vitisBerry: An Android-Smartphone Application to Early Evaluate the Number of Grapevine Berries by Means of Image Analysis

Arturo Aquino, Ignacio Barrio, Maria-Paz Diago, Borja Millán, Javier Tardáguila

Research Centre of Vine-and- Wine-related Science (University of La Rioja, CSIC, La Rioja Regional Government), 26006, Logroño, La Rioja, Spain.

Corresponding author:

Arturo Aquino (arturo.aquino@unirioja.es).

ABSTRACT

In agriculture, crop monitoring and plant phenotyping are mainly manually measured. However, this practice gather phenotyping information at a lower rate than genotyping evolves, thus producing bottleneck. This paper presents vitisBerry, a smartphone application for assessing in the vineyard, using computer vision, the berry number in clusters at phenological stages between berry-set and cluster-closure. The implemented image analysis algorithm is an evolution of a previous development, providing 1.63% and 7.57% of Recall and Precision improvement, respectively. The application was evaluated using two devices, taking and analysing 144 images from 12 different grapevine varieties. The Recall and Precision results ranged between 0.8762-0.9082 and 0.9392-0.9508, depending on the device. The average computational time required to analyse the 144 images varied from 3.14 to 8.40 sec. According to these results, vitisBerry constitutes a tool for viticulturists to acquire phenotyping information from their vineyards in an easy and practical way.

KEYWORDS

Plant phenotyping tool; computer vision; yield prediction; berry-set estimation; berries' segmentation; smart devices.

ACKNOWLEDGMENT

This work received funding from the European Community's Seventh Framework Program (FP7/2007–2013) under Grant Agreement FP7-311775, Project Innovine. Additional funding

was given by the ‘Agencia de Desarrollo Económico de La Rioja’ through project VINETICS (2012-I-IDD-00097). The authors would also like to thank Prof. Serge Delrot, Prof. Cornelius van Leewen and Agnès Destract from the ‘Institut des Sciences de la Vigne et du Vin’ (Bordeaux, France) for allowing us to acquire images in the VitAdapt vineyard.

INTRODUCTION

Plant phenotyping involves a quantitative description of a plant's physiological, biochemical and morphological traits (Walter et al. 2015). It evaluates the effects on the phenotype as a result of the different interactions between the diverse genotypes and the environmental conditions to which the plant has been exposed (Minervini et al. 2015). Classically, the collection of phenotypes has been manually performed, but more recently, non-invasive image-based methods are increasingly being used to non-invasively capture detailed information of key agronomical and physiological traits of the plant throughout its life cycle (Spalding and Miller 2013; Li et al. 2014; Diago et al. 2012; Diago et al. 2016; Rabatel and Guizard 2007; Cubero et al. 2015; Aquino et al. 2015). Indeed, Herzog et al. (2014), and more recently Klodt et al. (2015), have studied the potential of the use of image analysis for high-throughput phenotyping in vineyards.

Within plant phenotyping in viticulture by using image analysis, the topic that has received more attention from the scientific community is yield estimation. Either from a breeding perspective (to identify and develop genotypes with specific yielding capacity) or from a grapegrowing approach, accurate yield assessment is of key importance in viticulture, and has been identified as one of the most profitable and strategical fields of research in viticulture (Dunstone 2002). Berry yield is defined by the yield components, involving the number of clusters, the berry number per cluster and the berry size. While the number of clusters per vine or per linear meter is mostly established at winter pruning, and remains stable across seasons, the number of berries per cluster is a more labile variable, even within a given genotype. It is influenced by the number of flowers per inflorescence and the fruit-set rate (fertility indicator), both parameters highly dependent upon the weather conditions during inflorescence development (at bud dormancy) and berry-set, respectively (May 2004). The number of berries per cluster is fully established at berry-set and remains invariable until harvest, determining not only the final yield but also the cluster compactness or cluster architecture and degree of berry aggregation (Cubero et al. 2015), which impacts berry ripening and disease incidence among other features.

As mentioned before, there is considerable bibliography on yield prediction, with methods working either under laboratory or field conditions. On the one hand, previous image-based studies for berry number estimation per cluster carried out under controlled laboratory conditions (Diago et al. 2015; Liu et al. 2015) offer a limited applicability, as plants experience more heterogeneous situations in the field, including environmental and lightning changes and competition from adjacent plants. Moreover, the rate at which plant phenotyping information is gathered under laboratory or field conditions does not match the speed of genotyping and, as a result, a bottleneck is being produced (Houle et al. 2010). As a consequence, image analysis algorithms towards yield prediction working under 'real' uncontrolled conditions have been presented over the recent years (Fernandez et al. 2013; Font et al. 2015; Nuske et al.2011; Nuske et al. 2011b; Nuske et al. 2012; Nuske et al. 2014; Berenstein et al. 2010; Liu et al. 2013; Reis et al. 2011; Reis et al. 2012). However, these methods are hardly directly applicable by viticulturists, since they often use complex mobile sensing platforms or specialized capturing devices.

Currently, there is a very mature and stablished market of mobile devices or smartphones offering a wide range of options. At this time, even low-range devices with affordable prices offer outstanding computational processing and photographic capabilities, which have made possible the development of specialized applications in multiple fields. Nevertheless, viticulture is not prolific in this sense yet, since there are only a few examples of available applications for managing vineyard features. The applications by De Bei et al. (2015) and Fuentes et al. (2012) for measuring grapevine canopy architecture, and that by Aquino et al. (2015) and Millan et al. (2016) (vitisFlower) for assessing the number of flowers per inflorescence are surely the most remarkable among those presented.

This paper presents the development and testing of a novel smartphone application, called vitisBerry, for counting the berry number per cluster visible in images of clusters at a phenological stage between berry-set and cluster-closure; the number of berries in a cluster image is strongly correlated to the actual number in the real cluster (Aquino et al. 2016). The

application integrates the capturing and image analysis processes by exploiting the smartphone capabilities, providing the viticulturist with an easy-to-use tool to acquire phenotypic information directly in the field.

MATERIALS AND METHODS

Image analysis algorithm for berry counting in cluster images

The vitisBerry application developed for Android devices, is a novel tool for viticulturists to assess the number of berries in grapevine clusters at a phenological stage between berry-set and cluster-closure (stages K and L according to the scale proposed by Baggiolini (1952)). The application allows to take a cluster picture and implements an improved version of the algorithm presented by Aquino et al.(2016) for its analysis. As a result, the number of visible berries in the cluster is given.

The image analysis algorithm included in vitisBerry is based on mathematical morphology and pixel classification by means of supervised learning. As a pre-requisite, the photo to be analysed has to be taken by placing a dark background behind the cluster. Concretely, a simple capturing box as the one shown in Figure 1 made with a Din A3 cardboard was employed. The use of this artefact offered applicative advantages. On the one hand, it eased the individual capture of clusters, since very often they were located so close together within the grapevine, or even touching themselves, that it was impossible to photograph them individually without picking or damaging them. In addition, it also favoured the accurate extraction of a region of interest (ROI) by image analysis. Moreover, the capturing box also improved the experience with respect of using a flat cardboard, as the box's borders prevented from the generation of undesired bright spots on the berrys' surface originating from light reflections on leaves or other elements.

The algorithm is divided into three main steps:

- Image pre-processing: this step is basically aimed at extracting the cluster in the image from the background. Firstly, the image is downsized to a resolution of 1170 *x* 1578 pixels (1.84 Mpx) for computational-time optimization purposes (see an example of an image acquired

with the vitisBerry application in Figure 2-(a)). Then, it is converted from the native RGB to the CIE 1976 L*a*b* colour space (Connoly and Flies 1997), as this colour scheme offers the illumination and colour information decoupled, what favours the approach followed in the algorithm. Finally, the ROI is extracted by using colour discrimination criteria (see results on Figure 2-(b)); note that colour is invariant to light conditions in the CIE 1976 L*a*b* colour space).

- Image analysis: this step addresses the extraction of berry candidates. The detection of candidates relies on the light reflection pattern that takes place on the convex surface of berries. It is characterised by a maximum light reflection point representing the centre of a circle, progressively decreasing the reflection intensity around the centre for describing a circular pattern; this effect follows the Lambert's cosine law (Smith 2007). Hence, berry candidates are extracted, using the extended *h*-maxima transform (Soille 2004), by finding those connected components within the ROI being regional maxima in the lightness channel (L*) of the CIE 1976 L*a*b* colour space (check Figure 2-(c)).
- Image post-processing: this processing is focused on analysing the set of connected components representing berry candidates previously obtained for discarding false positives. For every candidate, a set of berry descriptors is calculated and given as input to a neural network (NN) trained with supervised learning. The NN architecture is analogue to the original one described in Aquino et al. (2016), producing as result a real output in the interval *[0,...,1]*, which is considered as the candidate's probability of belonging to an actual berry. Once all candidates are analysed, a probability map is created by composing an image in which the candidates are represented with their computed probability (see Figure 2-(d)). Finally, the image is binarized using the threshold automatically provided by the Otsu's method (Nobuyuki 1979), thus discarding false positives (see Figure 2-(e) and (f) to check the final result).

