time is shortened.

Keywords: banana stalk; dilated convolution; lightweight network; multi-feature structure; sandglass structure



Citation: Chen, T.; Zhang, R.; Zhu, L.; Zhang, S.; Li, X. A Method of Fast Segmentation for Banana Stalk Exploited Lightweight Multi-Feature Fusion Deep Neural Network. *Machines* **2021**, *9*, 66. <https://doi.org/10.3390/machines9030066>

Academic Editor: Marcelo H. Ang Jr.

Received: 2 February 2021

Accepted: 16 March 2021

Published: 18 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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

According to the Food and Agriculture Organization of the United Nations, currently there are 137 countries and regions planting banana, and the banana production is still rising [1–3]. In 2016, the world's total banana production reached 1113.23 million tons. In China, in 2018, banana production reached 11.21 million tons. However, at present, bananas are mainly harvested by manual labor, and a weight of more than 25 kg is hard for farmers to pick. With the emergence of the concepts of digital and intelligent agriculture, agricultural robots have become a research hotspot in the field of agricultural application [4–7]. The research on intelligent and automatic banana picking robots has important practical value and broad application prospects. For banana harvesting, the key point is to cut the banana stalk then harvest a whole bunch of bananas. Therefore, the automatic harvesting robot needs to first detect the banana stalk, and then transform it into a three-dimensional point cloud and process it as the target operation point [8]. Finally, the end-effector of the robot will automatically work to the target cutting point of the fruit stalk. In this process, the accurate and rapid recognition of banana stalk is the premise and foundation of robot automatic harvesting. If the banana fruit stalk is wrongly identified or the banana fruit is taken as the target recognition object, it will be difficult for the robot to cut to the banana stalk with the end-executing tool in the subsequent work, which will affect the accurate and automatic harvesting of the robot in the field of precision agriculture to some extent. As the core component of a picking robot, the visual system is a premise to realize an accurate operation of a fruit picking robot. In addition, the speed and accuracy of recognition affect

the subsequent picking directly. Therefore, the recognition and segmentation of banana stalk is an important step to realize the robot automatic picking.

The main problems of a visual system in picking robots are as follows:

- (1) Banana picking is mainly done at the position of the cutting stalk, and the target of the stalk is relatively small compared to the banana fruit. Also, the tilt degree is different, so it is difficult to use shape features such as that used for apple, orange, or tomato.
- (2) There are many interfering factors in an orchard environment, and the banana stalk is basically consistent in color with a background environment. Compared with citrus, litchi, strawberry and other fruit with an obvious color difference, banana stalks are more difficult to be accurately detected.
- (3) As a picking robot works outdoors, its visual system needs to be deployed using a mobile terminal. Although commonly used large network models have good detection performance, they are deployed with a mobile terminal with a low detection speed, which cannot meet the real-time requirements of the equipment. Thus, the development of lightweight and efficient recognition and segmentation algorithm is the main objective of this research.

In recent studies, many feature extraction methods for target crops were proposed, and different image processing algorithms were developed, achieving the segmentation accuracy of over 85% for multiple fruits [9–14]. However, these recognition algorithms based on shape and color features are not suitable for banana stalk-recognition and lack generalization due to different environment interference. Also, some researchers used a larger sample size for training recognition, but it is also based on the texture color features of the detection target [15,16]. In some studies on banana fruit recognition, bananas have been recognized based on structural features, such as color and texture, achieving the recognition accuracy of over 80% [17–20]. However, the banana stalk is smaller than the banana fruit, so the algorithm finds it difficult to recognize the banana stalk in a complex environment. Although some scholars proposed an effective method for non-destructive testing and achieved good results [21,22], these methods used thermographic inspection technology and pay more attention to the details of defects, so it is not suitable for the identification of banana stalk in outdoor environment. To solve these problems, many studies used the deep learning-based method to recognize different fruit or estimation [23–25]. For instance, the mask-RCNN algorithm was used, and the model with Resnet and Segnet as the backbone network built, which greatly improved the segmentation accuracy, reaching the recognition accuracy of 95% [26–28]. Also, the use of neural network has also achieved good results in some other applications [29,30]. Although a high-capacity network model can improve the segmentation accuracy, it sacrifices performance and running speed at the expense of mobile edge devices. In recent literature, many different network architectures have been constructed, the applicability of depth separable convolution in the lightweight YOLO has been proven [31,32], and it has been demonstrated that the usage of residual blocks can better extract features [33,34]. However, the detection network can only locate the target in a small area, but not achieve the pixel-level segmentation effect, and it is difficult to present the contour shape completely, affecting the precision of subsequent harvest. Some studies considered that the information distillation mechanism is not efficient and proposed improved recognition methods [35–37]. Although the lightweight performance was improved, the overall number of parameters is still small.

The main contributions of this paper can be summarized as follows:

- (1) A lightweight sandglass residual feature extraction network is proposed to extract image feature information. The segmentation accuracy of the proposed network is not affected when the number of network layers is reduced.
- (2) In the decoding network, the dilated convolution with different expansion rates is adopted for feature fusion, so that the banana stalk features are denser and decoding can be realized more effectively.

- (3) The quantitative analysis of five different networks shows that the proposed network model Sandglass_MFN has good performance. In a complex orchard environment, the banana stalk can be segmented effectively.

The remainder of this paper is organized as follows. Section 2 introduces material and methods. Section 3 describes deep learning network model training. Section 4 shows the experiment results and analysis and Section 5 concludes the paper.

2. Materials and Methods

2.1. Image Acquisition and Processing

To verify the effectiveness and feasibility of the proposed model algorithm, which used the sandglass structure in the encoding network combined with multi-feature fusion structure of the decoding network, the experiment was conducted using the data collected on the banana plantation base in Suixi County, Zhanjiang city in late June, 2020. A Huawei P30 (EI-A100) was used to take 2088 photos of banana fruit and stalks of Cultivar No.3, and the imaging distance between them and the canopy was about 1.0~2.0 m. To ensure the diversity of data, a variety of environmental background data were collected under different weather conditions, including sunny, cloudy, phototropic, and backlight conditions. At the same time, different angles were used in the data acquisition process. A manual labeling method was used to label the banana stalk of the collected data to obtain the training label images. An example of the data sample is shown in Figure 1.

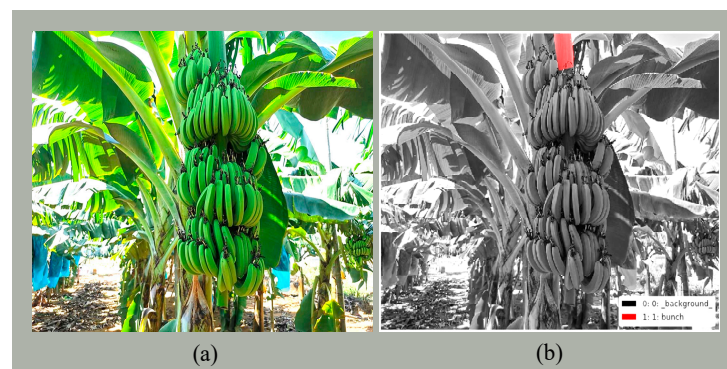


Figure 1. Data sample. (a) The original image; (b) The training label image.

