Segmentation Methods Evaluation on Grapevine Leaf Diseases

Szilárd Molnár
Automation Department
Technical University of Cluj-Napoca
Romania
Email: Szilard.Molnar@aut.utcluj.ro

Levente Tamás
Automation Department
Technical University of Cluj-Napoca
Romania
Email: Levente.Tamas@aut.utcluj.ro

Abstract—The problem of vine disease detection (VDD) was addressed in a number of research papers, however, a generic solution is not yet available for this task in the community. The region of interest segmentation and object detection tasks are often complementary. A similar situation is encountered in VDD applications as well, in which crop or leaf detection can be done via instance segmentation techniques as well. The focus of this work is to validate the most suitable methods from the main literature on vine leaf segmentation and disease detection on a custom dataset containing leaves both from the laboratory environment and cropped from images in the field. We tested five promising methods including the Otsu’s thresholding, Mask R-CNN, MobileNet, SegNet, and Feature Pyramid Network variants. The results of the comparison are available in Table I summarizing the accuracy and runtime of different methods.

I. INTRODUCTION

VINE DISEASE DETECTION plays an important role in the overall vineyard management allowing the loss reduction and the overcome of the pesticide overuse. The early stage VDD allows the degree of contamination reduction, which implicitly implies a positive economic impact as well.

Remote sensing plays an important role in precision agriculture, allowing the detection of different diseases, estimation of yield, or the computation of the fertilizer rates [29]. With the widespread of Unmanned Aerial Vehicles (UAV) in agriculture as well, close-range remote sensing expanded the range of applications for precision agriculture. The classical image processing algorithms were replaced by deep learning-based variants also for segmentation and object detection.

The most currently available solutions based on convolutional neural networks (CNN) are based on a sliding window approach, which allows the operations on smaller-sized image patches in favor of computational speed. However, for segmentation and detection purposes the whole image view could improve the segmentation boundaries and the accuracy of the detection.

In this paper, we propose to compare the already existing segmentation methods for masking diseased spots on grapevine leaves. For this, we create a mixture of datasets, which contains images from a laboratory environment as well as leaves cropped from images captured in the field from proprietary and publicly available datasets. Our proprietary dataset is captured with a mid-range commercial drone at low-altitude flight using a high-resolution (4K) camera.

The main contribution of this paper is the overview of the existing methods for this particular scenario with close-range remote sensing and the conclusions of the experimental finding in challenging datasets from various vineyards. The paper is organized as follows: the state of the art is presented in Section II, the dataset and method in Section III, and the comparison of the methods in Section IV.

II. STATE OF THE ART

Being an important aspect of precision viticulture, disease detection has a wide range of solutions in the literature. Many researchers seek a new way to stop the spread of diseases as early as possible, to reduce the chances of plant disposal and decreased quality. As far as the domain, multiple approaches exist. The first way to compare these approaches is to specify if the used images are from a laboratory environment or from the field. The approaches focusing on field image processing can be further split into proximal sensing, mainly using a conventional RGB camera, and remote sensing, using a variety of different mediums, such as RGB, multispectral, or hyperspectral. In this section, we provide a brief overview of the existing disease detection methods.

Cruz et al. [8] use transfer learning to detect grapevine yellow disease on single-leaf images while comparing multiple architectures. They experiment with numerous architectures, to conclude that ResNet-50 [26] has the best accuracy-to-complexity ratio.

Similarly, Liu et al. [16] detect grapevine diseases using images of grapevine leaves. The images are either from a laboratory or from the field, however, an image contains only one leaf in both cases. The different leaf sizes are resolved using dense inception convolutional neural network from GoogLeNet [26] and asymmetric factorization approach [27].

While Gutiérrez et al. [10] capture their images in the field, they manually segment their images, to contain only one leaf, which either represents downy mildew and spider mite...
symptoms. The RGB data is converted into HSV color space. The authors claim this color space change ensures robustness for their hue thresholding-based method.


Mousavi and Farahani [21] base their work on the mixture of VGG16 [25] and Faster R-CNN [23]. This method captures images of grapevines using a drone, however, the leaves are individually segmented before disease detection and localization.

Although all of these methods detect diseases, they do not create a binary mask to segment the diseased spots on the leaves. One example of this can be the work of Abdelghafour et al. [2], who detect downy mildew by capturing the images using a high-power flashlight, similar to Liu et al. [17], which causes an instantaneous segmentation, then converting the images into L*a*b color space. Local structure tensor [14] is used to extract geometric features.

III. MATERIAL AND METHODS

In this section, we provide a brief description of the used datasets, and methods.

A. Datasets

Data is a highly valuable asset in computer vision. It is used to calibrate and evaluate the model, therefore we need a dataset with high variability. In this section, we describe the used datasets.

As the primary dataset, we use the PlantVillage dataset created by Hughes et al. [13], with the codename: PV_data. Other versions of this dataset also exist, for example by Cruz et al. [8], however, ultimately we chose the one available on GitHub\(^1\), because in this case the background of the images is already blackened, Figure 1, unlike other versions, where the background is a gray table surface.

