UAV IMAGES AND DEEP-LEARNING ALGORITHMS FOR DETECTING FLAVESCENCE DOREE DISEASE IN GRAPEVINE ORCHARDS

M. A. Musci [1,2]*, C. Persello [3], A. M. Lingua[1,2]

[1] DIATI, Politecnico di Torino, 10124 Torino, Italy, (mariaangela.musci, andrea.lingua) @polito.it
[2] PIC4SeR, Politecnico di Torino Interdepartmental Centre for Service Robotics, Torino, Italy
[3] Department of Earth Observation Science, Faculty ITC, University of Twente, 7514AE Enschede, The Netherlands, c.persello@utwente.nl

Commission TCIII-IVc

KEYWORDS: Precision viticulture; Unmanned Aerial Vehicle (UAV); Flavescence dorée grapevine disease; Object Detection; Deep-Learning; Faster R-CNN

ABSTRACT:

One of the major challenges in precision viticulture in Europe is the detection and mapping of *flavescence dorée* (FD) grapevine disease to monitor and contain its spread. The lack of effective cures and the need for sustainable preventive measures are nowadays crucial issues. Insecticides and the plants uprooting are commonly employed to withhold disease infection, even if these solutions imply serious economic consequences and a strong environmental impact. The development of a rapid strategy to identify the disease is required to cover large portions of the crop and thus to limit damages in a time-effective way. This paper investigates the use of Unmanned Aerial Vehicles (UAVs), a cost-effective approach to early detection of diseased areas. We address this task with an object detection deep network, Faster R-CNN, instead of a traditional pixel-wise classifier. This work tests Faster R-CNN performance on this specific application through a comparative analysis with a pixel-wise classification algorithm (Random Forest). To take advantage of the full image resolution, the experimental analysis is performed using the original UAV imagery acquired in real conditions (instead of the derived orthomosaic). The first result of this paper is the definition of a new dataset for FD disease identification by UAV original imagery at the canopy scale. Moreover, we demonstrate the feasibility of applying Faster-R-CNN as a quasi-real-time alternative solution to semantic segmentation. The trained Faster-R-CNN achieved an average precision of 82% on the test set.

1. INTRODUCTION

Viticulture has great economic relevance in the EU (European Union) agriculture; however, it is also the sector with the highest pesticide use that has consequences such as chemical pollution and soil contamination. The insecticide adoption is caused by the presence of some diseases that constantly interfere with the production and influence grape yield both in terms of quantity and quality (Matese and Gennaro, 2015; Mazzetto et al., 2010; Micheloni, 2017). Among the vineyard diseases with heavy impact in the EU, Micheloni, 2017 includes the *Flavescence dorée* (FD). FD is a grapevine disease caused by a bacteria transmitted in the field by the leafhopper Scaphoideus titanus Ball; it is included in the A2 EPPO list (EC directive no. 2009/297EC) as a quarantined organism (Chuche and Thiéry, 2014). The identification of the FD disease area is currently based on visual inspection in the field and laboratory analysis carried out by teams of experts. Since this approach is time-consuming, especially for wide fields, the definition of rapid methods is the main challenge in this application field (Hruška et al., 2019).

In this context, UAV (Unmanned Aerial Vehicle) imagery combined with machine learning algorithms has shown great potentials for rapid, non-destructive, cost-effective detection, and localization of the disease. Indeed, this approach allows monitoring large areas, detecting biophysical, biochemical, and optical property changes of tissues and leaves with automatic analysis. However, the development of automatic algorithms is still in an early stage in this application field due to the lack of consistent datasets with a large number of labeled images acquired in real conditions (different lighting conditions, points of view, and with an inconsistent background) and due to the small size of the disease spots.

Previous works addressed the detection problem with semantic segmentation techniques, which perform a pixel-wise image labeling. Machine learning algorithms, such as random forest, support vector machines, and deep learning techniques such as CNNs (e.g., AlexNET, ResNet, U-Net) have been employed for this purpose (Cruz et al. 2019; Rançon et al. 2019). The algorithms cited above were used to analyze the problem at different levels of detail (from leaf to canopy scale). Previous studies used mainly leaf disease datasets detecting the disease on the individual leaves, and when UAV images were adopted, the detection was carried out on the orthomosaic (Albetis et al. 2017, Kerkech, Hafiane, and Canals 2018). However, the mosaicking procedure performed by photogrammetric software can produce geometric artifacts, which can lead to problems in the identification of the disease. The 3D canopy reconstruction and orthomosaic generation are affected by the difficulties of feature extraction algorithms to cope with the similarity between grape leaves and the grass on the ground and the different position of a single leaf in different views due to possible presence of the wind. Figure 1 shows an example of the orthomosaic and original image ground truth. On the left, due to the geometric artifacts and the homogeneity with the background, it is impossible to identify the FD in the red bounding box.