For network model training, additional data features were used to improve the accuracy of model training. Namely, the brightness, chroma, and contrast of the original image were enhanced and weakened, and the inclination of the fruit stalk was changed. The data of 2088 images collected are further expanded. Each image was executed with a probability of 0.8. Based on the original image, the brightness, contrast, and chroma were scaled and transformed randomly according to the transformation factor of 0.7~1.3. The stalk angle was also randomly rotated within the range of 10° to the left or right. The total sample number was set to be 10,440 images. To make the system acquire labels accurately during training, the label image was operated in the same way with the original image, and the corresponding label images can be obtained and put into the network training while obtaining the enhanced images. After data enhancement, a total of 10,440 images were obtained and randomly divided into training, validation, and test sets according to the proportion of 8:1:1. In the network model training process, in order to reduce the training difficulty, the resolution of images in the training set was reduced to 576×576 . The test set was used to verify the performance of the optimized network, and the validation set was used to evaluate the effectiveness of the trained network model.

2.2. Network Model Construction

2.2.1. Sandglass Encoding Network Design

The classical residual bottleneck structure was first proposed in Resnet, and it is mainly composed of three convolutional layers. In this structure, the channel is reduced by a 1×1 point-state convolution, one 3×3 convolution is used for spatial feature extraction, and another 1×1 convolution is used for channel expansion, as shown in Figure 2a. Although this structure has achieved great success in the weighted network, it is difficult to construct a lightweight network and deploy it in a mobile terminal since this model includes a standard 3×3 deep convolution, and has a large number of parameters and large amount of computation. It is more suitable for high precision detection and recognition rather than real time. The anti-residual block was first introduced in the MobilenetV2 which is used for the lightweight coding network, and in this structure linear bottlenecks are connected with shortcut keys, which greatly improves network performance and optimizes model complexity. However, a high-dimensional space is structurally compressed and mapped to the low dimensional feature tensor, so it is difficult to obtain enough feature information. Furthermore, the connection of shortcut keys between low-dimensional tensors also increases the probability of feature loss. This approach can be applied to the lightweight domain, but it is still difficult to tailor the number of network layers to some extent.

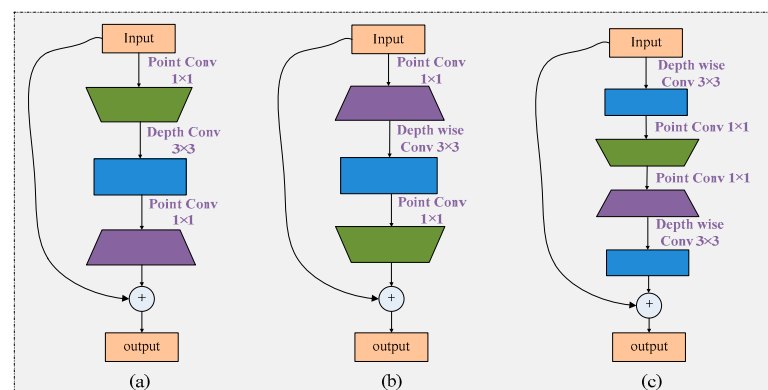


Figure 2. Different structural block designs. (a) Residual bottleneck design; (b) Reverse residual bottleneck design; (c) Residual sandglass design.

By combining the characteristics of the residual bottleneck and the anti-residual structure, a sandglass structure network was designed, and its structure is shown in Figure 2c. This network design has been mainly applied to the encoding network, and its structure is similar to the bottleneck structure. However, in order to reduce the number of network model parameters, depth separable convolution is introduced. The order between each module is adjusted, the connection mode of shortcut keys changed to join the high-dimensional space, and the traditional residual bottleneck structure is embedded into a new sandglass structure. The functions of using this structure are as follows:

- (1) More information from the bottom layer is retained when the data is propagating through the deep network, and the shortcut key connection is set on the high-dimensional features to extract richer target features.
- (2) Due to deep separable convolution and appropriate clipping of network modules, the network can be reduced.
- (3) The combination of this structure with the subsequent multi-feature fusion structure can give better play to the network performance.

The specific process of the sandglass structure is as follows. First, the tensor data is processed by a 1×1 pointwise convolution to adjust the channel into a higher dimension to be the input tensor. This is the first step. Then, the convolution kernel is used as a depth separable convolution learning feature. This is the second step. In addition, a bottleneck

structure is added to the middle of structure to encode the inter-channel information with point-state convolution. It should be noted that a 3×3 convolution kernel represents a depth separable convolution. Then it generates the feature layer prior to step 3. To ensure that feature information is not lost, step 3 only used the normalized function without using the activation function. Finally, according to difference stride of the convolution kernel, if convolution of step size 2 is used in the procedure, then step 4 is used; otherwise, step 5 is used, then shortcut keys is introduced to do channel splicing in high dimensions, and the output tensor data is obtained. The structure of the sandglass module is shown in Figure 3.

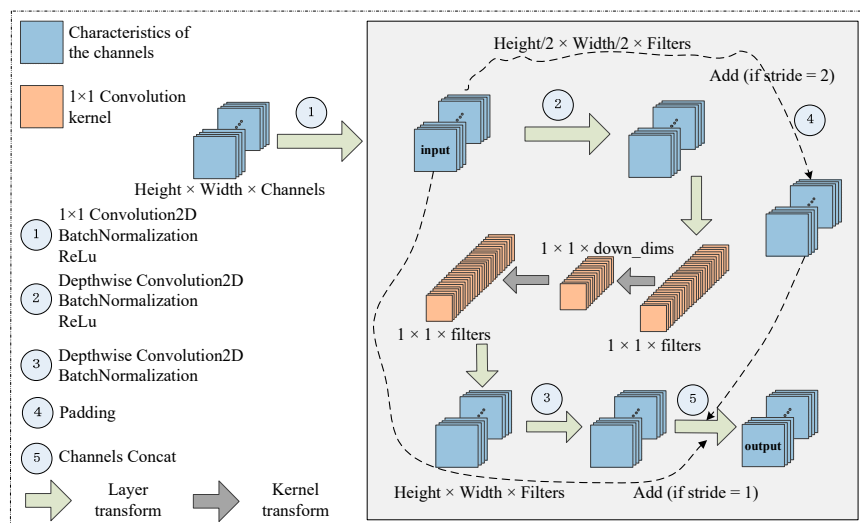


Figure 3. The Sandglass network structure.

Since point state convolution can be used to encode the information between channels, applying the bottleneck structure to the intermediate channel encoding is beneficial to reduce the number of network parameters. To capture spatial information, the channel and spatial features are extracted by combining the point-state and deep convolutions. A 3×3 lightweight kernel is introduced for deep convolution extraction, and the depth separable convolution method is used to reduce the number of network model parameters effectively. Based on the sandglass module structure, the encoding network is constructed, and its structure is shown in Figure 4.

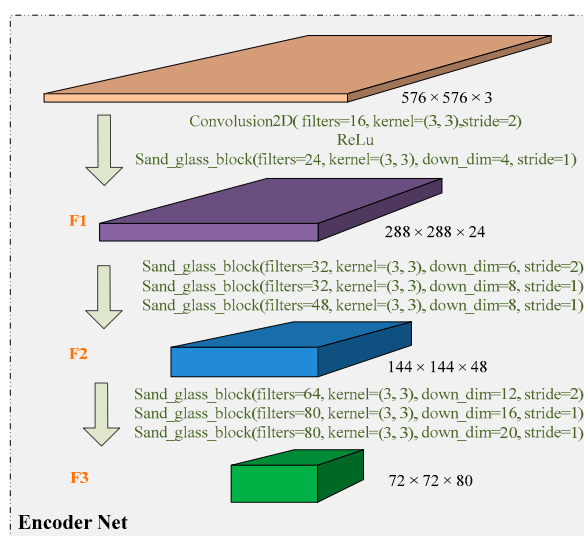


Figure 4. The encoding network structure.

