Shoot Counting System Based on SegNet

Funa Ito
School of Information and
Telecommunication Engineering
Tokai University
Tokyo, Japan
E-mail: funa.ito@outlook.jp

Duke Maeda
School of Information and
Telecommunication Engineering
Tokai University
Tokyo, Japan
E-mail: maedaduke@gmail.com

Naoki Nakamura
School of Information and
Telecommunication Engineering
Tokai University
Tokyo, Japan
E-mail: nakamura.naoki@star.tokai-u.jp

Kenta Morita
Graduate School of Engineering
Mie University
Mie, Japan
E-mail: k-morita@ip.elec.mie-u.ac.jp

Naoki Morita
School of Information and
Telecommunication Engineering
Tokai University
Tokyo, Japan
E-mail: morita@tokai.ac.jp

***Abstract*—In the present paper, we analyze an aerial photograph of a vineyard to count the number of shoots using SegNet, which is a tool for extracting the area of an object. However, occluded object regions make it difficult to recognize the hidden surface. The area of the object in the background is divided into two areas. It is thus necessary to connect the divided areas to obtain the correct number of shoots. Here, we propose a system that combines adjacent pixels and then connects all parts of the segmented area based on the segmentation output.**

Keywords - SegNet; Segmentation; Shoot; Pruning; Grapes.

I. Introduction

Some vineyards in Japan have adopted shelf cultivation, for which overhead work is required. Farmers can only see a tiny area at a given time when working on the vineyard. There is a project in Japan that helps vineyard farmers [1]. The goal of this project is to develop a system that can count the shoots on a tree from an aerial photograph of the vineyard taken by a drone.

Here, we propose a method that uses image recognition technology based on object area detection. In this study, we use SegNet [2] with the Caffe deep learning library for area detection. SegNet has an encoder, which has a general convolutional neural network structure, and a decoder. The encoder can be changed to ResNet, or other similar networks, to improve recognition accuracy. Figure 1 shows an aerial photograph taken by a drone. Figure 2 shows the result of area detection obtained with the VGG16 encoder for the photograph in Figure 1. Segmented branch pixels are shown in red, segmented shoot pixels are shown in yellow, and other pixels are shown in gray.

However, it is insufficient to count the number of new treetops per branch using only SegNet. As shown in Figure 2, there are two issues, namely shoots occlude branches and misrecognition. Although we use SegNet in the present study, the same issues arise using, for example, U-Net [3], DANet [4], and OCNet [5].

The purpose of this study is to determine the number of new treetops per branch from one aerial photograph.

The remainder of this paper is organized as follows. Section II describes the proposed concept. Section III validates the concept. Section IV gives the conclusions.

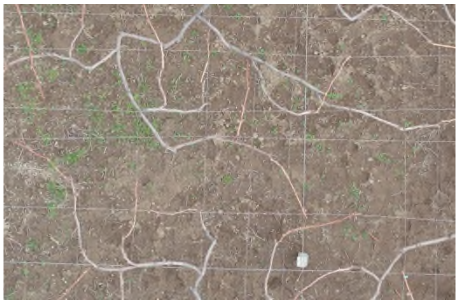

Figure 1. Original photograph.

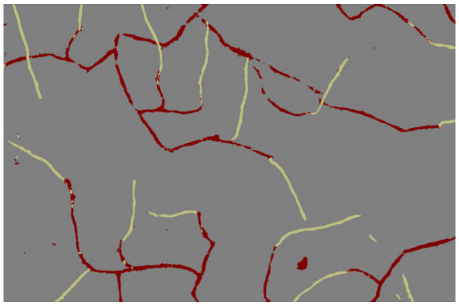

Figure 2. Segmentation result.

II. Proposed Concept

ID numbers are assigned to each red and yellow region and connected part pairs are counted to determine the number of shoots. The same ID number is assigned to parts of a divided area.

In this section, we describe a method for counting all treetops in the segmented image shown in Figure 2 and a method for assigning ID numbers to separate areas. We first describe the proposed processes and then demonstrate them with examples. The proposed process is as follows:

A. Allot ID numbers to all areas based on the segmentation results.
B. Detect divided areas.
C. Reassign ID numbers based on the results of B.
C-(1) Connect divided red areas.
C-(2) Connect separated branch regions using recognized shoot areas.
C-(3) Connect separated branch regions using unrecognized shoot areas.
D. Obtain the number of shoots for each branch.

A. Allot ID Numbers to all Areas Based on Segmentation Results

The aerial photograph of part of the vineyard is first segmented. Then, ID numbers are allotted to branch and shoot regions. A given region is assigned the same ID for adjacent pixels of the same color, and the neighbor list adds the shoot ID numbers adjacent to each branch ID number. The allotted branch ID numbers are given in red and the shoot ID numbers are given in yellow. Yellow areas that have fewer pixels than the threshold are changed to red. Red areas in contact with fewer yellow areas than the threshold are changed to yellow. Red areas not in contact with a yellow area are changed to gray. Figure 4 shows the results of assigning ID numbers to Figure 3.

B. Detect Divided Areas

The red region is enlarged by the thickness of the shoots and target areas are connected. If the pixels of the extended area are yellow, they are changed to orange. If they are gray, they are repainted blue. Figure 5 shows the diagram obtained after adding a connection target area in Figure 4.

C. Reassign ID Numbers Based on Results of B

1) Connect divided red areas: All branches in contact with each other via a blue area are reassigned to have the same ID. However, if a blue area is in contact with an orange area, it will not be a target. If a blue area is in contact with an orange area and also in contact with areas that are not in contact with the orange area, it will be a target. Figure 6 shows the diagram obtained after reassigning the same ID to selected areas relative to Figure 5.

