ACCEPTED VERSION

Expert system based on computer vision to estimate the content of impurities in olive oil samples

P. Cano Marchal *, D. Martínez Gila, J. Gámez García, J. Gómez Ortega

Robotics, Automation and Computer Vision Group, University of Jaén, Spain

A B S T R A C T

The determination of the content of impurities is a very frequent analysis performed on virgin olive oil samples, but the official method is quite work-intensive, and it would be convenient to have an alterna-tive approximate method to evaluate the performance of the impurity removal process. In this work we develop a system based on computer vision and pattern recognition to classify the content of impurities of the olive oil samples in three sets, indicative of the goodness of the separation process of olive oil after its extraction from the paste. Starting from the histograms of the channels of the *Red–Green–Blue* (*RGB*), *CIELAB* and *Hue-Saturation-Value* (*HSV*) color spaces, we construct an initial input parameter vector and perform a feature extraction previous to the classification. Several linear and non-linear feature extrac-tion techniques were evaluated, and the classifiers used were Support Vector Machines (SVMs) and Arti-ficial Neural Networks (ANNs). The best classification rate achieved was 87.66%, obtained using Kernel Principal Components Analysis (KPCA) and a grade-3-polynomial kernel SVM. The best result using ANNs was 82.38%, yielded by the use of Principal Component Analysis (PCA) with the Perceptron.

1. Introduction

The application of computer vision to develop tools to measure the quality of goods and the performance of processes is ubiquitous (Malamas et al., 2003), and the food industry is one of its main fields of application (Brosnan and Sun, 2002; Du and Sun, 2006; Ruiz-Altisent et al., 2010; Rosell and Sanz, 2012). However, there are not many applications of computer vision to the olive oil industry, being those mainly to detect defects in the olives (Diaz et al., 2004) and determine its maturity (Furferi et al., 2007, 2010).

The determination of the content of impurities in virgin olive oil is a very frequent analysis performed on virgin olive oil samples. It is useful whenever there is a bulk oil transaction, as the maximum content allowed of moisture and impurities is fixed, and if the olive oil contains higher levels than those, the price paid is decreased accordingly. Determination of impurities is also helpful during the olive oil elaboration process, as it enables the operator to check the performance of the moisture and impurities removal process held in the vertical centrifuge or the settling tanks. For this latter application of the analysis the required precision is lower, since it is used only to adjust the process. A sketch of the olive oil elaboration process is presented in Cano Marchal et al. (2011).

The official method established in the international norm ISO 663:2000 requires dissolving the olive oil sample with hexane and filtering it through a previously dried filtering paper. Afterwards, more hexane is filtered through this paper in order to remove any residue of oil, and the filtering paper is introduced into an oven to eliminate the hexane. Finally the hexane-free paper is weighed and the content of impurities determined. This method is quite work-intensive, and it would be convenient to have an alternative approximate method to evaluate the performance of the impurity removal process. To the best knowledge of the authors, results regarding the use of computer vision to estimate the content of impurities in olive oil samples are still scarce, and we have found no reference of any actual industrial application.

The goal of this research work is to develop an approximate method, less work-intensive than the official one, capable of discerning between different contents of impurities indicative of the wellness of the impurity removal process for its use as off-line feedback information by the operators of the process. For that objective we propose and develop a system based on computer vision and pattern recognition to classify the content of impurities of the olive oil samples in three sets.

1.1. Background and previous works

The application of computer vision to the olive oil elaboration process is still a fairly open field, since there are not many reported applications and, to the best knowledge of the authors, even less commercially available products used routinely in the sector.

* Corresponding author.
E-mail address: pcano@ujaen.es (P. Cano Marchal).