  Two important contributions have been added to this stage with respect to the original version detailed in Aquino et al. (2016): a linear transformation for colour homogenization and eight new descriptors added to the original set composed of six.

Let *L*, *A* and *B* be the images from the L, a* and b* channels of an image in the CIE 1976 L*a*b* colour space. First, a linear transformation in *A* and *B* is applied to every pixel within the ROI according to:

$$A'(i,j) = \frac{(A(i,j) * 108.78)}{\bar{A}}, B'(i,j) = \frac{(B(i,j) * 167.82)}{\bar{B}} \quad (1)$$

Where $\bar{A}$ and $\bar{B}$ are the average values of the pixels within the ROI for the image *A* and *B*, respectively. Additionally, 108.78 and 167.82 are the average values measured from the a* and b* channels, respectively, in the 26 images used to train the NN, by considering only the pixels within the corresponding ROIs (the set of 26 images was the same used to train the original version of the NN described in Aquino et al. (2016)). Hence, this transformation homogenizes colour information in images 'unknown' for the NN, by moving pixel values around the average calculated in training images.

With these definitions, the set of berry descriptors is formulated as:

- Shape (one descriptor): it is applied as defined in Aquino et al. (2016), and evaluates the circularity of a connected component representing a berry candidate by calculating the quotient between its minor and major axis length:
- Normality (one descriptor): this descriptor is computed on image *L* and strengthens the detection of berries by assuming that the light reflection produced on their surface describes a 2-dimensional Gaussian distribution. For a full description and study on this descriptor, consult Aquino et al. (2016).
- Colour discrimination (four descriptors): the vitisVerry application was designed to analyse clusters at phenological stages before veraison, when berries are green independently from the grapevine variety. This colour feature was used to define four statistical descriptors calculated on *A'* and *B'* as defined in Aquino et al. (2016). Note that in this first algorithm's version, these descriptors were calculated directly on the non-corrected images *A* and *B*.

- Hu moments (seven descriptors): the Hu moment invariants (Hu 1962) are a set of seven moments invariant under size, translation and rotation, and used for recognizing known patterns in images. Grape berries exhibit a circular shape in images, feature distinctive from other objects in the scene and exploited for discrimination by means of modelling using the mentioned set of seven moments. Concretely, the logarithm of every moment is calculated for all connected component representing a candidate on a gradient-magnitude subimage extracted around the centroid of the component. Thus, seven new descriptors were added to the classifier implemented in the previous version of the image analysis algorithm.
- Entropy (one descriptor): a value of entropy calculated from an image is a statistical measure of the randomness present in the distribution of pixel values or, in other words, it can be seen as a measure of 'coherence' of the distribution of pixel values. Applied to this work, a berry candidate represented by a connected component will show a more meaningful distribution of pixel values in its surroundings when it truly represents a berry, compared to when it comes from other artefacts in the image, such as for instance the rachis (see Figure 3 to graphically visualise this concept). This is why a descriptor based on the entropy measured in images was developed and added to the original classifier (the basis and calculation of entropy in images can be consulted in Gonzalez and Woods (Gonzalez and Woods 2003).

vitisBerry's technical overview

vitisBerry integrates the image capture, analysis and results storage processes to provide a comprehensive tool for the early manual assessment of berries per cluster. It was conceived as a 'twin' application of vitisFlower (Aquino et al. 2015), so it was developed for smartphones powered with 2.3 Android versions and above. Furthermore, the aim was to provide the vitisFlower's users with a complementary application with an analogue behaviour, so as to make them feel familiar with it from the beginning. With this in mind, vitisFlower was designed following a model-view-controller approach (Larman 2004). This made possible to develop

vitisBerry from the basis of vitisFlower, re-implementing the business logic but keeping the application's controller, hence also keeping the usability unchanged, and only slightly modifying the user's interface for showing the appropriated information.

For the reasons outlined above, the vitisBerry's architecture followed the same design principles applied for vitisFlower. Its key characteristics can be summarized as follows (a detailed description can be found in Aquino et al. (2015)):

- The image analysis algorithm described in Section 2.1 was implemented using the C/C++ OpenCV library for proving higher versatility than its Java version, OpenCV4Android.
- The application's controller and graphical interface were implemented in Java.
- The communication between the 'Java side' and the 'C++ size' carried out by the controller was made through Java Native Interface (JNI). This way, the controller in the 'Java side' invokes the analysis of an image to the 'C++ side' through JNI, which is really in charge of executing the appropriate methods and returning results.

vitisBerry's performance description

Figure 4 illustrates the performance of the vitisBerry application by means of an illustrated flow-chart diagram, which is outlined here as follows:

1. Home: it shows basic information about vitisBerry.
2. Instructions for image capture: it provides the user with some basic notions for appropriate image capture.
3. Image capture: the camera application available in the device (parametrized according to the user's settings) is invoked, thus allowing image capture.
4. Image analysis: this state is transparent to the user, and executes the image analysis algorithm for detecting and counting berries in the image previously taken.
5. Results display: the analysed image, highlighting the detected berries with red crosses, is displayed along with a brief text indicating the number of detected berries.

6. Image storage: upon user request, the processed image is saved in a folder called 'VitisBerryImages' located in the root folder of the device's internal storage. The image is saved with a name formatted as [name]_[date]_[detected number of berries].jpg, with the following meaning:
    - [name]: the name introduced by the user through the dialog box, or 'image' by default.
    - [date]: the capture date as day-month-year_hour.minutes.seconds.miliseconds.
    - [detected number of berries]: the berry number detected in the image.

vitisBerry's testing and validation

Evaluation of the image analysis algorithm's improvement

The image analysis algorithm presented here was firstly evaluated on the database used for assessing the original version developed in Aquino et al. (2016). By this way, a rigorous comparison between both versions could be carried out.

The dataset consisted on 152 cluster images of the grapevine varieties Tempranillo, Semillon, Merlot, Grenache, Cabernet Sauvignon, Chenin Blanc and Sauvignon Blanc. The images were taken in a grapevine variety collection within the experimental vineyards of the 'Institut des Sciences de la Vigne et du Vin' (ISVV, Villenave d'Ornon, Bordeaux, France). Clusters were sampled at a phenological stage previous to veraison, concretely varying depending on the variety between stages K and L according to the scale proposed by Baggiolini (1952); the images were taken under field conditions. The smartphone used for image acquisition was the BQ Aquaris E5 (Mundo Reader S.L., Madrid, Spain) detailed in Table 1. The camera was set in automatic mode, the flash was used depending on the illumination conditions, and the distance between the camera and the cluster was kept around 20 to 40 cm.

For evaluating the results, Recall and Precision were calculated per image as:

$$RC = \frac{TP}{TP + FN}; PC = \frac{TP}{TP + FP} \quad (2)$$

where *RC* stands for Recall (percentage of actual berries detected), and *PC* denotes Precision (percentage of berry misclassification). A gold standard set was created by manually labelling berries on each image for allowing the application of these metrics. Then, true positives (*TP*), false positives (*FP*) and false negatives (*FN*) were calculated per image following the definitions:

- *TP*: berries automatically detected corresponding to actual berries labelled in the gold standard.
- *FP*: berries automatically detected not corresponding to actual berries in the gold standard. Redundant *TP*s were also considered as *FP*.
- *FN*: actual berries labelled in the gold standard not found by the segmentation algorithm.

In Addition, one-way analysis of variance was used to compare the performance of both versions of the algorithm in terms of the defined metrics and among grapevine varieties for each version. Mean comparison was attempted using the Tukey's test (Tukey 1949) at $p < 0.05$.

vitisBerry's performance evaluation

The vitisBerry's testing was designed pursuing a double aim. On the one hand, its ability to accurately detect berries when using it directly in the vineyard running on a smartphone was assessed. On the other hand, its dependency in terms of robustness upon cameras of different qualities was also evaluated.

To achieve these goals, an experiment was performed at a grapevine variety collection located in the experimental vineyards of the 'Instituto de Ciencias de la Vid y el Vino' (Logroño, Spain). The experiment consisted on capturing and analysing using vitisBerry, running on two different devices, 144 clusters at phenological stages between K and L (Baggiolini 1952). Concretely, 12 clusters of the following 12 different grapevine varieties were studied: Airen, Chardonay, Cabernet Sauvignon, Grenache Blanc, Grenache, Merlot, Malvasía, Pinot Meunier, Pinot Noir, Tempranillo, Syrah and Viognier. The two devices employed for the experiment were those detailed in Table 1. The Sony Xperia Z5 (Sony Corp., Tokyo, Japan) is a high-end

smartphone mounting a camera widely considered as one of the finest available in the market, whereas the BQ Aquaris E5 is a low-end smartphone equipped with a more modest camera (camera considerations are formulated according to the DxOMark ranking[1]). Finally, the results on the 144 studied clusters produced with both smartphones were assessed in terms of Recall and Precision following the methodology described to this effect in the previous section. One-way analysis of variance using the Tukey's test (Tukey 1949) at $p < 0.05$ was used to compare the algorithm's performance running on both devices and among grapevine varieties for each devise.

vitisBerry's computational efficiency study

The vitisBerry application was also evaluated in terms of overall usability by analysing its efficiency. This efficiency is mainly conditioned by the computation time the image analysis algorithm takes to process an image. It was also interesting to assess how different devices with different hardware configurations affected the computation time.