In the encoding network, deep convolution is first used to extract data features, and then the channel of the feature layer is adjusted and the image resolution is compressed to obtain the feature layer F1. Then, the sandglass module is used and different parameters are adjusted to obtain the characteristic layer F2. Furthermore, while compressing the image resolution, using the sandglass module can extract the feature layer F3, so as to obtain high-level semantic information of a banana stalk. The purpose of using the sandglass structure in the encoding network is to realize the lightweight without losing the feature information, at the same time, combining with the subsequent multi-feature fusion can make the structure have a better play.

2.2.2. Multi-Feature Decoding Network Design

After the Segnet, which uses the max-pooling indices received from the corresponding encoders to perform the nonlinearity of the input feature map, encoding and decoder structures have been widely used in semantic segmentation model. In the encoding network, with the increase in the number of convolutional layers, image data size and the number of channels, the target features with high-level semantics are finally extracted. In contrast, in the decoding network, using the learned features, the feature map that retains the size of the original image is constructed to achieve the segmentation of the image pixel points. Through the learning of a large amount of image data, the network model parameters corresponding to the mapping between the original image and the label can be adjusted so as to achieve the accurate prediction of pixel points.

For the segmentation of small targets, it is particularly important to use effective features extracted from the encoding structure for deep decoding. Multi-scale fusion can improve the feature sensitivity to a certain extent, but its effect on small targets is not obvious. Likewise, pooling can increase the receptive field, but characteristic details are lost. In contrast, dilated convolution can increase the receptive field without loss of either resolution or coverage. An R parameter is added to the original convolution as expansion rate, and the visual field sense of the convolution kernel is expanded to extract different target features. The convolution kernels with different expansion rates are shown in Figure 5.

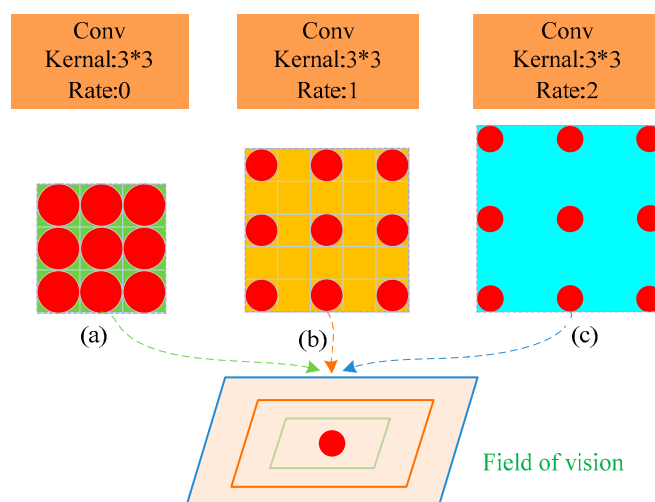


Figure 5. Expansion of the convolution kernel. (a) 3×3 convolution kernel at $R = 0$; (b) 3×3 convolution kernel at $R = 1$; (c) 3×3 convolution kernel at $R = 2$.

The viewing area of dilated convolution in every feature layer mapped on the original image region can be calculated as follow:

$$r_{i+1}^2 = [(r_i - 1) + (2l + 1)]^2 \quad (1)$$

where r_i is the length of the convolution kernel, l is the Expansion coefficient, and r_{i+1}^2 is the range of viewing area. In the case of the same size of convolution kernel, it has a larger receptive field.

And the dilated convolution in two dimensions can be defined as follows:

$$(F *_l k)(p) = \sum_{s+lt=p} F(s)k(t) \quad (2)$$

where $*_l$ is dilated convolution and its domain is p . F is the input image and s is its definition domain, namely the center of the convolution operation; k is the kernel, and t is the domain. Compared with ordinary convolution, the condition of void convolution changes from $s + t = p$ to $s + lt = p$, i.e., each convolution kernel only operates with the elements in the position multiple of l in image F .

In the decoding network, the image features extracted by the encoding network are further decoded, and the convolution kernels with the expansion rate l of 2, 4, 8, and 16 were used to conduct the convolution computation on the original image to obtain more stalk features, making the extracted features denser. In order not to lose the original image features, the original image features and the newly generated image features are fused into a parallel channel. Furthermore, the image features extracted from the intermediate layer of the coding network were introduced to construct a serial channel, and the serial and parallel channels were then merged, so as to obtain the fruit stalk features of the representational model categories, realized the multi-feature fusion of the fruit stalk, and propagated feature to the higher-level network.

The decoding network consists of six convolutional layers and two splicing layers. Firstly, based on the characteristics of the fruit stalk obtained from the encoding network, the parallel channel was constructed using four dilated convolution operations with different expansion rates. The splicing layer was used to fuse the features of different fruit stalk. Next, the output size of the splicing layer was adjusted, and the constructed channels and the characteristic layer of the encoding network constitute a series of parallel channels. Then, a splicing layer was used again to fuse the features of the series and parallel channels. Finally, two depth-separable convolutions were used to decode the features and adjust the number of channels which had been fused to obtain the final segmented image. The decoding network is shown in Figure 6.

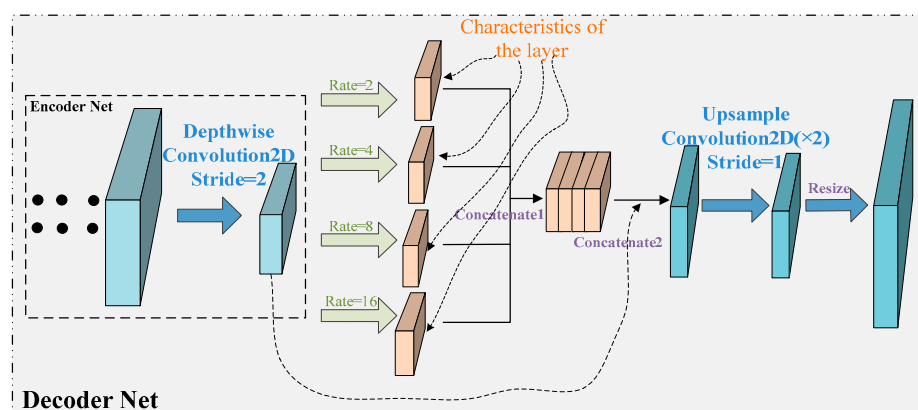


Figure 6. The decoding network structure.

Combined with the adopted encoding network and decoding network, the whole network model framework is constructed, as shown in Figure 7 and the algorithm block diagram is shown in Figure 8.

