agronomy

MDPI

Article

Leaf Segmentation and Classification with a Complicated Background Using Deep Learning

Kunlong Yang †, Weizhen Zhong † and Fengguo Li *

Guangdong Provincial Key Laboratory of Quantum Engineering and Quantum Materials, Guangdong Provincial Engineering Technology Research Center for Quantum Precision Measurement, National Demonstration Center for Experimental Physics Education, SPTE, South China Normal University, Guangzhou 510006, China; 2019021917@m.scnu.edu.cn (K.Y.); 2018021895@m.scnu.edu.cn (W.Z.)

* Correspondence: lifengguo@m.scnu.edu.cn

† These authors contributed equally to this work.

Received: 10 October 2020; Accepted: 2 November 2020; Published: 6 November 2020

Abstract: The segmentation and classification of leaves in plant images are a great challenge, especially when several leaves are overlapping in images with a complicated background. In this paper, the segmentation and classification of leaf images with a complicated background using deep learning are studied. First, more than 2500 leaf images with a complicated background are collected and artificially labeled with target pixels and background pixels. Two-thousand of them are fed into a Mask Region-based Convolutional Neural Network (Mask R-CNN) to train a model for leaf segmentation. Then, a training set that contains more than 1500 training images of 15 species is fed into a very deep convolutional network with 16 layers (VGG16) to train a model for leaf classification. The best hyperparameters for these methods are found by comparing a variety of parameter combinations. The results show that the average Misclassification Error (*ME*) of 80 test images using Mask R-CNN is 1.15%. The average accuracy value for the leaf classification of 150 test images using VGG16 is up to 91.5%. This indicates that these methods can be used to segment and classify the leaf image with a complicated background effectively. It could provide a reference for the phenotype analysis and automatic classification of plants.

Keywords: deep learning; image segmentation; Mask R-CNN; plant classification; VGG16

1. Introduction

To realize sustainable agriculture and boost agricultural yield, plant phenotyping is a significant process [1,2]. The color and shape of the leaf, plant height, leaf area index, and growth rate are important information for phenotype analysis. The automatic and non-destructive extraction of leaves from the plant images can boost the phenotype analysis.

The important information that leaves contain can be used to identify plant species. Plant identification is usually by their floral parts, fruits, and leaves. Flowers and fruits are not suitable for plant identification as they appear for a short interval. Leaves, on the other hand, are available for a longer duration and are available in abundance. Therefore, leaves are a suitable choice for the automatic classification of plants [3].

Recently, urbanization and biodiversity loss have made plant classification a significant problem for many professionals such as agronomists, gardeners, and foresters. Classification of the plant has great significance to explore the genetic relationship of plants and explain the evolution of plants. However, considering the great number of species, plant identification is a fairly difficult task, even for botanists [4].

Therefore, the automatic segmentation and classification of leaves in plant images with a complicated background have been further studied.

In recent years, some researchers developed many methods to segment and identify leaves. Wang et al. [5] presented a two stage approach for leaf image retrieval by using simple shape features. However, in some cases, it is impossible to differentiate one leaf from another by shape alone. To address this issue, H. Fu et al. [6] tried to classify leaves by their veins. They proposed an approach that combines a thresholding method and an artificial neural network classifier, to extract vein patterns from leaf images, and tried to apply it to leaf classification. Nevertheless, the two methods mentioned above are focused on single leaf image segmentation and classification with a simple or pure background. Commonly, the captured images of field-living plants usually contain a complicated background. To resolve this problem, Xiao-Feng Wang et al. [7] proposed a method that combines pre-segmentation and a morphological operation to segment leaf images with a complicated background, which obtained an average correct classification rate up to 92.6%. However, this method is not automatic. The optimum thresholding is different for each image, which makes the segmentation task time consuming. G. Alenya et al. [8] tried to segment leaves using time-of-flight data, which can even gather three-dimensional information. However, strong sunlight could affect the accuracy of time-of-flight data, which means the segmentation task can only be conducted under dim lighting conditions.

