Vitis **58**, 95–104 (2019) DOI: 10.5073/vitis.2019.58.95-104

Efficient identification, localization and quantification of grapevine inflorescences and flowers in unprepared field images using Fully Convolutional Networks

R. Rudolph[1)], K. Herzog[2)], R. Töpfer[2)] and V. Steinhage[1)]

[1)] Department of Computer Science IV, University of Bonn, Bonn, Germany
[2)] Institute for Grapevine Breeding Geilweilerhof, Julius Kühn-Institut (JKI), Federal Research Centre for Cultivated Plants, Siebeldingen, Germany

Summary

Yield and its prediction is one of the most important tasks in grapevine breeding purposes and vineyard management. Commonly, this trait is estimated manually right before harvest by extrapolation, which mostly is labor-intensive, destructive and inaccurate. In the present study an automated image-based workflow was developed for quantifying inflorescences and single flowers in unprepared field images of grapevines, *i.e.* no artificial background or light was applied. It is a novel approach for non-invasive, inexpensive and objective phenotyping with high-throughput.

First, image regions depicting inflorescences were identified and localized. This was done by segmenting the images into the classes "inflorescence" and "non-inflorescence" using a Fully Convolutional Network (FCN). Efficient image segmentation hereby is the most challenging step regarding the small geometry and dense distribution of single flowers (several hundred single flowers per inflorescence), similar color of all plant organs in the fore- and background as well as the circumstance that only approximately 5 % of an image show inflorescences. The trained FCN achieved a mean Intersection Over Union (IOU) of 87.6 % on the test data set. Finally, single flowers were extracted from the "inflorescence"-areas using Circular Hough Transform. The flower extraction achieved a recall of 80.3 % and a precision of 70.7 % using the segmentation derived by the trained FCN model.

Summarized, the presented approach is a promising strategy in order to predict yield potential automatically in the earliest stage of grapevine development which is applicable for objective monitoring and evaluations of breeding material, genetic repositories or commercial vineyards.

K e y w o r d s : *Vitis vinifera* ssp. *vinifera*; BBCH 59; Convolutional Neural Network (CNN); computer-based phenotyping; semantic segmentation.

Introduction

Grape yield is one of the most important traits in the scope of grapevine breeding, breeding research and vineyard management (Molitor *et al.* 2012, Preszler *et al.* 2013, Töpfer and Eibach 2016, Simonneau *et al.* 2017). It is affected by genetic constitution of cultivars, training system, climatic conditions, soil and biotic stress (Bramley *et al.* 2011, Kraus *et al.* 2018, Howell 2001). Several prediction models recently published are often based on destructive, laborious measurements and extrapolations right before harvest (detailed overview is given by de la Fuente *et al.* 2015). For targeted vineyard management, i.e. yield adjustments due to bunch thinning, early yield predictions between fruit set and veraison (begin of grape ripening), are required in order to achieve well-balanced leaf-area-to-fruit-ratios, which are essential for maximized grape quality (Auzmendi and Holzapfel 2014, de la Fuente *et al.* 2015).

Flower development, flowering and fruit set rate are directly linked to the amount of yield and thus are promising traits for comparative studies (Petrie *et al.* 2005). In grapevine breeding programs and research, investigations regarding the flower abscission (i.e. level of coulure or fruit set rate) and its genetic, physiological and environmental reasons are of peculiar interest (Boss *et al.* 2003, Lebon *et al.* 2004, Marguerit *et al.* 2009, Giacomelli *et al.* 2013, Domingos *et al.* 2015). However, phenotyping of such small and finely structured traits is commonly done by visual estimations and thus achieve phenotyping scores that are more or less inaccurate and subjective, depending on the experience, awareness and condition of the employees. Currently, more accurate measurements require much more labor-intensive and partially destructive measurements, which preclude repetitive monitoring studies of several hundreds of different grapevine genotypes, e.g. crossing progenies (Giacomelli *et al.* 2013, Keller *et al.* 2010).

The application of fast imaging sensors facilitates multiple field screenings of large experimental plots, breeding populations and genetic repositories. In combination with efficient and automated data analysis, objective, precise and

Correspondence to: Dr. V. Steinhage, Department of Computer Science IV, University of Bonn, Endenicher Allee 19A, 53115 Bonn, Germany. E-mail: steinhage@cs.uni-bonn.de

comparable phenotypic data can be produced with minimal user interaction. Fast, inexpensive and simple-to-apply sensors, e.g. consumer cameras, are promising for cost-benefit and user friendly approaches. Recently, different sensor-based methods were developed for flower quantification based on images of individual captured grapevine inflorescences (Diago *et al.* 2014, Aquino *et al.* 2015 a, b, Millan *et al.* 2017, Liu *et al.* 2018). All of these approaches require images of a single inflorescence in front of well distinguishable backgrounds, i.e. artificial background or soil, which makes screenings of large numbers of plants more difficult and laborious.

Further, exertions of tractor-based approaches (Nuske *et al.* 2014) or other field phenotyping platforms (Kicherer *et al.* 2015, 2017, Aquino *et al.* 2018) are not feasible by analysis of individual plant organs. Regarding to this crucial restriction, a novel image analysis strategy for images of whole grapevine canopies without artificial background and additional light is required. Efficient identifying and localizing of the image regions that depict inflorescences hereby is the most challenging step regarding the small geometry and dense distribution of single flowers within inflorescences, the similar color of all plant organs in the fore- and background, as well as the circumstance that only approximately 5 % of the image area shows single flowers (Fig. 1).

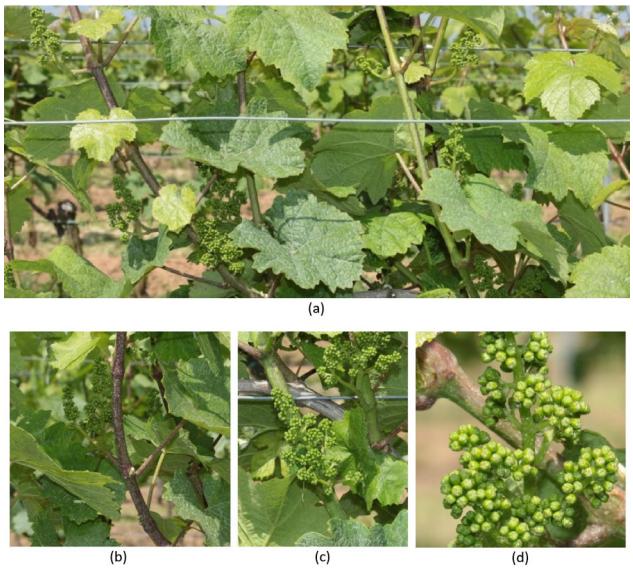

Fig. 1: Major challenges of unprepared grapevine images: Only 5 % of images are inflorescences and all plant organs are green (**a**); no standardized light conditions resulting in varying color characteristics of inflorescences, *i.e.* different green tones (**b**) (**c**); dense location of single flowers within one inflorescence (**d**).