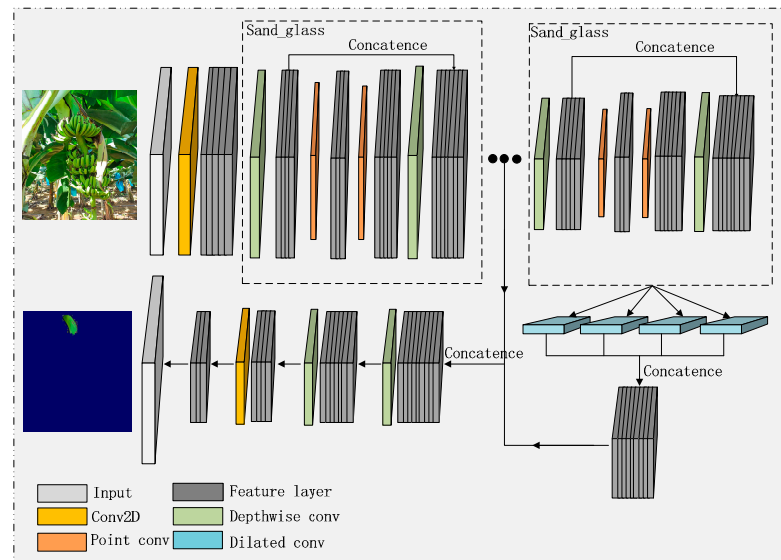


Figure 7. The Sandglass-MFN network model structure.

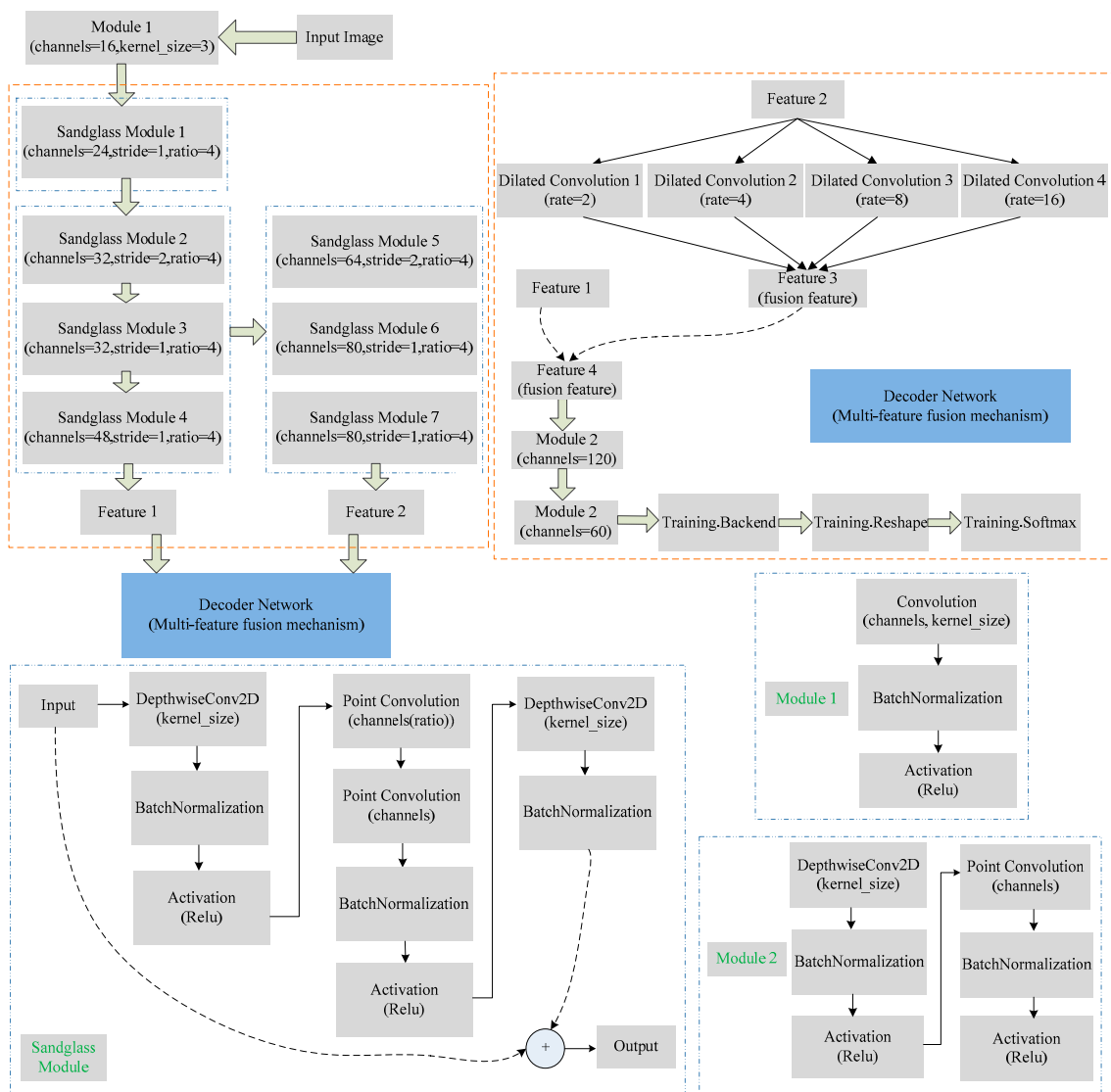


Figure 8. The algorithm block diagram of Sandglass-MFN network.

3. Deep Learning Network Model Training

About the whole network structure of the banana stalk segmentation, the specific parameters of each layer are shown in Table 1. Take the size of the input image of 576×576 as an example. The point convolution uses 1×1 convolution kernel, and the other convolution kernels use 3×3 lightweight convolutions. In all the sandglass blocks, the channel contraction of the input layer is four times the same as the number of bottleneck channels.

Table 1. The Sandglass-MFN structure parameters.

| Layer | Output Size | Network Layer Structure |
|------------------------------------|-------------------------------------|-----------------------------------|
| Original image | $576 \times 576 \times 3$ | None |
| Conv2D | $288 \times 288 \times 16$ | Stride = 1, ReLu6 |
| Sandglass Block1 | $288 \times 288 \times 24$ | Stride = 1, Bottleneck channel 4 |
| Sandglass Block2 | $144 \times 144 \times 32$ | Stride = 2, Bottleneck channel 6 |
| Sandglass Block3 | $144 \times 144 \times 32$ | Stride = 1, Bottleneck channel 8 |
| Sandglass Block4 | $144 \times 144 \times 48$ | Stride = 1, Bottleneck channel 8 |
| Sandglass Block5 | $72 \times 72 \times 64$ | Stride = 2, Bottleneck channel 12 |
| Sandglass Block6 | $72 \times 72 \times 80$ | Stride = 1, Bottleneck channel 16 |
| Sandglass Block7 | $72 \times 72 \times 80$ | Stride = 1, Bottleneck channel 20 |
| Dilated Convolution ($\times 4$) | $72 \times 72 \times 60 (\times 4)$ | Expansion rate = 2, 4, 8, 16 |
| Concentration Layer1 | $72 \times 72 \times 240$ | None |
| Upsample1 | $144 \times 144 \times 240$ | Linear interpolation |
| Point Convolution1 | $144 \times 144 \times 256$ | Stride = 1, ReLu6 |
| Point Convolution2 | $144 \times 144 \times 48$ | Stride = 1, ReLu6 |
| Concentration Layer2 | $144 \times 144 \times 304$ | None |
| Depthwise Convolution | $144 \times 144 \times 120$ | Stride = 1, ReLu6 |
| Upsample2 | $288 \times 288 \times 64$ | Linear interpolation |
| Depthwise Convolution | $288 \times 288 \times 32$ | Stride = 1, ReLu6 |
| Conv2D | $288 \times 288 \times 2$ | Stride = 1, ReLu6 |
| Resize | $576 \times 576 \times 2$ | None |

In this work, the entire experimental platform configuration used for the training and evaluation of all the neural network are presented in Table 2.

