ACCEPTED VERSION

Olive fly sting detection based on computer vision

Abdullah Beyaz[a], Diego Manuel Martínez Gila[b,*], Juan Gómez Ortega[b], Javier Gámez García[b]

[a] Department of Agricultural Machinery and Technologies Engineering, Faculty of Agriculture, Ankara University, Ankara, Turkey
[b] Robotics, Automation and Computer Vision Group, University of Jaén, Campus Las Lagunillas s/n, 23071 Jaén, Spain

A R T I C L E I N F O

Keywords:
Olive fruit
Olea europaea
Olive fly
Bactrocera oleae
Olive fly sting
Image processing
Computer vision

A B S T R A C T

Olive fly *(Bactrocera oleae* Rossi) is a parasite that infects olive fruit by stinging the fruit to deposit its eggs. The infection damages the fruit during its growth and affects the quality of the olive oil as a final product. Detection of olive fly stings in the virgin olive oil production process is therefore critical. This research aimed to develop automatic methodologies based on computer vision to detect fly stings in infected olive fruit samples. Different methodologies to detect defective areas on the surface of the fruit and classify them between stings and other bruises are described. The methodology to detect defective areas reached a success ratio of 93% and best classification algorithms reached a success ratio close to 80%. The proposed methodology could be applied at reception of fruit for input of the virgin olive oil production process.

1. Introduction

The production of olive oil has increased from 1.5 Mt to about 3 Mt from 1990 to 2015, with 90% of this quantity being from Spain, Italy, and Greece according to the International Olive Council. Olive fly (*Bactrocera oleae*) damage is increasing, affecting fruit properties and consequently, olive oil production (Daane and Johnson, 2010; Imperato et al., 2012; Spadafora et al., 2008). The damage related to the olive fly occurs when the adult female insect lays eggs in the fruit near the olive pericarp. The fly normally selects fresh and green olive fruit. The larvae consumes the flesh during fruit development (Gonçalves et al., 2012). Fig. 1 illustrates olive fruit affected by olive fly sting. Damage produces huge economic losses, e.g. Genc (2013) estimated losses at about eight hundred millions of dollars per year. The Greek Ministry of Agriculture has spent twenty millions of euros since 1937 in protection plans against the olive fruit fly (Varikou et al., 2016).

The olive fly damage can be classified in two types according to the quality and the quantity of olive oil produced (Malheiro et al., 2016, 2015; Moscetti et al., 2015). For olive oil quality, the concentrations of compounds such as phenols, volatiles, chlorophylls, carotenoids, tocopherols, fatty acids and sterols are negatively affected, and the oxidative stability of the oils decreased (Gucci et al., 2012).

Currently, olive fly stings are detected before or after the harvesting process by human experts manually looking at the fruit surface. In green fruit, the sting can be easily seen with the human eye, since a brown spot appears on the olive fruit skin after fruit fly infection. The aim of this work is to employ computer vision to develop automatic methodologies to detect olive fly stings in green olive fruit. As a final goal, the acquired knowledge through this research could be used together with a classification machine to remove infected olive fruit batches at the beginning of the virgin olive production process. In the literature, different examples of computer vision techniques have been applied to the detection of defects and diseases on fruit and vegetables (Ding and Taylor, 2016; Fukatsu et al., 2012). One example of this is the detection of similar defects on selected apple cultivars using hyperspectral and multispectral image analysis (Mehl et al., 2002). They obtained the spectral reflectance on apples related to bruises, *Gloeosporum* infection, soil contamination, side rot, scab, black pox, sooty blotch and insect stings. An automatic algorithm to detect infestations based on X-ray images of infected olive fruit obtained an accuracy of 88% for large infestations and over 50% for the smallest infestations (Haff and Pearson, 2007). The overall objective of our work was to evaluate the feasibility of an automatic system based on computer vision to detect fly stings on batches of olive fruit. Different methodologies based on computer vision were developed to quantify defects over the olive fruit skin and to classify them according to stings or bruises. Then, the first objective was focused on detect defective areas on the olive fruit surface (stings and other defects as bruises) and the second was to classify them.

* Corresponding author.
E-mail address: dmgila@ujaen.es (D.M. Martínez Gila).

https://doi.org/10.1016/j.postharvbio.2019.01.003
Received 18 September 2018; Received in revised form 6 January 2019; Accepted 6 January 2019

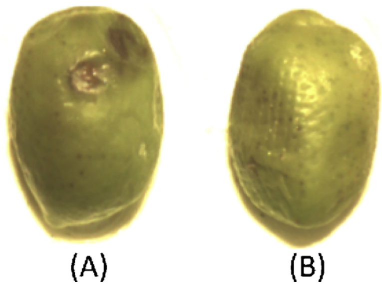

Fig. 1. Olive fruit samples with olive fly sting (A) and without olive fly sting (B).

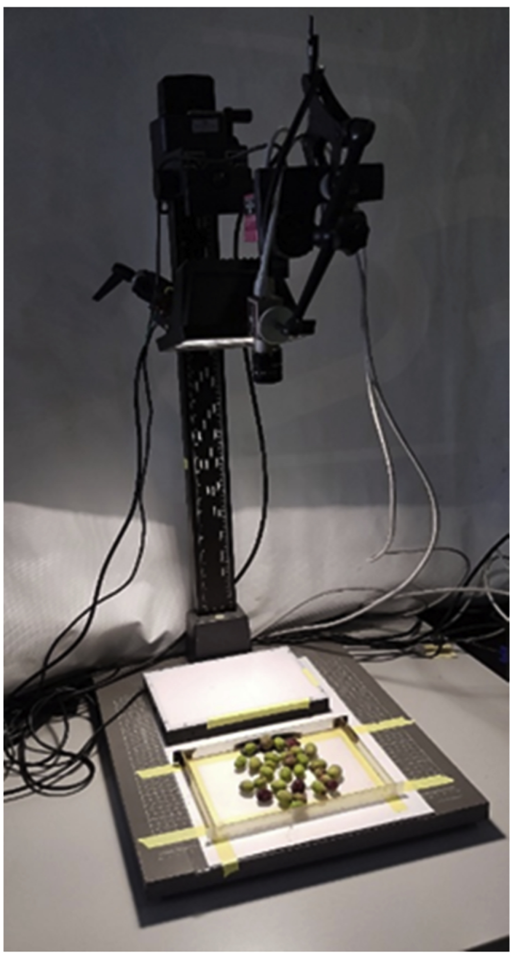

Fig. 2. Olive fruit image capture setup.