In this study, the task of identifying and localizing the inflorescence areas was understood as a segmentation task, *i.e.* a task of partitioning the image into the classes "inflorescence" and "non-inflorescence" by assigning a class label to each individual pixel. While traditional approaches to image segmentation employ handcrafted heuristic criteria (e.g., intensity and color distributions) to identify appropriate image regions, deep learning convolutional neural networks (CNNs) allow learning descriptive criteria of the desired image regions just from the image data itself. By training a CNN on inflorescence segmentation data, a segmentation model able to deal with the complex scenes of images of whole grapevines was generated. CNNs have established themselves as a state-of-the-art method for many tasks of image processing, including image classification (Krizhevsky *et al.* 2012, Simonyan *et al.* 2014) as well as, more recently, image segmentation (Long *et al.* 2015, Ronneberger *et al.* 2015).

CNNs used for image classification classify complete images, e.g. showing cars, buildings, dogs, etc. and generally follow a common structure that shows two phases: the feature extraction phase and the classification phase. In the feature extraction phase multiple convolution layers and pooling layers generate successively more complex class characteristic image features (in the convolution layers) thereby downsampling the image size (in the pooling layers). In the classification phase multiple fully connected layers derive class labels based on the derived image features. CNNs for image segmentation generally implement a classification of each pixel in an image. Two approaches to CNN-based image segmentations are most important here:

Long *et al.* 2015 introduced the Fully Convolutional Networks (FCNs) for image segmentation. The architecture of a classification network is modified in a way that its fully connected layers for the complete image classification are replaced by multiple convolutional layers and decoder layers. In this network, the up-convolutional layers upsample the output size and the up-convolutional layers learn localization of class labels by combining the more precise high resolution features from layers of the extraction phase with the upsampled output. Due to upsampling, this part can increase the spatial resolution up to the input-dimensions, providing per-pixel information on the input image. Therefore, an FCN shows the following two phases: the feature extraction phase (as in the classification networks) followed by a decoder phase that results in a classification on the original image resolution, *i.e.* assigns a class label to each pixel of the image. This kind of network is also called encoder-decoder network: a given input image is encoded in terms of features at different scales in the first phase while the second phase decodes all these features and generates a segmentation of the image.

U-Net (Ronneberger *et al.* 2015) is a popular architecture of FCNs and can be trained end-to-end. This was not possible for the FCN approach presented by Long *et al.* 2015, which requires the encoder part to be pre-trained before being able to train the decoder part.

In this study, a U-net-like architecture of an FCN was used for the segmentation. The major objective of the present study was the development and validation of an automated image analysis workflow in order to quantify the number of single flowers per grapevine inflorescences in unprepared field images for non-invasive, inexpensive and objective phenotyping with high-throughput. The workflow of our approach shows four steps (Fig. 2):

1. Fast, inexpensive and simple-to-handle image taking with consumer camera.
2. Identification and localization of inflorescences employing FCN-based image segmentation.

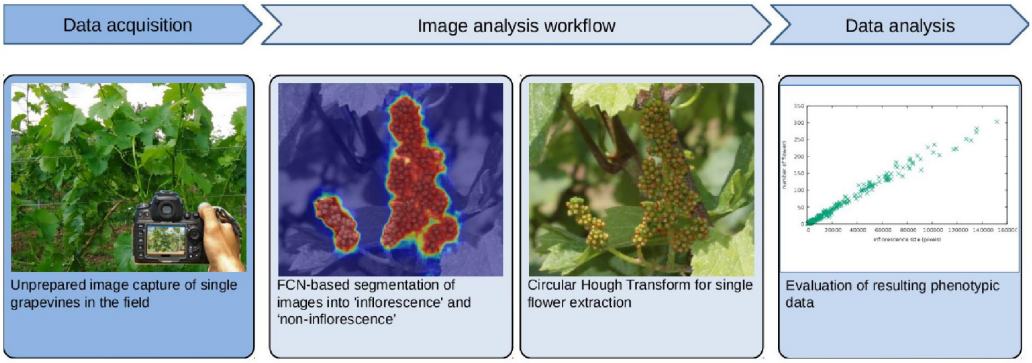

Fig. 2: Phenotyping workflow: Captured images of grapevines are analyzed by segmenting them into "inflorescence" and "non-inflorescence" *via* FCN and applying circle detection for flower extraction within the class "inflorescence". Finally, objective and precise phenotypic data are provided for further analysis.

3. Flower extraction by applying Circular Hough Transformation on segmented images.
4. Evaluation of resulting phenotypic data

Material and Methods

Plant material, image capture and pre-processing steps: For image capturing, *i.e.*, step 1 of our workflow (Fig. 2), a single-lens reflex (SLR) camera (Canon EOS 70D) and a focal length of 35 mm was used for image capture in the field under natural illumination conditions with manually controlled exposure. The distance to the plants was approximately 1 m.

108 field images of the *Vitis vinifera* ssp. *vinifera* 'Riesling' and 'Chardonnay' were captured at the end of May 2016 when plants reached the BBCH stage 59 (Biologische Bundesanstalt, Bundessortenamt und CHemische Industrie; Lorenz *et al.* 1995).

The field images were randomly divided into a training set (98 images) and an evaluation or test set (10 images). Both the training and evaluation set were manually annotated with bounding polygons around their inflorescences (Fig. 3). Additionally, the evaluation set was annotated with the center of each individual flower visible within the images. In order to reduce work load in the annotation process, the training set was less precisely annotated than the evaluation set. Since most of the image area shows non-inflorescence parts of plants, it was assumed that false positives would for the most part not be learned as positives during training.