To evaluate these features, an experiment consisting on studying the computation time consumed by four smartphones equipped with different hardware and software configurations for analysing the same set of images was developed. The devices used for this study were those detailed in Table 2, and were selected to cover a wide range of the market's spectrum in terms of price and performance, were used.

To obtain meaningful results, the four devices analysed the same dataset composed of the 144 images captured with the Sony Xperia Z5 described in the previous section. This analysis was performed with a simplified version of vitisBerry including only a home page with a single button for running the test. Prior to test starting, the following protocol was applied with all devices for the seek of avoiding the interference in the results of other applications and services:

1. Closing recent applications.
2. Selecting the flight mode.

[1] https://www.dxomark.com/Mobiles

3. Re-starting the device.
4. Waiting for 20 sec. for the operating system to completely loaded.
5. Starting the benchmarking version of vitisBerry.
6. Running the test.

The set of images, stored in the smartphone's internal storage, was analysed five times upon test starting, registering the time taken for each image in each of the five analysis. Thus, for a given image, the definitive computation time for its analysis was finally calculated as the average time taken for its analysis during the five performed iterations.

RESULTS AND DISCUSSION

Evaluation results of the improvement of the image analysis algorithm

Table 3 summarizes the results obtained by the original version of the image analysis algorithm for detecting berries on cluster images (Aquino et al.2016), and those given by the improved version developed in this paper. Results were calculated in terms of average Recall ($\overline{RC}$) and Precision ($\overline{PR}$), measured per variety and also considering all images together.

The results obtained with both versions, measured on the same set of images, indicated clear improved performance in terms of both metrics, giving Recall and Precision increasing of 1.63% and 7.57%, respectively. Furthermore, the analysis of variance of both metrics also showed more robust behaviour for the improved version attending to results per variety and also considering all images as a whole. This improvement can be explained by the two main contributions added to the original version of the algorithm. On the one hand, the green tone of the berries greatly varies among varieties, development stage and many other factors related to in-the-field image capture. To this respect, the addition of the linear transformation for homogenizing colour in images 'unknown' for the classifier improved the behaviour of the descriptors based on colour assessing. On the other hand, two sets of new descriptors, the one based on the seven Hu's moment invariant and the one based on entropy, were also added. The obtained results argue for the inclusion of the new descriptors increased the descriptive skills of

the algorithm, thus making it able to effectively detect more berries, but also producing less false positives at the same time.

Figure 5 includes two examples of images analysed with the two versions of the image analysis algorithm. The berries detected are represented by a blue connected component representing the validated bright spot on the berry's surface. Images (a) and (c) were produced by the original version of the algorithm, whereas (b) and (d) were results given by the improved version. From a visual inspection of image (a) vs. (c), it can be identified how the improved version increased the number of correctly detected berries (the additional berries found can be identified in purple colour in image (c)). By checking image (b) vs. (d), it can be assessed how the new version outstandingly reduced the number of false positives without penalising the number of real berries detected (the discarded berries can be identified in red colour in image (d)).

Results of the vitisBerry's performance evaluation

The in-the-field results obtained with vitisBerry running of the two selected devices are given in Table 4 and expanded in Figure 6. Better results were produced by the Sony Xperia Z5, in terms of absolute values of $\overline{RC}$ and $\overline{PR}$, and also attending to the analysis of variance per variety. In this last case, the analysis indicated more robust behaviour among varieties for this device. However, when considering all images as a unique set for each smartphone, the analysis of variance showed no significant differences between both. This result argues for an analogue performance of vitisFlower running of both devices in spite of belonging to opposite market ranges and, therefore, despite offering technological solutions of very differentiated qualities.

Notwithstanding the above, better experience was achieved with the Xperia device than with the Aquaris E5 when working with the application in the vineyard. This last device incorporates a camera that showed inconsistent behaviour in terms of light exposure, white balance, auto-focus and other aspects. This fact sometimes provoked retaking some captures, which happened quite less frequently with the Xperia. Going deeper in the application usability, generally speaking, the experience of the application in the field and the visual analysis of the acquired images led

to define the image acquisition settings producing the best application behaviour. These included:

- Analysing inflorescences facing the Sun. The opposite orientation leads to light reflection and refraction patterns that can negatively affect the results.
- Casting a shadow on the cluster to create a homogeneous illumination. If it is poor due to low natural-light conditions, the use of the camera flash is recommended.

Additionally, especially the results obtained with this device were comparable to those discussed in the previous experiment. This takes special relevance when considering that the images used to train the NN were taken during the same working session and with the same device (the BQ Aquaris E5) than the ones used for validating results in the previous section. This meant that both set of images inevitably shared certain characteristics like illumination, colour temperature, etc., fact that did not happen with the images validated here from the Xperia Z5. Indeed, this particular emphasizes confidence in the flexibility and descriptive potential of the algorithm.

Results of the study of vitisBerry's computational efficiency

Figure 7 illustrates the measured results for evaluating the computational efficiency of the vitisBerry application running on the four devices listed in Table 2. The Xiaomi Redmi Note 3 Pro (Xiaomi Inc., Beijing, China) offered the best computation time distribution, with an average and standard deviation of 3.14 sec. and 0.92 sec., respectively. Attending to price and specs this result was surprising at a first view, since the Xperia Z5 was expected to be the most efficient for equipping the most powerful CPU (it gave 4.47 sec. and 1.43 sec. of average time and standard deviation). However, taking into account that vitisBerry was not parallelized and, therefore, runs exclusively on the main CPU's core, the result was coherent as the main core of the CPU of the Redmi Note 3 Pro (Cortex A72) is a more recent and powerful design than the one in the CPU of the Xperia (Cortex A57). The measures for the Xiaomi Mi4 were 5.87 sec. and 1.76 sec. of average and standard deviation, whereas the BQ Aquaris E5 produced the worst

result with 8.40 sec. of average time and 2.53 sec. of standard deviation. These results confirm that vitisBerry is an efficient tool, providing good usage experience in terms of efficiency even when working with modest devices in terms of price and specs. Note that, attending to the box and whisker plot in Figure 7-(b), the lowest device (BQ Aquaris E5) analysed the 75% of images in less than 11 sec. in average.

Significance of the vitisBerry application for the wine industry

The presented smartphone application, allows to anyone going into the vineyard to acquire objective knowledge on two valuable agronomic parameters. On the one hand, the viticulturist may select a representative number of vines (this can be variable upon the degree of variability within a vineyard plot) and 2-3 clusters per vine for estimating fruit-set (number of flowers that become berries). It can be carried out by using vitisBerry together with vitisFlower[9] to monitor the selected clusters from preflowering to veraison. On the other hand, since the number of berries in a cluster image is strongly correlated to the actual number in the real cluster (Aquino et al. 2016), vitisBerry can be used to obtain information about future yield by sampling the selected representative set of clusters in the plot.

CONCLUSIONS

This paper presents a smartphone application for Android devices, called vitisBerry, which allows assessing grapevine clusters by means of image analysis. This application incorporates an improved version of a previous image analysis algorithm for detecting berries in cluster images. The evolved algorithm is also described here and compared to the original version, providing clear improvement in terms of both Recall and Precision. According to the results of the experiments developed in this paper, vitisBerry is a practical and efficient tool for the viticulturists to analyse phenotypic information in their vineyards.

Future improvements of vitisBerry will include the creation of a wider and more variated training set of images composed of images taken with different devices, at different sunlight expositions and many other variated situations. This is expected to improve the application's

usability by reducing the frequency of needing to retake images, as well as to homogenize even more its performance with devices with different cameras.

REFERENCES

Aquino, A., Diago, M.P., Millan, B., & Tardáguila, J. (2016). A new methodology for estimating the grapevine-berry number per cluster using image analysis. *Biosystems Engineering, 156*, 80-95.

Aquino, A., Millan, B., Gastón, D., Diago, M.P., & Tardáguila, J. (2015). vitisFlower®: Development and Testing of a Novel Android-Smartphone Application for Assessing the Number of Grapevine Flowers per Inflorescence Using Artificial Vision Techniques. *Sensors-Basel, 15*, 21204-21218.

Aquino, A., Millan, B., Gutiérrez, S., & Tardáguila, J. (2015). Grapevine flower estimation by applying artificial vision techniques on images with uncontrolled scene and multi-model analysis. *Computers and Electronics in Agriculture, 119*, 92-104.