2. Material and methods

2.1. Fruit samples

The University of Jaén (southern Spain) has an experimental grove with more than 100 olive trees, cv. Picual, for research purposes. During November 2017, five trees with signs of olive fly attack were selected and 280 fruit with stings and other bruises were manually harvested. Since the fruit were collected in November they were immature, being green and turgid. Maturity of the fruit was evaluated after acquiring the images according to the standard method (Roca and Mínguez-Mosquera, 2001). Also, the fruit were visually inspected, and 417 defects (stings and bruises) were measured. Most fruit had more than one defect on their surface.

2.2. Experimental setup

The image acquisition platform (Fig. 2) was configured with an industrial charge-coupled device (CCD) vision camera (Mako G223C PoE) with 2048 × 1088 pixels of resolution. The color gain parameter for each color channel was adjusted automatically by using the Sapera software through three color patterns, one for each color channel. The general gain parameter was set to 0 to reduce the background noise. The focal length of the camera lens was 35 mm and the lighting system was a halogen spotlight lamp with 500 W. The camera was placed 70 cm far away from the tray with the fruit samples and it was connected to the PC through a gigabit Ethernet link. The images were acquired on demand with the Sapera CamExpert software and they were stored in bitmap (BMP) format. The olives were divided in 10 batches of 28 fruit. One image was acquired for each batch placed in the tray.

2.3. Regions of interest identification

This approach was designed to get two aims; firstly, the experimental work was focused on identifying defects over the fruit skin, that is, unconventional shapes, which modify the green surface of the olive fruit; and secondly, the morphology and the colour of these shapes were employed to classify defects according to its origin (stings versus others). This methodology was coded in Matlab®.

The fruit images were acquired in the RGB colour space and they converted to different colour spaces such as HSV and CIELab in order to get more information. Fig. 3 shows one olive fruit with sting defect surrounded by a dark shape (bruise) in colour spaces. The a* and b* channels from the CIELab colour space have a strong contrast between the sting and the rest of the fruit surface. Channel a* illustrated the circular shape of the sting defect well and therefore was used in following processing steps.

The images were then binarized through a locally adaptive threshold chosen using local first-order image statistics around each pixel (Bradley and Roth, 2007). The adaptive threshold was based on the local mean intensity around the pixel. Algorithms based on global thresholds were discarded because they compute thresholds by using

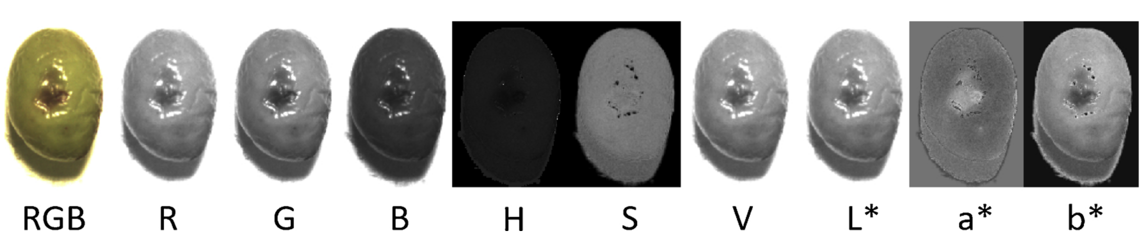

Fig. 3. Olive fruit in different color spaces.

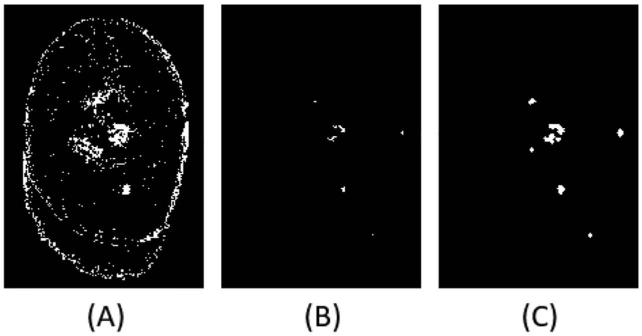

Fig. 4. Results according to the binarization process (A), erosion procedure (B) and dilation process (C).

the undesired information related to the background of the image. Fig. 4A shows the results of the binarization process where the white pixels are the potential defects.

The erosion morphological operator was applied to the binarized images with a disk structuring element to remove small white areas related to the border part of the olive fruits. Fig. 4B shows the results of the erosion procedure. Finally the dilation morphological operator was applied to the images with the same structuring element to increase the defective areas. Fig. 4C presents the resultant image where white areas or ROIs denote the defects.

2.4. Classification algorithms

Morphological and colour properties of the detected ROIs were investigated for their usefulness to classify between bruises and stings. The first step to implement classifiers was to build the feature vectors for each defect. Feature vectors using only morphological properties were hereinafter referred to as $X^M_{n,m}$ and, using just colour information, were referenced as $X^C_{n,m}$. The subscripts "n" and "m" denote the number of sample and the number of feature respectively. The $X^M_{n,m}$ vectors are described in Table 1.

The mask obtained in Fig. 4C was applied to the colour channels that appear in Fig. 3. After that, the $X^C_{n,m}$ vectors were built by using the mean intensity for each channel. The configuration of these vectors are shown in Table 2.

These features were employed as inputs to train and validate different multivariable classification algorithms. The outputs of these classifiers were the class for each sample, that is, sting or bruise. For this work, the version 5.0 of the freeware classification toolbox released by Milano Chemometrics and QSAR research Group (Ballabio and Consonni, 2013) was used. Different classification models were trained as follows.