Table 2. Experimental platform configuration.

| Specification | Details |
|--------------------------|--|
| Operating System | Ubuntu 18.04, 64-bit Operating System |
| CPU | Intel Xeon(R) Gold 5218 CPU@2.3 GHz \times 64 |
| GPU | GeForce RTX2080 256-Bit HDMI/DP/DVI 8GB GDDR6 |
| GPU acceleration library | Tensorflow-gpu 2.0, CUDA 10.2, CUDNN 8.0 |

To compare different network models including Resnet_segnet, Mobilenet_segnet, Sandglass_segnet, Mobilenet_MFN, Sandglass_MFN, the batch size, learning rate, iteration number, and initial weight of the network models were all set the same. The batch size was set to four. The Adam algorithm of driving quantity was adopted to optimize the gradient, and the learning rate was set to 0.0001. The maximum number of iterations was set to 5000. The loss function was the cross-entropy, and it was defined as:

$$L = -\frac{1}{c} \sum_{i=1}^c [y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i)] \quad (3)$$

where L denotes the loss value and y_i , \hat{y}_i denoted the label of banana stalk and the model output respectively. The parameter c is the number of the pixels which is 576×576 . When $y_i = 0$, it stands for the background and when $y_i = 1$, it stands for the banana stalk.

At the same time, the method of learning rate decline was set. If the accuracy rate does not decrease for three times, the learning rate will be reduced by a reduction factor of 0.5 and the training will continue. When the loss value does not decrease for 10 consecutive times, it means that the basic training of the model is completed, the training is stopped.

The memory occupied by the network and the length of training time were evaluated for different networks. In the training process, the memory occupied by Resnet_segnet, Mobilenet_segnet, Sandglass_segnet, Mobilenet_MFN and Sandglass_MFN were 5.4 G, 4.9 G, 5.1 G, 3.8 G, 3.9 G. About the training time, the longest training time of Resnet_segnet was 13.7 h, and the shortest training time of Mobilenet_MFN was 5.8 h. In addition, MobileNet_Segnet, Sandglass_Segnet, and Sandglass_MFN were 10.1, 11.2, and 6.4 h, respectively. It can be seen that both Mobilenet_MFN and Sandglass_MFN have significantly improved in terms of resource occupancy and training time after lightweight treatment.

4. Experiment and Results

4.1. Performance Indices

The data set described in Section 2.1 was used for the evaluation of image segmentation results obtained by the proposed network model. The precision, recall, comprehensive evaluation index (F1) and accuracy were used as the evaluation metrics, and they were respectively defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

$$\text{Accuracy} = \frac{T}{R} \quad (7)$$

where TP denoted the number of properly segmented pixels, FP denoted the number of wrongly segmented pixels, and FN represented the number of unsegmented pixels of the stalk area. T were the intersection pixel of the banana stalk segmentation region and the actual label region, and R were the union pixel of the partition region and the actual label region of the banana stalk.

The framerate denoted the speed at which a camera captures an image and feeds it to the network for segmentation. In addition, execution time was the average time required for the network to segment ten images at one time.

$$\text{Framerate} = (\text{Framerate} + \frac{1}{t_0 - t_s})/2 \quad (8)$$

$$\text{Execution_time} = \frac{\sum_{j=1}^i [t_{end}(j) - t_{start}(j)]}{i} \quad (9)$$

where the initial value of Framerate is 0, t_0 was the initial time when the image was passed in and t_s was the ending time when the image segmentation completed. Also t_{start} was the start time of image segmentation and t_{end} was the time at the end of one image segmentation completed. i was the total number of image segmentation. In addition, j stands for the image number.

4.2. Results and Analysis

The trained deep learning models were deployed and ran on a GPU1660 graphics card, and the framerate, number of parameters and image segmentation time of each model

were counted. Meanwhile, the accuracy rate, precision rate, recall rate, and F1 value of the banana stalk segmentation results was calculated.

(1) F1 and recall

RSN, MSN, SSN, and MMFN denote Resnet_Segnet, Mobilenet_Segnet, Sandglass_Segnet, and Mobilenet_MFN, respectively, as comparison methods. Meanwhile, SGMFN denotes our proposals called Sandglass_MFN. The comparison results of recall rate and the value of F1 of different network models are presented in Table 3, where it can be seen that each network had a good evaluation value in recall rate and comprehensive evaluation index F1 except the light-weight Mobilenet_MFN (MMFN). To some extent, it shows that the lightweight network MMFN cannot give full play to its performance.

Table 3. Comparison results of recall rate and F1 of different network models.

| Model | Recall Rate | F1 |
|--------|-------------|-------|
| RSN | 99.06 | 99.33 |
| MSN | 98.01 | 98.75 |
| SSN | 99.07 | 99.30 |
| MMFN | 9.57 | 14.62 |
| SGMMFN | 99.08 | 99.32 |

(2) Precision

The segmentation precision of different deep learning network models is shown in Figure 9. It shows that the high-capacity RSN network achieved the best segmentation effect on the banana stalk, while the light-weight MMFN had the worst effect, and other network segmentation effects did not differ significantly of precision. It can be seen that with the deepening of the deep network layer, the model can achieve a certain effect in detection and segmentation. However, to achieve the goal of being lightweight by cutting the number of network layers pays more attention to the structure of the network, and not arbitrary cutting can achieve the effective segmentation of the target.

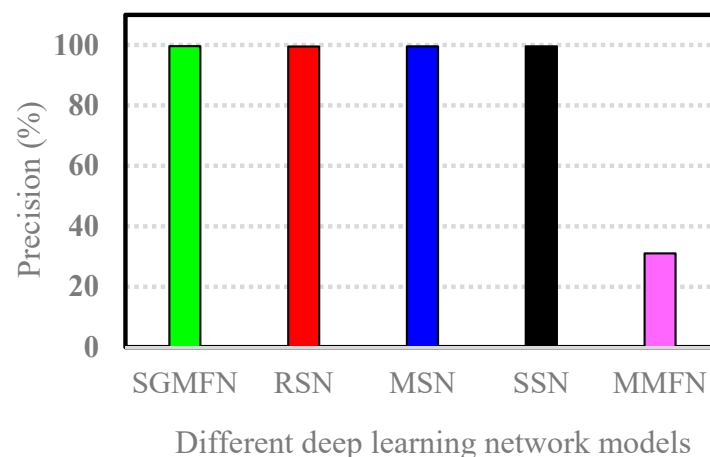


Figure 9. Comparison results of precision of different network models.

(3) Accuracy

In addition to the above three indicators, the accuracy also was used to evaluate the performance. It is shown in Figure 10. It is obvious that when the segmentation effect is good; accuracy and precision have similar performance. It can be seen from the four networks as regards SGMFN, RSN, MSN, SSN. In addition, as regards MMFN, after the lightweight processing, the segmentation effect is poor, reflected in the accuracy of only 18%.

