Method for detecting 2D grapevine winter pruning location based on thinning algorithm and Lightweight Convolutional Neural Network

Qinghua Yang, Yuhao Yuan, Yiqin Chen, Yi Xun*

(*Key Laboratory of Special Purpose Equipment and Advanced Processing Technology, Ministry of Education & Zhejiang Province, College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou 310023, China*)

Abstract: In viticulture, there is an increasing demand for automatic winter grapevine pruning devices, for which detection of pruning location in vineyard images is a necessary task, susceptible to being automated through the use of computer vision methods. In this study, a novel 2D grapevine winter pruning location detection method was proposed for automatic winter pruning with a Y-shaped cultivation system. The method can be divided into the following four steps. First, the vineyard image was segmented by the threshold two times Red minus Green minus Blue (2R−G−B) channel and S channel; Second, extract the grapevine skeleton by Improved Enhanced Parallel Thinning Algorithm (IEPTA); Third, find the structure of each grapevine by judging the angle and distance relationship between branches; Fourth, obtain the bounding boxes from these grapevines, then pre-trained MobileNetV3_small×0.75 was utilized to classify each bounding box and finally find the pruning location. According to the detection experiment result, the method of this study achieved a precision of 98.8% and a recall of 92.3% for bud detection, an accuracy of 83.4% for pruning location detection, and a total time of 0.423 s. Therefore, the results indicated that the proposed 2D pruning location detection method had decent robustness as well as high precision that could guide automatic devices to winter prune efficiently.

Keywords: grapevine winter pruning, Lightweight Convolutional Neural Network, thinning algorithm, detection method
DOI: 10.25165/j.ijabe.20221503.6750

Citation: Yang Q H, Yuan Y H, Chen Y Q, Xun Y. Method for detecting 2D grapevine winter pruning location based on thinning algorithm and Lightweight Convolutional Neural Network. Int J Agric & Biol Eng, 2022; 15(3): 177–183.

1 Introduction

Winter grapevine pruning is a significant step in viticulture that can preserve high-quality buds, improve light utilization, and ultimately increase the yield of grapes in the coming year. In recent years, automation has been a major trend in agricultural development. Many automated agricultural devices have been developed[1], while grape pruning is still mainly done manually. Therefore, it is of great importance to develop automation and intelligent grapevine pruning devices.

To achieve automatic pruning, machine vision technology is an inevitable part. There have been many machine vision methods already proposed for viticulture, including grape bunch detection[2], finding the structure of vine[3], grapevine yield, and leaf area estimation[4].

In the grapevine pruning field, Xu et al.[5] presented a bud detection method based on the Rosenfeld thinning algorithm[6] and Harris algorithm[7]. The color threshold method was used to convert the RGB images captured indoors into binary images and then Rosenfeld thinning algorithm was applied to extract the grapevine’s skeleton. Because of the similarity between the buds and the corners, they applied the Harris algorithm to detect buds from the skeleton image. The recognition rate of their method reached 70.2%. The limitations of this work laid in that the images must be captured indoors and the accuracy of bud detection by Harris algorithm was insufficient. Pérez et al.[8] detected the grapevine buds using Scale-Invariant Feature Transform[9] for calculating low-level features, Bag of Features[10] for building an image descriptor, and Support Vector Machines[11] for training a classifier. To be more specific, the images were captured in natural field conditions without artificial background. They reported a recall higher than 0.9 and a precision of 0.86 when the sorting images containing at least 60% of a bud and scaled up to window patches containing a proportion of 20%-80% of bud versus background pixels. Based on the method of Pérez et al.[8], Díaz et al.[12] outlined an approach for the localization of buds in 3D space.

Deep learning is the state-of-art method for image tasks in viticulture, lots researchers have applied deep learning to improve the performance of algorithms. Zabawa et al.[13] took advantage of a convolutional neural network to detect single berries in images by performing a semantic segmentation. Palacios et al.[14] proposed a non-invasive method combining deep learning and machine vision technology to count grapevine flowers by on-the-go image acquisition. Cruz et al.[15] used convolutional neural networks to detect grapevine yellow symptoms end-to-end.

Deep learning was also applied in the latest literature on grapevine pruning. Marset et al.[16] presented a method for grapevine bud detection based on Fully Convolutional Networks MobileNet architecture (FCN-MN). This was a semantic segmentation network that had the capacity of segmenting the full shape of the buds from the grapevine images as the detection precision and recall of this approach were 95.6% and 93.6% for buds nearby and prominent, respectively.

The above studies represented a great advance in relation to the problem of detecting grapevine buds, nevertheless, limitations did exist to meet the needs of automatic grapevine pruning in winter. First, they had not got the order of the buds on the grapevine while it was essential to determine the pruning location. Second, the

Received date: 2021-05-09 **Accepted date:** 2021-09-15

Biographies: Qinghua Yang, PhD, Professor, research interest: agriculture robot, Email: robot@zjut.edu.cn; **Yuhao Yuan**, Master candidate, research interest: machine vision, Email: 2111902105@zjut.edu.cn; **Yiqin Chen**, Master candidate, research interest: machine vision, Email: chen_yiqin@zjut.edu.cn.