The first approaches evaluated were the Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) (Hastie et al., 2017). Discriminant analysis models the distribution of the predictors X separately in each of the responses classes (sting or bruise defects), and then uses the Bayes' theorem to flip these around into estimates for the probability of the response category given the value of X.

Table 2
Features using only color properties.

Feature	Description Mean intensity of:
$X^C_{n,1}$	Red channel (RGB space)
$X^C_{n,2}$	Green channel (RGB space)
$X^C_{n,3}$	Blue channel (RGB space)
$X^C_{n,4}$	Hue channel (HSV space)
$X^C_{n,5}$	Saturation channel (HSV space)
$X^C_{n,6}$	Value channel (HSV space)
$X^C_{n,7}$	Lightness channel (CIELAB space)
$X^C_{n,8}$	a* component (CIELAB space)
$X^C_{n,9}$	b* component (CIELAB space)

The LDA algorithm develops discriminant scores for each observation to classify between classes. These scores are obtained by finding linear combinations or canonical variates of the independent variables. For a single predictor variable x the LDA classifier is estimated as Eq. (1) denotes.

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k) \tag{1}$$

where $\hat{\delta}_k(x)$ is the estimated discriminant score, $\hat{\mu}_k$ is the average of all the training observations from the k_{th} class, $\hat{\sigma}^2$ is the weighted average of the sample variances for each of the K classes and $\hat{\pi}_k$ is the prior probability that an observation belongs to the k_{th} class.

This classifier assigns an observation to the k_{th} class according to the largest $\hat{\delta}_k(x)$ discriminant score. The canonical variable loadings of our model and the scores can be seen in the Fig. 5A and B, respectively. Fig. 5A denotes that the features $X^C_{n,1}$ and $X^C_{n,6}$ that reached the highest weights for classification purposes and Fig. 5B shows that there is not a clear border between the two classes, explaining that the percentage of misclassified samples by the model in training phase was 30%.

As for the LDA, the QDA classifier assumes that the observations from each class of Y are drawn from a Gaussian distribution. However, unlike LDA, QDA assumes that each class has its own covariance matrix (Hastie et al., 2017). In this case, the classifier assigns a sample to the class according to the discriminant function that appear in Eq. (2).

$$\hat{\delta}_k(x) = -\frac{1}{2}x^T \sum\nolimits_k^{-1} x + x^T \sum\nolimits_k^{-1} \hat{\mu}_k - \frac{1}{2}\hat{\mu}_k^T \sum\nolimits_k^{-1} \hat{\mu}_k - \frac{1}{2} log \left| \sum\nolimits_k \right| + \log(\hat{\pi}_k) \tag{2}$$

Table 1
Features for morphological properties (*regionprop* algorithm of Matlab).

Feature	Description
$X^M_{n,1}$	Area property containing a scalar that specifies the actual number of pixels in the defective region of the olive fruit.
$X^M_{n,2}$	Major axis length value that specifies the length (in pixels) of the major axis of the ellipse and has the same normalized second central moments as the region.
$X^M_{n,3}$	Minor axis length value.
$X^M_{n,4}$	Eccentricity of the ellipse that has the same second-moments as the region. The eccentricity is the ratio of the distance between the foci of the ellipse and its major axis length. The value is between 0 and 1. (0 and 1 are degenerate cases. An ellipse whose eccentricity is 0 is actually a circle, while an ellipse whose eccentricity is 1 is a line segment).
$X^M_{n,5}$	The perimeter that contains a scalar that specifies the distance around the boundary of the region.

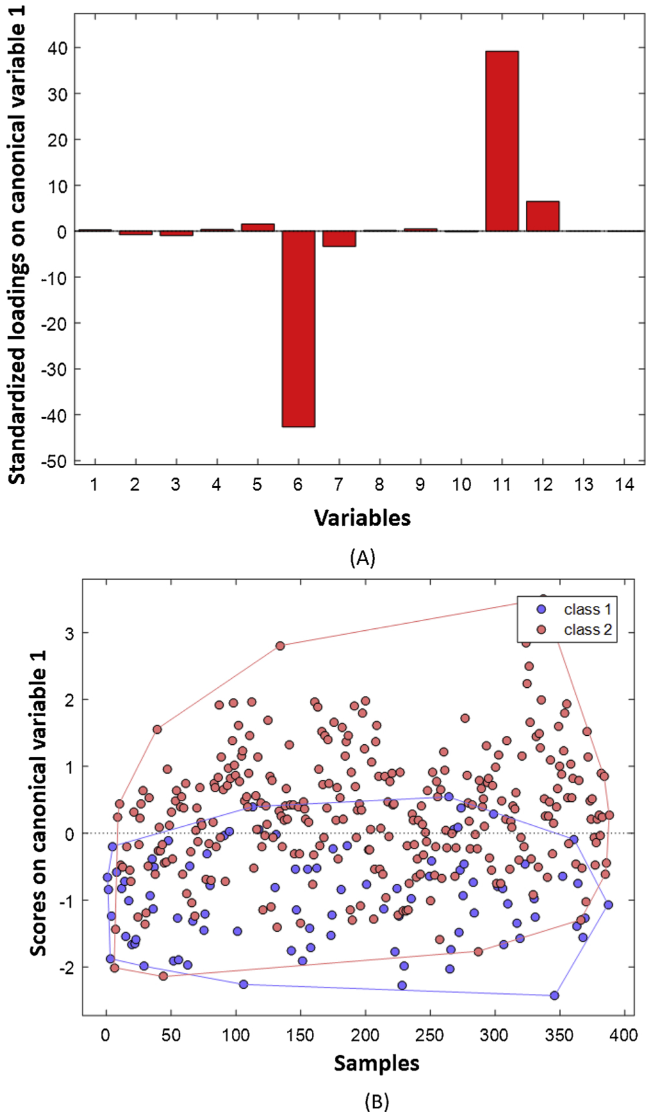

Fig. 5. Canonical variable loadings (A) and the scores (B), where class 1 belongs to sting defects and class 2 belongs to bruise defects.