With the development of image processing technology and deep learning, some researchers tried to identify or segment multiple leaves in an image using deep learning [9–29]. For example, S. Aich et al. [10] used a deep learning architecture to count the leaves in plant images. D. Kuznichov et al. [13] tried to improve the accuracy of leaf segmentation using the data augmentation method. H. Scharr et al. [15] compared four methods using deep learning to segment the leaves of digital plant images. They found that the leaf segmentation using these methods can reach an average accuracy above 90%. However, they also pointed out that the complications in the background could lower the accuracy. J. Bell et al. [17] introduced an approach using a relatively shallow convolutional neural network to segment and classify the leaf images. This approach is strong in distinguishing occluding pairs of leaves where one leaf is largely hidden. Although obtaining some encouraging results, the main limitation of these methodologies is the use of shallow Convolutional Neural Networks (CNNs). K. Simonyan et al. [30] proposed a CNN by adding more convolutional layers and using very small convolution filters in all layers. The result showed that the increased depth led to better performance. S. Ren et al. [31] proposed a CNN by adding a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network. It improves region proposal quality and thus the overall object detection accuracy. Although these CNNs are not proposed to perform leaf segmentation and classification, it is possible to apply these networks to leaf segmentation and classification or other similar tasks with enough training and fine-tuning [13].

J. Gené-Mola et al. [32] used a Kinect v2 RGB-D camera and Faster Region-based Convolutional Neural Networks (Faster R-CNNs) for apple fruit detection. It can accurately identify apples in an image with a complicated background. Zhou et al. [33] optimized VGG16 and built an eight layer network to extract features of the main organs of tomatoes, such as the stem, flower, and fruit. To realize real-time recognition of apple fruits in the field, Tian et al. [34] optimized the YOLO-V3 architecture and used the dense convolutional network to deal with the low-resolution feature layers. They found that the improved model performed better than the original YOLO-V3 model and Faster R-CNN. However, these neural network algorithms (such as R-CNN, Fast R-CNN, Faster R-CNN, and YOLO) can only roughly frame the target using the bounding box. These algorithms are unable to extract contour and shape information. However, the shape of the leaf is among the key information for plant phenotyping. Therefore, a high precision of leaf contour and shape recognition is necessary. Nevertheless, the Mask Region Convolutional Neural Network (Mask-RCNN), proposed by Kaiming et al. [35], has been able to segment objects with masks.

In this paper, the segmentation and classification of leaf images with a complicated background using deep learning are studied. Because Mask R-CNN can recognize and extract object regions from the background at the pixel level, it is suitable for the leaf segmentation task. We used VGG16 to develop a leaf classifier. Compared to other CNNs (such as VGG19 and Inception ResNetV2 [36]), it has fewer parameters and less depth, which is better for training with a limited dataset [30].

2. Materials and Methods

2.1. Image Acquisition

Many state-of-the-art models take a vast amount of labeled data to obtain a better result [13]. Therefore, it is essential to collect sufficient training images. However, there is no open-source training set of leaf images with a complicated background. Plants affiliated with the South China Normal University were selected as our research objects. The images were captured in early November 2017 with a mobile phone and a digital camera. The mobile phone was the iPhone 6s. The camera was the NIKON D610. The images were stored in JPEG format with a resolution of 3024×4032 and 4512×3008, respectively. Due to the growing seasons and climate changes, the plant types available were very limited. We chose 15 species that have a minor phenotypic difference and were available in abundance for our study, including Gardenia jasminoides, Callisia fragrans, Psidium littorale, etc. Figure 1 shows the collection of the 15 species. More than 15,000 leaf images were captured and about 1000 images for each species.

Figure 1. The collection of the 15 species.

2.2. Segmentation

2.2.1. Annotation

The training set used in Mask R-CNN must be labeled. Because of the limited video memory of our GPU, the resolution of the training images should be no more than 850 × 850. We reduced the resolution of all training images first. Due to the complicated background and overlapping of the leaves, the challenge of segmentation was increased significantly. Accordingly, it would be a huge effort to label all the images. We chose more than 2000 of them for our training and labeled them with Labelme-3.3.6, which was developed by MIT's Computer Science and Artificial Intelligence Laboratory. Figure 2 shows some images during the labeling process. Because Mask R-CNN is used for segmentation in our experiment, it only has two classifications in our study: leaf and background. The label data were saved as JSON format by Labelme. Later, we converted them to the COCO dataset format and input them into the neural network. Three of the labeled images are presented in Figure 3.

Figure 2. The images during the labeling process.