PUBLISHER VERSION: http://dx.doi.org/10.1016/j.jfoodeng.2013.05.032

The majority of applications of computer vision to this field have been devoted to classify olives according to the presence of defects or to infer different properties of them, such as ripening index or oil content. Diaz et al. (2004) presented a comparison of three algorithms – Partial Least Squares (PLS), Mahalanobis distance and Artificial Neural Networks (ANNs) – to classify table olives according to their defects, and concluded that the best results are obtained using ANNs, with a classification accuracy of 90%. Riquelme et al. (2008) proposed using three consecutive discriminant analysis for the same objective, yielding a 75% of correct classification rate in the validation phase.

Regarding the ripening index of olives, Furferi et al. (2010) developed a system to predict it based on computer vision and a further refinement of the results using an ANN. This ANN combined the preliminary ripening index provided by the initial algorithms with chemical parameters obtained from historical curves from the region where the olives came from.

For the prediction of the oil content of olives, Ram et al. (2010) constructed two models based on linear regressions and ANNs. The best results were achieved using ANNs, obtaining linear correlations of 0.81 for Souri and 0.87 for Picual olives. To conclude with the computer vision applications to olives reported, it is noteworthy the work by Gatica et al. (2011), where they presented a method to recognise the diameter of the olives from images of the olive tree.

There are some works related to visual characteristics of olive oil that do not apply computer vision techniques, but that are worth mentioning. Moyano et al. (2008b,a) related the relationship between color parameters of the oils in different color spaces to their chlorophylls and carotenoids indexes using a spectrophotometer. In turn, Gordillo et al. (2011) studied the influence of turbidity grade on the color and appearance of virgin olive oil samples, using a turbidimeter and a spectrophotometer.

Concerning the applications of computer vision techniques to other liquid products in the food industry, it has been reported its usage to quantify the total quantity of bacteria in juice (Jin and Yin, 2010). In this work, the authors constructed a feature vector with different parameters, such as Area, Perimeter and Circularity, and used biomimetic pattern recognition to segment the bacteria and distinguish it from cells and impurities. Afterwards, a count of the segmented regions was held to determine the quantity of bacteria.

Lastly, it is relevant to highlight the work by Hepworth et al. (2004), where the authors employed computer vision to determine the size and velocity of bubbles in beer.

1.2. Problem description

The content of impurities is a determination that is usually demanded jointly with the determination of the moisture content of the sample, as these are substances exogenous to the olive oil and affect negatively its conservation. The method to determine the moisture content is to weigh a certain quantity of the sample and introduce it to a drying oven until the weigh of the sample is constant. Then, the moisture content can be calculated simply by means of the weigh difference of the sample before and after the drying process (Fig. 1).

During this drying process, the viscosity of the oil decreases as a result of the increase in the oil temperature – typically, the drying is held at 110 °C. This decrease of the viscosity allows the impurities to sediment in the porcelain evaporating basin. Fig. 1 shows an olive sample before it is introduced in the drying oven and after the drying process, where the deposition of the impurities is visible. This phenomena highlights visual differences between samples of different impurity content, thus suggesting the use of images of the samples after the drying process to construct an approximate method for rapid estimation of the impurity content.

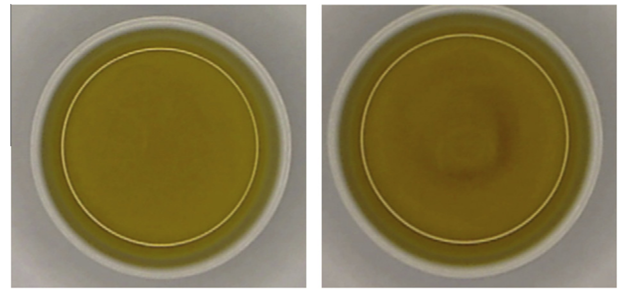

Fig. 1. Olive oil sample before (left) and after (right) drying process, illustrating the sedimentation of the impurities after the drying process.

The color of the olive oil samples is not constant and it is not related to the content of impurities. Besides, the patterns that the impurities form vary within the same level of impurities. Fig. 2 shows different samples for the three different levels of impurities considered, where the aforementioned differences between samples can be noticed.