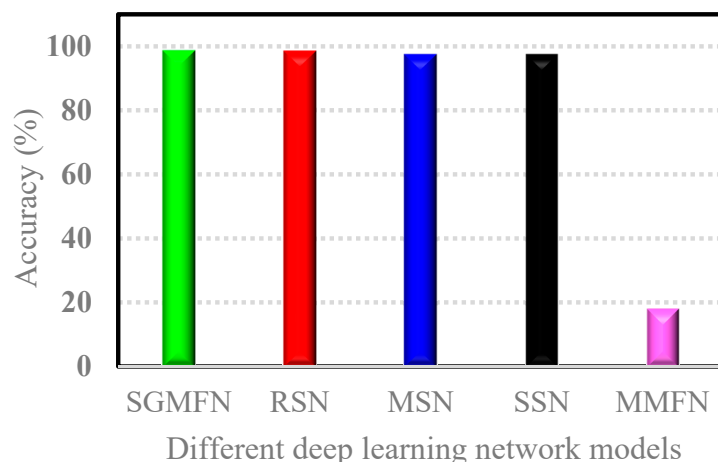


Figure 10. Comparison results of accuracy of different network models.

(4) The number of network model parameters

The number of parameters is an important index to realize network lightweight and different deep learning network models are shown in Figure 11. According to the figure, the number of parameters of the MMFN and SGMFN decreased significantly, realizing the effect of network parameter lightening. Compared with network structure of MMFN and SGMFN, depth separable convolution was used to reduce the number of parameters. Also, by appropriately trimming the number of network layers, both of them achieve model lightweight. However, because of the different features extracted by the encoding structure, the decoding network plays a different role.

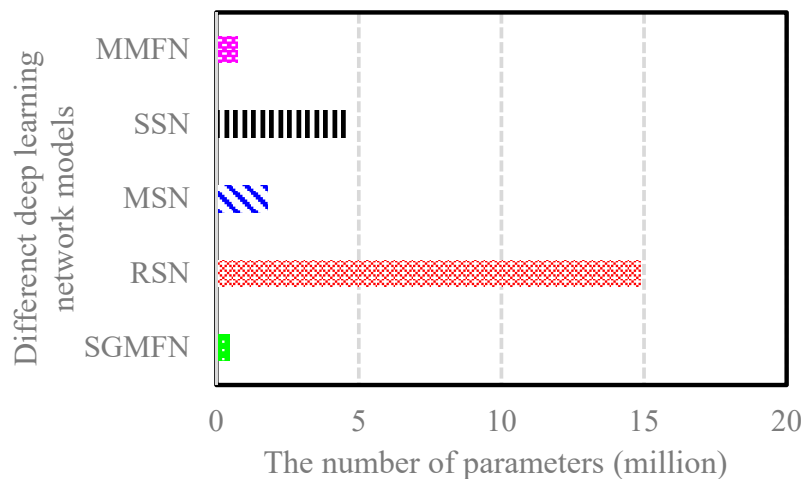


Figure 11. Comparison results of parameters of different network models.

(5) Framerate and average execution time

Based on Figures 12 and 13, It is obvious that the network SGMFN run the fastest frame rate. Also, the average processing speed of the SGMFN is the fastest per 10 images, with the minimum time consumption. Similarly, the realization of lightweight network MMFN also has a faster running speed, the image segmentation time is shorter. In addition, other networks run slower, especially RSN.

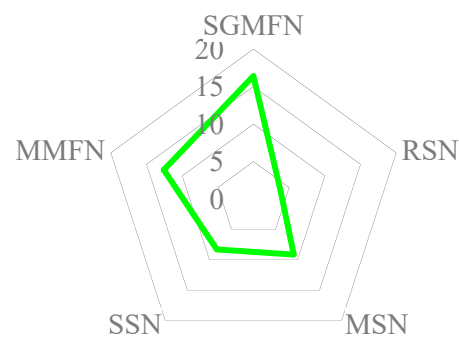


Figure 12. Comparison results of framerate of different network models.

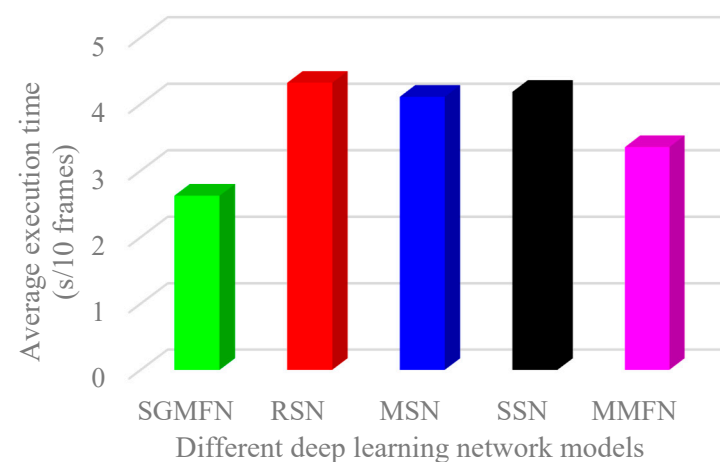


Figure 13. Comparison results of average execution time of different network models.

Based on the results presented in all figures, the high-capacity RSN network achieved the best segmentation effect on the banana stalk. Meanwhile, due to the large network model and a large number of network parameters, the running frame rate of this model was slow, and the average execution time was longer than the other. The main reason is that the bottleneck structure is adopted in the network, the number of channels is compressed between the bottlenecks, and the deep convolution is used to learn the features. However, deep convolution can better learn the stalk features, it is not conducive to network lightweight. The MSN and SSN networks' capacities were almost of the same level, and the difference in each evaluation index was also almost close. Using the depth separable convolution and decreasing the number of network layers to achieve the purpose of network lightweight. The Mobilenet and Sandglass structures were considered to be the encoding network, without cutting the characteristic layers, the network could better extract stalk features and achieve segmentation.

Also, in Figures 9–13, it can be seen that the MMFN network was appropriately trimming the number of network layers, which greatly reduced the recognition and segmentation ability of this network. The main reason for this was that the shortcut keys in the Mobilenet network were connected between the low-dimensional tensors, which increased the probability of feature loss, and the reduced number of network layers made the learned stalk features inadequate, so the multi-feature fusion mechanism of the decoding network was difficult to perform.

The network proposed for banana stalk segmentation in this paper uses the Sandglass structure, the channel used the point convolution to extract the channel information, and the shortcut key was connected to the high dimension to learn the stalk features more fully. Therefore, an appropriate reduction in the number network layers and the addition of the multi-feature fusion mechanism to the decoding network could provide a better segmentation effect to a certain extent.

(6) Image Segmentation effects

The segmentation effects of the SGMFN are shown in Figure 14. Compared to other models, the segmentation effects of different models on the banana stalk under different external conditions are shown in Figure 15, where it can be seen that the RSN and SGMFN relatively completed segmentation of the banana stalk, and MSN and SSN could also segment the banana stalk accurately, but only the subrange of the banana stalk. However, the MMFN network with a light weight had a poor segmentation effect, shown obvious that the number of wrongly segmented image was large and that some images could not be segmented. Comparing all the results, it can be discovered that using the sandglass structure in the encoding network and combining with the multi-feature fusion structure of the decoding network, the SGMFN model realized the lightweight without reducing the segmentation accuracy, and sped up the running frame rate and image segmentation average execution time.