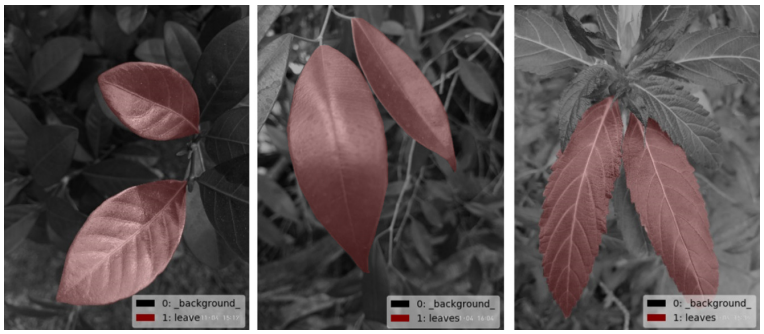

Figure 3. The labeled images.

2.2.2. Mask-RCNN

Mask R-CNN can be divided into two parts. The first part can extract features and select a large number of candidate object regions, then feed them into the second part. The second part can produce the box classification, the box regression, and the object mask. The flowchart is presented in Figure 4. According to the flowchart, Mask R-CNN consists of several modules. The explanations are as follows:

(1) ResNet-Feature Pyramid Network (ResNet-FPN): The ResNet-FPN is the backbone of Mask R-CNN, which is the integration of ResNet and the Feature Pyramid Network (FPN). ResNet is a standard convolutional neural network, which can extract the features from the images. The first few layers extract the low-level features, then the following layers extract higher level features. To improve upon the feature map, the authors who developed Mask R-CNN introduced the FPN as an extension [35], which can better

present the object in the feature map at multiple scales. The FPN improves the extraction ability by adding the second pyramid to the standard feature extraction pyramid. The second pyramid can take high-level features from the first pyramid and feed them into the lower layers, which can fully integrate features from different levels. In our experiment, ResNet101 + the FPN backbone were used.

(2) Region Proposal Network (RPN): Using a sliding window, the RPN module can select large numbers of areas that contain objects from the features map. The selected regions are called anchors, which are the boxes that frame the objects. In practice, there are more than 200 K anchors with different sizes and aspect ratios, and they will cover objects of the image as much as possible. The RPN prediction will select the anchors that are likely to contain the objects and resize the frame to fit it. If some anchors are overlapping too much in a region, the RPN prediction will keep the one with the highest foreground score and discard the rest of them.

(3) ROI align: Using the bilinear interpolation, this can crop a part of the feature map, where the Region Of Interest (ROI) is, and pass it to the ROI classifier and bounding box regressor.

(4) Box regression and classification: The ROI from ROI align are fed into this stage, which includes the ROI classifier and bounding box regressor. The ROI classifier is a deeper network, which has the ability to refine the classification of the ROI. However, in this experiment, the ROI classifier is only used to classify two classes (foreground and background). The function of the bounding box regressor is similar to the RPN. However, it can further refine the box to fit the object.

(5) Segmentation mask: This branch is a convolutional network, which can mask the positive region given by the ROI classifier. In order to keep the mask branch light, the generated masks are low in resolution (28×28 pixels). However, in the output image, the mask is scaled up to the size of the bounding box.

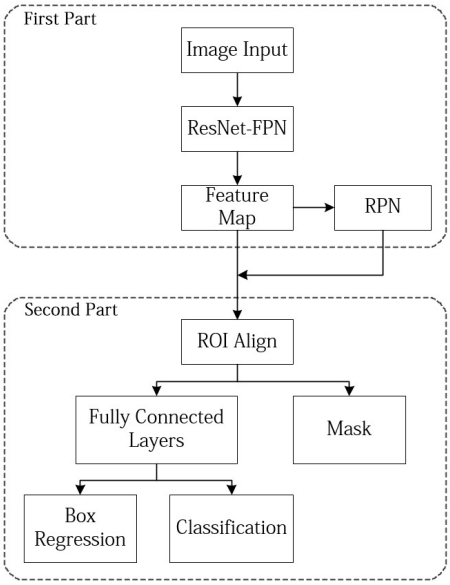

Figure 4. The flowchart of Mask R-CNN.

We trained the model using 2000 training images. We tested the model with a variety of parameter combinations. At first, the max epoch, learning rate, and momentum were set as 12, 0.01, and 0.9, respectively. However, this model was underfitting because of the low epoch. After, the max epoch, learning rate, and momentum were set as 24, 0.02, and 0.9. This time, the model was overfitting and oscillation occurred, because the learning rate was too high. Therefore, in our study, the max epoch,

learning rate, and momentum, were set as 24, 0.01, and 0.9, respectively. The total training time was about 11 h.

2.3. Classification

The VGGnet model was used for classification. There are several preparatory tasks that have to be done before training. First, the input image size should be 224 × 224. Therefore, all the training images were transformed to this size. Then, all the images should be labeled. We ran a script written with Python-2.7 to label all the images and stored the labeled data as an HDF5 file format.