The main limitations of available datasets for this application, as underlined by Arsenovic et al., 2019 are related to images

* Corresponding author

subjects and scale. In literature, indeed, the datasets are composed of images acquired in the laboratory with uniform background and objects on a big scale (leaves scale). To overcome these limitations, this study proposes to use the original UAV imagery acquired in real conditions to take advantage of the full image resolution.
A dataset of UAV original imagery for the FD diseased plant is prepared with images of the diseased plant acquired in real condition at the canopy scale.
In this study, FD detection is addressed with an object detection network, i.e., Faster R-CNN (Ren et al., 2016), instead of a traditional pixel-wise classifier. Faster R-CNN is selected because of its capability to deal with objects of various sizes at multiple scales. Faster R-CNN is composed of two different modules: a deep convolutional neural network that proposes regions and a Fast-R-CNN module that uses the previous regions to predict bounding boxes and class labels.
The class labels output needs to distinguish multiple objects in images. It becomes useless information, instead, for single object detection, because only one class has to be identified.
The paper aims to compere Faster RCNN with a classification algorithm such as Random Forest and point out the main differences between the two strategies.

Figure 1. On the left a patch of the orthomosaic and on the right an original oblique image of the same area

2. METHODS

A binary classification task is applied for testing the two different approaches: object detection and semantic segmentation. In both cases, two classes are considered: FD diseased plants and background. The background class includes all the other objects that it is possible to identify in the environment such as healthy grapevine, terrain, poles, buildings, and other species of vegetation. The dataset is built by collecting and annotating UAV original images. Then, the dataset is divided into a training and a test set. After that Faster R-CNN and Radom Forest hyperparameters are tuned. Finally, the algorithms are tested for addressing FD diseased detection. In this section, and Random Forest algorithm (2.1) and Faster RCNN architecture (2.2) are briefly described. Moreover, evaluation metrics for semantic segmentation and object detection are presented (2.3.1 and 2.3.2.).

2.1 Semantic Segmentation: Random Forest

Random Forest (RF) algorithm combines multi-decision trees that operate as an ensemble trained with a bagging mechanism (Breiman, 2001). The bagging mechanism samples N random bootstraps of the training set with replacement. The higher number of trees makes the algorithms more accurate than a simple decision tree (Zhang and Ma, 2012). Random Forest is based on the binary recursive partitioning trees using individual variables. In the classification task, the typical criterion for node splitting is the GINI index (Q) (1):

$$Q = \sum_{k \neq k'}^{k} \hat{p}_k \ \hat{p}_{k'} \quad (1)$$

Where p_k is the proportion of class k observations in the node as defined in (2):

$$\hat{p}_k = \frac{1}{n} \sum_{i=1}^{n} I(y_i = k) \quad (2)$$

The GINI index measures the "purity" of classification at a node. Large values of a GINI index represents an impure node. According to the splitting criteria, a candidate split creates two descendant nodes and the splitting is chosen to minimize the following (3):

$$Q_{split} = n_L Q_L + n_R Q_R \quad (3)$$

Where Q_L and Q_R are the two descendants and n_L and n_R are the sample size. The trees grow without pruning until the terminal node.

2.2 Object detection: Faster R-CNN

Faster R-CNN framework is a two-stage detector and is composed of three components: the backbone convolutional network (e.g. ResNET, AlexNET), a Region proposal Network (RPN), and Fast R-CNN detector (Ren et al., 2016). Figure 2 shows a generic RPN-based architecture for generic object detection.
The backbone is a feature extraction network, pretrained in standard practice. In the first stage, the RPN, a deep fully convolutional network, predicts object locations and scores at the same time. In the second stage, Fast R-CNN handles region detection.
The input of the backbone is an image that is resized: the height is 600 pixels and the width is not exceeding 1024. The backbone output features (H × W) are smaller than the original image depending on the backbone stride. In Liu et al., 2019, the stride is 16.
RPN has to identify the object and discard the background. Then, for each object, RPN has to learn the location and the estimated size. To achieve these goals, it uses the last layer of feature map extracted by the CNN backbone and for each location in the feature map initializes k reference boxes, called anchors. Anchors indicates possible objects in a defined location with different scales (area of the bounding box) and aspect ratio (H/W). The scale and the aspect ratio sets allow dealing with different shapes and scales of the detection window (Liu et al., 2019).