Additionally, we create an infeld-dataset, which contains cropped images from vineyards from various locations. This ensures a wide variety of camera angles and lighting conditions. The first two such datasets are the ones we have access to, each of them located in Romania, courtesy of the University of Agricultural Sciences and Veterinary Medicine. Our main vineyard is located in Cluj-Napoca (codename: Cj_data), then less data is from Apoldu de Sus (codename: Ap_data). These images are captured using a DJI Mini 2 drone, using the onboard 4K camera.

The next dataset is from Abdelghafour et al. [1] (codename: Ab_data). This is a vineyard near Bordeaux. The uniqueness of this dataset is that while the images are captured from a camera mounted on a tractor, the creators use a high-power flashlight, Figure 3a. The result is a highly detailed canopy, with a dark, almost invisible background, all this with consistency, independently from weather or time of day.

The fifth dataset is created by Alessandri et al. [4] (codename: Al_data), using an Italian vineyard, focusing on leaves with esca disease, from different distances and angles, Figure 3b.

The last dataset is created by Casado et al. [7], named S3CavVineyardDataset (codename: S3_data), based on a swiss vineyard, Figure 3c. The images are perpendicular to the vines, captured from a tractor.

1) Data Organization: Since the task in this work is disease detection on single-leaf images, we need to have a ground truth mask for each image, which is created by us manually using GIMP [28].

From the dataset, we use 648 images of leaves with some sort of disease (black rot, esca, and grapevine yellow, or dry leaf), and 433 images of healthy leaves. Additionally, we crop leaves from other datasets: Cj_data, Al_data, Ap_data, and S3_data. We call this latter group infeld images, hence their background is not black, but the real environment, Figure 4. In the infeld group, 118 diseased leaves, and 100 healthy leaves are included. The task is disease detection, hence in the case of healthy images, the mask is just a black image, meaning that no diseased parts are present. The PV_data images are considered as group1, with an 80-20 train-test image ratio. The infeld images are considered group2 with

\(^1\)https://github.com/shreyansh-kothari/Grapes-Leaf-Disease-detection
a 20-80 train-test image ratio. We plan three test cases. In the first case, we train only on the images from group1 and test only on group1. In the second test, we train only on the images from group1 and test on group2. In the third test, we train on images from group1 and group2 and test on group2. All of these images are sized 255×255 pixels.

B. Methods

The disease detection task segments a region of interest, for this we choose both neural network-based methods, as well as a classical method to analyze their performance. We choose different architectures, to provide a wider analysis.

1) Mask R-CNN: The first machine learning algorithm that we include is the Mask R-CNN [11], which is used for precision viticulture by many researchers, for example, Ghiani et al. [9] and Santos et al. [24]. This is a well-known method, together with its other variants, such as Faster R-CNN ([23]). The base for implementing this method can be found at the link 2.

2) MobileNetV3: The idea of using MobileNetV3 [12] comes from Aghi et al. [3], who use it for canopy segmentation and row detection. The base for implementing this method is available 3. The main advantage of this model is its simplicity and lightness, making it more suitable for running on embedded devices.

3) Feature Pyramid Network: The Feature Pyramid Network FPN [15] architecture stands as a middle-ground between the lightness of MobileNetV3 and the accuracy of Mask R-CNN. We have seen the FPNs perform decently in surface normal estimation application [18], and canopy segmentation [19], since different support sizes are analogous on some levels to vine leaves. The base for implementing this method can be found at the link 4.

4) SegNet: As the name suggests, SegNet [5] is a neural network designed for segmentation. Similarly to Mask R-CNN, SegNet is also well-known and widely used. Since it is based on an encoder-decoder architecture, the latent space could be helpful in encoding the diseased parts. The base for implementing this method is available 5.

5) Otsu’s thresholding: Otsu’s thresholding [22] is a dynamic thresholding application, meaning that instead of choosing a static value, and masking the image according to this value, Otsu’s thresholding analyses each image, and chooses a thresholding value that is more decisive. The main drawback of this method is that despite the RGB color space using 3 channels, Otsu’s thresholding only works with monochromatic images. One solution would be to mimic the work of Abdelghafour et al. [2], who convert the input signal into HSV color space and apply Otsu’s thresholding only on the hue channel. However, because RGB does not have a hue value, we conduct a series of tests, to define the best solution. This phase is similar to the training phase in the case of a neural network since we use the training data for estimating an optimal set of parameters, which are later applied to the test data.

We run the thresholding method for each channel, which results in 3 binary masks. Then we combine these masks with each other, achieving a total of 7 masks. Then we do the same thing, but this time inverting the binary masks, since it is possible, that the region of interest might fall into the lower end of the thresholding. We compare the binary masks with the ground truth masks to determine the combination which gives us the best accuracy. Additionally, we create another set of estimation masks, where each individual channel is either inverted or not, depending on the previous results, and then combine these masks to determine the best combination. The ideal combination is noted for each case, and this parameter is used at the time of evaluation. Rather interestingly, from these initial tests, the optimal combination is between the red and blue channels, while the green channel results in slightly worse accuracy. The base for implementing this method is the OpenCV library [6].