In our study, the VGGnet model with 16 layers (VGG16) was used [30]. To reduce the training time and improve the robustness, the transfer learning method was used. It can get the pre-trained configurations of the model from the ImageNet dataset, which contains more than 1.4 million labeled images with more than 1000 different classes. Because of the huge dataset, the spatial hierarchy of features learned from the dataset was huge.

The VGG16 model contains 13 convolutional layers, 2 fully connected layers, and 1 softmax classifier. The architecture of the VGG16 model is presented in Figure 5. According to the architecture, the explanations are as follows.

(1) Convolutional layer: In this layer, a 3 × 3 matrix called the kernel will slide over the input matrix. During the sliding process, at every location, an element-wise matrix multiplication (convolution) is performed and sums the result on the feature map. After this process, a feature map is created. If the input image is 2-dimensional, the convoluted matrix can be calculated as follows:

$$S(i,j) = \sum_m \sum_n I(m,n)k(i-m, j-n) \quad (1)$$

where I is the matrix of the input image, k is the kernel, S is the convoluted matrix, and m and i are the row number of the input matrix and the convoluted matrix, respectively. n and j are the column number of the input matrix and the convoluted matrix, respectively.

(2) Non-linear activation functions (ReLU): ReLU is a node that comes after the convolutional layer, which can do a nonlinear transformation over the input signal. When the input is positive, it will output the input; otherwise, it will output zero.

(3) Pooling layer: The feature map acquired from the convolutional layer has a drawback. Every position of the feature map is an accurate reflection of the corresponding position of the input image. Therefore, when the input image has minor changes like cropping or rotation, the output feature map will be completely different. To cope with this problem, a pooling layer is applied after ReLU. The pooling layer can make the output of ReLU approximately invariant to a small alternation of the input image.

(4) Fully connected layer: This can connect every node in the first layer to the nodes in the second layer. Usually, at the end of a convolutional neural network, the input of the fully connected layer is the output of the pooling layer, and the number of fully connected layers can be one or more.

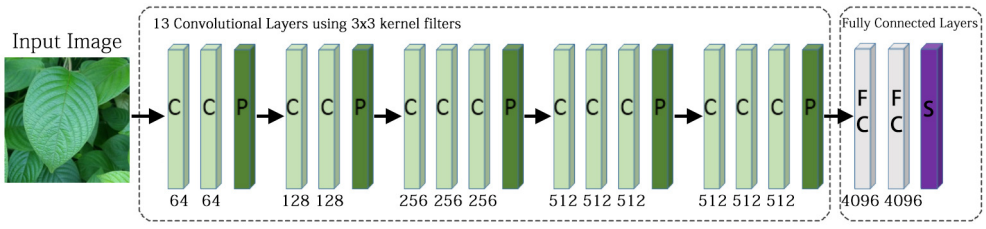

Figure 5. VGG16 model architecture.

To get a better result, we fine-tuned the whole base model. We re-trained it on our data with a very low learning rate. This can achieve meaningful improvements, by incrementally adapting the pre-trained features to the new data. The learning rate was initially set to 0.01 and then decreased by a factor of 10 when the validation set's accuracy stopped improving. We decreased it 3 times, and then, it reached the best performance. VGG16 required fewer epochs to converge due to the implicit regularization imposed by the greater depth and smaller convolution [30]. Therefore, the epoch was set to 10.

3. Results and Discussion

3.1. Segmentation

We performed the segmentation training and evaluation on a Ubuntu 16.04 system. The system was equipped with an NVIDIA Tesla P4 GPU (video memory was 8 G).

The segmentation targets were the relatively larger leaves on the image. The relatively smaller leaves in the background of the images were not the targets for segmentation. Mask R-CNN only separates the target leaves from the background with masks of different colors. In the results, each leaf was framed in a green box. One of the images after segmentation is presented in Figure 6. The number 0.98 means that the probability of the correct recognition of the leaf is 98%.

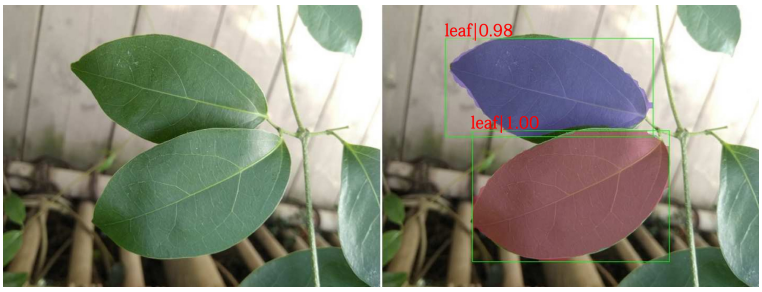

Figure 6. Result of segmentation.