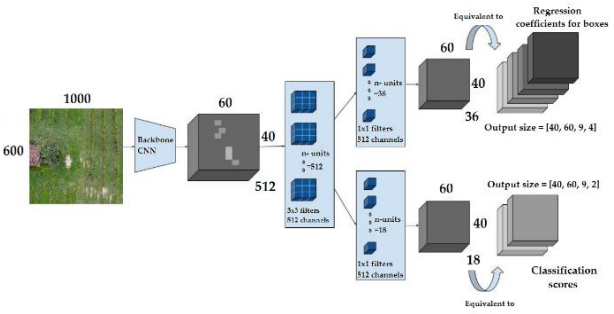

Figure 2. The Region Proposal Network (RPN) architecture (adapted from Liu et al., 2019)

The anchor box number k is defined considering the possible combinations of scales and aspect ratio (Figure 3). For a set of 3 scales and 3 aspect ratio, 9 anchors box is used.

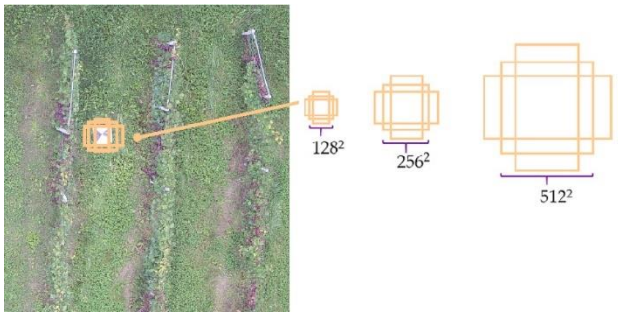

Figure 3. Example of anchors on real case images on the left. On the right, anchor scales (128^2, 256^2, 512^2) and aspect ratios (1:1- squared shape, 1:2- horizontal rectangular shape and 2:1- vertical rectangular shape) for the PASCAL challenge.

Each anchor is mapped with an objectness score to a lower-dimensional vector. The objectness score indicates the membership to a set of object classes (s_{obj}) versus background (s_b). The positive score is assigned according to two different conditions: (i) the highest Intersection-over-Union (IoU) overlap with a ground-truth or (ii) an IoU overlap higher than a threshold (in literature it is set to 0.7) with any ground-truth. The negative label is assigned to a non-positive anchor with an IoU less than a threshold (0.3). Moreover, it is possible that for a single ground-truth box, positive labels are assigned to multiple anchors, thus A Non-Maximum Suppression algorithm (NMS) is applied to reduce the redundancy of the anchor. It uses the Intersection of Union (IoU) between each proposal and the most likely proposal. The IoU values have to be greater than a threshold (0.7) to select the ROIs with the highest probability to contain an object.
After defined the object proposal, a 3×3 convolutional layer with 512 units is applied to return a 512-d feature map for every location. The output of this last step is fed into two sibling fully-connected layers which are 1×1 convolution layer with 18 units for object classification and 1×1 convolution with 36 units for bounding box regression. The classification branch gives an output of size (H×W×18) and indicates, for each feature map point, the probability to contain an object within all k anchor boxes (confidence score). The regression branch gives and output of size (H×W×36) and indicates bounding box coordinates.
The Faster-RCNN, as defined also for the Fast RCNN (Girshick, 2015), loss function combines the losses of classification and bounding box regression as defined in (4):

$$L(\{pi\},\{ti\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{box}}\sum_i p_i^*\ L_{reg}(t_i, t_i^*) \quad (4)$$

where

i = index of an anchor
N_cls, Greg and λ the normalization terms and the weight respectively

L_{cls}, as defined in (5), is the log loss function over two classes that are object and background in a binary case.

$$L_{cls}(p_i, p_i^*) = -p_i^* log, p_i - (1-p_i^*)(1 - log, p_i) \quad (5)$$

where p_i = predicted probability of anchor I being an object.
p_i^* = ground truth object label (1 for an object, 0 for not object)