Methodology: Step 2 of our workflow (Fig. 2) addresses the identification and localization of inflorescences in the image. This is done by applying a trained FCN, as introduced by Long *et al.* 2015 (see "ROI segmentatin", right column), to the input image. The FCN was trained on the annotated inflorescence segmentation data of the training set (see above "Image capture and pre-processing steps"). After training, the FCN is able to derive a segmentation of the input image, determining for each pixel whether it is part of an inflorescence, or if it can be ignored. Generally, the pixels depicting inflorescence form coherent regions. Since these regions are of interest for further processing (*i.e.* single flower extraction and deriving phenotyping data) these regions are called "regions of interest" (ROI). In Fig. 2 the result of an image segmentation is depicted in terms of a heat map where the red areas show the identified ROIs, *i.e.* the inflorescences.

Fig. 3: Annotation of an inflorescence using bounding polygons (fuchsia) and of single flowers using points (red) in the test set. The training set inflorescence polygons were annotated less precise in order to reduce workload.

In the third step of our workflow single flowers are extracted from all detected ROIs of an image. This was done by applying a Circular Hough Transform (CHT) on the image areas of the ROIs (see page 99 on flower extraction). For this study, the CHT was modified to consider the gradient direction, similar to the modification presented by Roscher *et al.* 2014.

ROI segmentation: Due to large areas of the image containing non-inflorescence parts of vines and varying lighting conditions throughout the images, filtering the image by the color of individual pixels, as it was done by Aquino *et al.* 2015 as a first step of finding inflorescences, would not produce reliable results.

Instead, we employ an U-net-like architecture, as proposed by Ronneberger *et al.* 2015, to identify and localize inflorescences in the image. The network architecture is based upon the AlexNet architecture (Krizhevsky *et al.* 2012) as an encoder part, with a short decoder part added to

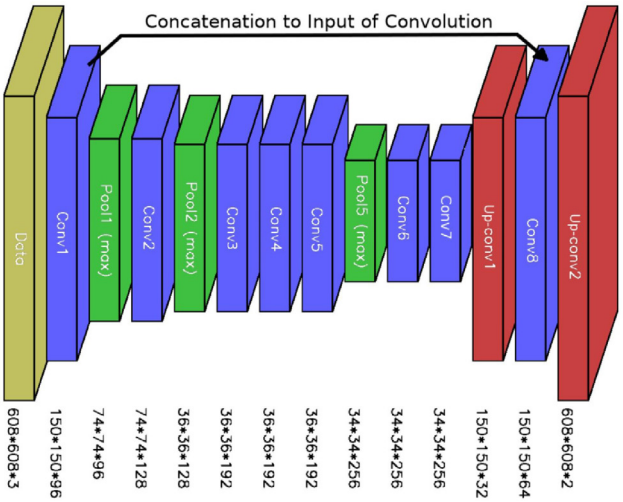

Fig. 4: The AlexNet-based FCN with up-convolution-based decoder part. Below spatial resolution and number of outputs are shown. The arrow denotes the concatenation of the channels of the outputs of Conv1 and Up−conv1, as input for Conv8. This is a visualization of the architecture presented in Tab. 1.

it. The architecture is visualized in Fig. 4, where each vertical blocks depicts layers of the full convolutional network. The first olive block labeled as "data" depicts input images of size 608 x 608 pixels with three layers for the RGB components. The second, blue block labeled as "Conv1" depicts 96 convolutional filters. Each filter encodes the image data in different types of trained features and shows an output size of 150 x 150 pixels. The third, green block labeled as Pool1 (max) depicts the downsampling of the output of the 96 layers convolutional filters down to 74 x 74 pixels. All following blue and green boxes up to the box labeled as "Conv7" depict convolutional and pooling layers that encode the input image successively in more and more complex and abstract feature representations, thereby downsampling the output size to 34 x 34 units. This downsampling part can be seen as the left (downgoing) part of the shape of the character "U". The following layers labeled as "Conv8", "Up-Conv1" and "Up-Conv2", respectively, show the upgoing part of the shape of the character "U". From this the name of the so-called U-net architecture was given by RONNEBERGER *et al.* 2015. The upgoing part fuses and upsamples all feature representations to the original image size, thereby deriving class labels (inflorescence / non-inflorescence) for all pixels. For the sake of completeness, Tab. 1 provides more detailed information for those who are familiar with convolutional networks and interested in the technical design of our U-net-based architecture.

For this network, the numbers of outputs of most layers were reduced from the values used in the AlexNet architecture in order to reduce memory requirements.

For the implementation the caffe-segnet (BADRINARAYANAN *et al.* 2015) fork of the caffe library (JIA *et al.* 2014) was used. This inflorescence segmentation network uses only two upsampling layers (denoted by RONNEBERGER *et al.* 2015 as "up-convolution"). Layer Up-conv1 upsamples to the resolution of layer Conv1. The outputs of layer Up-conv1 and layer Conv1 are then appended and a convolution is applied in layer Conv8. Layer Up-conv2 then upsamples

Table 1

The network structure of the FCN used for inflorescence segmentation by layers. For the last output layer the softmax function (Prob layer) is used to map the resulting two output values for each pixel to probabilities for the two output classes (inflorescence and non-inflorescence respectively). The Concat-layer combines the channels of the outputs of Conv1 and Up−conv1.
Technical note: all convolution layers are followed by ReLUs. As in the AlexNet definiton, Local Response Normalizations follow the Pool1 and Pool2 layers. Both up-convolutions use a stride of 4, Conv8 uses a padding of 1. All further parameters were chosen according to the Alexnet definition

Name	Type	Output	Ksize
Data	Input	3 × 608 × 608	
Conv1	Convolution	96 × 150 × 150	11 × 11
Pool1	Max. Pooling	96 × 75 × 75	3 × 3
Conv2	Convolution	128 × 75 × 75	5 × 5
Pool2	Max. Pooling	128 × 37 × 37	3 × 3
Conv3	Convolution	192 × 37 × 37	3 × 3
Conv4	Convolution	192 × 37 × 37	3 × 3
Conv5	Convolution	192 × 37 × 37	3 × 3
Pool5	Max. Pooling	192 × 35 × 35	3 × 3
Conv6	Convolution	256 × 35 × 35	3 × 3
Conv7	Convolution	256 × 35 × 35	3 × 3
Up-conv1	Up-convolution	32 × 150 × 150	14 × 14
Concat	Concatenate	128 × 150 × 150	
Conv8	Convolution	64 × 150 × 150	3 × 3
Up-conv2	Up-convolution	2 × 608 × 608	12 × 12
Prob	Softmax	2 × 608 × 608	

to the resolution of the input image and produces output of the two classes, inflorescence and non-inflorescence. This architecture design was chosen on the assumption that the information of the first convolution is sufficient to find a fine separation between inflorescence and non-inflorescence, given the context of the surrounding image was provided by the last layer of the encoder network.