The leaf images artificially labeled with target pixels and background pixels were used as the ground truth data in the experiment. To acquire a quantitative evaluation of the segmentation, the Misclassification Error (ME) was used to evaluate the result. It can be determined by the formula:

$$ME = 1 - \frac{|B_O \bigcap B_T| + |F_O \bigcap F_T|}{M \times N} \quad (2)$$

The image was segmented into the foreground and background. The foreground is the target leaves in the experiment, and the rest is the background. B_O is the number of pixels of the background of the ground truth image. B_T is the number of pixels of the background segmented by Mask R-CNN. F_O is the number of pixels of the foreground of the ground truth image. F_T is the number of pixels of the foreground segmented by Mask R-CNN. $M \times N$ means the total pixels of the test image. The smaller the ME is, the better the segmentation result is.

We compared the segmentation results of the proposed method with two other segmentation algorithms. These were the Otsu segmentation algorithm [37] and Grabcut [38]. We chose 80 test images for the comparison, with eight images per species. The average ME of each method is shown in Table 1. The ME of each image is shown in Figure 7. The average ME of this designed segmentation method, Grabcut, and Otsu segmentation algorithm was 1.15%, 28.74%, and 29.80%, respectively. It can be seen that the algorithm proposed in this paper had a good effect on the ME.

Table 1. The average Misclassification Error (*ME*) of different segmentation methods.

Method	Average *ME*
The designed segmentation method	1.15%
Grabcut	28.74%
Otsu segmentation algorithm	29.80%

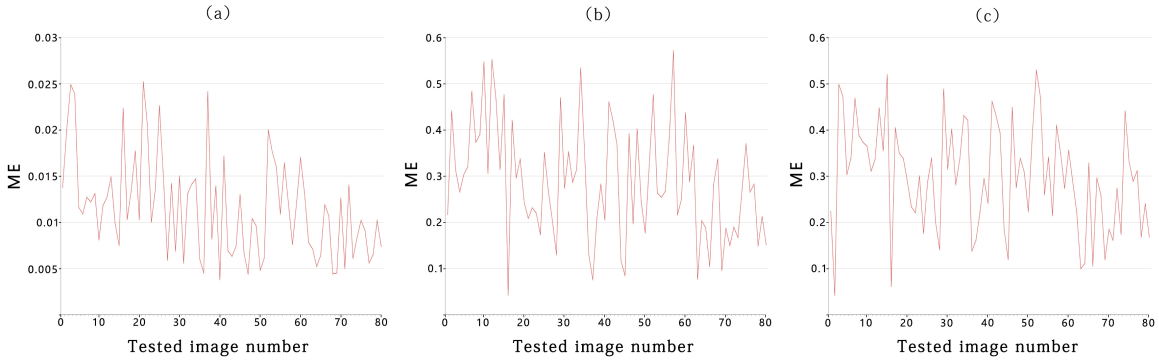

Figure 7. The *ME* of each image: (**a**) the designed segmentation method results, (**b**) Grabcut results, and (**c**) Otsu segmentation algorithm results.

3.2. Classification

The classification training and evaluation were performed on a Ubuntu 18.04 system. The system was equipped with an Intel i7-7700 CPU.

We collected our own dataset in this study. The leaf dataset had 1500 images of 15 species classes with 100 images per class. These images were separated into the training set and the test set, respectively, with a ratio of 9:1. We compared the classification results of the proposed method with two other neural networks. These were VGG19 and Inception ResNetV2 [36]. The total training time of VGG16, VGG19, and Inception ResNetV2 was 192 minutes, 381 minutes, and 461 minutes, respectively. The experiment results are shown in Table 2.

The results show that VGG19 achieved the highest classification accuracy. We observed that the classification accuracy of VGG16 was slightly lower than VGG19. This is because VGG19 has a deeper network. However, the computation speed of VGG19 was about twice slower than VGG16 in our experiment, and its detection accuracy was not significantly higher than that of VGG16. The classification accuracy of Inception ResNetV2 was slightly lower than VGG16. We believe this may be due to the fact that it was developed with a focus on ImageNet and thus overfit this specific task. Based on the above results, we can conclude that VGG16 can perform well in the classification of leaf images with a complicated background at a relatively faster computation speed.