***Corresponding author: Yi Xun**, PhD, Lecturer, research interest: machine vision. Zhejiang University of Technology, Hangzhou 310023, China. Tel: +86-13064728616, Email: xunyi@zjut.edu.cn.

specific shape of the buds was not necessary for the automatic grape pruning since the pruning location was always on the branch between the buds. Third, none of them combined their algorithms with the grapevine cultivation system while a considerable cultivation system can effectively reduce the background complexity and improve the accuracy of the vision algorithm. Combining cultivation systems with mechanical automation is an inevitable trend for the development of agricultural automation.

In this study, a novel 2D grapevine pruning location detection method was proposed for automatic grapevine pruning in winter with a Y-shaped cultivation system, which prepared for directly establishing the 3D model of the pruning location in the future work. And there were four key steps in this study including image acquisition and segmentation, extracting the grapevine skeleton, finding the structure of each grapevine, and finding the pruning location by classification with Lightweight Convolutional Neural Network.

2 Materials and methods

2.1 Image acquisition

The images were captured in the Mingcheng Vineyard in Hangzhou, Zhejiang Province, China. The Mingcheng Vineyard adopted a Y-shaped cultivation system, in which the fruiting branches of the grapevine extended to both sides in a Y shape, and the ipsilateral fruiting branches were roughly on the same plane, as shown in Figure 1. This Y-shaped cultivation system can effectively avoid most of the interference when taking photos at the elevation angle.

A mobile phone with a Sony IMX586 webcam was utilized to take these grapevine images, which also showed that the method we proposed did not have high requirements for the camera. The images were captured between 1:00 p.m. and 4:00 p.m. between December 16, 2020, and January 6, 2021, when the leaves were completely withered and had fallen, but before the plants began to sprout again.

When shooting images for finding the location of pruning, it was required that the main grape branches in the images contained at least the number of buds that need to be retained during pruning. In order to obtain a better shooting effect, the shooting plane of the camera parallel was often set to the plane where the branches were located.

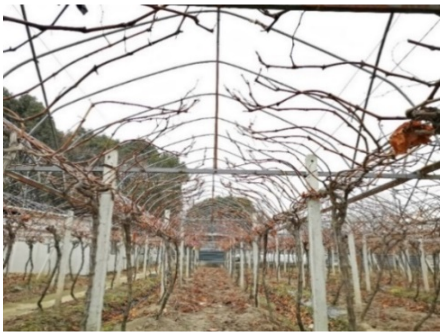


Figure 1 Mingcheng vineyard

2.2 Image segmentation

Color information is the most significant feature of images. Since the vineyard where the images were captured applied the Y-shaped cultivation system, the colors of the foreground and the background in captured images were clearly distinguished, which was very suitable for the color-based threshold segmentation.

There were mainly three objects in captured images, including grapevine branches, greenhouse brackets, and background, as shown in Figure 2. RGB and HSV color spaces were utilized to segment the grapevine branches, while (2R−G−B) was also added to the analysis since the segmented foreground was biased towards reddish-brown in color.

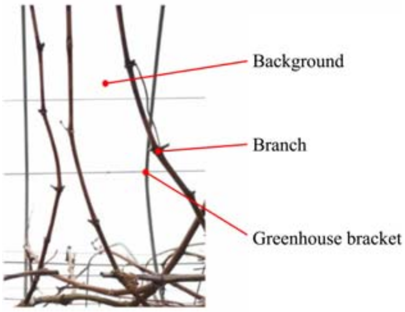
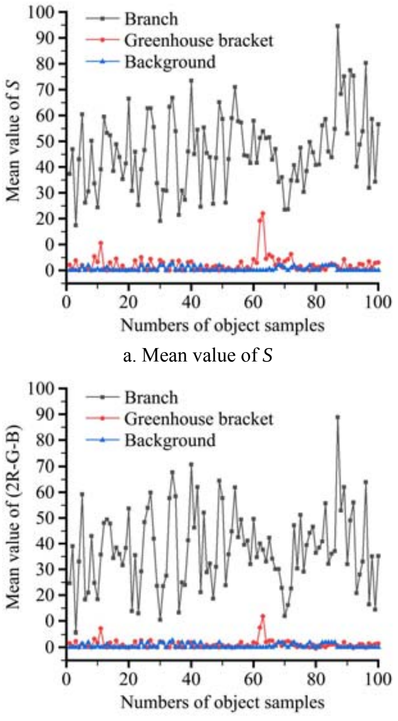


Figure 2 Objects of vineyard image

100 areas in captured images were sampled for each type of object to analyze the color distinction in R, G, B, H, S, V, (2R−G−B) color channels and it was finally found that there was a significant difference between the foreground and the background in S and (2R−G−B) channels, as showed in Figure 3. Based on the above results, S and (2R−G−B) channels were finally chosen as the basis for threshold segmentation.



a. Mean value of *S*

b. Mean value of (2R-G-B)

Figure 3 Mean value of *S* and (2R-G-B) of image samples

2.3 Extracting of the grapevine skeleton

To process the grapevine information and find the buds more efficiently, the thinning algorithm was applied to extract the skeleton of the grapevine. The thinning algorithms widely used include ZHANG-SUEN thinning algorithm (ZS)[17], Hilditch thinning algorithm[18], Lu-Wang thinning algorithm (LW)[19,20], and Rosenfeld thinning algorithm[6]. In addition to these classic thinning algorithms, the Improved Enhanced Parallel Thinning Algorithm (IEPTA) proposed by Zhao et al.[21] was also tested in the grapevine images.