The impurity content of the classes created for the classifier were chosen to be indicative of the goodness of the separation process, according to typical values indicated by practising olive oil elaboration experts. Impurity content below 0.04% may be regarded as low, a content between 0.04% and 0.05% (both included) is acceptable and values over 0.05% are indicative that the separation process can be improved.

2. Methods

The approach used was to select a candidate input parameter vector derived from the data contained in the images of the samples, then perform a feature extraction before feeding the data to the classifier. The following subsections detail each of these steps.

2.1. Experimental set-up

To get the measurement process constant during all the tests, different issues must be checked before acquiring the information. They are: no dust or dirt, low external light compared to our lighting system (reducing the noise coming from surround light), a constant temperature and no shock or vibration of the setup. All these recommendations are normally fulfilled in a chemical lab.

Fig. 3 shows the experimental set-up used for the image acquisition. The system was composed of a LZ836BP Logitech webcam of 2 MP resolution and a ring-shaped LED illumination device. The ring shaped lighting source used had a power of 17 W, a diameter of 18 cm, and was placed 25 cm above the samples.

The images were acquired using the software provided with the webcam with a Pentium IV standard desktop PC. The image processing, feature extraction and training of the classifiers were developed using an Intel Q9550 CPU, 4 GB of RAM, Windows 7 running PC. The software used to perform the experiments was Matlab, and the library used for the SVM implementation was LIBSVM (Chang and Lin, 2011).

The available sample set was composed of 154 samples, where 82 samples belong to the first class, 48 to the second and 24 to the third. The reason for such an unbalanced frequency of each of the samples is due to the typical occurrence of values which naturally occurs in the oil elaboration process, since the samples used were supplied by a laboratory specialized in olive oil analysis from the ones they received. An attempt was made to construct artificial samples for the higher level of impurities, but the samples