Table 2. The classification accuracy of different neural network architecture.

Species	Recognition Accuracy Rate (%)		
	VGG19	**VGG16**	**Inception ResNetV2**
Gardenia jasminoides	94.6%	93.1%	92.5%
Callisia fragrans	89.3%	91.8%	90.2%
Psidium littorale	75.3%	85.3%	83.1%
Osmanthus fragrans	84.4%	88.4%	85.8%
Bixa orellana	90.1%	88.6%	84.3%
Ficus microcarpa	91.9%	87.2%	89.8%
Calathea makoyana	100%	99.2%	98.6%
Rauvolfia verticillata	96.7%	92.4%	91.3%
Ardisia quinquegona	95.2%	90.5%	87.6%
Baccaurea ramiflora	86.5%	85.5%	80.1%
Synesepalum dulcificum	97.3%	96.8%	96.4%
Hydnocarpus anthelminthicus	98.2%	94.5%	92.7%
Daphne odora	96.5%	93.3%	91.6%
Dracaena surculosa	94.1%	92.5%	89.8%
Mussaenda pubescens	96.1%	93.4%	90.5%
Average Value	92.4%	91.5%	89.6%

3.3. Discussion

It can be seen from Figure 8 that the Otsu segmentation algorithm cannot classify the leaf with a dark streak very well, and it cannot segment the overlapping area. This was mainly because the color of the streak of some leaves was similar to the background color in a dark environment, and it was hardly possible to segment them simply by color. Grabcut is not affected by the streak on the leaf. However, it still cannot segment the overlapping area. The designed segmentation method can segment overlapping leaves correctly and has the best result among these methods. According to the results given above, the algorithm studied in this paper can segment multiple overlapping leaves with a complicated background accurately.

As can be seen from Table 2, the image classification methods based on deep learning achieved good results in plant recognition with a complicated background. Nevertheless, VGG16 achieved this result at a relatively faster computation speed. This has great significance when the species classes and training dataset are huge. It is always greatly laborious to label all the images and train the model with them. Compared to VGG16, VGG19 and Inception ResNetV2 are deeper models with more parameters. This will increase the difficulty of fine-tuning. In conclusion, VGG16 had the best comprehensive performance for leaf image classification. The results show that deep learning can be robustly applied to complicated leaf image classification.

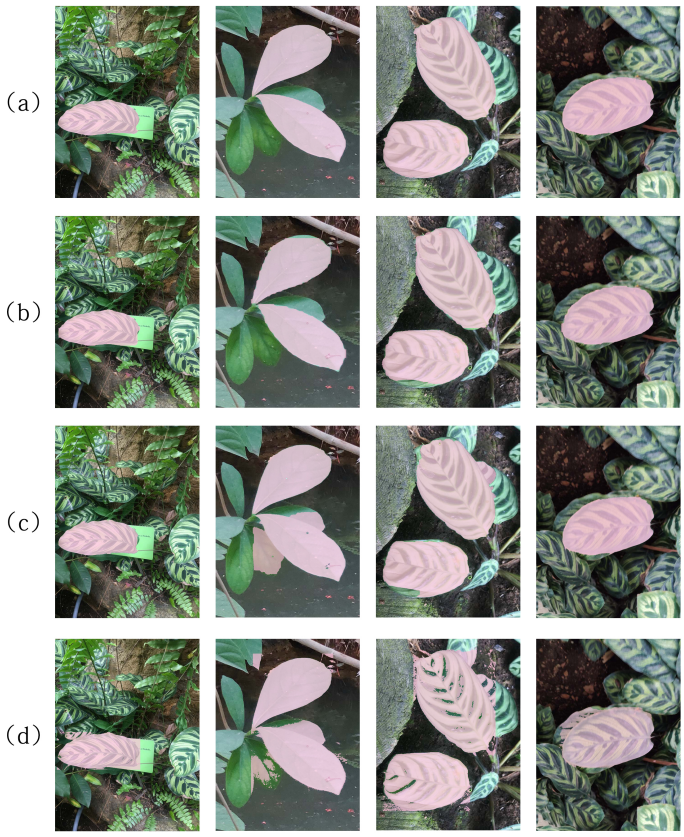

Figure 8. Manual labeling results and algorithm segmentation results: (**a**) manual labeling results, (**b**) the designed segmentation method results, (**c**) Grabcut results, and (**d**) Otsu segmentation algorithm results.

