Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1081–1085
DOI: 10.15439/2023F7053
ISSN 2300-5963 ACSIS, Vol. 35

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.

[1]The authors are thankful for the support of Analog Devices Romania, for the equipment list and Nvidia for graphic cards offered as support to this work. This work was financially supported by the Romanian National Authority for Scientific Research, project nr. PN-III-P2-2.1-PED-2021-3120. The authors are also thankful to the Domus Foundation for their support.

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*

(a) *PV_data-disease* (b) Binary mask (c) *PV_data-healthy*

Fig. 1: Samples from the *PV_data* dataset.

(a) *Cj_data*-disease (b) *Cj_data*-healthy (c) *Ap_data*

Fig. 2: Samples from our dataset.

(a) *Ab_data* (b) *Al_data* (c) *S3_data*

Fig. 3: Samples from datasets: *Ab_data*, *Al_data*, *S3_data*

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.

Morellos et al. [20] detect (*esca* and *powdery mildew* using transfer learning. Comparing multiple architectures, *Inception v3* [27] provides the overall best classification accuracy.

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 *infield-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.

[1]https://github.com/shreyansh-kothari/Grapes-Leaf-Disease-detection

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 Alessandrini 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*, *Ab_data*, *Ap_data*, and *S3_data*. We call this latter group *infield* images, hence their background is not black, but the real environment, Figure 4. In the *infield* 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 *infield* images are considered *group2* with

(a) Diseased sample. (b) Binary mask. (c) Healthy sample.

Fig. 4: Samples cropping *infield* leaves for 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].

[2]https://github.com/matterport/Mask_RCNN
[3]https://github.com/MrD1360/deep_segmentation_vineyards_navigation
[4]https://github.com/molnarszilard/canopy_segmentation
[5]https://github.com/say4n/pytorch-segnet

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.

Method	Test1[%]	Test2[%]	Test3[%]	Time[s]
Otsu	62.4	46.7	46.7	**0.0004**
Mask R-CNN	**93.64**	61.04	*86.48*	0.160
MobileNetV3	83.76	**81.89**	82.97	0.088
FPN	90.52	50.24	85.57	0.015
SegNet	63.3	59.12	65.1	0.007

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

Method	FP_{t1}	FN_{t1}	FP_{t2}	FN_{t2}	FP_{t3}	FN_{t3}
Otsu	35	2	48	6	48	6
Mask R-CNN	3	3	38	1	10	3
MobileNetV3	0	16	11	8	3	14
FPN	8	1	48	2	4	12
SegNet	35	2	40	0	31	4

perform relatively well, in a much shorter time, which can be important for real-time applications on the field.

Another aspect that we want to check is the amount of increase in accuracy if a few *infield* 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

[1] Florent Abdelghafour, Barna Keresztes, Aymeric Deshayes, Christian Germain, and Jean-Pierre Da Costa. "An annotated image dataset of downy mildew symptoms on Merlot grape variety". In: *Data in Brief* 37 (2021), page 107250. DOI: https://doi.org/10.1016/j.dib.2021.107250.

[2] Florent Abdelghafour, Barna Keresztes, Christian Germain, and Jean-Pierre Da Costa. "In Field Detection of Downy Mildew Symptoms with Proximal Colour Imaging". In: *Sensors* 20.16 (2020), page 4380. DOI: 10.3390/s20164380.

[3] Diego Aghi, Simone Cerrato, Vittorio Mazzia, and Marcello Chiaberge. "Deep Semantic Segmentation at the Edge for Autonomous Navigation in Vineyard Rows". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - October 1, 2021*. IEEE, 2021, pages 3421–3428. DOI: 10.1109/IROS51168.2021.9635969.

[4] M. Alessandrini, R. Calero Fuentes Rivera, L. Falaschetti, D. Pau, V. Tomaselli, et al. "A grapevine leaves dataset for early detection and classification of esca disease in vineyards through machine learning". In: *Data in Brief* 35 (2021), page 106809. DOI: https://doi.org/10.1016/j.dib.2021.106809.

[5] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (2017), pages 2481–2495. DOI: 10.1109/TPAMI.2016.2644615.

[6] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).

[7] A. Casado-García, J. Heras, A. Milella, and R. Marani. "Semi-supervised deep learning and low-cost cameras for the semantic segmentation of natural images in viticulture". In: *Precision Agriculture* (2022), pages 1–26. DOI: 10.1007/s11119-022-09929-9.

[8] Alberto Cruz, Yiannis Ampatzidis, Roberto Pierro, Alberto Materazzi, Alessandra Panattoni, et al. "Detection of grapevine yellows symptoms in *Vitis vinifera* L. with artificial intelligence". In: *Computers and Electronics in Agriculture* 157 (2019), pages 63–76. DOI: 10.1016/j.compag.2018.12.028.