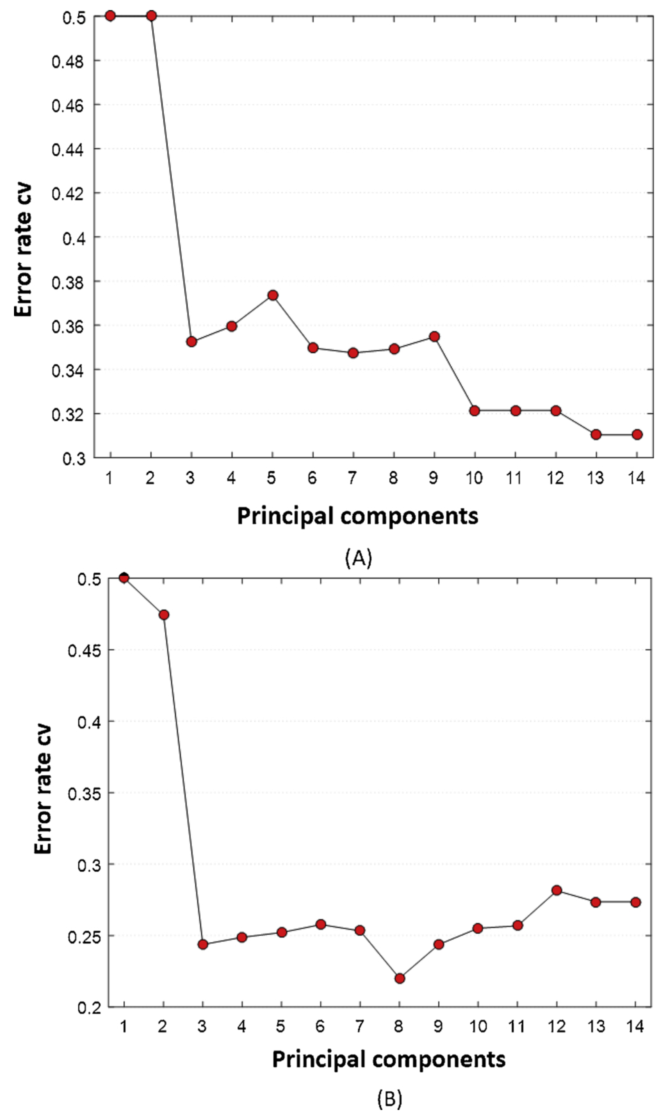

Fig. 6. Optimal number of principal components for LDA (A) and QDA (B).

where $\sum_k$ is a covariance matrix for the kth class. In this case, the percentage of misclassified samples in training phase was the 25% and consequently QDA provides more accurate non-linear classification decision boundaries.

After that, Principal Component Analysis (PCA) (Wold et al., 1987) was evaluated together with LDA and QDA to reduce the number of features by selecting a linear combination of those that explain the variance of the whole dataset. The optimal number of principal components was evaluated according to the error rate in cross validation. Figs. 6A and B show optimal numbers for LDA and QDA models, respectively. PCA improved the results and the best model was PCA-QDA using eight principal components, reducing the misclassified ratio to the 22%.

PCA analysis identifies linear combinations of the original variables which maximize the variance but the class information is not employed, as is available from Partial Least Square Discriminant Analysis (PLSDA) (Ballabio and Consonni, 2013). The general underlying model of multivariate PLSDA appears in the Equations 3 and 4.

$$X = TP^T + E \tag{3}$$

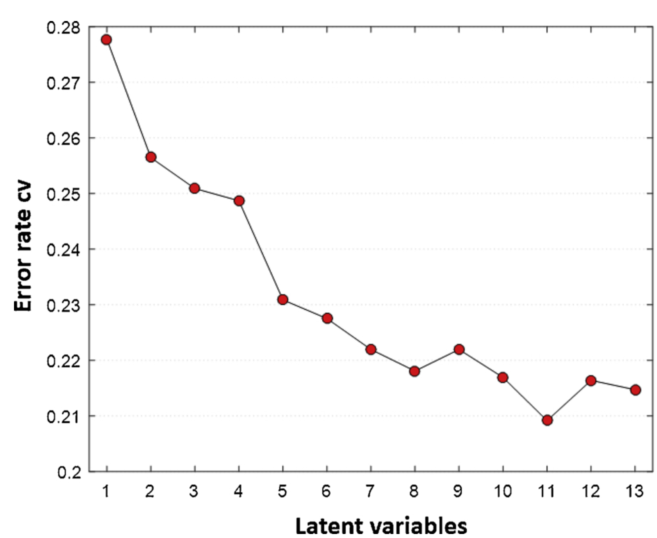

Fig. 7. Error rate in cross validation versus latent variables in the PLSDA model.

$$Y = UQ^T + F \quad (4)$$

where T represents the scores of X and P is the matrix with the loadings of X. The same occurs with Eq. (4). PLSDA searches for latent variables TP^T and UQ^T with a maximum covariance with the class variables Y and produces essentially the same results as LDA, but with the noise reduction and variable selection advantages of Partial Least Squares (Barker and Rayens, 2003). Fig. 7 presents the results when PLSDA was trained to select the optimal number of latent variables and it denotes that the minimum percentage of error (21%) was reached with 11 variables. In this case, X and Y matrices were mean centered prior to analysis.