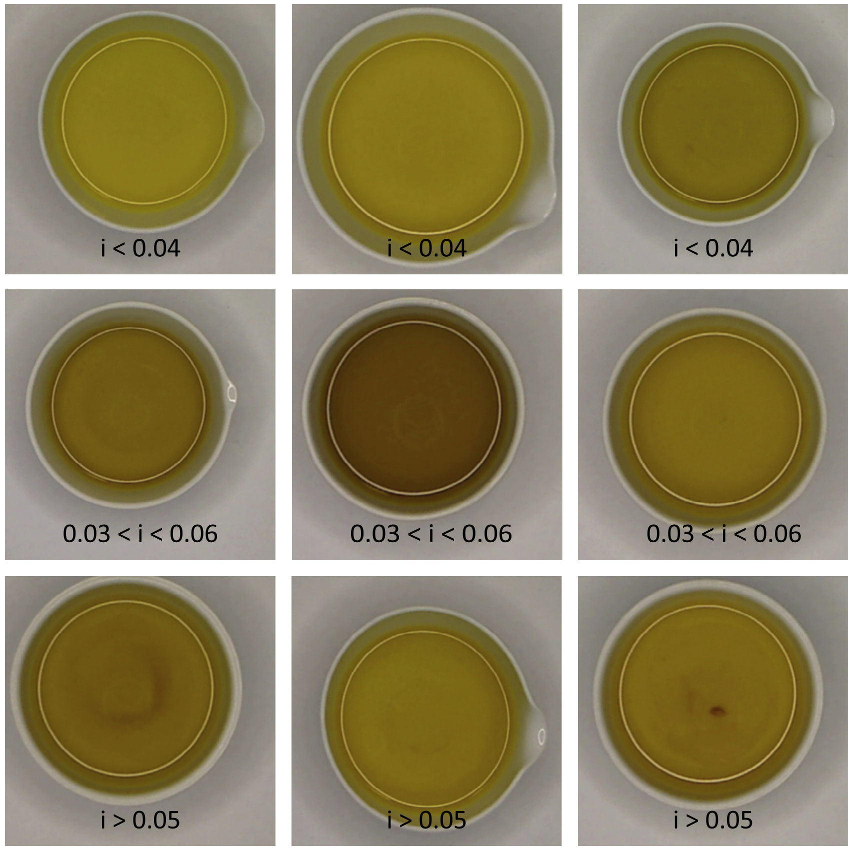

Fig. 2. Different olive oil samples after drying process with their content of impurities, showing the diversity in the color of the oil and patterns formed by the sedimented impurities. *i* denotes the percentage of impurities in the samples. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

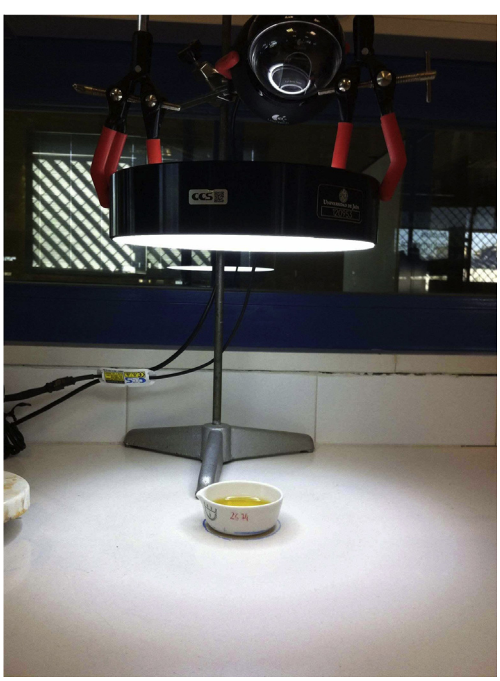

Fig. 3. Image acquisition set-up, composed by the illumination ring and the camera (top). In the bottom of the image is the porcelain basin containing the olive oil sample.

obtained showed notorious differences with the natural ones, so they were discarded.

2.2. Image data and input parameter vector

The data considered as initial input parameter vector was the histogram of each channel of the image in three color spaces: *RGB*, *CIELAB* and *HSV* (Sangwine and Horne, 1998). The election of this preliminary input parameter vector was justified in the fact that it is the color of the pixels what determines if there are any impurities and the total amount of them, independently of their position. We considered these three different color spaces to construct an input parameter vector that contained several non-linear combinations of the same original data (set 1). Fig. 4 shows the histogram of the different channels for each color space of all the olive oil samples used.

However, after testing several feature extraction techniques and parameters for the classifier, the results obtained using this input parameter were not satisfactory.

For the sake of reducing the influence of the color tone of the sample, we considered using a subset of the initial input vector, including only the channels *a* and *b* from the *CIELAB* color space, and *H* and *S* from *HSV* (set 2). Again, the results obtained with this approach were far from adequate.

After some visual analysis of the histograms plots, a new input parameter vector was constructed as follows: the first subset of parameters was defined as the values of each element of each channel of the histogram in descending order; the second subset was defined as the index that the element of the first subset had in the original histogram (set 3) (see Table 1). Fig. 5 shows