The training was done on the per-pixel class information provided by the manual annotation. Due to the high memory requirements, the network could not be trained on the full 5472×3648 pixel images. Instead, the network was trained on 5292 non-overlapping images patches of 608 × 608 pixels (as depicted in the olive input data layer of Fig. 4) produced from the training set. "Complete image segmentation" (page 99) describes how we resolved this problem of memory footprint. The network was trained using a Stochastic Gradient Descent solver, with a fixed learning rate of $5 \cdot 10^{-5}$, a momentum of 0.9, a weight decay of 10^{-4} and a batch size and iteration size of 1.

Since the detection and localization of inflorescences results in regions of interest (ROI), mean Intersection Over Union (IOU) was used as a quality measure. IOU is defined per class as the cardinality of the intersection of the detected areas and actual areas of a class divided by cardinality of the union of these areas.

$$IOU(c) = \frac{|T_c \cap P_c|}{|T_c \cup P_c|} \qquad (1)$$

The development of the mean IOU on the validation set during the training is shown in Fig. 5. The best model produced by the training achieved a mean IOU of 87.6% after 285500 iterations. This best-performing model was used as the segmentation model for the single flower extraction.

Fig. 6 shows an example of the segmentations produced by the trained model.

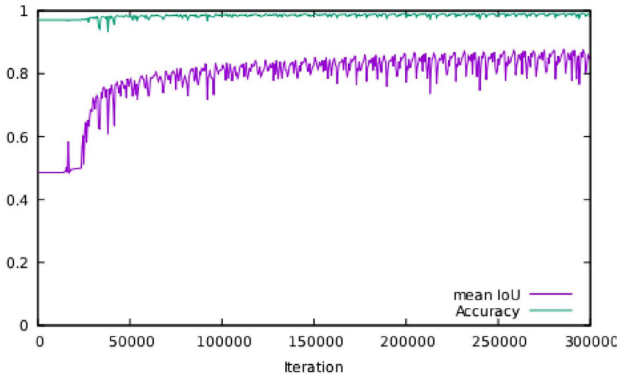

Fig. 5: Accuracy and mean Intersection Over Union (IOU) of the Fully Convolutional Network (FCN) during training. The flat behavior at the end of the graph indicates that further training would have not yielded much improvement of the model and that no overfitting occurred. The best-performing model was found after training for 285500 iterations (IOU of 87.6 %).

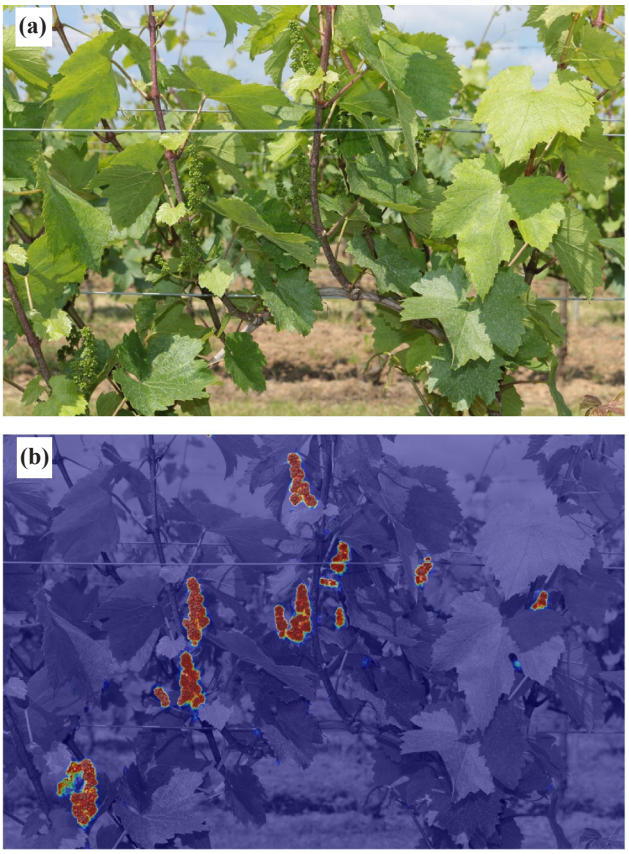

Fig. 6: Example of the segmentation produced by the trained FCN model. (**a**) Original input image of the grapevine 'Chardonnay'. (**b**) Segmentation heatmap with detected "inflorescences".

Complete image segmentation: While the designed network architecture is fully convolutional and therefore can scale it's output with it's input, in practice this is not be possible with arbitrary high-resolution input sizes, due to memory requirements. Instead of processing a complete image at once, it might be required to divide an image into smaller patches, as it was done previously for training the model. For prediction, the segmentations produced for the patches then have to be recombined to produce a segmentation of the complete image. This approach is a common workaround for this kind of bottleneck and was applied similarly by Ronneberger *et al.* 2015. The image patch size used during training does not limit the image patch size available for prediction. If sufficient memory is available, a larger image patch size can be chosen in order to increase runtime performance, without requiring a new model to be trained. As opposed to Ronneberger *et al.* 2015, the network used here was designed to produce an output of the same spatial size as the input by using padding and choosing the up-convolution kernels accordingly. This resulted in the network producing sub-optimal results at the boundary edges between two patches. This effect is shown in Fig. 7, in which