The last approach was to evaluate the performance of Support Vector Machine (SVM) (Cortes and Vapnik, 1995) classifiers to find the hyperspace where our two classes of defects could be optimally separated. This approach is based on a decision boundary, which can be described as a hyper-plane that is expressed in terms of a linear combination of functions parameterized by support vectors that give the

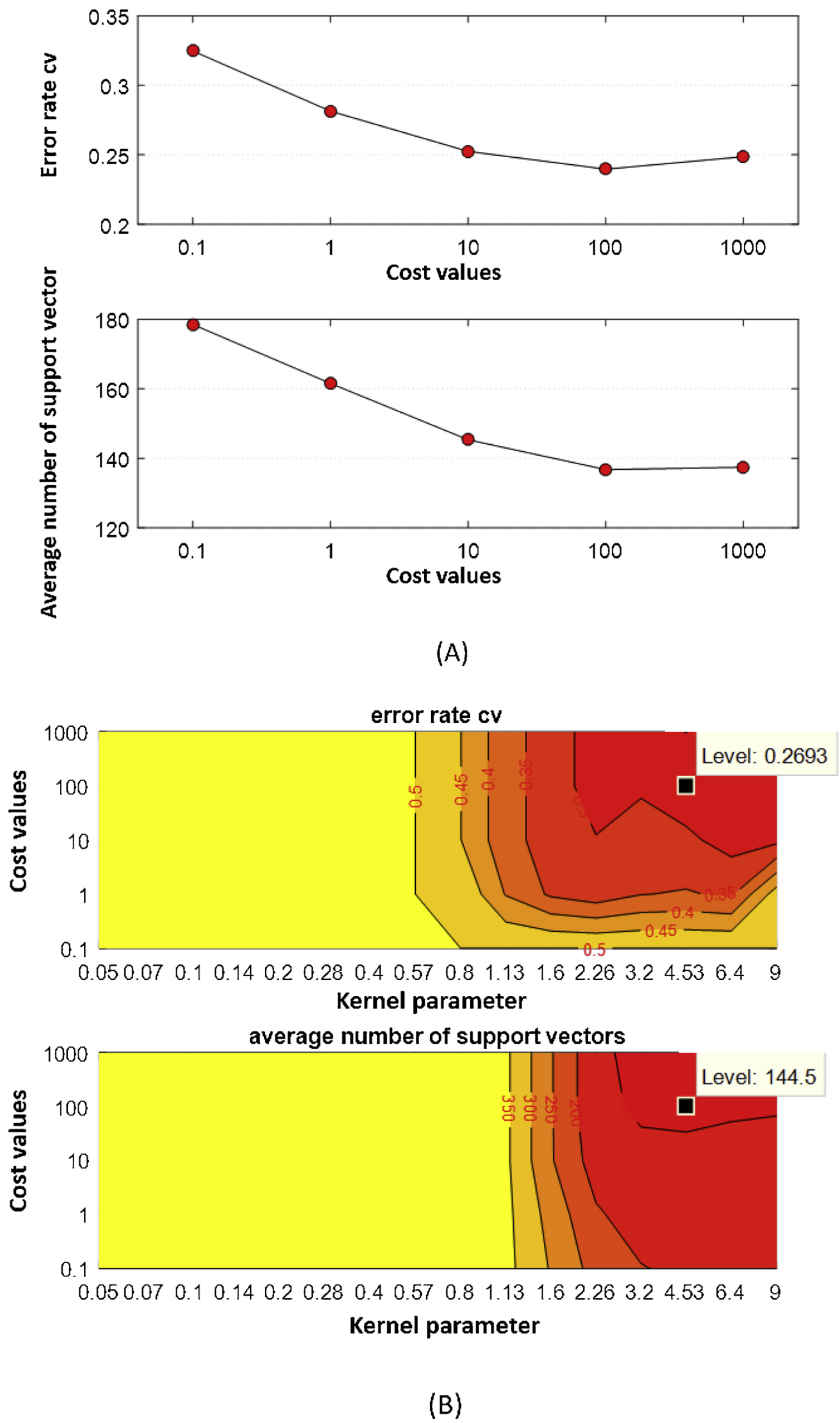

Fig. 8. Error rate with C value ranging from 0.1 to 1000, using the linear kernel (A) and RBF kernel (B).

best separating hyper-plane using a kernel function (Eqs. (5) and (6)).

$$\min_{\beta,\beta_0} \frac{1}{2} \|\beta\|^2 + C\sum_{i=1}^{N} \xi_i \quad (5)$$

$$subject\ to\ y_i(x_i^T\ \beta + \beta_0) \geq 1 - \xi_i\ \ \forall\ i \quad (6)$$

In our context, two kernels were compared: linear and radial basis functions (RBF). The optimal values of the kernels parameters were selected by means of reducing the error in cross-validation. Fig. 8A shows the error rate with cost values (C) ranging from 0.1 to 1000 and using the linear kernel. In this case, the minimum error rate was 24% and this value was reached with a C value equal to 100 and also 100 support vectors. Likewise, Fig. 8B presents the results when RBF kernel was employed. For this kernel the minimum error value was 27% and it was obtained when C values was equal to 100 and the RBF kernel parameter named γ was equal to 4.53. In both cases the results were worse than the previous algorithms.

The results of the different classifiers were then compared. The validation algorithm selected was venetian blinds cross validation 10-fold where the dataset is divided in 10 groups. The cross validation method is an iterative process and for each iteration 9 groups are used to train the model and 1 group is employed to test it.

The performance of the different classifiers were expressed in terms of precision (Pr_g) (Eq. (7)), sensitivity (Sn_g) (Eq. (8)), specificity (Sp_g) (Eq. (9)), accuracy (AC) (Eq. (10)), non-error rate (NER) (Eq. (11)) and error rate (ER) (Eq. (12)).

$$Pr_g = \frac{n_{gg}}{n_g'} \quad (7)$$

$$Sn_g = \frac{n_{gg}}{n_g} \quad (8)$$

$$Sp_g = \frac{\sum_{k=1}^{G}(n_k' - n_{gk})}{n - n_g}\ for\ k \neq g \quad (9)$$

$$AC = \frac{\sum_{g=1}^{G} n_{gg}}{n} \quad (10)$$

$$NER = \frac{\sum_{g=1}^{G} Sn_g}{G} \quad (11)$$

$$ER = 1 - NER \quad (12)$$

where n_g' is the total number of samples assigned to the g-th class, n_{gg} is the number of samples belonging to class g and correctly assigned to class g, n_{gk} is the number of samples belonging to class g and incorrectly assigned to class k, n_g is the total number of samples belonging to the g-th class, n_k' is the total number of samples assigned to the k-th class (Eq. (13)), n is the total number of samples and G is the total number of groups.