Baggiolini, M. (1952). Stades repères de l'abricotier. *Revue Romande d'Agriculture, de Viticulture et d'Arboriculture, 8*, 28-29.

Berenstein, R., Shahar, O.B., Shapiro, A., & Edan, Y. (2010). Grape clusters and foliage detection algorithms for autonomous selective vineyard sprayer. *Intelligent Service Robotics, 3(4)*, 233-243.

Connolly, C., & Fliess, T. (1997). A study of efficiency and accuracy in the transformation from RGB to CIELAB color space. *IEEE Transactions on Image Processing, 6*, 1046-1047.

Cubero, S., Diago, M.P., Blasco, J., Tardaguila, J., Prats-Montalbán, J.M., Ibáñez, J., et al. (2015). A new method for assessment of bunch compactness using automated image analysis. *Australian Journal of Grape and Wine Research, 21*, 101-109.

De Bei, R., Hook, J., Fuentes, S., Gilliham, M., Tyerman, S., & Collins, C. (2015). Linking Canopy Architecture to Grape Quality using the LAI Canopy App. In *Proceedings of the 19th International Meeting of Viticulture GIESCO* (pp. 585–588). Montpellier, France.

Diago, M.P., Correa, C., Millán, B., Barreiro, P., Valero, C., & Tardaguila, J. (2012). Grapevine yield and leaf area estimation using supervised classification methodology on RGB images taken under field conditions. *Sensors-Basel, 12*, 16988-17006.

Diago, M.P., Krasnow, M., Bubola, M., Millan, B., & Tardaguila, J. (2016). Assessment of vineyard canopy porosity using machine vision. *American Journal of Enology and Viticulture, 67*, 229-238.

Diago, M.P., Tardaguila, J., Aleixos, N., Millan, B., Prats-Montalbán, J.M., Cubero, S., et al. (2015). Assessment of cluster yield components by image analysis. *Journal of the Science of Food and Agriculture, 95*, 1274-1282.

Dusntone, R.J. (2002). Winegrape crop forecasting module. In *Final report for project DNR 02/02* (pp. 19). Adelaida, Australia.

Fernández, R., Montes, H., Salinas, C., Sarria, J., & Armada, M. (2013). Combination of RGB and Multispectral Imagery for Discrimination of Cabernet Sauvignon Grapevine Elements. *Sensors-Basel, 13*, 7838-7859.

Font, D., Tresanchez, M., Martínez, D., Moreno, J., Clotet, E., & Palacín, J. (2015). Vineyard Yield Estimation Based on the Analysis of High Resolution Images Obtained with Artificial Illumination at Night. *Sensors-Basel,* 15, 8284-8301.

Fuentes, S., de Bei, R., Pozo, C., & Tyerman, S. (2012). Development of a Smartphone Application to Characterise Temporal and Spatial Canopy Architecture and Leaf Area Index for Grapevines. *Wine and Viticulture Journal, 27***:** 56–60.

Gonzalez, R.C., & Woods, R.E. (2003). *Digital Image Processing - 3rd ed*. Upper Saddle River, New Jersey, USA: Prentice Hall.

Herzog, K., Roscher, R., Wieland, M., Kicherer, A., Läbe, T., Förstner, W., et al. (2014). Initial steps for high-throughput phenotyping in vineyards. *Vitis, 53*, 1-8.

Houle, D., Govindaraju, D.R., & Omholt, S. (2010). Phenomic: The next challenge. *Nature Reviews Genetics, 11*, 855-866.

Hu, M.K. (1962). Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory, IT-8*, 179-187.

Klodt, M., Herzog, K., Töpfer, R., & Cremers, D. (2015). Field phenotyping of grapevine growth using dense stereo reconstruction. *BMC Bioinformatics, 16*, 143.

Larman, C. (2004). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development - 3rd ed*. Upper Saddle River, New Jersey, USA: Prentice Hall.

Li, L., Zhang, Q., & Huang D. (2014). A review of imaging techniques for plant phenotyping. *Sensors-Basel, 14*, 20078-20111.

Liu, S., Marden, S., & Whitty, M. (2013). Towards Automated Yield Estimation in Viticulture. In *Proceedings of the Australasian Conference on Robotics and Automation*. Sydney, Australia.

Liu, S., Whitty, M., & Cossel, S. (2015). A lightweight method for grape berry counting based on automated 3D bunch reconstruction from a single image. In *Proceedings of the Workshop on Robotics in Agriculture.* Seattle, USA.

May, P. Development after fertilization (2004). In P. May (Ed.), *Flowering and Fruitset in Grapevines*. Adelaide, Australia: Lythrum Press.

Millan, B., Aquino, A., Diago, M.P., & Tardaguila, J. (2016). Image analysis-based modelling for flower number estimation in grapevine. *Journal of the Science of Food and Agriculture, 97(3)*, 784-792.

Minervini, M., Scharr, H., & Tsaftaris, S. (2015). Image analysis: The new bottleneck in plant phenotyping [Applications corner]. *IEEE Signal Processing Magazine, 32*, 126-131.

Nobuyuki O, A Threshold Selection Method from Gray-Level Histograms. *IEEE T Syst Man Cyb* **9:** 62-66 (1979).

Nuske, S., Achar, S., Bates, T., Narasimhan, S.G., & Singh, S. (2011). Yield Estimation in Vineyards by Visual Grape Detection. In IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 2352-2358). San Francisco, California, USA: IEEE.

Nuske, S., Achar, S., Bates, T., Narasimhan, S.G., & Singh, S., (2011b). *Visual Yield Estimation in Vineyards: Experiments with Different Varietals and Calibration Procedures (CMU-RI-TR-11-39).* Pittsburgh, Pennsylvania, USA: Robotics Institute, Carnegie Mellon University.

Nuske, S., Gupta, K., Narasihman, S., & Singh, S. (2012). Modeling and calibration visual yield estimates in vineyards. In *Proceedings of the 8th International Conference on Field and Service Robotics*. Matsushima, Miyagi, Japan.

Nuske, S., Wilshusen, K., Achar, S., Yoder, L., Narasimhan, S., & Singh, S. (2014). Automated visual yield estimation in vineyards. *Journal of Field Robotics, 31*, 837-860.

Rabatel, G., & Guizard, C. (2007). Grape berry calibration by computer vision using elliptical model fitting, In *Proceedings of the 6th European Conference on Precision Agriculture* (pp. 581-587). Skiathos, Greece.

Reis, M.C., Morais, R., Pereira, C., Soares, S., Valente, A., Baptista, J., et al. (2011). Automatic Detection of White Grapes in Natural Environment Using Image Processing. In *Proceedings of the 6th International Conference on Soft Computing Models in Industrial and Environmental Applications* (pp. 19-26). Salamanca, Spain.

Reis, M.J.C.S., Morais, R., Peres, E., Pereira, C., Contente, O., Soares, S., et al. (2012). Automatic detection of bunches of grapes in natural environment from color images. *Journal of Applied Logic, 10(4)*, 285-290.

Smith, W.J. (2007). *Modern Optical Engineering. The Design of Optical Systems - 4th ed*. NY City, USA: McGraw-Hill Education – Europe.

Soille, P. (2004). *Morphological Image Analysis. Principles and Applications - 2nd ed*. Berlin, Germany: Springer – Verlag.

Spalding, E.P., & Miller, N.D. (2013). Image analysis is driving a renaissance in growth measurement. *Current Opinion in Plant Biology, 16*, 100-104 (2013).

Tukey, J. (1949). Comparing individual means in the analysis of variance. *Biometrics, 5*, 99-114.

Walter, A., Liebisch, F., & Hund, A. (2015). Plant phenotyping: from bean weighing to image analysis. *Plant Methods*, 11-14.

TABLES

Table 1. Main relevant features of the two smartphones used for evaluating performance of the vitisBerry application.

Feature / Device	Price/Release date	Sensor model	Resolution	Lens Size	Aperture	ISO
Sony Xperia Z5	479.0 €/2015	Sony IMX300	23 Mpx	1/2.3''	f/2.0	50-12800
BQ Aquaris E5	185.99 €/2014	Sony IMX214	13 Mpx	1/3.2''	f/2.2	100-1600

Table 2. Main relevant features of the four devices used for evaluating computational efficiency of the vitisBerry application.

Feature / Device	Price/Release date	Chipset	CPU	GPU	RAM	Android version
Sony Xperia Z5	479.0 €/2015	Qualcomm MSM8994 Snapdragon 810	Octa-core 2.0 GHz; 4xARM Cortex A57, 4xARM Cortex A53	Adreno 430	3 GB	6.0.1
BQ Aquaris E5	185.99 €/2014	Qualcomm MSM8916 Snapdragon 410	Quad-core 1.2 GHz ARM Cortex-A53	Adreno 306	1 GB	5.1.1
Xiaomi Mi4	180.75 €/2014	Qualcomm MSM8974AC Snapdragon 801	Quad-core 2.5 GHz 4xKrait 400	Adreno 330	3 GB	6.0.1
Xiaomi Redmi Note 3 Pro	198.83 €/2016	Qualcomm MSM8956 Snapdragon 650	Hexa-core 1.8 GHz; 2xARM Cortex A72, 4xARM Cortex A53	Adreno 510	3 GB	5.1

Table 3. Comparison of the original and improved version of the image analysis algorithm for berry counting on cluster images. Results were measured on the same set of 126 images used in Aquino et al., (2016) for evaluating the original version of the algorithm. Figures are given detailed per grapevine variety.