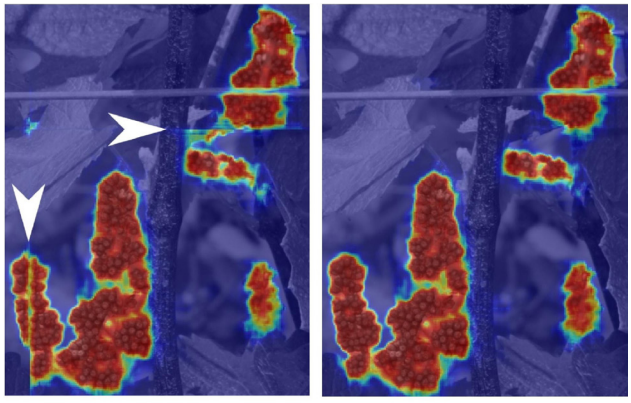

Fig. 7: Assembly of FCN-segmented image patches into whole images. (**a**) Assembly of image patches without overlapping, resulting in artifactual, inaccurate edges (white arrowheads). (**b**) Assembly of image patches with overlapping resulting in correctly segmented regions without missing information.

segmentation errors can be seen along the edges between processed image patches. As a workaround, the sub-optimal boundary edges of the prediction for each individual image patch were discarded. This effectively results in the predicted segmentation covering a smaller area within the input image patch. In order to produce a segmentation of the complete image the images were processed in overlapping patches in such a way that the smaller segmentations produced from the patches cover the complete input image. At the boundary edges of the complete input image artificial context was provided by mirroring the input image. This was done analog to the method described by Ronneberger *et al.* 2015. This results in a refined segmentation, which then can be used for the third step of our workflow, *i.e.* the flower extraction.

Flower extraction: For the extraction of single flowers from the previously found ROIs we approximate the contours of the single flowers by two-dimensional spheres. Due to this approximation we can apply the Circular Hough Transform (CHT) to detect the flowers. The CHT is a well-established approach to find imperfect instances of spheres by a voting procedure that is carried out in the

parameter space of two-dimensional spheres. The parameter space of two-dimensional spheres shows three dimensions: two dimension for the two-dimension coordinates of the position (x,y) of center of a circle hypothesis and the third dimension for the radius of a circle hypothesis.

This approach is similar to that of Roscher *et al.* 2014, where CHT was used to find berries of grape vines and determine their size. This third stage of processing of our workfl w (Fig. 2) includes some pre-processing of the image, applying edge detection, removing any edges not within an ROI, applying the CHT and extracting the candidates according to the voting analysis in the parameter space. The implementation was done using the OpenCV library (Itseez 2017). For preprocessing a local contrast normalization, as described by Jarrett *et al.* 2009, was applied, in order to allow for a single set of edge thresholds of the edge detection for flower contours throughout the image. This was required, since different areas of the images could be more or less blurry, either due to depth of field and distance, or due to slight movement of the inflorescences by wind.

For edge detection the Canny Operator was used. After using the implementation provided by the OpenCV library, edges not within an ROI were removed.

For the Hough Transform, single flowers of the radii between domain-specific minimum and maximum values of flower radii were checked. This interval represents the size of most single flowers within the field images. As a modification to the standard Hough Transform, each edge point only casts votes for a circle arc in its gradient direction, as well as the opposing direction, in order to reduce noise within the Hough Transform. Here, the arc in which votes were cast was chosen at $\gamma = \frac{\pi}{8} = 22.5°$. Additionally the voting values were normalized by dividing them by the number of possible votes they could achieve in total, in order to allow for direct comparison between values of different radii.

For the extraction of candidate single flowers all candidates above a certain threshold were sorted according to their value. By iterating over this sorted list of candidates, starting with those of highest value, the final resulting single flowers were selected. This was done by maintaining an occupancy map. If the center point of a candidate is not marked as occupied within the map, it is selected and a circle with $r = 1.5 \cdot r_{candidate}$ is marked as occupied on the map. This increased radius was chosen to allow for slight overlapping of candidates. After an iteration over all candidates, the selected candidates are returned as result. This is shown in algorithm 1. An example of this CHT-based flower extraction is shown in Fig. 8.

For validation, the candidate single flowers produced by the extraction were compared against the annotated single flowers by iterating over the candidates and finding the closest annotated single flower. If a single flower was within a certain radius-dependent distance of the candidate, the candidate was considered a true positive. Annotated single flowers selected for one candidate were ignored for future candidates. Candidates without a matching annotated single flower were considered false positives and annotated single flowers without a candidate near them were considered false negatives. Using these measures, F1 score, recall and precision were determined.

Data: Sorted list of candidate circles C, Image size S, Radius factor a
Result: A List of circles
Image *O*(S): = Unset;
List Result ← Ø;
foreach c ∈ *C* **do**
  if *O*(c.position) = Unset **then**
    Result.append(*c*);
    DRAW_CIRCLE (*O*, *c.position, c.radius* · *a*);
  end
end

Algorithm 1: Algorithm for selecting the final circles (Result) from the circle candidates (C) produced by the Circular Hough Transform. By maintaining an occupancy map O strongly overlapping circles are prevented.

Fig. 8: Example of the Circular Hough Transform-based flower extraction, using the segmentation previously shown in Fig. 6. (**a**) section of heatmap showing class "Inflorescence", (**b**) result of flower extraction within the region classified as "Inflorescence".

Results

The focus of this study is set on the efficiency of the image processing procedures, *i.e.* steps 2 and 3 of our workflow. Therefore, we first present the evaluation of the identification and localization of inflorescences by the trained FCN-segmentation model. Then, we present the evaluation of flower detection and quantification.

Identification and localization of inflorescences: The trained FCN-segmentation model achieved a mean Intersection Over Union (IOU) of 87.6 %, with class-specific IOUs of 76.0 % for inflorescences and 99.1 % for non-inflorescence. Examining the segmentations on the test set predicted by the segmentation model, it can be shown that most wrong classifications are false positives occurring around actual flower areas (Fig. 9a), while individ-