4. Conclusions

In this paper, the Mask R-CNN model and the VGG16 model are used to segment and classify leaf images with multiple targets and a complicated background. More than 4000 images were used for model training and testing. The results show that the average *ME* of segmentation is up to 1.15% using the Mask R-CNN model, and the average classification accuracy is up to 91.5% using the VGG16 model. This shows that the Mask R-CNN model and the VGG16 model could reliably be used in the segmentation and classification of leaf images with a complicated background. Further study is recommended to be performed with different deep learning algorithms and a greater number of data, which may lead to a better result. Besides the algorithm's development, improving the image quality with better devices can also contribute to better performance. What is more, it will be possible to segment and classify leaves automatically in real-time by using an embedding system.

Author Contributions: Conceptualization, K.Y. and F.L.; methodology, K.Y., F.L. and W.Z.; formal analysis, K.Y. and W.Z.; investigation, K.Y. and W.Z.; resources, K.Y. and W.Z.; data curation, K.Y. and F.L.; writing, original draft preparation, K.Y., F.L. and W.Z.; writing, review and editing, K.Y., F.L. and W.Z.; visualization, W.Z. and F.L.; supervision, F.L. All authors read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 11575064) and the Natural Science Foundation of Guangdong Province, China (Grant No. 2016A030313433).

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

References

1. Lei, L.; Qin, Z.; Danfeng, H. A Review of Imaging Techniques for Plant Phenotyping. *Sensors* **2014**, *14*, 20078–20111. [CrossRef] [PubMed]
2. Achim, W.; Frank, L.; Andreas, H. Plant phenotyping: From bean weighing to image analysis. *Plant Methods* **2015**, *11*, 14.
3. Abdolvahab, E.R. Plant Classification Based on Leaf Recognition. *Int. J. Comput. Sci. Inf. Secur.* **2010**, *8*, 78–81.
4. Pierre, B.; Ben, Y.S.; Kai, F.M.; Volker, S. LeafNet: A computer vision system for automatic plant species identification. *Ecol. Inform.* **2017**, *40*, 50–56.
5. Wang, Z.; Chi, Z.; Feng, D. Shape based leaf image retrieval. *IEE Proc. Vis. Image Signal Process.* **2003**, *150*, 34. [CrossRef]
6. Fu, H.; Wang, Z. Combined thresholding and neural network approach for vein pattern extraction from leaf images. *IEE Proc. Vis. Image Signal Process.* **2006**, *153*, 881. [CrossRef]
7. Wang, X.; Huang, D.; Du, J.; Xu, H.; Heutte, L. Classification of plant leaf images with complicated background. *Appl. Math. Comput.* **2008**, *205*, 916–926. [CrossRef]
8. Alenya, G.; Dellen, B.; Foix, S.; Torras, C. Robotized Plant Probing: Leaf Segmentation Utilizing Time-of-Flight Data. *IEEE Robot. Autom. Mag.* **2013**, *20*, 50–59. [CrossRef]
9. Kumar, J.P.; Domnic, S. Image based leaf segmentation and counting in rosette plants. *Inf. Process. Agric.* **2019**, *6*, 233–246.
10. Aich, S.; Stavness, I. Leaf counting with deep convolutional and deconvolutional networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 22–29 October 2017.
11. Itakura, K.; Hosoi, F. Automatic leaf segmentation for estimating leaf area and leaf inclination angle in 3d plant images. *Sensors* **2018**, *18*, 3576. [CrossRef]
12. Turkoglu, M.; Hanbayi, D. Leaf-based plant species recognition based on improved local binary pattern and extreme learning machine. *Phys. A Stat. Mech. Appl.* **2019**, *527*, 121297. [CrossRef]
13. Kuznichov, D.; Zvirin, A.; Honen, Y.; Kimmel, R. Data Augmentation for Leaf Segmentation and Counting Tasks in Rosette Plants. *arXiv* **2019**, arXiv:1903.08583.
14. Ozguven, M.M.; Adem, K. Automatic detection and classification of leaf spot disease in sugar beet using deep learning algorithms. *Phys. A Stat. Mech. Appl.* **2019**, *535*, 122537. [CrossRef]
15. Scharr, H.; Minervini, M.; French, A.; Kramer, D. Leaf segmentation in plant phenotyping: A collation study. *Mach. Vis. Appl.* **2016**, *27*, 585–606. [CrossRef]
16. Salvador, A.; Bellver, M.; Campos, V.; Baradad, M.; Marques, F.; Torres, J.; Giro-i-Nieto, X. Recurrent Neural Networks for Semantic Instance Segmentation. *arXiv* **2017**, arXiv:1712.00617.
17. Bell, J.; Dee, H.M. Leaf segmentation through the classification of edges. *arXiv* **2019**, arXiv:1904.03124.
18. Viaud, G.; Loudet, O.; Cournde, P. Leaf segmentation and tracking in arabidopsis thaliana combined to an organ-scale plant model for genotypic differentiation. *Front. Plant Sci.* **2016**, *7*, 2057. [CrossRef]
19. Al-Shakarji, N.; Kassim, Y.; Palaniappan, K. Unsupervised learning method for plant and leaf segmentation. In Proceedings of the 2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Washington, DC, USA, 10–12 October 2017.
20. Arvidsson, S.; Prez-Rodrguez, P.; Mueller-Roeber, B. A growth phenotyping pipeline for arabidopsis thaliana integrating image analysis and rosette area modeling for robust quantification of genotype effects. *New Phytol.* **2011**, *191*, 895–907. [CrossRef]
21. Camargo, A.; Papadopoulou, D.; Spyropoulou, Z.; Vlachonasios, K.; Doonan, J.; Gay, A. Objective definition of rosette shape variation using a combined computer vision and data mining approach. *PLoS ONE* **2014**, *9*, e96889. [CrossRef]
22. Dobrescu, A.; Giuffrida, M.; Tsaftaris, S. Leveraging multiple datasets for deep leaf counting. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 22–29 October 2017.
23. Giuffrida, M.V.; Doerner, P.; Tsaftaris, S.A. Pheno-deep counter: A unified and versatile deep learning architecture for leaf counting. *Plant J.* **2018**, *96*, 880–890. [CrossRef]