To evaluate these thinning algorithms, 110 grapevine images with a pixel size of 125×188 were utilized to calculate the average single-pixel ratio and average thinning time of each thinning algorithm, as listed in Table 1. The single-pixel ratio refers to the proportion of lines with a single-pixel width in the thinned images, as stated as follows:

$$S_{\text{pixel}} = \left(1 - \frac{n_r}{n}\right) \times 100\% \qquad (1)$$

where, S_{pixel} is the single-pixel ratio, %; n_r is the number of redundant pixels that are non-endpoint pixels and do not affect the original connectivity of the thinned image after removal; n is the total number of pixels.

Table 1 Evaluation of thinning algorithms

Item	Rosenfeld	ZS	Hilditch	LW	IEPTA
Average single-pixel ratio/%	100.00	91.69	90.45	91.28	100.00
Average time/s	0.521	0.266	0.530	0.237	0.328

Note: ZS: ZHANG-SUEN thinning algorithm; LW: Lu-Wang thinning algorithm; IEPTA: Improved Enhanced Parallel Thinning Algorithm.

To make the subsequent steps more convenient and efficient, the thinning algorithm we used must have a high single-pixel ratio. Both the Rosenfeld algorithm and the IEPTA achieve a single-pixel ratio of 100%, while IEPTA was finally adopted since its less average processing time.

In addition, it was found that the wider the object to be thinned, the more time it took. In the Y-shape cultivation system, the grapevine growing horizontally at the bottom of the images was not the part needed to extract the skeleton, but its thinning wasted a long time because it was often the widest in the images. Therefore, these horizontal branches were removed before thinning to improve the thinning speed. The average thinning time of IEPTA was shortened from 0.328 s to 0.165 s after optimization, a reduction of 49.695%.

2.4 Finding the structure of each grapevine

According to the principle of grapevine pruning in winter, a certain number of buds on a grapevine should be kept, which requires counting the buds on each grapevine from the bottom up to determine the pruning location. So it was necessary to find the structure of each grapevine that needed to be pruned from the skeleton images.

Eliminate the intersections to convert lines in the image into branch vectors and then filter out too short or extending horizontally branches. The length thresh was based on the average width of branches.

These branches were reconnected into complete grapevines by judging their relative location relationship and angle relationship.

In the process of reconnecting the complete grapevines, some branches would also be filtered out because they did not meet the requirements.

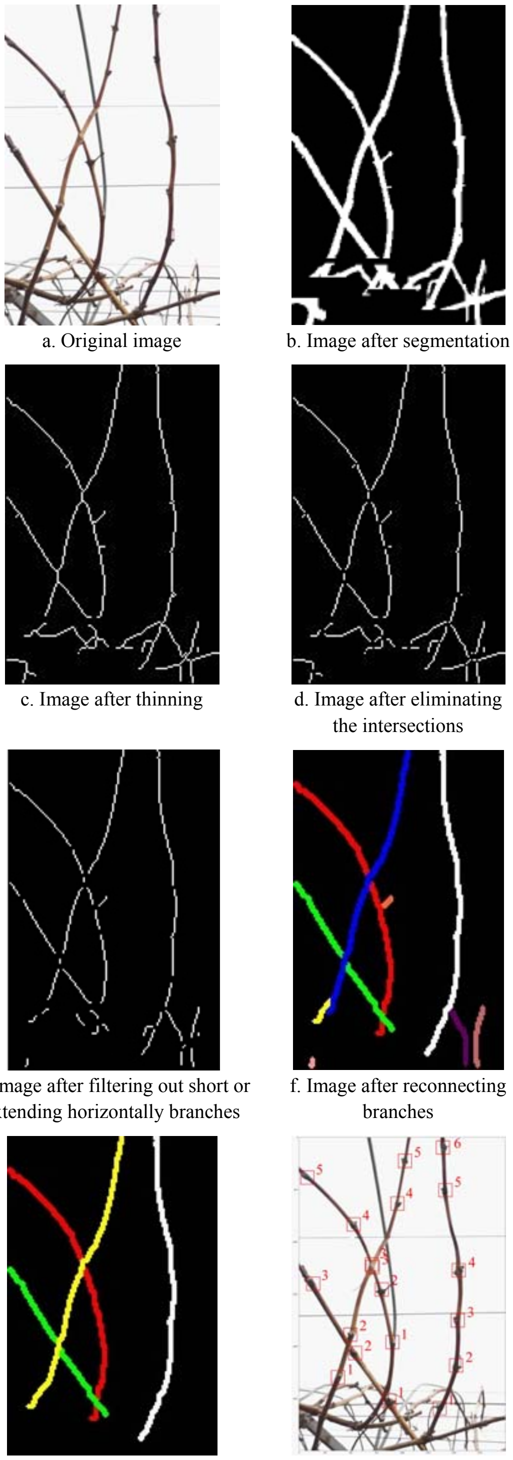
After the above steps, the structure of each grapevine was found successfully from the skeleton images. The whole process is shown in Figure 4.