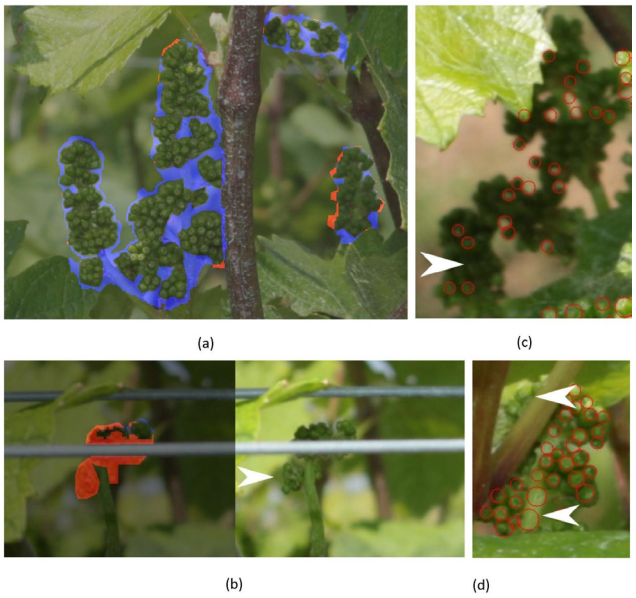

Fig. 9: (**a**) False positives (blue) within an inflorescence and false negatives (red) at the edge of an inflorescence produced by the segmentation model. (**b**) False negatives (red) on a small inflorescence produced by the segmentation model. (**c**) Inconsistent flower extraction on a blurry background inflorescence detected by the segmentation. (**d**) False positives on a branch (lower arrowhead) and false negatives (upper arrowhead) of the flower extraction.

ual false positives occur rarely. The false positives around flower areas are likely to be a result of the evaluation set being more precisely annotated than the training set. False negatives occur more rarely and usually occur at very small inflorescences of only a few single flowers (Fig. 9b) and occasionally at the edges of larger inflorescences (Fig. 9a).

While the FCN-segmentation model achieves high IOU values on the test set and in practice performs well as a basis for the flower extraction (see below), the performance could still be improved. Training the model on more images would, most likely, allow the network to learn a better generalization. Additionally, data augmentation (e.g. modified brightness, hue, saturation) during training could further improve the results.

Since the model was not yet tested on other grapevine varieties, it is unknown how well it generalizes to those. However, it can be assumed that it performs best on the varieties it was trained on. Incorporating more different grapevine varieties into the training set would improve the generalization of the inflorescence detection, allowing for the application on other varieties

Flower detection and quantification: The single flower detection and quantification was evaluated separately on (a) the complete image without providing segmentation, (b) the segmentation produced by the trained model and (c) the manually generated ground truth segmentation. The performance measures F1 score, Recall and Precision using each of the segmentations are shown in Tab. 2. Additionally, the 'EOA' column shows the mean amount of single flowers estimated over the amount annotated. The standard deviation of this value over the validation set is given in the 'σ(EOA)' column.

The flower extraction is prone to producing false positives over false negatives, generally resulting in a higher recall than precision. False negatives can occur in regions not labeled as ROI by the previous step (Fig. 9d, top) and at inflorescences which were labeled as ROI, but which are too blurred for the CHT to detect the single flowers (Fig. 9c). False positives often occur on other small plant structures within ROIs, e.g. the stems of inflorescences (Fig. 9d, bottom).

Since the flower extraction tends to overestimate the number of single flowers (Tab. 2, column EOA), a linear regression was fitted to correct for it. Tab. 3 shows these measurements as well as the absolute number of single flowers estimated and annotated for each individual test sample for the segmentation model. This linear model (shown in Fig. 10) achieved a coefficient of determination of $R^2 = 0.930$. However, due to the small sample size this relation might not generalize well and should be examined again utilizing more samples, including samples with fewer flowers.

While the best performance was achieved on the ground truth segmentation, the F1 score on the segmentation produced by the trained model is lower by only 4.8 %. This relatively small difference in performance of the flower extraction using the automatically generated segmentation data and the annotated segmentation data shows that the application of an FCN-based segmentation model is a promising strategy.

To meet the challenge of overestimation in the flower extraction step, future work will investigate the employment of a trained CNN in the flower extraction step instead of the

Table 2

The performance of the flower extraction using different segmentations. The performance measures include F1 score, recall, precision, mean estimated over actual visible number of single flowers (EOA) and standard deviation of EOA over the test set

Segmentation	F1 (%)	Recall (%)	Precision (%)	EOA (%)	σ(EOA) (%)
None	9.8	85.5	5.2	1718.7	336.7
Segmentation model	75.2	80.3	70.7	115.4	8.66
Ground truth	80.0	84.3	76.1	112.7	10.17

Table 3

The performance of the flower extraction using the trained segmentation model for all images of the test set. The measures include F1 score, recall, precision, estimated over actual number of single flowers (EOA), as well as the raw numbers of annotated and estimated single flowers

Image	F1 (%)	Recall (%)	Precision (%)	EOA (%)	Annotated	Estimated
Chardonnay Frontal 04	78.3	82.2	74.7	110.1	1157	1274
Chardonnay Frontal 08	76.4	86.2	68.5	125.8	839	1056
Chardonnay Frontal 11	76.4	80.1	69.3	115.6	1074	1242
Chardonnay Upwards 02	77.1	83.4	71.7	116.3	1312	1527
Chardonnay Upwards 05	72.1	70.2	74.2	94.5	1876	1774
Chardonnay Downwards 07	72.1	83.3	69.9	119.0	935	1113
Riesling Frontal 09	75.8	83.1	69.6	119.3	1138	1358
Riesling Upwards 04	72.6	81.0	65.8	123.1	1110	1367
Riesling Upwards 13	72.9	80.9	66.4	121.8	971	1183
Riesling Upwards 04	77.6	80.9	74.5	108.4	1320	1432

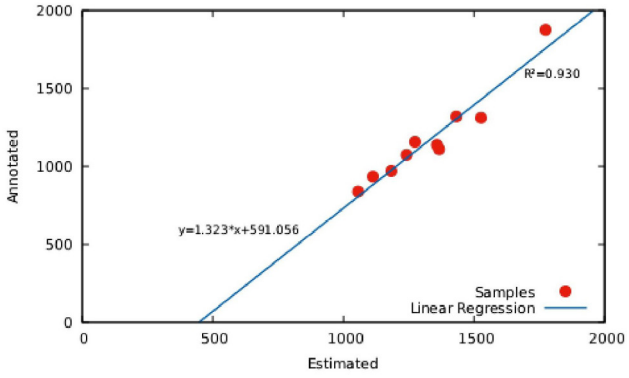

Fig. 10: Linear regression model of estimated *vs.* annotated single flower numbers within the test set. The test set consists of 10 randomly chosen images (6 'Chardonnay', 4 'Riesling') of differing perspectives.