	Original Version (Aquino *et al.*, 2016)		Improved Version (this work)	
Variety	$\overline{RC}$	$\overline{PR}$	$\overline{RC}$	$\overline{PR}$
Tempranillo	0.9511abc	0.9101*a*	0.9663a	0.9556*a*
Semillon	0.9664ab	0.8349*bc*	0.9680a	0.9488*a*
Sauvignon Blanc	0.9928a	0.8169*c*	0.9734a	0.9658*a*
Merlot	0.9296bc	0.8999*ab*	0.9767a	0.9351*a*
Grenache	0.9105c	0.8715*abc*	0.9664a	0.8974*b*
Cabernet Sauvignon	0.9668ab	0.870*8abc*	0.9803a	0.9506*a*
Chenin Blanc	0.9832a	0.8852*ab*	0.9821a	0.9696*a*
Overall	0.9572A	0.8705*A*	0.9735B	0.9462*B*

**Dissimilar low-case letters within rows represent statistically different means among varieties, while dissimilar capital letters indicate statistically different means between the two versions of the algorithm; the Tukey (1949) test was used at* $p < 0.05$.

Table 4. Results of the vitisBerry application measured on 144 images using two different smartphones. Results are given detailed per grapevine variety.

	Sony Xperia Z5		BQ Aquaris E5	
Variety	$\overline{RC}$	$\overline{PR}$	$\overline{RC}$	$\overline{PR}$
Airen	0.8865a	0.9475*ab*	0.8291a	0.9524*ab*
Chardonay	0.9043a	0.9391*ab*	0.8658abc	0.9308*ab*
Cabernet Sauvignon	0.9078a	0.9499*ab*	0.8660abc	0.9424*ab*
Grenache Blanc	0.9169a	0.9506*ab*	0.8741abc	0.9159*ab*
Grenache	0.8837a	0.9121*a*	0.8538ab	0.9342*ab*
Merlot	0.8870a	0.9632*b*	0.8303ab	0.9727*b*
Malvasía	0.9202a	0.9388*ab*	0.8994abc	0.9060*a*
Pinot Meunier	0.9113a	0.9698*b*	0.9313c	0.9273*ab*
Pinot Noir	0.9207a	0.9501*ab*	0.9071bc	0.9481*ab*
Tempranillo	0.9003a	0.9627*b*	0.8697abc	0.9575*ab*
Syrah	0.9387a	0.9689*b*	0.8820abc	0.9427*ab*
Viognier	0.9209a	0.9575*b*	0.9059abc	0.9407*ab*
Overall	0.9082A	0.9508*A*	0.8762A	0.9392*A*

**Dissimilar low-case letters within rows represent statistically different means among varieties, while dissimilar capital letters indicate statistically different means of the algorithm's performance running on each device; the Tukey (1949) test was used at* $p < 0.05$.

FIGURES

(a) (b)

Figure 1. (a) Capturing box hand-made with a Din A3 black cardboard. (b) Example of image capture.

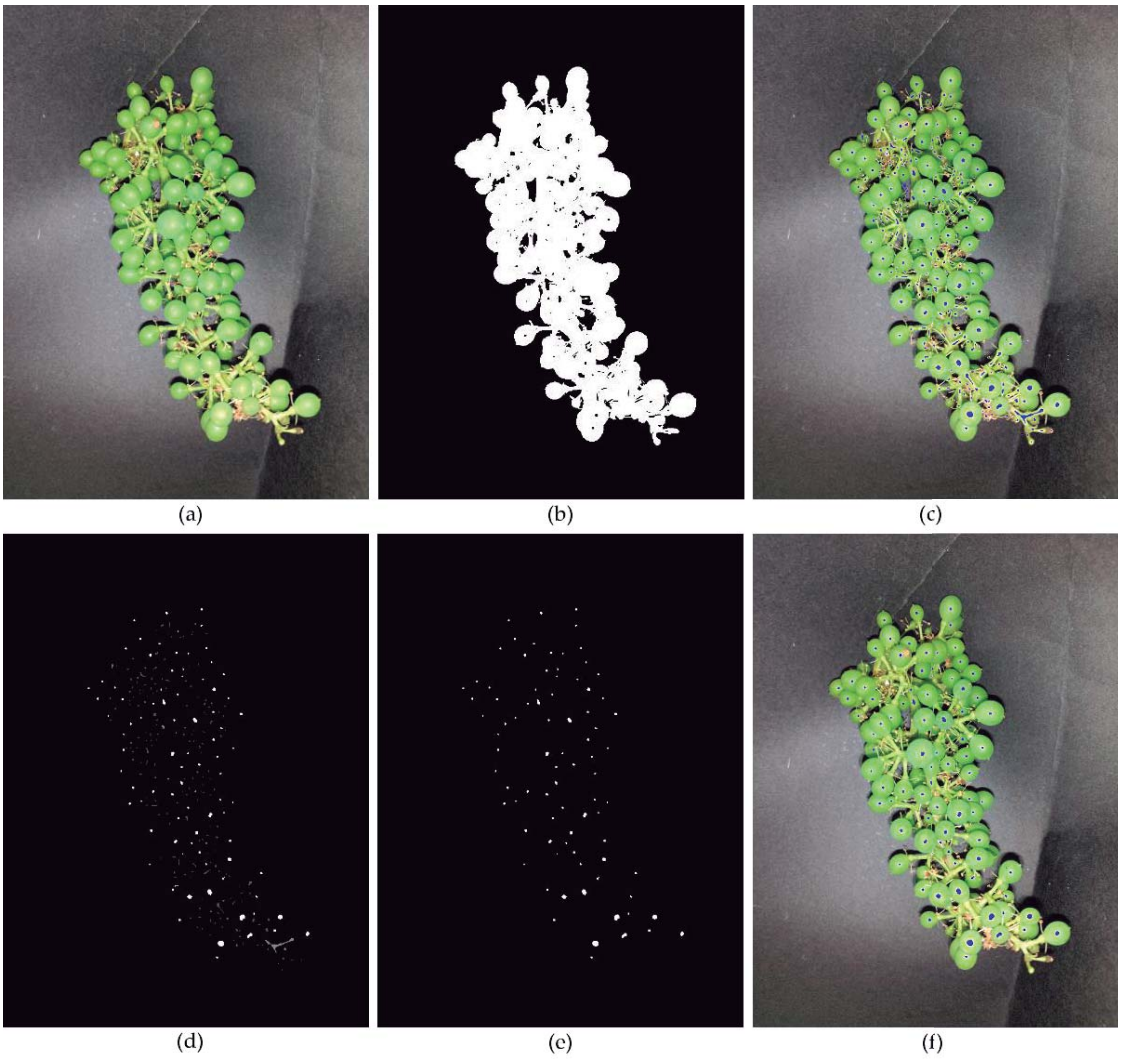

Figure 2. Illustration of the algorithm for berry segmentation on cluster images included in the vitisBerry application: (a) original image; (b) extracted ROI; (c) set of berry candidates represented in blue colour on the original image (a); (d) image in which each candidate is represented according to its computed probability of being berry (the brighter, the higher probability); (e) binary image illustrating the candidates confirmed as berry after thresholding (d); (f) final result showing the found berries.

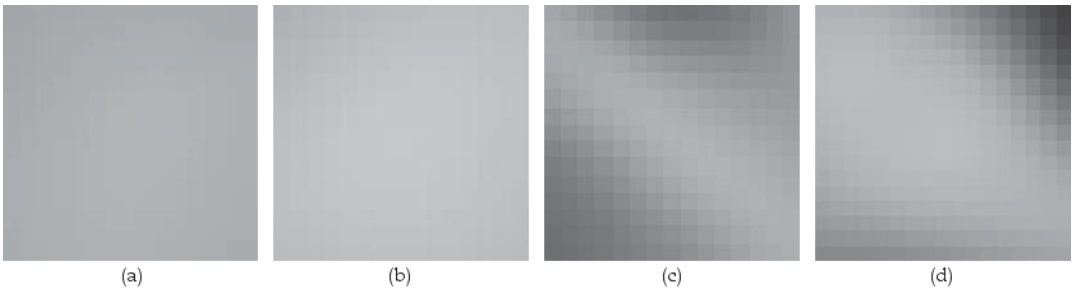

Figure 3. Application of the principle of entropy to berry detection. From (a) to (d), subimages extracted, centred in the centroid of a berry candidate, from the L channel of the original image are shown. (a) and (b) correspond to real berries, showing consistent distribution of pixel values. (c) and (d) come from false positives located in the rachis of the cluster, showing a more randomly distribution of pixel values. The entropy values calculated in subimages (a) to (d) were 3.7363, 4.0292, 5.3078 and 5.6714, respectively. Note that higher entropy values indicate more randomly distributed values.