IV. Evaluation

In this section, we show the results of the conducted tests. For each task, the accuracy is calculated on the percentage of the pixels correctly estimated, compared to the ground truth. At first sight, this task might seem trivial, because of the small images, yet, the shade difference and the varying spot shapes add a layer of complexity to it. As we described previously, we conduct 3 tests: 1) train on PV_data (864 images), test on PV_data (217 images); 2) train on PV_data (864 images), test on infield images (174 images); 3) train on PV_data with added infield images (908 images), test on infield images (174 images). The last test case is to see how much the accuracy rises by adding 5% more images from the test domain. Accuracy can be seen in Table I, and the range of false positives and false negatives in Table II.

From our tests, we can see that SegNet is not suitable for understanding healthy leaves, where it should not extract any region of interest, yet it does, which pulls back the performance by at least 20%. Furthermore, Otsu’s thresholding is extremely unstable. On the other hand, both of these methods are the fastest. On the first test, Mask R-CNN performs the best, although, it is the slowest, while MobileNetV3 and FPN...
TABLE I: Accuracy of the various methods for disease segmentation, including the runtime.

<table>
<thead>
<tr>
<th>Method</th>
<th>Test1[%]</th>
<th>Test2[%]</th>
<th>Test3[%]</th>
<th>Time[s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Otsu</td>
<td>62.5</td>
<td>46.7</td>
<td>46.7</td>
<td>0.0004</td>
</tr>
<tr>
<td>Mask R-CNN</td>
<td>93.64</td>
<td>61.14</td>
<td>86.49</td>
<td>0.160</td>
</tr>
<tr>
<td>MobileNetV3</td>
<td>83.76</td>
<td>81.89</td>
<td>82.97</td>
<td>0.088</td>
</tr>
<tr>
<td>FPN</td>
<td>90.52</td>
<td>50.24</td>
<td>85.57</td>
<td>0.015</td>
</tr>
<tr>
<td>SegNet</td>
<td>63.3</td>
<td>59.12</td>
<td>65.1</td>
<td>0.007</td>
</tr>
</tbody>
</table>

TABLE II: The approximate percentage of false positives and false negatives for the various methods for disease segmentation in the three test cases

<table>
<thead>
<tr>
<th>Method</th>
<th>FP(_1)</th>
<th>FN(_1)</th>
<th>FP(_2)</th>
<th>FN(_2)</th>
<th>FP(_3)</th>
<th>FN(_3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Otsu</td>
<td>35</td>
<td>2</td>
<td>48</td>
<td>6</td>
<td>48</td>
<td>6</td>
</tr>
<tr>
<td>Mask R-CNN</td>
<td>3</td>
<td>3</td>
<td>38</td>
<td>1</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>MobileNetV3</td>
<td>0</td>
<td>16</td>
<td>11</td>
<td>8</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>FPN</td>
<td>8</td>
<td>1</td>
<td>48</td>
<td>2</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>SegNet</td>
<td>35</td>
<td>2</td>
<td>40</td>
<td>0</td>
<td>51</td>
<td>4</td>
</tr>
</tbody>
</table>

Another aspect that we want to check is the amount of increase in accuracy if a few infeld images are added to the training. In the case of Otsu’s method, we find virtually no difference, while for the other methods, we see an increase in accuracy between 10-20%, which is significant for such little data. This test is an indication that it is worth pretraining a model with general images, from various grape leaves, and then training a few epochs with a few additional images from the domain of application. However, we think that in the case of MobileNetV3 we see an anomaly in the second test because the result is too accurate.

Additionally, we also observed that on average the number of false positives is higher for Otsu’s method, Mask R-CNN, FPN, and SegNet, while for MobileNetV3 the false negatives are higher. We generally prefer false positives, because in VDD an image flagged as infected should be further investigated by a specialist, therefore, be corrected, however, an infected leaf that is not flagged is unnoticed.

V. Conclusion

In this work, we compared the performance of existing segmentation algorithms from the state of the art for vine disease leaf segmentation and detection. Overall, the CNN-based methods performed well except for SegNet, while the Otsu’s thresholding gave poor results, even if it is the fastest method. We also proved that adding just a few images from the target domain to the general dataset, yields significantly better performance. While Mask R-CNN provides relatively good accuracy, the FPN-based method offers much faster execution without an increased loss of accuracy and an overall smaller memory footprint for the model. The latter aspect is relevant for the embedded implementation of the methods.

For future work, we would like to experiment with different color spaces, as the color spaces can affect the performance of the CNN methods. Although, our raw data is in RGB, a neural network could be capable of optimizing the data in a latent layer better than a simple color conversion.

The disease segmentation can be extrapolated on entire grapevine canopies, which removes the necessity for individual leaf extraction. Additionally, further tests should be done using more variable datasets, including synthetic datasets, and more vine species captured from different angles from different vineyards.

REFERENCES