2.5 Finding the pruning location by classification

2.5.1 Obtaining bounding boxes

The bounding boxes were obtained based on the grapevine structures found in Section 2.4. The size of bounding boxes was determined based on the average width of the branches in the image, but it needs to be multiples of 32 to facilitate mapping to the feature map. To find the buds effectively, two bounding boxes with different side lengths of 128 pixels and 192 pixels were used.

In order to reduce the repeated calculation of the bounding boxes, the entire images were input into the convolutional layers. The bounding boxes' corresponding areas were obtained in the output feature maps, and then they were classified after Adaptive Average Pooling.



a. Original image

b. Image after segmentation

c. Image after thinning

d. Image after eliminating the intersections

e. Image after filtering out short or extending horizontally branches

f. Image after reconnecting branches

g. Image after filtering out short or high repetition grapevines

h. Image after classification

Figure 4 Whole process of the classification method used in this study

2.5.2 Classification with Lightweight Convolutional Neural Network

The classification was the most significant step in this study to find the bounding boxes which contained buds. The proposed method was for automatic pruning devices so the lightweight convolutional neural network which had a small number of parameters was more suitable for these devices, and was adopted in this study.

Lightweight convolutional neural networks commonly include MobileNetV2[22], MobileNetV3[23], ShuffleNetV2[24], EfficientNet[25], SqueezeNet[26], etc. In Section 3, the image dataset collected in the vineyard was utilized to test the effect of the above-mentioned neural networks.

MobileNetV3 is a state-of-art neural network algorithm for mobile tasks proposed by Howard et al.[23] Compared with MobileNetV2, MobileNetV3 can use fewer computing resources to obtain higher accuracy. MobileNetV3 has two versions, MobileNetV3_small and MobileNetV3_large, which are targeted for high and low resource use cases. According to the experimental results listed in Table 2, MobileNetV3_small with a width coefficient of 0.75 is the most appropriate decision in the application scenario of this study.

2.5.3 Non-Maximum Suppression

Due to the multi-scale dense sampling in the step of obtaining bounding boxes, a bud may be contained by multiple bounding boxes. So Non-Maximum Suppression (NMS)[27] was applied to search for the bounding box with the highest score from the surroundings of the same grapevine as the final result.

2.5.4 Find the pruning location

According to the staff of the Mingcheng Vineyard, 4-5 buds were retained on each grape branch when pruning in winter.

Since it was known that the highest probability of missed detection in the proposed method was the lowest bud, apply the principle of keeping 4 buds when detecting the pruning location so that even if there was one missed detection below, the result was still within the allowable error range.

The optimal pruning location was 2-4 cm upwards from the last bud along the branch. However, this study's method can only obtain the location of the last bud and its upwardly extending branch in the 2D image. The real coordinates of the pruning location still need to be obtained through the subsequent binocular vision method combined with the pruning location detection results of multiple images. The method proposed in this study helps to directly establish the 3D model of the branch at the pruning location, rather than finding the pruning location after establishing the 3D model of the entire grapevine.

3 Results and discussion

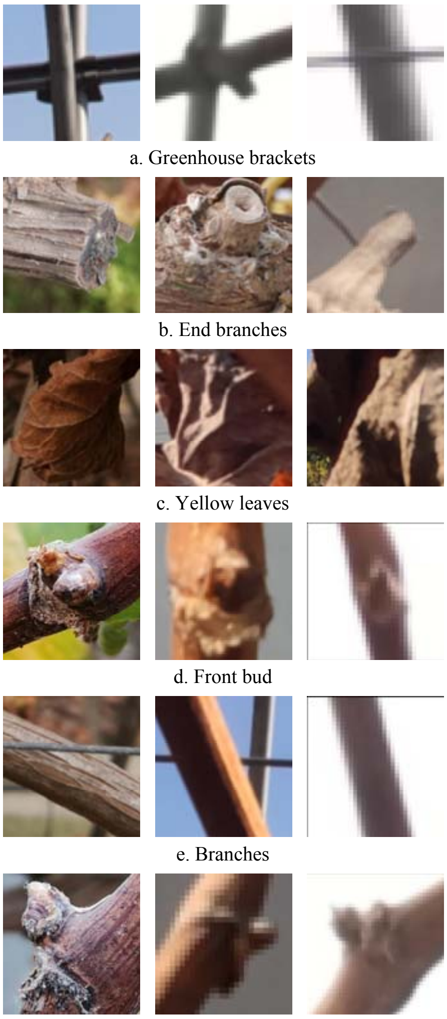
3.1 Experimental condition

Python 3.6, Open Source Computer Vision Library (OpenCV3.1.0, Intel Corporation) were used to realize the proposed detection algorithm on an Intel(R) Core(TM) i9-7900 CPU @ 3.30 GHz, 3.31 GHz, 32 GB RAM desktop with NVIDIA GeForce RTX 2080 Ti GPU. To speed up processing, the original images with a pixel size of 2000×3000 were compressed to 250×375 in segmentation and 125×188 when thinning. However, the inputs of classification were still the original images.