L_{reg} is defined as (6). The regression loss is activated only for positive anchors.

$$L_{reg}(t_i, t_i^*) = R\ (t_i - t_i^*) \quad (6)$$

where

t_i = vector of 4 parametrized coordinates of the predicted box
t_i^* = ground truth box coordinate
R= robust loss function (smooth L_1)

2.3 Evaluation Metrics

2.3.1 Segmentation

To characterize the performance of semantic segmentation: out of bag score and testing accuracy have been selected. The out-of-bag score (OOB) is computed as the number of a correctly predicted sample from the out of bag samples. The out-of-bag samples are excluded from training observations and they allow to estimate the generalization of the model (Zhang and Ma, 2012). The accuracy measures the set of labels predicted for a sample that exactly match the corresponding set of ground truth labels.

2.3.2 Object Detection

Average precision (AP), recall, and Intersection-Over- Union are used as evaluation metrics for Faster RCNN. The Average Precision is defined as the mean precision at a set of eleven equally spaced recall levels [0,0.1,...,1] (7) (Everingham et al., 2010) :

$$\text{AP} = \frac{1}{11}\sum_{r\in\{0,0.1,\ldots,1\}} p_{interp}\,(r) \quad (7)$$

where p is the precision at each recall level r, interpolated by taking the maximum precision measured as in (8):

$$p_{inter} = \max p(r) \quad (8)$$

where p(r) is the measured precision at recall r. The case in which the bounding box sufficiently overlaps the ground truth is defined as true positives (TP). False Positive (FP) is the case in which the bounding box overlaps with the ground truth insufficiently. False negatives (FN) are the ground-truth that could not be detected. The IoU determines whether the proposed bounding box overlaps with the ground truth sufficiently and it is used to measure the accuracy of object detection. It is defined as (9) (Liu et al., 2019):

$$\text{IOU} = \frac{area\ (b\ \cap b^g)}{area\ (\ b \cup b^g)} \quad (9)$$

where b = predicted Bounding Box b
b^g = ground-truth

The IOU must be greater than a fixed threshold, typically set at 0.5 (50%).

3. CASE STUDY

For our experimental analysis, a vineyard case study was selected in Baldichieri d'Asti (Piedmont, Italy). The vineyard, surrounded by a dense forest, is composed of two types of grape (Fresia and Barbera) and it covers two hectares of a hilly area (Figure 4). The presence of several types of grapes implies different responses to the FD, detected through the color transformation of the leaves from green to dark red (Figure 6).

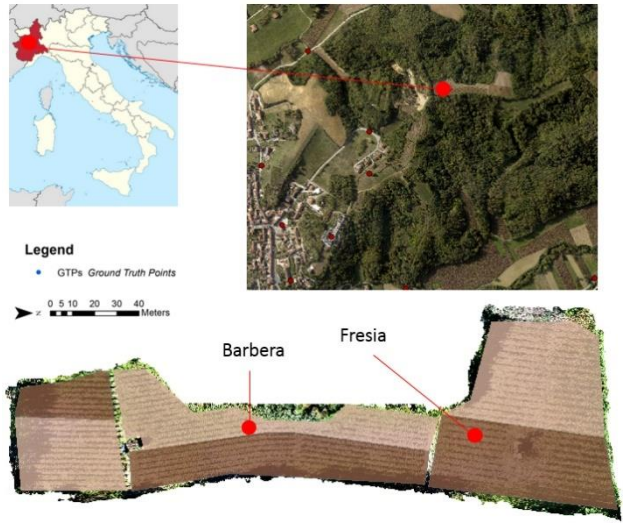

Figure 4. The case study area. On the left an overview of the position of the area. On the right, the map of the vineyard (in pink the Barbera grapes in brown the Fresia grapes)

Figure 5. Example of FD diseased plants. The FD diseased parts are reddish.

3.1 Data collection

To build a consistent dataset, a measurement campaign was carried out in September 2019, a period in which the symptoms of the disease are more evident. During this campaign, four acquisitions with different platforms, sensors (Table 1), flight altitude (20 and 25 m), and patterns (nadir and oblique) were carried out (Table 2). The UAV was chosen according to the size of the study area, flight time, and expected output of the survey. Based on this, it was used DJI Phantom 4 Pro and the DJI Matrice 210 v2. The flight height was determined and computed to obtain a centimeter resolution of the final model. Therefore, 522 images were acquired with a resolution of 4000×3000 pixels. To georeference the images, the coordinates of 16 plastic markers have been acquired using the Network Real-Time Kinematic (NRTK) Global Navigation Satellite System (GNSS) technique (Cina et al., 2015). The georeferencing procedure allows retrieving positions of the bounding boxes in a global reference system after the object detection procedure on the original images. Moreover, 20 well-distributed ground-truth points have been acquired adopting the same technique for an accurate and quick dataset sampling operation (Figure 6).