Circular Hough Transform. The trade-off will be additional annotation work in the training data: instead of labeling inflorescences by bounding polygons (Fig. 3) single flowers must be labelled - e.g. by bounding circles. This new CNN-based flower extraction could be either implemented as a separate processing step operating on the inflorescence areas or directly as part of the segmentation network.

Runtime performance: The runtime behavior of both the segmentation model and the flower extraction were evaluated on the following system: MSI GE60-OND Gaming Notebook; Intel Core i7-3630QM 2.4GHz, 8GB DDR3 RAM, GeForce GTX 660M (2GB GDDR5 SDRAM). The operating system run was Arch Linux (64 bit Linux kernel, version 4.12.10). The libraries used were the at the time of writing most recent git version of caffe-segnet (rc2-338-gdba43980), cuda 8.0.61, cudnn 7.0.1 and opencv 3.3.0.

The segmentation model was run using the GPU mode of the caffe library. In order to make the best use of the massive parallelization possible with GPUs, the image patches were chosen as large as possible. One of the spatially largest networks able to fit in the 2GB graphics memory of the test system was that of an input size of 1216 × 1216 pixels. This allowed for processing of a complete image of 5472 × 3648 pixels in 20 image patches.

Including mirroring at edges, disassembling into patches and reassembling, the mean time of a segmentation was measured at 7.8 s. It can be expected that using upcoming, more modern graphics cards the runtime performance of the segmentation would significantly increase, due to availability of more memory, allowing for larger patches, as well as general increases in speed in modern hardware.

The flower extraction was run as a single-thread process on the CPU. On the segmentation produced by the trained model the mean runtime per image was 4.161s. Since the memory footprint of the flower extraction is relatively low, when used in practice, the throughput of the flower extraction could be massively sped up by using more threads/cores. This possibility makes the segmentation the main bottleneck of the complete system.

However, even without optimization of the flower extraction step, the total required time of about 12s per image should still allow for a practical application of the system.

Conclusions

In the present study, a low-cost and commercial available consumer camera was used in order to reveal simple-to-apply image acquisition of normal growth grapevines directly in vineyards. Further, an efficient, automated image analysis was developed for reliable single flower detection and quantification. It is the first study facilitating efficient and contactless screening of large sets of grapevines receiving objective and high-quality phenotypic data. This is important for further studies regarding the development of reliable early yield prediction models for objective characterization and multi-year monitoring of breeding material, e.g. crossing populations and genetic repositories. Early yield prediction is a promising strategy for grapevine training

systems showing more complex canopy architectures, e.g. semi-minimal pruned hedges. Further, the developed strategy makes carrying of artificial backgrounds or invasive treatments of grapevines due to defoliation unnecessary which opens up possible vehicle-based phenotyping applications.

In order to make predictions about the complete plant using the developed strategy, it further needs to be shown that the information gained about the inflorescences and flowers visible in the images can be extrapolated to all inflorescences and flowers of the plant.

Further, training the FCN on grapevine images of early stages of fruit development, *i.e.* fruit set (BBCH 71) or groat-sized berries (BBCH 73), will enable comparison of quantified single flowers and quantified young berries in order to phenotype susceptibility to fruit abscission, *i.e.* level of coulure, objectively and with high throughput. However, the system has to be robust for its reliable application on high diversity phenotypes. Therefore, further large data sets for different stages of plant development, different grapevine cultivars and phenotypic variable breeding populations are required for validation. Finally, the development of an intuitive graphical user interface will improve usability for potential users, *i.e.* breeders or scientists.

Acknowledgements

We gratefully acknowledge the German Research Foundation (Deutsche Forschungsgemeinschaft (DFG), Bonn, Germany (Automated Evaluation and Comparison of Grapevine Genotypes by means of Grape Cluster Architecture, STE 806/2-1, TO 152/6-1), as well as the German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung (BMBF), Bonn, Gemany (NoViSys: FKZ 031A349E)).

References

Aquino, A.; Millan, B.; Diago, M. P.; Tardaguila, J.; 2018: Automated early yield prediction in vineyards from on-the-go image acquisition. Comp. Electron. Agric. **144**, 26-36.

Aquino, A.; Millan, B.; Gaston, D.; Diago, M. P.; Tardaguila, J.; 2015a: vitisFlower®: development and testing of a novel Android-smartphone application for assessing the number of grapevine flowers per inflorescence using artificial vision techniques. Sensors **15**, 21204-21218.

Aquino, A.; Millan, B.; Gutiérrez, S.; Tardáguila, J.; 2015b: Grapevine flower estimation by applying artificial vision techniques on images with uncontrolled scene and multi-model analysis. Comp. Electron. .Agric. **119**, 92-104.

Auzmendi, I.; Holzapfel, B. P.; 2014: Leaf area to fruit weight ratios for maximising grape berry weight, sugar concentration and anthocyanin content during ripening, 127-132. XXIX Int. Hortic. Congr.n Hortic.: Sustaining Lives, Livelihoods and Landscapes (IHC2014): IV 1115. Brisbane, Australia.

Badrinarayanan, V.; Kendall, A.; Cipolla, R.; 2015: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561.

Boss, P. K.; Buckeridge, E. J.; Poole, A.; Thomas, M. R.; 2003: New insights into grapevine flowering. Funct. Plant Biol. **30**, 593-606.

Bramley, R. G. V.; Ouzman, J.; Boss, P. K.; 2011: Variation in vine vigour, grape yield and vineyard soils and topography as indicators of variation in the chemical composition of grapes, wine and wine sensory attributes. Aust. J. Grape Wine Res. **17**, 217-229.

Diago, M. P.; Sanz-Garcia, A.; Millan, B.; Blasco, J.; Tardaguila, J.; 2014: Assessment of flower number per inflorescence in grapevine by image analysis under field conditions. J. Sci. Food Agric. **94**, 1981-1987.

Domingos, S.; Scafidi, P.; Cardoso, V.; Leitao, A. E.; Di Lorenzo, R.; Oliveira, C. M.; Goulao, L. F.; 2015: Flower abscission in *Vitis vinifera* L. triggered by gibberellic acid and shade discloses differences in the underlying metabolic pathways. Front. Plant Sci. **6**, Art. 457.