3.2 Lightweight neural networks test experiment

To find the model most suitable for the application of this study, a comparative test was presented on several lightweight convolutional neural networks, including MobileNetV2[22], MobileNetV3[23], ShuffleNetV2[24], EfficientNet[25], and SqueezeNet[26] with different widths.

In the vineyard, thanks to the Y-shaped cultivation system, there were only five types of objects that mainly contained: yellow leaves, branches, greenhouse brackets, end branches, and buds. Due to the obvious difference in the characteristics, the buds were divided into two types, the front buds, and the side buds. For these six types of objects, more than 5000 images captured in the vineyard were taken to train and evaluate the above-mentioned lightweight convolutional neural networks. To make the training results more robust, the taken images were a mixture of clear images and blurry images, as shown in Figure 5. These images and their labels were used during the training and evaluation of the classification models. For this purpose, the image set was separated into two disjoint subsets: the train set with 70% and the evaluation set with the remaining 30%.



a. Greenhouse brackets

b. End branches

c. Yellow leaves

d. Front bud

e. Branches

f. Side bud

Figure 5 Image sets of six different classes

Because of the small number of images in the train set, two techniques widely used in practice were employed to achieve robust training: transfer learning[28] and data augmentation[29]. Pre-trained models on ImageNet were applied to each model.

During training, the optimization was Adam and the initial learning rate was set to 0.001, which is adjusted to 0.0001 after 40 epochs. Using these parameters, the above-mentioned lightweight convolutional neural networks were trained over 100 epochs with a batch size of 12.

The six-class classification was applied first, and then the six-class result was converted into two-class results for the final comparison because we found that the accuracy in this way was a little higher than the direct two-class classification. The results are listed in Table 2. Params refer to the total number of network parameters, which represent the complexity of the networks.

According to the results, except for SqueezeNet_1.1, ShuffleNetV2×1.0, ShuffleNetV2×0.5, and MobileNetV2, the accuracies of the remaining models were all above 99.5%. Among them, the highest accuracy was EfficentNet_b2, whose accuracy was as high as 99.810%. Among these networks with an accuracy higher than 99.5%, the params of MobileNetV3_small× 0.75 was the lowest, even lower than a quarter of the params of

EfficentNet_b2. So MobileNetV3_small×0.75 with the best comprehensive performance was applied to the method of this study.

Table 2 Accuracy and params of Lightweight Neural Networks

Name	Six-class to Two-class accuracy/%	Params /million	Two-class accuracy/%	Params /M
MobileNetV2	99.241	2.232	99.115	2.228
MobileNetV3_small×1.0	99.621	1.665	99.305	1.512
MobileNetV3_small×0.75	99.621	1.023	99.431	1.019
SqueezeNet_1.1	98.293	0.726	98.102	0.724
EfficentNet_b0	99.684	4.015	99.241	4.010
EfficentNet_b1	99.726	6.521	99.431	6.516
EfficentNet_b2	99.810	7.709	99.558	7.704
MobileNetV3_large×1.0	99.684	4.210	99.362	4.205
MobileNetV3_large×0.75	99.621	2.720	99.241	2.715
ShuffleNetV2×1.0	99.121	1.575	98.989	1.571
ShuffleNetV2×0.5	98.609	0.402	98.293	0.398

3.3 Pruning location detection experiment

In this experiment, the quality of the proposed method was systematically evaluated in 110 test images.

In order to show its superiority, this algorithm was compared with three other existing algorithms on the same test images. To make the comparison more intuitive, all comparison algorithms complemented the step from the bud detection result to the pruning location.

1) Algorithm 1: the algorithm in this study.