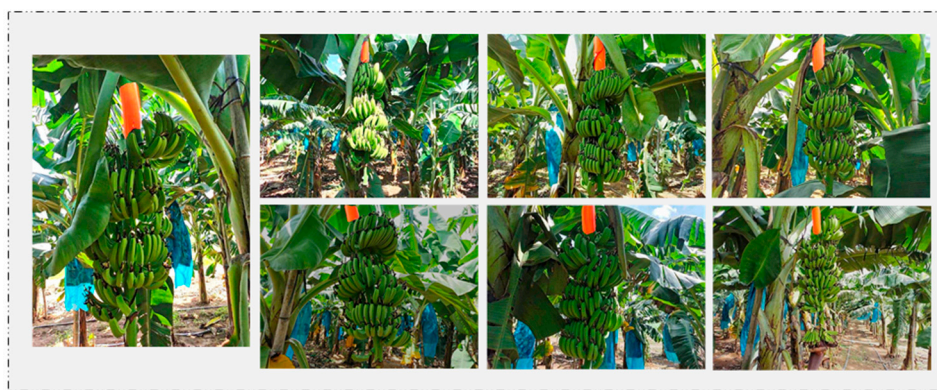


Figure 14. Segmentation effects of Sandglass_MFN network model.

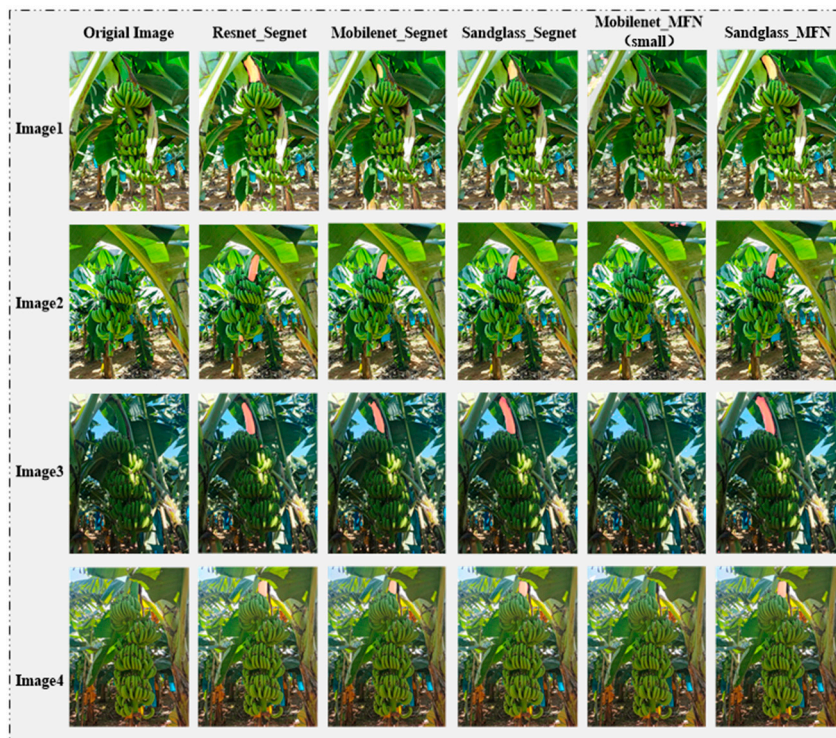


Figure 15. Segmentation effects of all the network models.

5. Conclusions

The segmentation of banana stalk in natural environment is of great significance to the picking robot. In this paper, a lightweight network model based on a sandglass structure and multi-feature fusion is proposed for banana stalk segmentation. Depth separable convolution is used and the number of network layers is appropriately clipped. At the same time, the dilated convolution with different expansion rates is used to extract the features and perform fusion to achieve network lightweight with high segmentation accuracy and execution speed. According to the experimental results, the following conclusions can be drawn:

- (1) The characteristics of the residual structure, reverse residual structure and sandglass results were analyzed, and it was found that the reverse residual and sandglass structures results are suitable for a lightweight network, but after a reduction in the network layer number, the deep neural network using reverse residual structure has reduced performance in feature extraction.
- (2) Adding the multi-feature fusion mechanism to the decoder network can make the features extracted by the encoding network be more fully integrated, learn the banana stalk features with high-level semantic segmentation ability, and effectively improve the segmentation ability of the network model in recognition of a banana stalk.
- (3) The proposed network model is verified by the experiment with the banana stalk images under different environment interference, and the banana stalk can be better segmented. In addition, on the premise of having no reduction in the accuracy and recall rate, the number of model parameters is effectively reduced and the operating efficiency of the proposed network model is improved, which is helpful for porting the model to mobile devices. Therefore, the proposed lightweight multi-feature fusion network model cannot only quickly identify and segment the banana stalk, but also be more easily deployed in the edge equipment.

In the future research, the proposed deep neural network will be deployed on an edge computer and put on the robot to pick bananas in a complex orchard environment. Also, it will be applied to other kinds of fruit recognition and segmentation experiments for more agriculture application.

Author Contributions: Conceptualization, L.Z. and X.L.; methodology, R.Z.; software, S.Z.; validation, S.Z.; writing—original draft preparation, T.C. All authors have read and agreed to the published version of the manuscript.

Funding: The paper is supported by the Key Area Research and Development Program in Guangdong Province of China under Grant 2019B020223003, the Natural Science Foundation of Guangdong Province of China under Grant 2019A1515011346, the Guangzhou Science and Technology Program under Grant 201804010480 and 202002030092 and the Modern Agricultural Industry Technology System Innovation Team Program in Guangdong Province of China under Grant 2019KJ139.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the privacy policy of the organization.

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

References

1. Panigrahi, N.; Thompson, A.J.; Zobelzu, S.; Knox, J.W. Identifying opportunities to improve management of water stress in banana production. *Sci. Hortic.* **2021**, *276*, 109735. [[CrossRef](#)]
2. Fida, R.; Pramafisi, G.; Cahyana, Y. Application of banana starch and banana flour in various food product: A review. *IOP Conf. Ser. Earth Environ. Sci.* **2020**, *443*, 012057. [[CrossRef](#)]
3. Khayer, A.; Eti, F.S.; Istiaq, A.; Hasan, M.M. Bio Fertilizer on Rachis: A New Method Facilitates Higher Banana (*Musa sapientum*) Production. *Preprints* **2019**. [[CrossRef](#)]
4. Hui, Y.; Liu, H.; Zhang, H.; Wu, Y.; Li, Y.; Li, X.; Wang, D. Application status and development trend of agricultural robot. In Proceedings of the 2018 ASABE Annual International Meeting, Detroit, MI, USA, 29 July–1 August 2018.
5. Mazurkiewicz, J.; Marczuk, A.; Pochwatka, P.; Kujawa, S. Maize straw as a valuable energetic material for biogas plant feeding. *Materials* **2019**, *12*, 3848. [[CrossRef](#)] [[PubMed](#)]