$$n_k' = \sum_{k=1}^{G} n_{gk} \quad (13)$$

3. Results and discussion

We focused on the number of defects detected by the image processing methodology and on the ratio of success in classification between stings and bruises defects. The methodology was computed for

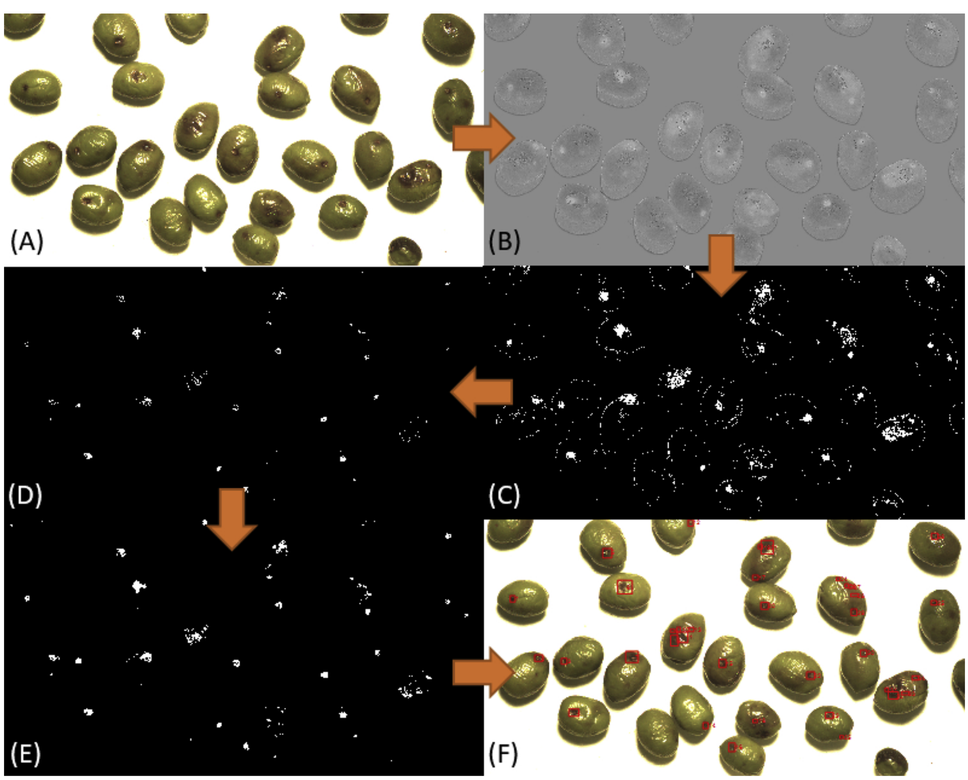

Fig. 9. Images for each step where (A) is the original image, (B) is the *a channel for the same image, (C) is the image after the binarization algorithm, (D) is the resultant image after the erosion algorithm, (E) presents the image after the dilation algorithm and (F) shows the identified defective areas.

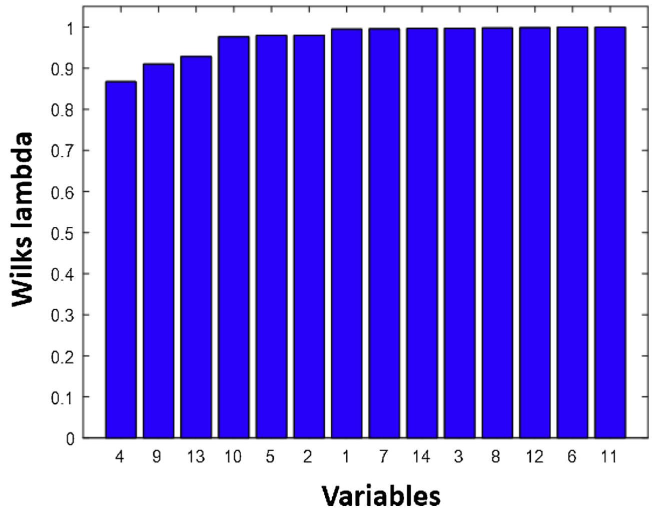

Fig. 10. Wilks' lambda values.

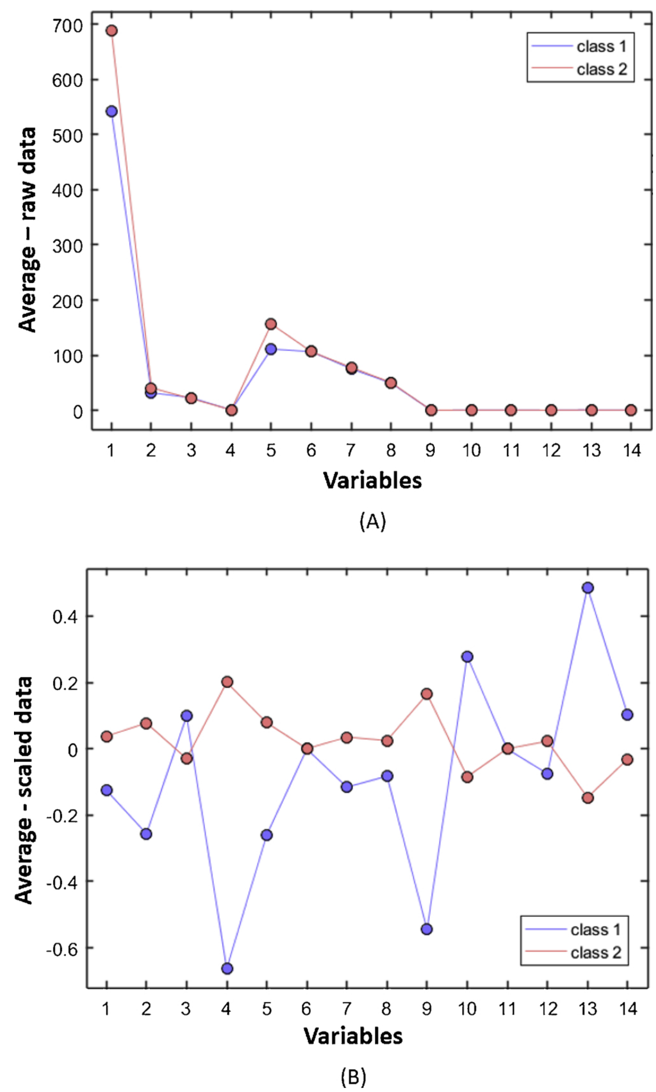

Fig. 11. Dispersion of the dataset according to each feature where class 1 belongs to sting defects and class 2 belongs to bruise defects: (A) presents raw values and (B) shows scaled data by using the mean value and the standard deviation for each feature.