24. Giuffrida, M.V.; Minervini, M.; Tsaftaris, S.A. Learning to count leaves in rosette plants. In Proceedings of the Computer Vision Problems in Plant Phenotyping Workshop 2015, Swansea, UK, 10 September 2015.
25. Giuffrida, M.V.; Scharr, H.; Tsaftaris, S.A. Arigan: Synthetic arabidopsis plants using generative adversarial network. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 22–29 October 2017.
26. Pape, J.M.; Klukas, C. 3-d histogram-based segmentation and leaf detection for rosette plants. In *Computer Vision—ECCV 2014 Workshops*; Springer: Cham, Switzerland, 2014.
27. Pape, J.M.; Klukas, C. Utilizing machine learning approaches to improve the prediction of leaf counts and individual leaf segmentation of rosette plant images. In Proceedings of the Computer Vision Problems in Plant Phenotyping Workshop 2015, Swansea, UK, 10 September 2015.
28. Ren, M.; Zemel, R. End-to-end instance segmentation with recurrent attention. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
29. Paredes, B.R.; Torr, P.H.S. Recurrent instance segmentation. In *Computer Vision ECCV (2016)*; Springer: Cham, Switzerland, 2016; pp. 312–329.
30. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
31. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
32. Gené-Mola, J.; Vilaplana, V.; Rosell-Polo, J.R.; Morros, J.R.; Ruiz-Hidalgo, J.; Gregorio, E. Multi-modal deep learning for Fuji apple detection using RGB-D cameras and their radiometric capabilities. *Comput. Electron. Agric.* **2019**, *162*, 689–698. [CrossRef]
33. Yuncheng, Z.; Tongyu, X.; Wei, Z.; Hanbing, D. Classification and recognition approaches of tomato main organs based on DCNN. *Trans. Chin. Soc. Agric. Eng.* **2017**, *33*, 219–226.
34. Yunong, T.; Guodong, Y.; Zhe, W.; Hao, W.; En, L.; Zize, L. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comput. Electron. Agric.* **2019**, *157*, 417–426.
35. Kaiming, H.; Georgia, G.; Dollar, P.; Girshick, R. Mask rcnn. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *42*, 386–397.
36. Christian, S.; Sergey, L.; Vincent, V.; Alex, A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv* **2016**, arXiv:1602.07261.
37. Dong, Y.X. An Improved Otsu Image Segmentation Algorithm. *Adv. Mater. Res.* **2014**, *989–994*, 3751–3754. [CrossRef]
38. Siyang, H.; Ping, S. GrabCut color image segmentation based on region of interest. In Proceedings of the 2014 7th International Congress on Image and Signal Processing, Dalian, China, 14–16 October 2014.