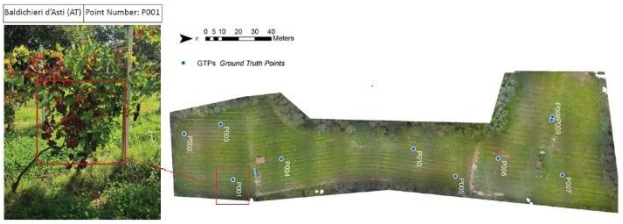

Figure 6. Example of ground truth point monography.

The NRTK survey was performed with a Trimble SP80 GNSS receiver, using the real-time correction of the permanent GNSS station in Canelli (AT).

Camera	RGB: FC330	DJI ZenMuse XT2
Sensors	1/2.3" CMOS	1/1.7" CMOS
Lens	FOV 94° 20 mm	FOV 57.12°× 42.44°
Focal length	8.8 mm	8 mm
Pixel size	2.4 µm	-
Dimensions	-	125.06 x 109.15 x 90.98 mm

Table 1. Sensors specifications.

Imaging Sensors	Flight Height (m) and schema	GSD (cm/pix)	N of images
RGB: FC330	20 nadir	0,70	584
RGB: FC330	20 oblique (45°)	1.40	322
RGB: FC330	25 nadir	255	255
ZenMuse XT2	25 nadir	0,65	255

Table 2. Acquisition configurations.

3.2 Grapevine orchard disease dataset and data annotation

A subset of 200 images was extracted from several acquisitions with different UAV platforms as described in 3.1. Indeed, the diseased grapevine dataset is built with images with different scales, conditions of lighting, points of view, and with an inconsistent background (Figure 7).

Figure 7. Example of original images obtained in different acquisition conditions.

In this way, we addressed the problem of insufficient training images enhancing, at the same time, the model generalization capability. Starting with the dataset of images, the images containing the FD diseased areas were split in overlapped tiles with a size of 1024x1024 pixels to avoid the resize that the Faster applies before the first step as described in section 2.2. Then all the images are manually annotated with bounding box and class. All images contain multiple diseased areas and all diseased areas are of the same size approximately. The annotation outputs were coordinates of bounding boxes of different sizes with their corresponding class.
The annotation process is different for the two algorithms. For the Random Forest, two classes were selected: FD diseased areas (FD) and No diseased FD area (NFD). In our case, NFD has included all the possible classes that it is possible to define in this environment.
However, for the Faster RCNN, only the FD diseased areas were annotated. The annotation operations were performed using LabelImage API open-source tool developed from MIT (Tzutalin, 2015). The labeling procedure was particularly difficult for the lack of shapes (the leaves shape is not evident at this scale). Thus the bounding boxes were digitized aiming at covering the small diseased areas in a precise way. It is evaluated also the possibility to define bigger bounding boxes, however, the presence of sparse green leaves does not allow to use it. Thus, 4575 polygons were manually annotated and exported in TFRecord format. This is a format for storing a sequence of binary records and a requirement for reading data efficiently (TensorFlow, 2020) As shown in Figure 8, the background prevailing on the diseased area is making the problem more challenging. To increase the accuracy, the detection task is addressed with images in which objects cover the main part of the image.

Figure 8. Data annotation examples for Random Forest. The blue boxes show the diseased areas of the vineyard, and pink boxes refer to the background.