each acquired image. Then, the images were processed to transform the colour space from "RGB" to "CIELab". Then, an adaptive binarization process was applied to the a* channel and non-desirable components in this binary image were removed through morphological operators. Finally the resultant white areas that denote defects were labelled as sting (class 1) or bruise (class 2). This labelling process was visually performed. Fig. 9 shows the resultant images for each step.

The ROI identification method of the 280 fruit defects had a success ratio of 93.05% with a false negative ratio of 6.95% and false positive ratio of 0%. It means that this approach could be applied for fruit used in the virgin olive oil production process.

The second goal was to classify defects between stings and bruises. The morphological and colour features were extracted from the ROIs, and the discrimination power of these features was evaluated by means of Wilks' lambda test (Lawless and Heymann, 2010). The Wilks' lambda is related to the likelihood ratio criterion and ranges between 0 and 1, where values close to 0 indicate that the class means are different. Consequently, variables with low Wilks' lambda values can be considered as optimal features for separating the considered classes. Fig. 10 shows the results for each feature in $X^{M+C}_{388,14}$ and it can be seen that all variables presented Wilks' lambda values higher than 0.8. Therefore, no optimal variable for separating the two classes was identified.

The feature matrixes $X^{M}_{388,5}$ and $X^{C}_{388,9}$ were composed. Fig. 11 shows the dispersion of the dataset according to each column for the matrix $X^{M+C}_{388,14}$; Fig. 11A shows raw data and Fig. 11B presents scaled data using the mean value and the standard deviation for each feature.

In view of the results in the Fig. 11B, the more discriminant property could be the features 4 (eccentricity), 9 (Hue channel in HSV) and 13 (a* value). These results are due mainly to the circularity of the sting defects (see Fig. 1) and the sensitivity of these colour channel to code in grey levels depth differences between sting defects and bruises (see Fig. 3).

Different multivariable classification algorithms were evaluated by using $X^{M}_{388,5}$, $X^{C}_{388,9}$ and $X^{M+C}_{388,14}$ independently as input variables (Table 3). The approach was to contrast information about morphological and colour features of the defects together or separately. For each dataset and classification method resulted in different results; the best results were obtained when the algorithm PLSDA was trained and cross-validated with the $X^{M+C}_{388,14}$ dataset. A non-error rate of 79% was obtained. The total number of defects was 388 and, from these, the number of misclassified stings was 15 while 75 bruises were considered as stings. These false positives were mainly due to similar morphologies and colours between defects.

4. Conclusions

Image processing methodology to detect stings on the olive surface caused by olive flies has been developed. A total of 280 samples of fresh green fruit were picked from an infected crop and the images were acquired by a computer vision setup. 93% of defective areas were detected. To classify defects according to their origin, different morphological and color features were composed and different sorting algorithms were evaluated. The best results were obtained with a PLSDA classification algorithm where the success ratio was close to 80%. The proposed methodologies could be implemented at fruit arrival at the processing factory to evaluate the olive fruit condition before the milling process and classify batches according the fruit quality.

Table 3
Results when different classification algorithms were evaluated. (LDA: Linear Discriminant Analysis; QDA: Quadratic Discriminant Analysis; PCA: Principal Component Analysis; PLSDA: Partial Least Square Discriminant Analysis; SVM: Support Vector Machine; RBF: Radial Basis Function; Pr: precision; Sn: sensitivity; Sp: specificity; AC: accuracy; NER: non-error rate; ER: error rate).

		Pr_g(g = 1/g = 2) (%)	Sn_g(g = 1/g = 2) (%)	Sp_g(g = 1/g = 2) (%)	***AC*** (%)	***NER***(%)	***ER***(%)
$X^{M}_{388,5}$	LDA	31 / 92	55 / 82	82 / 55	78	62	38
	QDA	31 / 92	89 / 40	40 / 89	51	64	36
	PCA-LDA	44 / 79	20 / 92	92 / 20	76	56	44
	PCA-QDA	43 / 90	76 / 70	70 / 76	71	73	27
	PLSDA	46 / 90	74 / 73	73 / 74	74	74	26
	SVM-Linear	70 / 88	58 / 93	93 / 58	85	75	25
	SVM-RBF	64 / 88	61 / 90	90 / 61	83	75	25
$X^{C}_{388,9}$	LDA	61 / 80	22 / 96	96 / 22	79	59	41
	QDA	40 / 86	63 / 71	71 / 63	69	67	33
	PCA-LDA	71 / 81	24 / 97	97 / 24	80	61	39
	PCA-QDA	42 / 87	66 / 72	72 / 66	71	69	31
	PLSDA	37 / 90	77 / 61	61 / 77	64	69	31
	SVM-Linear	61 / 80	19 / 96	96 / 19	78	58	42
	SVM-RBF	56 / 85	50 / 88	88 / 50	79	69	31
$X^{M+C}_{388,14}$	LDA	62 / 85	46 / 92	92 / 46	81	69	31
	QDA	38 / 92	83 / 60	60 / 83	65	72	28
	PCA-LDA	62 / 85	47 / 91	91 / 47	81	69	31
	PCA-QDA	52 / 92	78 / 78	78 / 78	78	78	22
	PLSDA	50 / 94	83 / 75	75 / 83	77	79	21
	SVM-Linear	73 / 88	58 / 94	94 / 58	85	76	24
	SVM-RBF	66 / 89	64 / 90	90 / 64	84	77	23

Acknowledgements

This work was partially supported by the Government of Spain through the research Projects DPI2016-78290-R. Also, the authors want to thank Picualia oil mill for their cooperation in the project.