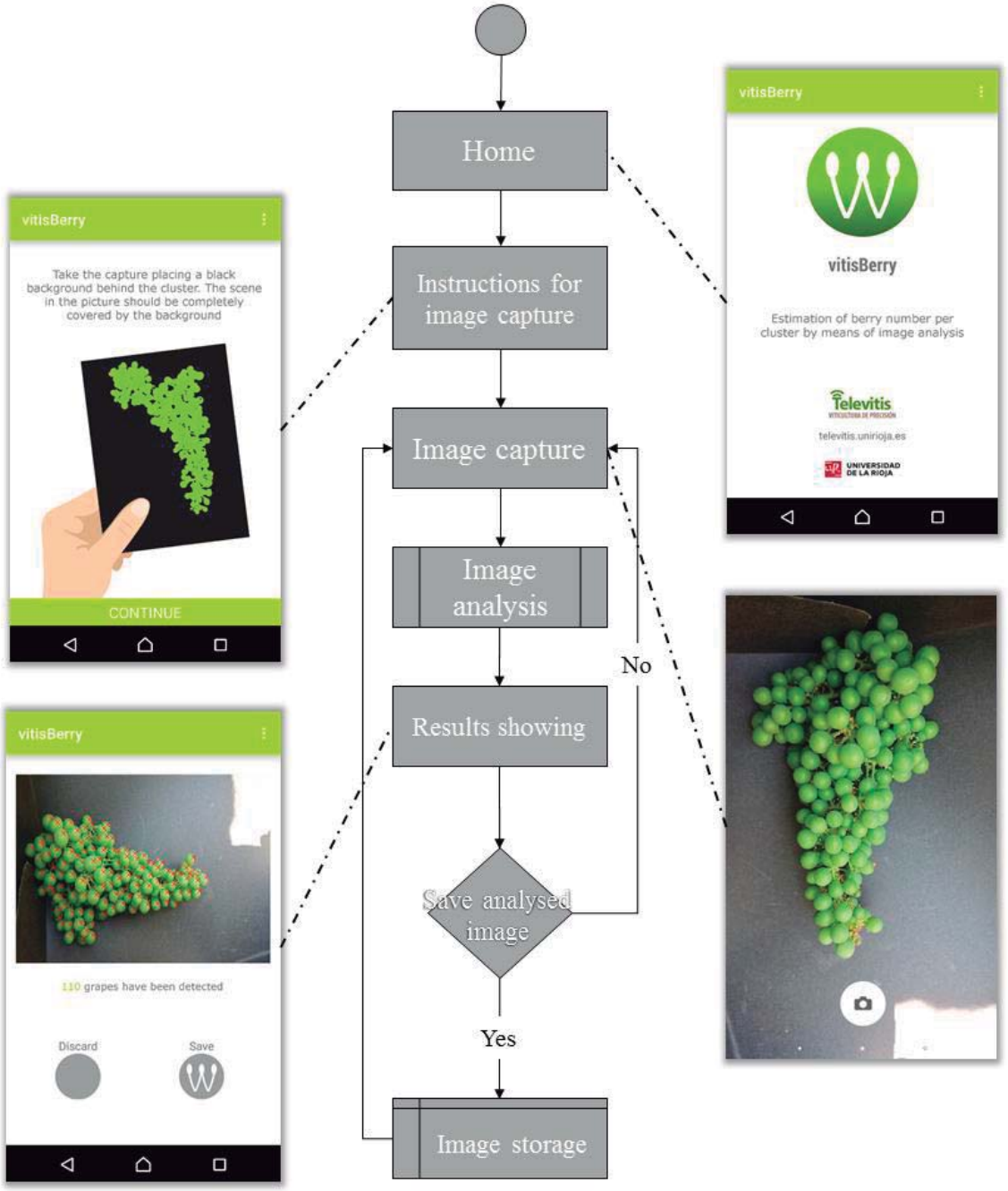

Figure 4. vitisBerry's flow-chart diagram illustrated with application's screenshots.

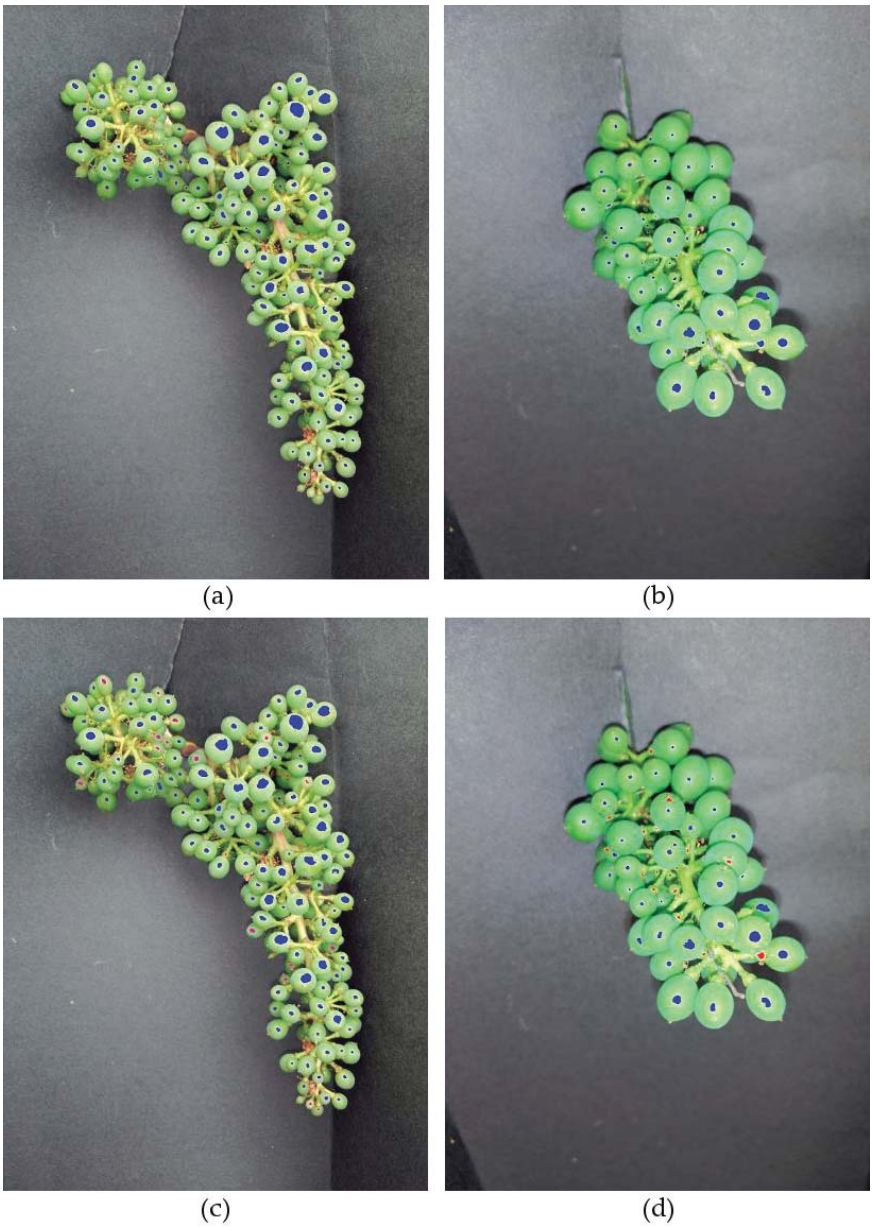

Figure 5. Comparison of results obtained with the original image analysis algorithm presented in Aquino et al. (2016) and the improved version described here. (a) and (c) are results from the original version, in which the found berries are represented in blue colour. (b) and (d) are results from the improved version in which purple candidates were rejected by the original algorithm but validated by the improved version, red candidates were validated by the original algorithm but rejected by the improved version and blue candidates were validated by both versions.