6. Kujawa, S.; Mazurkiewicz, J.; Czekala, W. Using convolutional neural networks to classify the maturity of compost based on sewage sludge and rapeseed straw. *J. Clean. Prod.* **2020**, *258*, 120814. [[CrossRef](#)]
7. Kujawa, S.; Janczak, D.; Mazur, A. Image Analysis of Sewage Sludge and Barley Straw as Biological Materials Composted under Different Conditions. *Materials* **2019**, *12*, 3644. [[CrossRef](#)]
8. Zhang, Q.; Gao, G. Prioritizing robotic grasping of stacked fruit clusters based on stalk location in RGB-D images. *Comput. Electron. Agric.* **2020**, *172*, 105359. [[CrossRef](#)]
9. Lv, J.; Wang, Y.; Xu, L.; Gu, Y.; Zou, L.; Yang, B.; Ma, Z. A method to obtain the near-large fruit from apple image in orchard for single-arm apple harvesting robot. *Sci. Hort.* **2019**, *257*, 108758. [[CrossRef](#)]
10. Wei, X.; Jia, K.; Lan, J.; Li, Y.; Zeng, Y.; Wang, C. Automatic method of fruit object extraction under complex agricultural background for vision system of fruit picking robot. *Optik* **2014**, *125*, 5684–5689. [[CrossRef](#)]
11. Zhuang, J.; Luo, S.; Hou, C.; Tang, Y.; He, Y.; Xue, X. Detection of orchard citrus fruits using a monocular machine vision-based method for automatic fruit picking applications. *Comput. Electron. Agric.* **2018**, *152*, 64–73. [[CrossRef](#)]
12. Tao, Y.; Zhou, J. Automatic apple recognition based on the fusion of color and 3D feature for robotic fruit picking. *Comput. Electron. Agric.* **2017**, *142*, 388–396. [[CrossRef](#)]
13. Xiong, J.; Lin, R.; Liu, Z.; He, Z.; Yang, Z.G.; Bu, R. Visual technology of picking robot to detect litchi at nighttime under natural environment. *Trans. Chin. Soc. Agric. Mach.* **2017**, *48*, 28–33.
14. Xu, S.; Lu, K.; Pan, L. 3D reconstruction of rape branch and pod recognition based on RGB-D camera. *Trans. Chin. Soc. Agric. Mach.* **2019**, *50*, 21–27.
15. Raghavendra, A.; Guru, D.; Rao, M.K.; Sumithra, R. Hierarchical approach for ripeness grading of mangoes. *Artif. Intell. Agric.* **2020**, *4*, 243–252. [[CrossRef](#)]
16. Raghavendra, A.; Guru, D.S.; Rao, M.K. An Automatic Predictive Model for Sorting of Artificially and Naturally Ripened Mangoes. In *Advances in Intelligent Systems and Computing*; Springer International Publishing: New York, NY, USA, 2020; pp. 633–646.
17. Hu, M.-H.; Dong, Q.-L.; Liu, B.-L.; Malakar, P.K. The Potential of Double K-Means Clustering for Banana Image Segmentation. *J. Food Process. Eng.* **2014**, *37*, 10–18. [[CrossRef](#)]
18. Prabha, D.S.; Kumar, J.S. Assessment of banana fruit maturity by image processing technique. *J. Food Sci. Technol.* **2013**, *52*, 1316–1327. [[CrossRef](#)] [[PubMed](#)]
19. Fu, L.; Duan, J.; Zou, X.; Lin, G.; Song, S.; Ji, B.; Yang, Z. Banana detection based on color and texture features in the natural environment. *Comput. Electron. Agric.* **2019**, *167*, 105057. [[CrossRef](#)]
20. Hu, M.-H.; Dong, Q.-L.; Liu, B.-L.; Pan, L.-Q.; Walshaw, J. Image Segmentation of Bananas in a Crate Using a Multiple Threshold Method. *J. Food Process. Eng.* **2016**, *39*, 427–432. [[CrossRef](#)]
21. Gao, B.; Li, X.; Woo, W.L.; Tian, G.Y. Physics-Based Image Segmentation Using First Order Statistical Properties and Genetic Algorithm for Inductive Thermography Imaging. *IEEE Trans. Image Process.* **2017**, *27*, 2160–2175. [[CrossRef](#)] [[PubMed](#)]
22. Hu, B.; Gao, B.; Woo, W.L.; Ruan, L.; Jin, J.; Yang, Y.; Yu, Y. A Lightweight Spatial and Temporal Multi-Feature Fusion Network for Defect Detection. *IEEE Trans. Image Process.* **2020**, *30*, 472–486. [[CrossRef](#)]
23. Sulisty, S.B.; Woo, W.L.; Dlay, S.S. Regularized Neural Networks Fusion and Genetic Algorithm Based On-Field Nitrogen Status Estimation of Wheat Plants. *IEEE Trans. Ind. Inform.* **2016**, *13*, 103–114. [[CrossRef](#)]
24. Majeed, Y.; Zhang, J.; Zhang, X.; Fu, L.; Karkee, M.; Zhang, Q.; Whiting, M.D. Deep learning based segmentation for automated training of apple trees on trellis wires. *Comput. Electron. Agric.* **2020**, *170*, 105277. [[CrossRef](#)]
25. Wan, S.; Goudos, S. Faster R-CNN for multi-class fruit detection using a robotic vision system. *Comput. Netw.* **2020**, *168*, 107036. [[CrossRef](#)]
26. Yu, Y.; Zhang, K.; Yang, L.; Zhang, D. Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. *Comput. Electron. Agric.* **2019**, *163*, 104846. [[CrossRef](#)]
27. Jia, W.; Tian, Y.; Luo, R.; Zhang, Z.; Lian, J.; Zheng, Y. Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot. *Comput. Electron. Agric.* **2020**, *172*, 105380. [[CrossRef](#)]
28. Kang, H.; Chen, C. Fast implementation of real-time fruit detection in apple orchards using deep learning. *Comput. Electron. Agric.* **2020**, *168*, 105108. [[CrossRef](#)]
29. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors* **2017**, *17*, 2022. [[CrossRef](#)] [[PubMed](#)]
30. Picon, A.; Alvarez-Gila, A.; Seitz, M.; Ortiz-Barredo, A.; Echazarra, J.; Johannes, A. Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild. *Comput. Electron. Agric.* **2019**, *161*, 280–290. [[CrossRef](#)]
31. Zhang, S.; Wu, Y.; Men, C.; Li, X. Tiny YOLO Optimization Oriented Bus Passenger Object Detection. *Chin. J. Electron.* **2020**, *29*, 132–138. [[CrossRef](#)]
32. Khokhlov, I.; Davydenko, E.; Osokin, I.; Ryakin, I.; Babaev, A.; Litvinenko, V.; Gorbachev, R. Tiny-YOLO object detection supplemented with geometrical data. In Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 25–28 May 2020.
33. Mehri, A.; Ardakani, P.B.; Sappa, A.D. MPRNet: Multi-Path Residual Network for Lightweight Image Super Resolution. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Manchester, UK, 11 September 2020; pp. 2704–2713.

34. Fooladgar, F.; Kasaei, S. Lightweight residual densely connected convolutional neural network. *Multimed. Tools Appl.* **2020**, *79*, 25571–25588. [[CrossRef](#)]
35. Kong, L.; Yang, J. Fdflownet: Fast Optical Flow Estimation Using A Deep Lightweight Network. In Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 25–28 October 2020; pp. 1501–1505.
36. Liu, J.; Tang, J.; Wu, G. Residual Feature Distillation Network for Lightweight Image Super-Resolution. *arXiv* **2020**, arXiv:2009.11551.
37. Jia, S.; Lin, Z.; Xu, M.; Huang, Q.; Zhou, J.; Jia, X.; Li, Q. A Lightweight Convolutional Neural Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, 1–14. [[CrossRef](#)]