3.3 Disease detection

After the data preparation, the images were randomly divided into a training set and test set as described in 3.2. For the Random Forest, 2360 polygons have been annotated and from them, 5000-pixel samples are randomly selected from different images for the training and 2000-pixel samples for the testing. Furthermore, because of the imbalance between FD disease class and the background, a class balance rectification was made. For the Faster R-CNN, 3660 and 915 polygons are chosen for training and testing the model, respectively. Moreover, the training and the validation set are split assuring that the images were equally distributed through the different types of images (Figure 8). Random forest and Faster R-CNN have been trained and evaluated on the same set of images. The two algorithms were experimentally compared to support the possibility to use an object detection approach. This task was implemented in Python programing language with the sklearn library for Random Forest (Pedregosa et al., 2011) and Tensorflow object detection API (Huang et al., 2017). The tests were performed on an Ubuntu workstation (18.04.4 LTS distribution) with an Intel(R) Xeon(R) CPU E5-1650 v4@3.60GHz (12 CPU with 6 cores per socket) and an NVIDIA GP102 -TitanX with 12 GB memory.

3.3.1 Random Forests: hyperparameter tuning and results

For the RF algorithm, we tuned two hyperparameters: the number of estimators (trees) and the number of features. The number of estimators, which is the maximum number of trees in the forest, was tuned according to the accuracy. For this application, the best fit is achieved with the number of estimators equal to 21. Moreover, the number of features to consider for the best split was set to all features, that in our case are three: Red, Green, Blue (RGB). According to a feature analysis importance, indeed, the RGB features are equally relevant. The RF test accuracy and the out-of-bag scores are 88% and 89%, respectively. The OOB score shows that even if the images with different scales, light conditions, the model achieved a promising level of generalization. From the qualitative point of view, as shown in Figure 9, the FD areas are well classified. Some false positive zone is located on the forest trees, that are characterized also by a red color. It is, however, possible that also the forest is affected by the disease.

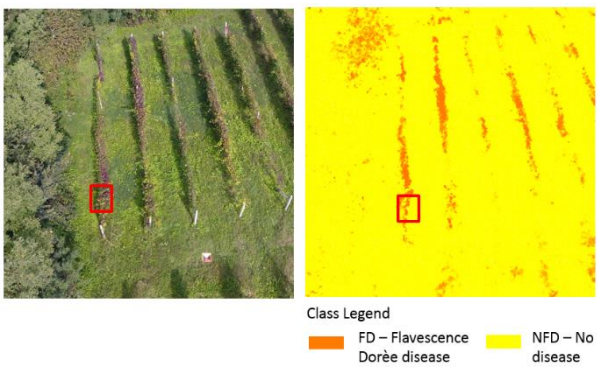

Figure 9. Semantic segmentation test result. On the left the original image, on the right the classified image. In the red box is underlined an FD diseased areas that are detected from RF.

3.3.2 Faster R- CNN: hyperparameter tuning and results

The experiments are performed based on the Faster R-CNN detector which employs as backbone ResNet-50 and ResNet-101. The model is pre-trained on ImageNet classification weight and fine-tuned on the FD dataset. The use of a pre-trained model, in this case, was necessary to reduce the computational time. In all the experiments, we trained both the RPN and Fast R-CNN branch. For the training optimization, the momentum was set to 0.89 and the learning rate was set to 0.001 according to the literature. Moreover, a maximum number of proposals per class (FD class) was decreased from 300 to 100 according to the number of bounding boxes annotated per images, which in the best case is 100 on average. For tuning the hyperparameters, 4 tests were performed as summarized in
Table 3.
The main tuned hyperparameters were: the anchor box scales and ratios, the IoU, and the number of training steps. These parameters were set according to the following considerations.

- Anchor boxes scale and ratio: two groups of parameters were used. The first has three scales [128^2,256^2,512^2] and three ratios: [0.5,1,2] and the second has four scales [0.25^2, 0.5^2, 1^2, 2^2] and three ratios [0.5,1,2]. The

first set was the starting point following the literature as described in 2.2. However, the second group was tuned according to the areas of the annotated bounding box in this application case study. Since the diseased plants/leaves could be classified as a small object due to the small scale, their areas are been estimated (between10^2 pixels and 50^2 pixels). The ratios instead have been maintained with the same value, because in the annotation shapes are still with the same ratios.

- IoU was set at 0.7 and 0.5. The intersection over Union was decreased compared with the literature, to increase the possibility to detect more objects. Indeed, with an IoU equal to 0.7 the half number of boxes was discarded.

- The training steps number was selected according to the loss function of the training dataset and the validation accuracy.