References

Ballabio, D., Consonni, V., 2013. Classification tools in chemistry. Part 1: linear models. PLS-DA. Anal. Methods 5, 3790. https://doi.org/10.1039/c3ay40582f.

Barker, M., Rayens, W., 2003. Partial least squares for discrimination. J. Chemom. 17, 166–173. https://doi.org/10.1002/cem.785.

Bradley, D., Roth, G., 2007. Adaptive thresholding using the integral image. J. Graph. Tools 12, 13–21. https://doi.org/10.1080/2151237X.2007.10129236.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20, 273–297. https://doi.org/10.1007/BF00994018.

Daane, K.M., Johnson, M.W., 2010. Olive fruit fly: managing an ancient pest in modern times. Annu. Rev. Entomol. 55, 151–169. https://doi.org/10.1146/annurev.ento.54.110807.090553.

Ding, W., Taylor, G., 2016. Automatic moth detection from trap images for pest management. Comput. Electron. Agric. 123, 17–28. https://doi.org/10.1016/j.compag.2016.02.003.

Fukatsu, T., Watanabe, T., Hu, H., Yoichi, H., Hirafuji, M., 2012. Field monitoring support system for the occurrence of *Leptocorisa chinensis*Dallas (Hemiptera: Alydidae) using synthetic attractants, field servers, and image analysis. Comput. Electron. Agric. 80, 8–16. https://doi.org/10.1016/j.compag.2011.10.005.

Genc, H., 2013. An alien species commonly known as native: the olive fruit fly (*Bactrocera oleae* (Rossi)) (Diptera: Tephritidae). 4th ESENIAS Workshop: International Workshop on IAS in Agricultural and Non-Agricultural Areas in ESENIAS Region.

Gonçalves, M.F., Malheiro, R., Casal, S., Torres, L., Pereira, J.A., 2012. Influence of fruit traits on oviposition preference of the olive fly,*Bactrocera oleae* (Rossi) (Diptera: Tephritidae), on three Portuguese olive varieties (Cobrançosa, Madural and Verdeal Transmontana). Sci. Hortic. (Amst.) 145, 127–135. https://doi.org/10.1016/j.scienta.2012.08.002.

Gucci, R., Caruso, G., Canale, A., Loni, A., Raspi, A., Urbani, S., Taticchi, A., Esposto, S., Servili, M., 2012. Qualitative changes of olive oils obtained from fruits damaged by *Bactrocera oleae* (Rossi). HortScience 47, 301–306.

Haff, R.P., Pearson, T.C., 2007. An automatic algorithm for detection of infestations in X-ray images of agricultural products. Sens. Instrum. Food Qual. Saf. 1, 143–150. https://doi.org/10.1007/s11694-007-9018-9.

Hastie, T., Tibshirani, R., Friedman, J., 2017. Linear methods for classification. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, Stanford, California p. 745.

Imperato, A., Corrado, G., Alagna, F., Varricchio, P., Baldoni, L., Rao, R., 2012. Olive molecular response to attack of *Bactrocera oleae*: identification of up-regulated genes in infested olive fruits. Acta Hortic. 125–128.

Lawless, H.T., Heymann, H., 2010. Data Relationships and Multivariate Applications. pp. 433–449. https://doi.org/10.1007/978-1-4419-6488-5_18.

Malheiro, R., Casal, S., Baptista, P., Pereira, J.A., 2015. A review of *Bactrocera oleae* (Rossi) impact in olive products: from the tree to the table. Trends Food Sci. Technol. https://doi.org/10.1016/j.tifs.2015.04.009.

Malheiro, R., Casal, S., Cunha, S.C., Baptista, P., Pereira, J.A., 2016. Identification of leaf volatiles from olive (*Olea europaea*) and their possible role in the ovipositional preferences of olive fly, *Bactrocera oleae*(Rossi) (Diptera: Tephritidae). Phytochemistry 121, 11–19. https://doi.org/10.1016/j.phytochem.2015.10.005.

Mehl, P.M., Chao, K., Kim, M., Chen, Y.R., 2002. Detection of defects on selected apple cultivars using hyperspectral and multispectral image analysis. Appl. Eng. Agric. 18, 219–226. https://doi.org/10.13031/2013.7790.

Moscetti, R., Haff, R.P., Stella, E., Contini, M., Monarca, D., Cecchini, M., Massantini, R., 2015. Feasibility of NIR spectroscopy to detect olive fruit infested by Bactrocera oleae. Postharvest. Biol. Technol. 99, 58–62. https://doi.org/10.1016/j.postharvbio.2014.07.015.

Roca, M., Mínguez-Mosquera, M.I., 2001. Changes in chloroplast pigments of olive varieties during fruit ripening. J. Agric. Food Chem. 49, 832–839. https://doi.org/10.1021/jf001000l.

Spadafora, A., Mazzuca, S., Chiappetta, F.F., Parise, A., Perri, E., Innocenti, A.M., 2008. Oleuropein-specific-β-glucosidase activity marks the early response of olive fruits (*Olea europaea*) to mimed insect attack. Agric. Sci. China 7, 703–712. https://doi.org/10.1016/S1671-2927(08)60105-4.

Varikou, K., Garantonakis, N., Birouraki, A., Ioannou, A., Kapogia, E., 2016. Improvement of bait sprays for the control of *Bactrocera oleae*(Diptera: Tephritidae). Crop Prot. 81, 1–8. https://doi.org/10.1016/j.cropro.2015.11.007.

Wold, S., Esbensen, K., Geladi, P., 1987. Principal component analysis. Chemometr. Intell. Lab. Syst. 2, 37–52. https://doi.org/10.1016/0169-7439(87)80084-9.