[9] Luca Ghiani, Alberto Sassu, Francesca Palumbo, Luca Mercenaro, and Filippo Gambella. "In-Field Automatic Detection of Grape Bunches under a Totally Uncontrolled Environment". In: *Sensors* 21.11 (2021), page 3908. DOI: 10.3390/s21113908.

[10] Salvador Gutiérrez, Inés Hernández, Sara Ceballos, Ignacio Barrio, Ana M. Díez-Navajas, et al. "Deep learning for the differentiation of downy mildew and spider mite in grapevine under field conditions". In: *Computers and Electronics in Agriculture* 182 (2021), page 105991. DOI: 10.1016/j.compag.2021.105991.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. "Mask R-CNN". In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017, pages 2980–2988. DOI: 10.1109/ICCV.2017.322.

[12] Andrew Howard, Ruoming Pang, Hartwig Adam, Quoc V. Le, Mark Sandler, et al. "Searching for MobileNetV3". In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pages 1314–1324. DOI: 10.1109/ICCV.2019.00140.

[13] David P. Hughes and Marcel Salathé. "An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing". In: *Computing Research Repository* abs/1511.08060 (2015).

[14] Hans Knutsson, Carl-Fredrik Westin, and Mats T. Andersson. "Representing Local Structure Using Tensors II". In: *Image Analysis - 17th Scandinavian Conference, SCIA 2011, Ystad, Sweden, May 2011. Proceedings*. Edited by Anders Heyden and Fredrik Kahl. Volume 6688. Lecture Notes in Computer Science. Springer, 2011, pages 545–556. DOI: 10.1007/978-3-642-21227-7_51.

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, et al. "Feature Pyramid Networks for Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pages 2117–2125. DOI: 10.1109/CVPR.2017.106.

[16] Bin Liu, Zefeng Ding, Liangliang Tian, Dongjian He, Shuqin Li, et al. "Grape Leaf Disease Identification Using Improved Deep Convolutional Neural Networks". In: *Frontiers in Plant Science* 11 (2020), page 1082. DOI: 10.3389/fpls.2020.01082.

[17] Ertai Liu, Kaitlin M. Gold, David Combs, Lance Cadle-Davidson, and Yu Jiang. "Deep semantic segmentation for the quantification of grape foliar diseases in the vineyard". In: *Frontiers in Plant Science* 13 (2022), page 978761. DOI: 10.3389/fpls.2022.978761.

[18] Szilárd Molnár, Benjamin Kelényi, and Levente Tamás. "Feature Pyramid Network Based Efficient Normal Estimation and Filtering for Time-of-Flight Depth Cameras". In: *Sensors* 21.18 (2021), page 6257. DOI: 10.3390/s21186257.

[19] Szilárd Molnár, Barna Keresztes, and Levente Tamás. "Feature Pyramid Network based Proximal Vine Canopy Segmentation". In: *IFAC-PapersOnLine* (2023).

[20] Antonios Morellos, Xanthoula Eirini Pantazi, Charalampos Paraskevas, and Dimitrios Moshou. "Comparison of Deep Neural Networks in Detecting Field Grapevine Diseases Using Transfer Learning". In: *Remote Sensing* 14.18 (2022), page 4648. DOI: 10.3390/rs14184648.

[21] Seyed Amirhossein Mousavi and Gholamreza Farahani. "A Novel Enhanced VGG16 Model to Tackle Grapevine Leaves Diseases With Automatic Method". In: *IEEE Access* 10 (2022), pages 111564–111578. DOI: 10.1109/ACCESS.2022.3215639.

[22] Nobuyuki Otsu. "A Threshold Selection Method from Gray-Level Histograms". In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (1979), pages 62–66. DOI: 10.1109/TSMC.1979.4310076.

[23] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2017), pages 1137–1149. DOI: 10.1109/TPAMI.2016.2577031.

[24] Thiago T. Santos, Leonardo L. de Souza, Andreza A. dos Santos, and Sandra Avila. "Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association". In: *Computers and Electronics in Agriculture* 170 (2020), page 105247. DOI: 10.1016/j.compag.2020.105247.

[25] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Edited by Yoshua Bengio and Yann LeCun. Association for Computing Machinery, 2015.

[26] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, et al. "Going deeper with convolutions". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 2015, pages 1–9. DOI: 10.1109/CVPR.2015.7298594.

[27] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. "Rethinking the Inception Architecture for Computer Vision". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pages 2818–2826. DOI: 10.1109/CVPR.2016.308.

[28] The GIMP Development Team. *GIMP*. Version 2.10.12. June 12, 2019. URL: https://www.gimp.org.

[29] M. Weiss, F. Jacob, and G. Duveiller. "Remote sensing for agricultural applications: A meta-review". In: *Remote Sensing of Environment* 236 (2020), page 111402. DOI: https://doi.org/10.1016/j.rse.2019.111402.