	Test	Anchor box (scale and ratios)	IoU	N of steps	AP@IoU50
Faster RCNN - ResNet -50	1	[128^2,256^2,512^2] [0.5, 1, 2]	0.7	60K	20%
	2	[0.25^2, 0.5^2, 1^2, 2^2] [0.5,1,2]	0.5	80K	65%
Faster RCNN - ResNet -101	3	[128^2,256^2,512^2] [0.5, 1, 2]	0.7	60K	40%
	4	[0.25^2, 0.5^2, 1^2, 2^2] ratio: [0.5,1,2]	0.5	80K	82%

Table 3. Test configurations. IoU stands for Intersection over Union and AP@50 is Average precision at 50% of Intersection over Union.

As Table 3 shows, the average precision for both tested architectures (Faster RCNN_ResNet 50 and Faster RCNN_ResNet 101) increases with smaller anchor boxes, because the area of the proposal and the ground truth is almost the same and the model is capable to learn the real size of objects. The number of detected instances and the associated confidence score increases with the accuracy percentage rising. In Figure 10, some visual test results are shown for Faster RCNN_ResNet 101. It is possible to notice that the confidence scores of the detected instances in most of the cases are greater than 80% for tests 2 and 4. The model achieved also a great generalization because it can find instances in all types of original images.
An analysis of the computing time shows that the Faster RCNN_ResNet 50 takes less time for the training compared to the Faster RCNN_ResNet 101. Indeed, the Faster RCNN_ResNet 50 trains with a speed of 0.3 sec/step, instead of 0.45 sec/step of the Faster RCNN_ResNet 10.

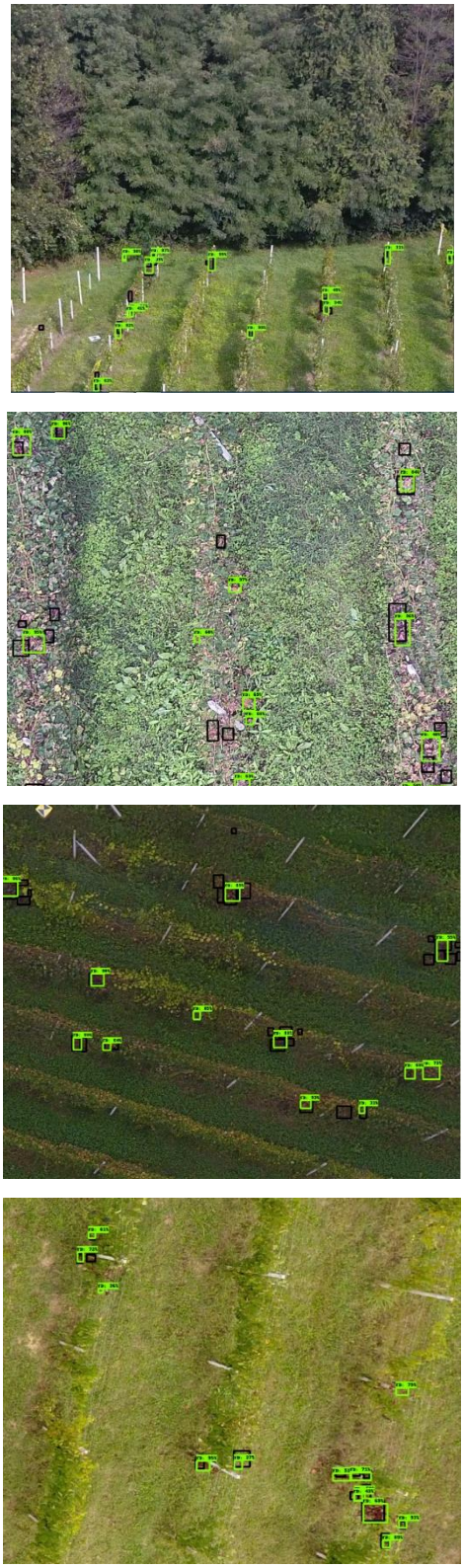

Figure 10. Object detection test results with the Faster RCNN_ResNet 101 architecture (test 4) on different types of images. In black, the reference data and in green the bounding box detected and the confidence score.

4. DISCUSSION

Since all the results related to the RF and Faster R CNN have been explained in the previous section, the focus of the discussion is the comparison between the two applied approaches: semantic segmentation and object detection approach. First, the two approaches present a different annotation process. For the RF, in a binary classification, all the possible classes in the image have to be annotated. For this reason, FD diseased area and all the other classes (buildings, trees, people, and streets) in the image were annotated. Instead, for the Faster- RCNN, only the FD diseased class was labeled. More effort has to be applied in the annotation process for the RF.