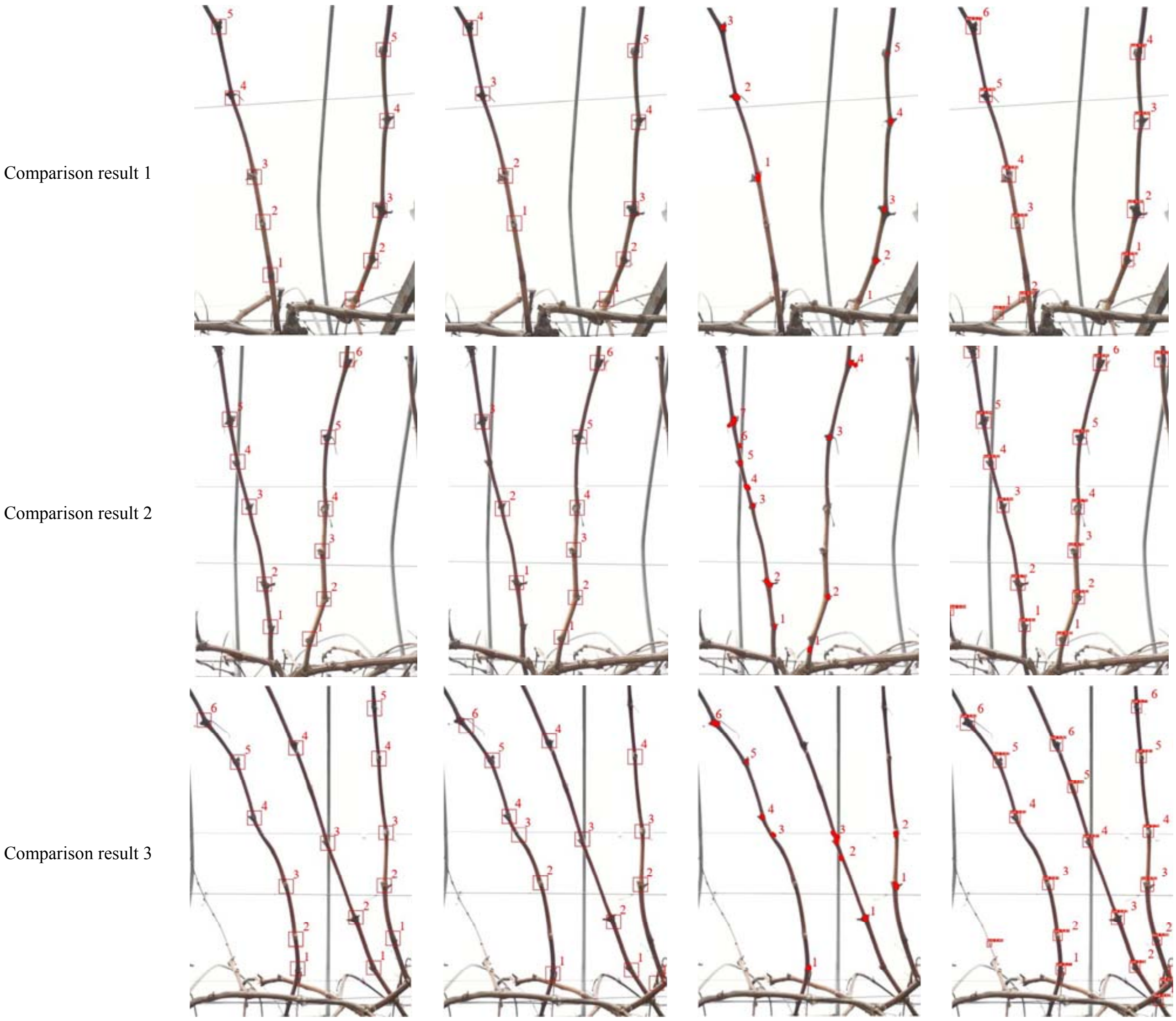
2) Algorithm 2: in which the classification used SVM as a classifier and Bag of Features to compute visual descriptors, otherwise, the rest was consistent with algorithm 1.

3) Algorithm 3: the algorithm using Rosenfeld algorithm for thinning and Harris algorithm to detect buds.

4) Algorithm 4: faster-RCNN-mobileNetV3_small×0.75 was used for bud detection. The steps to obtain branch information were the same as Algorithm 1, and finally, get the pruning location results according to the bud detection results and the branch information. (The batch size of Algorithm 1 and Algorithm 4 when testing was both 1)

Faster-RCNN[30] can only get the locations of the bud, while to find the pruning location, the help of the branch structure information was still essential.

The output images of the four algorithms are shown in Figure 6, and the measures of the four algorithms are listed in Table 3.



a. The result of Algorithm 1 b. The result of Algorithm 2 c. The result of Algorithm 3 d. The result of Algorithm 4

Note: The number next to the bounding box is the bottom-up order of the bud on the grapevine

Figure 6 Comparison of the effects of different algorithms

Table 3 Precision, recall, and accuracy for four algorithms

Algorithm name	Precision of bud detection/%	Recall of bud detection/%	Accuracy of pruning location/%	Total time/s
Algorithm 1	98.8	92.3	83.4	0.423
Algorithm 2	90.2	73.5	42.1	8.316
Algorithm 3	69.9	64.3	4.8	0.256
Algorithm 4	95.8	95.4	82.6	0.444

According to Table 3, Algorithm 1 achieved the best performance, while Algorithm 3 was the worst performer.

The worst performance of Algorithm 3 showed that only using corner features to represent the buds was imperfect. First, the intersections between the branches and the ends of the branches also showed obvious corner features. Second, some buds were the front buds, which did not show obvious corner features and cannot be detected by the Harris Algorithm. Third, Algorithm 3 had extremely high requirements for segmentation. Any inaccuracies in the edge of segmentation might be falsely detected as buds.

Compared with the results given in Pérez's research[8], the precision and recall of Algorithm 2 in this study's application were lower, because the buds in their images were near and prominent, while most buds in our images were far and small. In addition, some detection errors came from other steps rather than classification. In addition, due to repeated calculations from two different sizes of bounding boxes and usage of CPU, Algorithm 2 cost an extremely long time for the classification step.

The precision of Algorithm 4 was lower than Algorithm 1 because it erroneously detected the end branches on the horizontally growing branches and some unwanted buds on the short non-target branches, which were removed during Section 2.3 and Section 2.4 in Algorithm 1. However, the reason for the higher recall of Algorithm 4 was also the same. It can detect the buds that Algorithm 1 missed due to Section 2.3 and Section 2.4.

Algorithm 4 took a longer time than Algorithm 1 because Algorithm 1 has no RPN and bounding box regression compared to Algorithm 4, and the number of bounding boxes generated directly based on branch information was also less than that of faster-RCNN automatically generated by RPN. In addition, Algorithm 4 had one more step than Algorithm 1, which was to add the bud to the branch.