2) Connect separated branch regions using recognized shoot areas: If there is an orange region between two branch ID numbers, the two branches areas are reassigned the same ID. This branch ID is excluded as a new tree adjacent ID.

3) Connect separated branch regions using unrecognized shoot areas: If the orange area is adjacent to more than three branch ID numbers, the user determines which two regions are the separated areas. At this time, the image of the vertical and horizontal threshold pixels is displayed as a pop-up window centered on the orange area, and the same branch area is selected, as shown in Figure 7. Then, the ID of the area determined to be the adjacent area is changed to the same ID. This branch ID is excluded as the newest adjacent ID. Figure 8 shows the diagram obtained after reassigning the same ID to selected areas relative to Figure 6.

D. Obtain Number of Shoots for Each Branch

All shoot ID numbers in contact with branch ID numbers are counted to determine the yellow area. The results are shown in Figure 9.

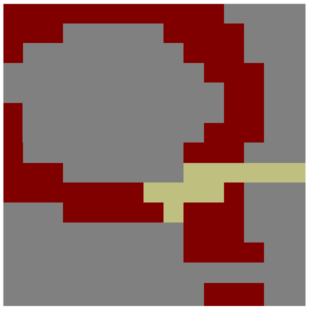

Figure 3. Base image.

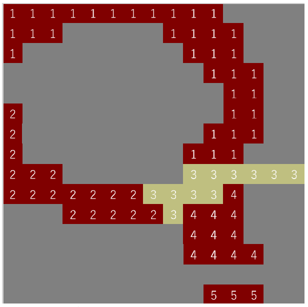

Figure 4. Process A.

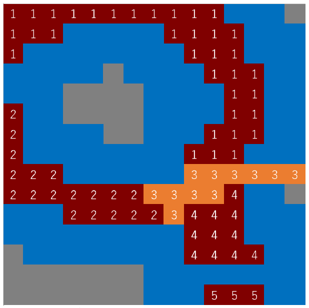

Figure 5. Process B.

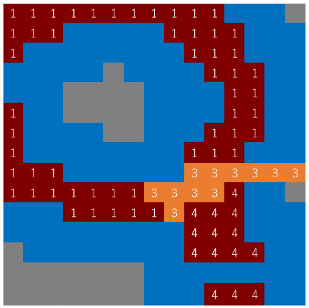

Figure 6. Process C-(1).

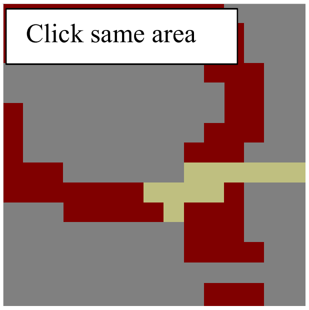

Figure 7. Pop-up Window.

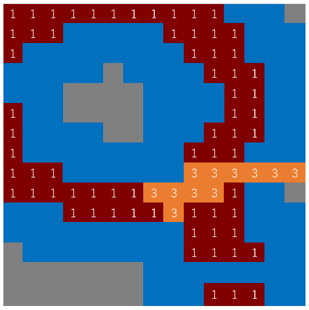

Figure 8. Process C-(3).

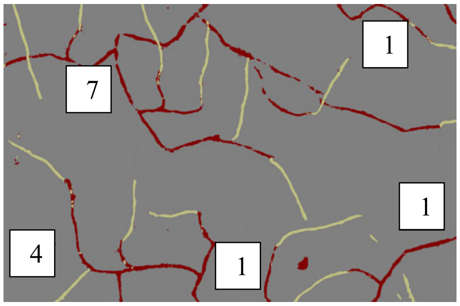

Figure 9. Count result.

ISBN: 978-1-61208-765-8

III. Validation of Proposed Concept

To implement the system, we confirmed the thresholds used in the proposed processes. We first obtained the size of the branch and shoot areas for repainting misrecognized areas. The results of the verification show that the red and yellow areas had at least around 3000 and 5000 pixels, respectively. The replacement target red and yellow areas had at most 1000 and 2100 pixels, respectively.

We also obtained the maximum width of the segmented shoot area for process B. The results of the verification show that the maximum width was 40 pixels. A threshold of 80 pixels also worked. The threshold thus has a wide range.

IV. Conclusion

This study proposed a method for counting the number of shoots on branches based on an aerial photograph taken by a drone. We used SegNet to detect and count shoots on a vine. However, this does not yield the correct number because shoots overlapped branches and thus the branch region was separated into several areas. To overcome this problem, we enlarged branch regions and detected the juncture branches. In future work, we will connect multiple photographs taken by a drone to count all shoots in a vineyard.

References

[1] http://www.miraikikin.org/activities/agriculture/rihoku.html [retrieved: Februaty 2020]

[2] V. Badrinarayanan et al., :”SegNet :A Deep Convolutional Encoder- Decoder Architecture for Image Segmentation.”, IEEE trans. on PAMI, vol. 39, pp. 2481-2495, 2017. [retrieved: February, 2020]

[3] O. Ronneberger, P. Fischer and T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, arXiv:1505.04597, 2015. [retrieved: February, 2020]

[4] J. Fu et al., Dual Attention Network for Scene Segmentation, arXiv:1809.02983v4, 2019 [retrieved: February, 2020]

[5] Y. Yuan and J. Wang, OCNet: Object Context Network for Scene Parsing, arXiv:1809.00916v3, 2019[retrieved: February, 2020]

ISBN: 978-1-61208-765-8