Moreover, as it has been underlined, regarding the achieved results, both algorithms show a good generalization of the model, despite the challenging task. The task can be considered difficult for three main reasons: the various nature of the dataset (images with different illumination conditions, scales, and prospectives), the sizes of the FD area at canopy scale, and the presence of similarity between plants and background.

Figure 9 and Figure 10 display the great difference between the two approaches: the semantic segmentation classified the whole images in FD class and background, instead the object detection localizes the diseased spots. Since the disease is localized in precise spots and on the plants, in most common cases, object detection can fit better to the task of avoiding the classification of useless classes.

5. CONCLUSION

In this paper, an approach to object detection for FD disease detection using UAV original images was tested. To address this purpose, UAV images have been collected during the most critical period for FD and fully annotated, because there is none ready-to-use dataset to face this specific task. To manage the FD detection in quasi-real-time at different scales, the Faster RCNN architecture was selected, because it is considered the trade-off between speed and accuracy in the categories of high-speed detection algorithms. The main challenges of this work were the hyperparameter tuning for the detection of small size objects and the mixture of provided images. Even if the image resolution in this application was 1 cm/ pixel in the worst case, the FD areas were still small compared to the prevalence of the background and it is not possible to identify the fixed shape (leaves or plant shape) at canopy scale. A large number of these architecture applications aim to detect objects with a shape such as people and building, etc. In this study case, only the radiometric information is used as features (reddish color of the leaves).

This experimental analysis demonstrates the feasibility of using this model for the proposed goal with an average precision value of 82%.

As future work, we plan to use multispectral and hyperspectral images. The use of spectral information could help both for better distinguishing diseased areas and background and identifying different stages of FD disease.

ACKNOWLEDGEMENTS

The study was carried out within the activities of the PoliTO Interdepartmental Centre for Service Robotics (PIC4SeR) and thanks to the concession of the “Azienda Agricola Ciabot” farm.

REFERENCES

Breiman, L., 2001. Random Forests. Machine Learning 45, 5–32. https://doi.org/10.1023/A:1010933404324

Chuche, J., Thiéry, D., 2014. Biology and ecology of the Flavescence dorée vector Scaphoideus titanus: a review. Agron. Sustain. Dev. 34, 381–403. https://doi.org/10.1007/s13593-014-0208-7

Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2010. The Pascal Visual Object Classes (VOC) Challenge. Int J Comput Vis 88, 303–338. https://doi.org/10.1007/s11263-009-0275-4

Girshick, R., 2015. Fast R-CNN. arXiv:1504.08083 [cs].

Hruška, J., Adão, T., Pádua, L., Guimarães, N., Peres, E., Morais, R., Sousa, J.J., 2019. Evaluation of machine learning techniques in vine leaves disease detection: a preliminary case study on Flavescence Dorée. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLII-3/W8, 151–156. https://doi.org/10.5194/isprs-archives-XLII-3-W8-151-2019

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K., 2017. Speed/accuracy trade-offs for modern convolutional object detectors. arXiv:1611.10012 [cs].

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M., 2019. Deep Learning for Generic Object Detection: A Survey. arXiv:1809.02165 [cs].

Matese, A., Gennaro, S.F.D., 2015. Technology in precision viticulture: a state of the art review [WWW Document]. International Journal of Wine Research. https://doi.org/10.2147/IJWR.S69405

Mazzetto, F., Calcante, A., Mena, A., Vercesi, A., 2010. Integration of optical and analogue sensors for monitoring canopy health and vigour in precision viticulture. Precision Agric 11, 636–649. https://doi.org/10.1007/s11119-010-9186-1

Micheloni, C., 2017. EIP-AGRI Focus Group . Diseases and pests in viticulture. EIP-AGRI Focus Group . Diseases and pests in viticulture 18.

Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. arXiv:1506.01497 [cs].

TensorFlow, 2020. . TensorFlow Core. URL https://www.tensorflow.org/tutorials/load_data/tfrecord?hl=it (accessed 5.4.20).

Tzutalin, L., 2015. LabelImg. Git code [WWW Document]. URL https://github.com/tzutalin/labelImg (accessed 5.4.20).

Zhang, C., Ma, Y. (Eds.), 2012. Ensemble Machine Learning. Springer US, Boston, MA. https://doi.org/10.1007/978-1-4419-9326-7