The accuracy of the pruning location was completely dependent on the bud detection precision and recall, and a small amount of missed or wrongly detected buds might have a great impact on the accuracy of the pruning location. Therefore, the accuracy of the pruning location of Algorithm 1 and Algorithm 4 was much greater than that of Algorithm 2 and Algorithm 3.

Algorithm 1 showed the best performance among the four algorithms.

The bud detection precision of Algorithm 1 was 98.8%, which meant that very few non-buds were classified as buds. Algorithm 1 achieved a recall of 92.3%, which was much lower than its precision. To Figure out the reasons for the low recall, we further analyzed the reason for each missed bud. The main reasons leading to errors were as follows:

Reason 1: Some grapevines were so thin to be completely segmented during segmentation resulting in the missing of these branches.

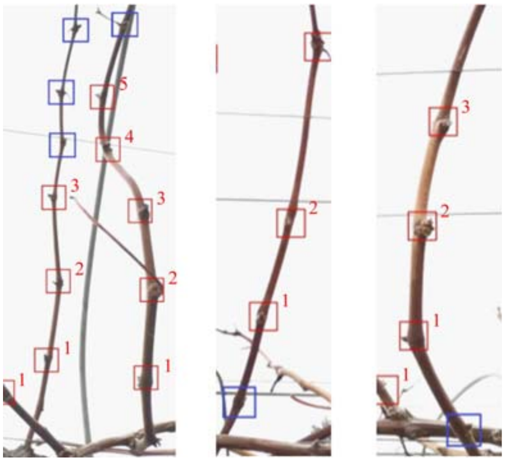
Reason 2: Some of the buds were too unobvious which generally occurs at the front buds.

Reason 3: Part of the branches at the bottom was not connected with the rest of the branches due to the obstruction of horizontal branches.

As shown in Figure 7, among the above three reasons, Reason 1 pointed to the errors in the segmentation step, Reason 2 meant the errors of classification, Reason 3 came from the step of finding the structure of each grapevine.

Count the number of these error reasons, the number of Reason 1 was 27, the number of Reason 2 was 21, and the number of Reason 3 was 53.

As above, Reason 3 provided the largest number of false negatives (FN), which accounted for 52.475% of the total number, and the second was Reason 1 with 26.733%. Reason 2 had the least, accounting for only 20.792%, which was relatively in line with the high accuracy achieved by the classifier in the networks test experiment. So the future work we first need to do is to improve the accuracy of Section 2.4.



a. Reason 1 b. Reason 2 c. Reason 3

Note: The red boxes indicate correct detections, and the blue boxes indicate missed detections

Figure 7 Missed detection of different reasons

4 Conclusions

In this study, a novel 2D grapevine winter pruning location detection method was proposed for automatic grapevine pruning in winter with a Y-shaped cultivation system. In this method, the images were segmented by the threshold in (2R−G−B) and S channel first, and then IEPTA was applied to extract the skeletons of branches. The skeletons were utilized to find the structure of each grapevine by judging the angle and distance relationship. After that, the bounding boxes were obtained on these grapevines. MobileNetV3_small×0.75 was used as the classifier to find the buds and finally get the winter pruning location in each grapevine.

Aiming at finding a lightweight neural network that was more suitable for our application, several lightweight neural networks were compared in Section 3.2. Among these networks, MobileNetV3_small×0.75 with an accuracy of 99.621% and params of 1.023 M showed the best comprehensive performance. The bud detection experiment was applied to test the overall performance of the method we proposed. This experiment showed that the method of this study achieved a precision of 98.8% and a recall of 92.3% for bud detection, an accuracy of 83.4% for pruning location detection and a total time of 0.423 s. And then we further analyzed the sources of errors in our method and found that these errors mainly come from the step of Section 2.4.

At present, there still have some steps in the method of this study that need further improvement. First, it is believed that the lightweight neural network could be further simplified. Second, the accuracy of Section 2.4 needs to be further improved. In the future work, the networks should be managed to simplify by Neural

Architecture Search to further reduce the amount of calculation. And SVM will also be tried to train for determining the connection between branches so that the errors in Section 2.4 can be reduced.

Acknowledgements

This work was financially supported by the Basic Public Welfare Research Project of Zhejiang Province (Grant No. LGN20E050007).

[References]

[1] Shamshiri R R, Weltzien C, Hameed I A, Yule I J, Grift T E, Balasundram S K, et al. Research and development in agricultural robotics: A perspective of digital farming. Int J Agric & Biol Eng, 2018; 11(4): 1–14.

[2] Liu S, Whitty M. Automatic grape bunch detection in vineyards with an SVM classifier. Journal of Applied Logic, 2015; 13(4): 643–653.

[3] Botterill T, Green R, Mills S. Finding a vine's structure by bottom-up parsing of cane edges. In: 2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013), Wellington, New Zealand: IEEE, 2013; pp.112–117. doi: 10.1109/IVCNZ.2013.6727001.

[4] Diago M-P, Correa C, Millán B, Barreiro P, Valero C, Tardaguila J. Grapevine yield and leaf area estimation using supervised classification methodology on RGB images taken under field conditions. Sensors, 2012; 12(12): 16988–17006.