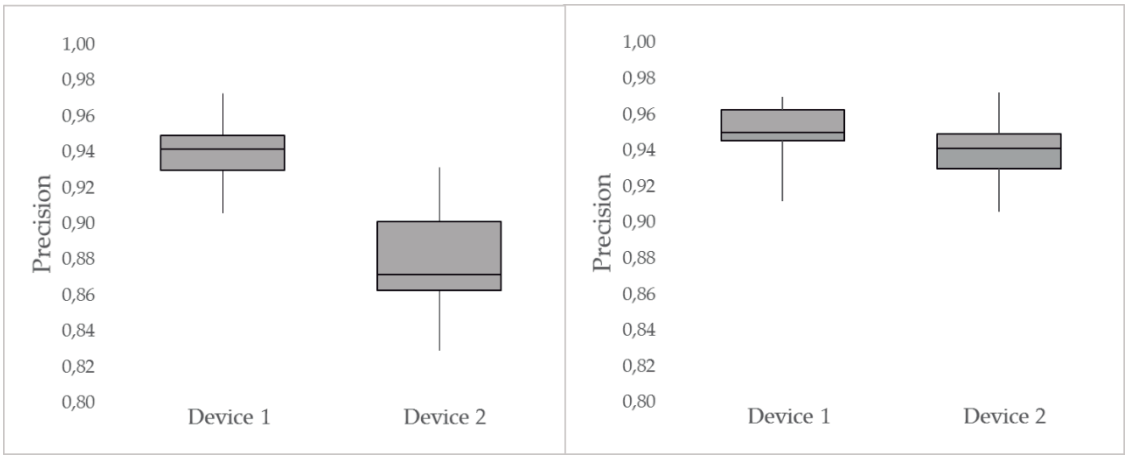

Figure 6. Box and whisker plots comparing the performance of the vitisBerry application running on two different devices in terms of Recall (a) and Precision (b).

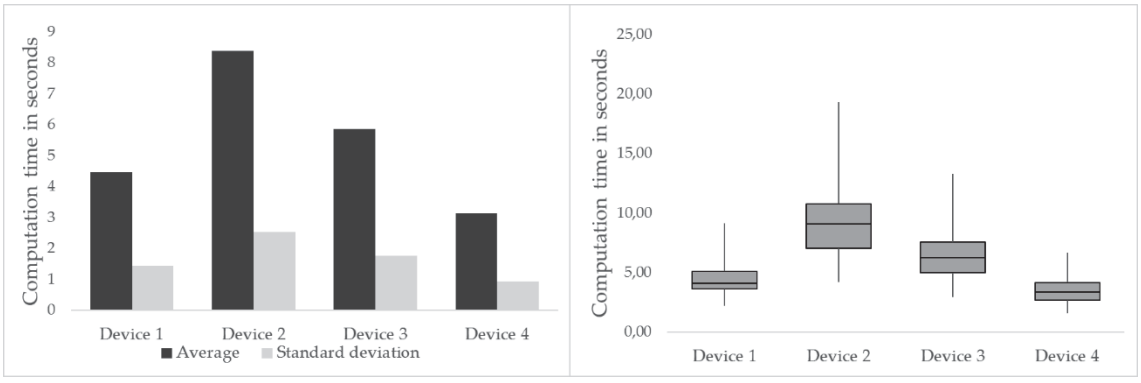

Figure 7. VitisBerry's computational efficiency study consisting on analysing 144 images with four different devices: (a) measured average and standard deviation computation time for the four devices used for the experiment; (b) box and whisker plots for the for the same experiment shown in (a).

Figure 1

Click here to download Figure Figure1.tif

(a) (b)