Fuente de la, M.; Linares, R.; Baeza, P.; Miranda, C.; Lissarrague, J.; 2015: Comparison of different methods of grapevine yield prediction in the time window between fruitset and veraison. OENO One **49**, 27-35.

Giacomelli, L.; Rota-Stabelli, O.; Masuero, D.; Acheampong, A. K.; Moretto, M.; Caputi, L.; Vrhovsek, U.; Moser, C.; 2013: Gibberellin metabolism in *Vitis vinifera* L. during bloom and fruit-set: functional characterization and evolution of grapevine gibberellin oxidases. J, Exp. Bot. **64**, 4403-4419.

Howell, G. S.; 2001: Sustainable grape productivity and the growth-yield relationship: A review. Am. J. Enol. Vitic. **52**, 165-174.

Itseez; 2017: Open Source Computer Vision Library. <https://github.com/itseez/opencv>, visited: 2017-10-27.

Jarrett, K.; Kavukcuoglu, K.; Ranzato, M. A.; LeCun, Y.; 2009: What is the best multi-stage architecture for object recognition? 2146-2153. IEEE 12th Int. Conf. Computer Vision, Kyoto, Japan.

Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T.; 2014: Caffe: Convolutional architecture for fast feature embedding, 675-678. Proc. 22nd ACM Int. Conf. Multimedia, Orlando, FL, USA.

Keller, M.; Tarara, J. M.; Mills, L. J.; 2010: Spring temperatures alter reproductive development in grapevines. Aust. J. Grape .Wine Res. **16**, 445-454.

Kicherer, A.; Herzog, K.; Bendel, N.; Klück, H. C.; Backhaus, A.; Wieland, M.; Rose, J. C.; Klingbeil, L.; Läbe, T.; Hohl, C.; Petry, W.; Kuhlmann, H.; Seiffert, U.; Töpfer, R.; 2017: Phenoliner: A new field phenotyping platform for grapevine research. Sensors **17**, Art. e1625.

Kicherer, A.; Herzog, K.; Pflanz, M.; Wieland, M.; Rüger, P.; Kecke, S.; Kuhlmann, H.; Töpfer, R.; 2015: An automated field phenotyping pipeline for application in grapevine research. Sensors **15**, 4823-4836.

Kraus, C.; Pennington, T.; Herzog, K.; Hecht, A.; Fischer, M.; Voegele, R. T.; Hoffmann, C.; Töpfer, R.; Kicherer, A.; 2018: Effects of canopy architecture and microclimate on grapevine health in two training systems. Vitis **57**, 53-60.

Krizhevsky, A.; Sutskever, I.; Hinton, G. E.; 2012: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 1097-1105. Lake Tahoe, Nevada, USA.

Lebon, G.; Duchêne, E.; Brun, O.; Magné, C.; Clément, C.; 2004: Flower abscission and inflorescence carbohydrates in sensitive and non-sensitive cultivars of grapevine. Sexual Plant Reprod. **17**, 71-79.

Liu, S.; Li, X.; Wu, H.; Xin, B.; Petrie, P. R.; Whitty, M.; 2018: A robust automated flower estimation system for grape vines. Biosyst. Engin. **172**, 110-123.

Long, J.; Shelhamer, E.; Darrell, T.; 2015: Fully convolutional networks for semantic segmentation, 3431-3440. Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition. Boston, MA, USA.

Lorenz, D. H.; Eichhorn, K. W.; Bleiholder, H.; Klose, R.; Meier, U.; Weber, E.; 1995: Growth stages of the grapevine: phenological growth stages of the grapevine (*Vitis vinifera* L. ssp. *vinifera*) - Codes and descriptions according to the extended BBCH scale. Aust. J. Grape wine Res. **1**, 100-103.

Marguerit, E.; Boury, C.; Manicki, A.; Donnart, M.; Butterlin, G.; Némorin, A.; Wiedemann-Merdinoglu, S.; Merdinoglu, D.; Ollat, N.; Decroocq, S.; 2009: Genetic dissection of sex determinism, inflorescence morphology and downy mildew resistance in grapevine. Theor. Appl. Genet. **118**, 1261-1278.

Millan, B.; Aquino, A.; Diago, M. P.; Tardaguila, J.; 2017: Image analysis-based modelling for flower number estimation in grapevine. J. Sci. Food .Agric. **97**, 784-792.

Molitor, D.; Behr, M.; Hoffmann, L.; Evers, D.; 2012: Impact of grape cluster division on cluster morphology and bunch rot epidemic. Am. J. Enol. Vitic. **63**, 508-514.

Nuske, S.; Wilshusen, K.; Achar, S.; Yoder, L.; Narasimhan, S.; Singh, S.; 2014: Automated visual yield estimation in vineyards. J. Field Robot. **31**, 837-860.

Petrie, P. R.; Clingeleffer, P. R.; 2005: Effects of temperature and light (before and after budburst) on inflorescence morphology and flower number of Chardonnay grapevines (*Vitis vinifera* L.). Aust. J. Grape Wine Res. **11**, 59-65.

Preszler, T.; Schmit, T. M.; Vanden Heuvel, J. E.; 2013: Cluster thinning reduces the economic sustainability of Riesling production. Am. J. Enol. Vitic. **64**, 121-123.

Ronneberger, O.; Fischer, P.; Brox, T.; 2015: U-net: Convolutional networks for biomedical image segmentation, 234-241. Int. Conf. Medical Image Computing and Computer-Assisted Intervention, Munich, Germany.

Roscher, R.; Herzog, K.; Kunkel, A.; Kicherer, A.; Töpfer, R.; Förstner, W.; 2014: Automated image analysis framework for high-throughput determination of grapevine berry sizes using conditional random fields. Comp. Electron. Agric. **100**, 148-158.

Simonneau, T.; Lebon, E.; Coupel-Ledru, A.; Marguerit, E.; Rossdeutsch, L.; Ollat, N.; 2017: Adapting plant material to face water stress in vineyards: which physiological targets for an optimal control of plant water status? OENO One 51, 167-179.

Simonyan, K.; Zisserman, A.; 2014: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Töpfer, R.; Eibach, R.; 2016: Pests and diseases: Breeding the next-generation disease-resistant grapevine varieties. Wine Vitic. J. **31**, 47-49.

Received November 22, 2018
Accepted July 2, 2019