[5] Xu S, Xun Y, Jia T M, Yang Q H. Detection method for the buds on winter vines based on computer vision. In: 2014 Seventh International Symposium on Computational Intelligence and Design, Hangzhou: IEEE, 2014; pp.44–48. doi: 10.1109/ISCID.2014.26

[6] Rosenfeld A. A characterization of parallel thinning algorithms. Information and Control, 1975; 29(3): 286–291.

[7] Harris C G, Stephens M J. A combined corner and edge detector. Alvey Vision Conference, 1988; pp.147–152. doi: 10.5244/c.2.23.

[8] Pérez D S, Bromberg F, Diaz C A. Image classification for detection of winter grapevine buds in natural conditions using scale-invariant features transform, bag of features and support vector machines. Computers and Electronics in Agriculture, 2017; 135: 81–95.

[9] Lowe D G. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 2004; 60(2): 91–110.

[10] Csurka G, Dance C R, Fan L, Willamowski J, Bray C. Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision ECCV 2004, 2004; pp.1–22.

[11] Vapnik V N, Mulier F M. Statistical learning theory. In: Learning from Data: Concepts, Theory, and Methods, IEEE, 2007; pp.99–150.

[12] Díaz C A, Pérez D S, Miatello H, Bromberg F. Grapevine buds detection and localization in 3D space based on structure from motion and 2D image classification. Computers in Industry, 2018; 99: 303–312.

[13] Zabawa L, Kicherer A, Klingbeil L, Toepfer R, Kuhlmann H, Roscher R. Counting of grapevine berries in images via semantic segmentation using convolutional neural networks. ISPRS Journal of Photogrammetry and Remote Sensing, 2020; 164: 73–83.

[14] Palacios F, Bueno G, Salido J, Diago M P, Hernandez I, Tardaguila J. Automated grapevine flower detection and quantification method based on computer vision and deep learning from on-the-go imaging using a mobile sensing platform under field conditions. Computers and Electronics in Agriculture, 2020; 178: 105796. doi: 10.1016/j.compag.2020.105796.

[15] Cruz A, Ampatzidis Y, Pierro R, Materazzi A, Panattoni A, Bellis L D, et al. Detection of grapevine yellows symptoms in Vitis vinifera L. with artificial intelligence. Computers and Electronics in Agriculture, 2019; 157: 63–76.

[16] Marset W V, Perez D S, Diaz C A, Bromberg F. Towards practical 2D grapevine bud detection with fully convolutional networks. Computers and Electronics in Agriculture, 2021; 182: 105947. doi: 10.1016/j.compag.2020.105947.

[17] Zhang T Y, Suen C Y. A fast parallel algorithm for thinning digital patterns. Communications of the ACM, 1984; 27(3): 236–239.

[18] Naccache N J, Shinghal R. An investigation into the skeletonization approach of Hilditch. Pattern Recognition, 1984; 17(3): 279–284.

[19] Lu H E, Wang P S P. A comment on "A fast parallel algorithm for thinning digital patterns". Communications of the ACM, 1986; 29(3): 239–242.

[20] Lu H E, Wang P S P. An improved fast parallel thinning algorithm for digital patterns. In: Proceedings CVPR '85: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1985; pp.364–367.

[21] Zhao D D, Wang H B, Tao L, Zhou J. Improved EPTA parallel thinning algorithm. Computer Engineering and Applications, 2016; 52(9): 196–201. (in Chinese)

[22] Sandler M, Howard A, Zhu M L, Zhmoginov A, Chen L-C. MobileNetV2: Inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA: IEEE, 2018; pp.4510–4520. doi: 10.1109/CVPR.2018.00474.

[23] Howard A, Sandler M, Chu G, Chen L-C, Chen B, Tan M, et al. Searching for MobileNetV3. In: 2019 IEEE/CVF International Conference on Computer Vision, IEEE, 2019; pp.1314–1324.

[24] Ma N, Zhang X, Zheng H-T, Sun J. ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y (Eds). Computer Vision - ECCV 2018, Springer, 2018; pp. 122–138. doi: 10.1007/978-3-030-01264-9_8.

[25] Mingxing T, Le Q V. EfficientNet: Rethinking model scaling for Convolutional Neural Networks. arXiv, 2019; arXiv: 1905.11946 doi: 10.48550/arXiv.1905.11946.

[26] Iandola F N, Han S, Moskewicz M W, Ashraf K, Dally W J, Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. arXiv, 2016; arXiv: 1602.07360. doi: 10.48550/arXiv.1602.07360.

[27] Rosenfeld A, Thurston M. Edge and curve detection for visual scene analysis. IEEE Transactions on Computers, 1971; C-20(5): 562–569.

[28] Panigrahi S, Nanda A, Swarnkar T. A survey on transfer learning. IEEE Transactions on Knowledge and Data Eengineering, 2010; 22(10): 1345–1359.

[29] Shorten C, Khoshgoftaar T M. A survey on image data augmentation for deep learning. Journal of Big Data, 2019; 6(1): 60. doi: 10.1186/s40537-019-0197-0.

[30] Ren S Q, He K M, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017; 39(6): 1137–1149.