Figure 2

Click here to download Figure Figure2.tif

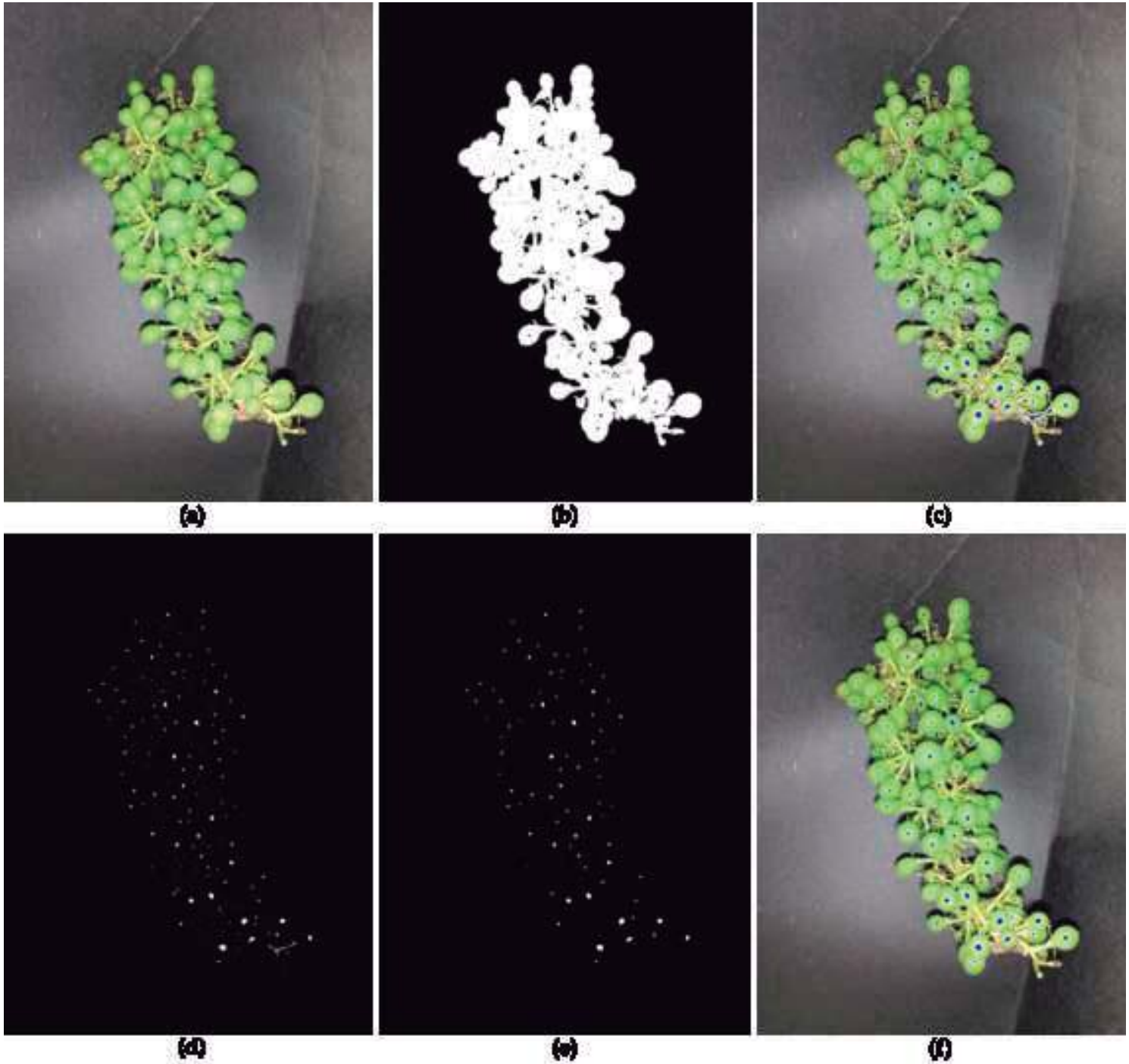

Figure 3

Click here to download Figure Figure3.tif

Figure 4

Click here to download Figure Figure4.tif

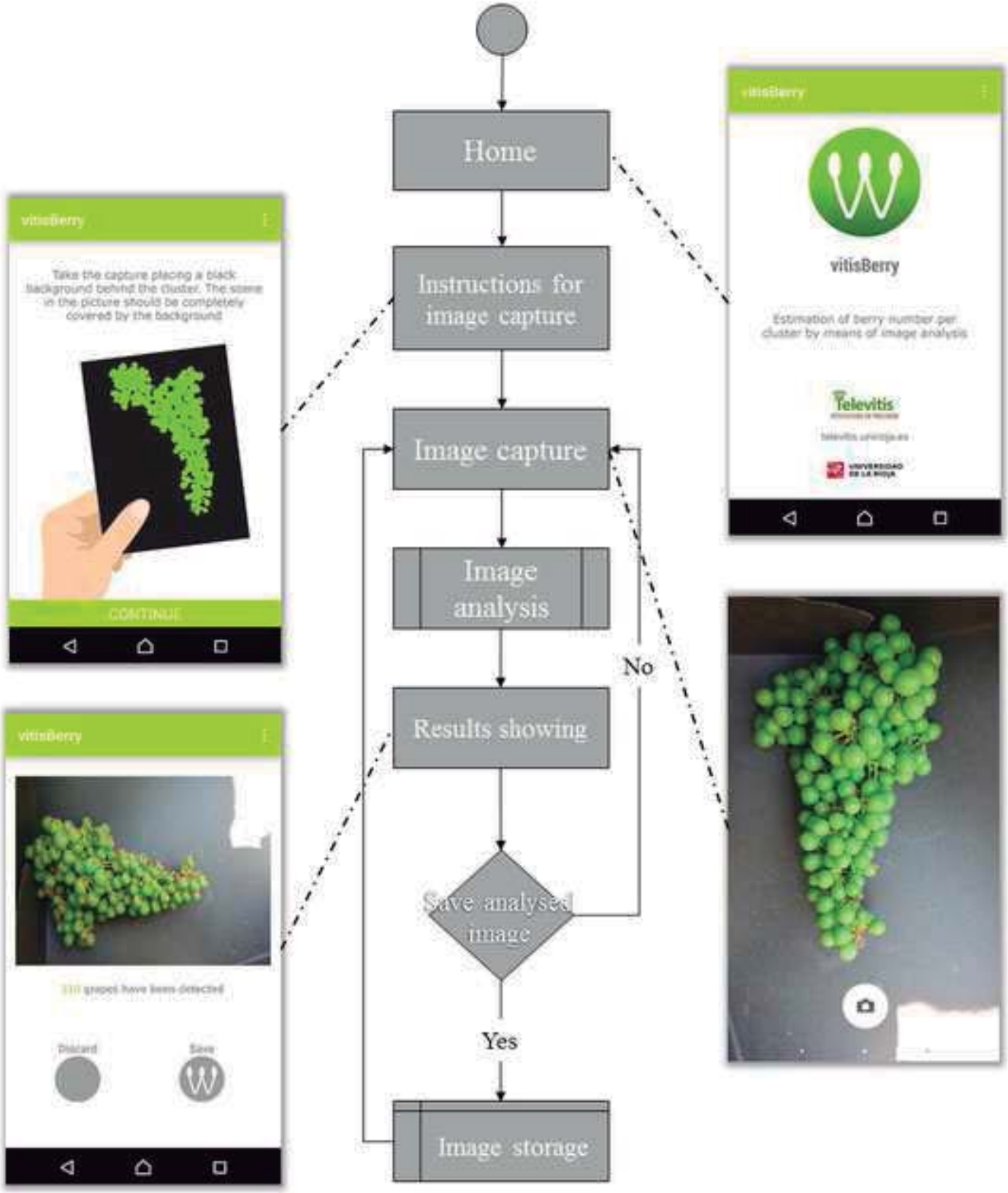

Figure 5

Click here to download Figure Figure5.tif

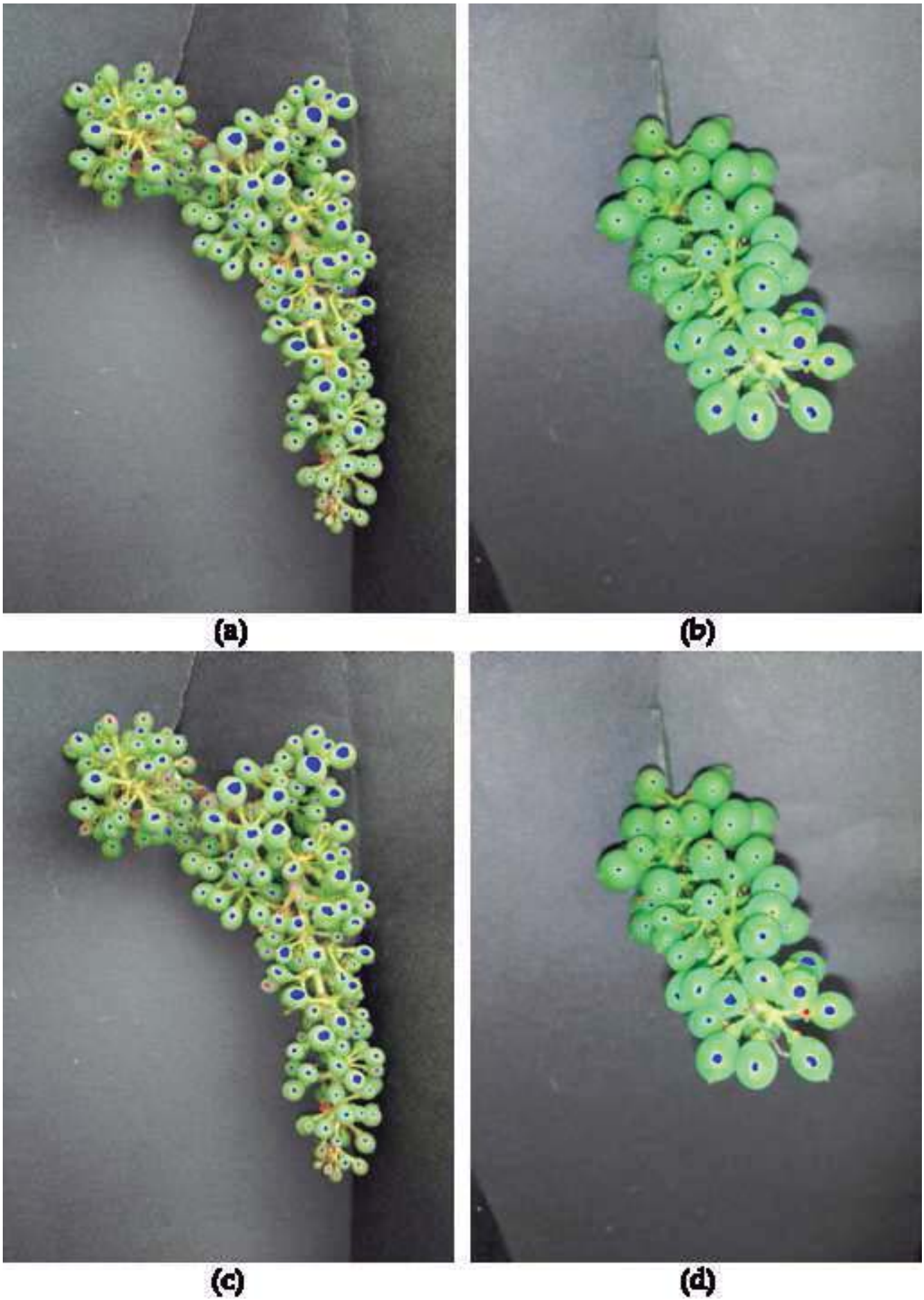

Figure 6

Click here to download Figure Figura6.tif

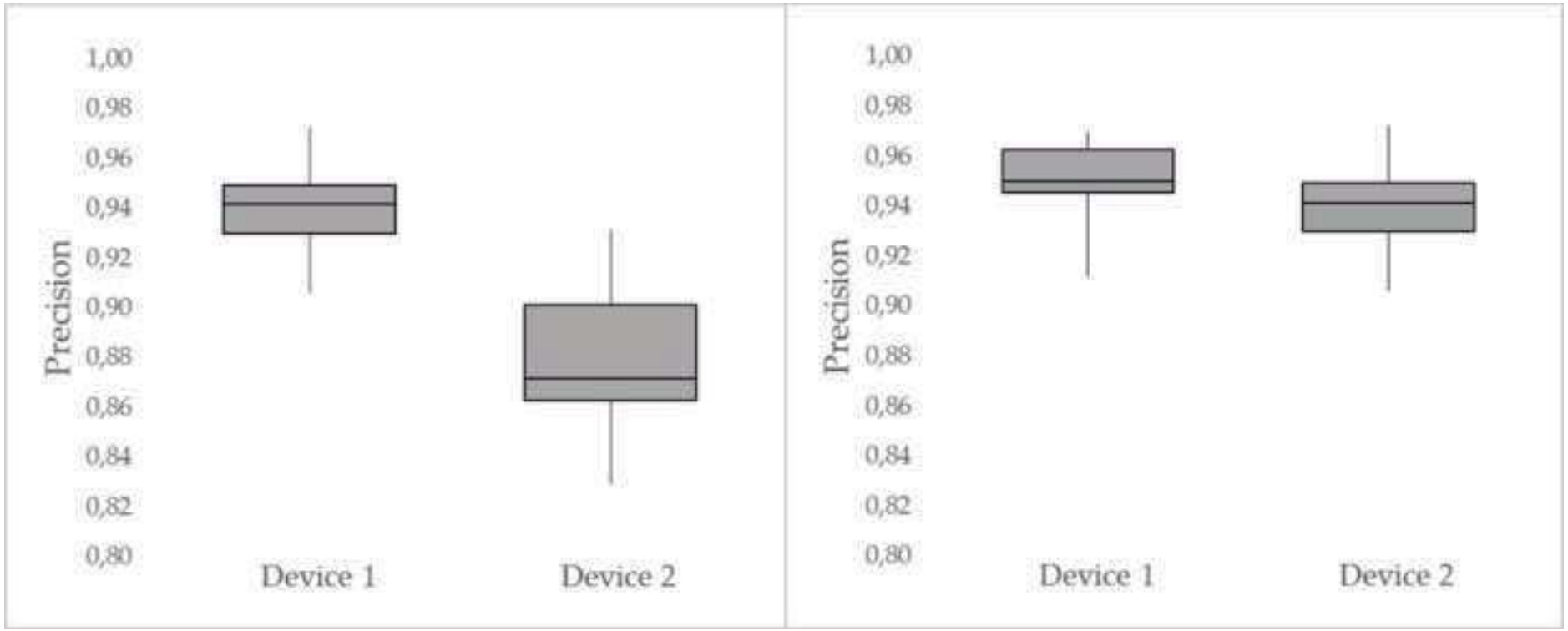

Figure 7

Click here to download Figure Figura7.tif