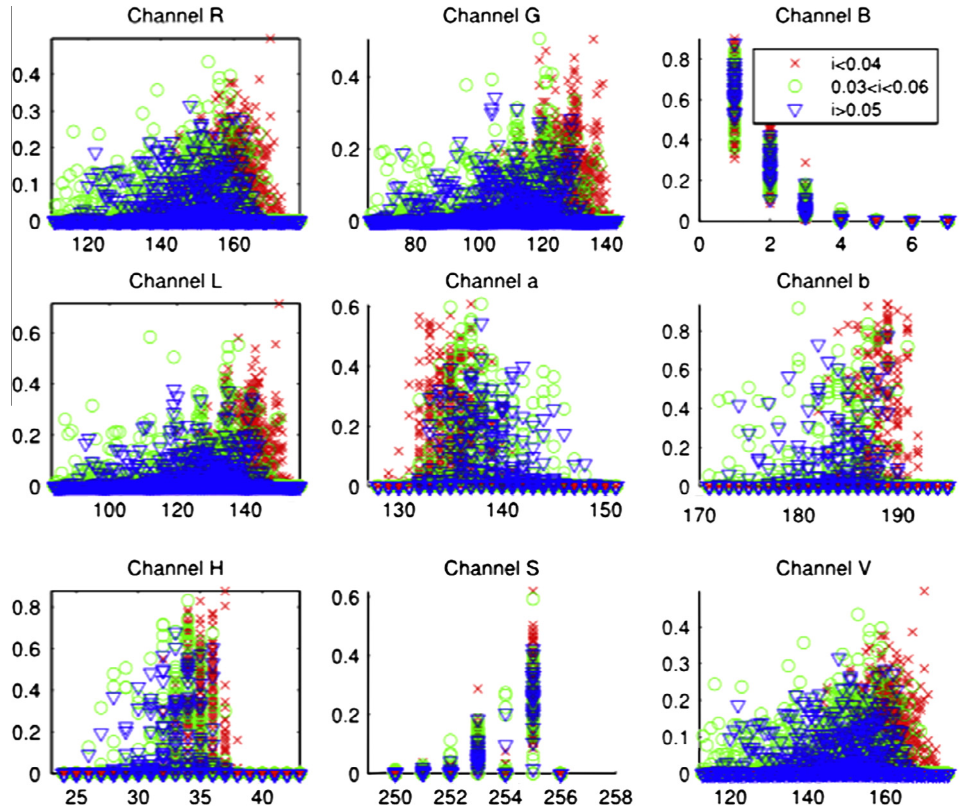

Fig. 4. Histogram of the different *RGB*, *Lab* and *HSV* color spaces channels for all the samples. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

schematically the construction of the input vector, and Fig. 6 depicts a plot of it.

This approach seemed more robust accounting for displacements of the curves of the histogram. An offset of the histogram curve due to variations of the base olive oil color implies the change of the values of the whole set of variables, and, due to the shape of the curve, the change in the value of the variable would be of the same order of magnitude that the value itself. Coding the information of the histogram in the way described allows that only the value of the index variables would change with the offset of the curve. Besides, the magnitude of the change would be small compared to the value of the variable. The main drawback of this approach is the increase of the number of input variables.

All the components in the different sets considered that were zero for all the samples were removed, and the input sets mapped to the $[-1,1]$ interval and centered before feeding them to the feature extraction methods.

The final sizes of the parameter vectors were 322 for set 1, 60 for set 2 and 91 for set 3. The results obtained using each one of these input vectors are presented and discussed in Section 3.

2.3. Feature extraction

In order to reduce the dimensionality of the input parameter vector, several techniques were evaluated: Principal Component

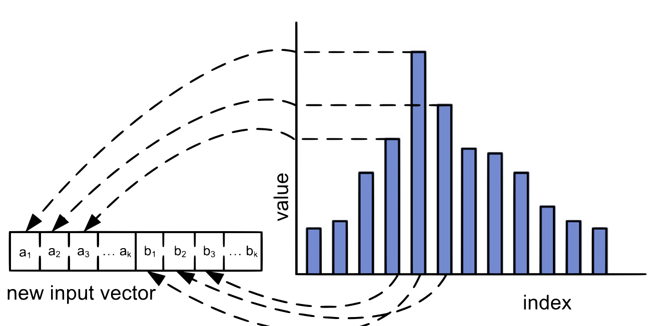

Fig. 5. Construction of the final input vector from the original histogram of a channel. The first subset of the vector are the values of the histogram in descending order and the second subset are the indexes of the original histogram components.

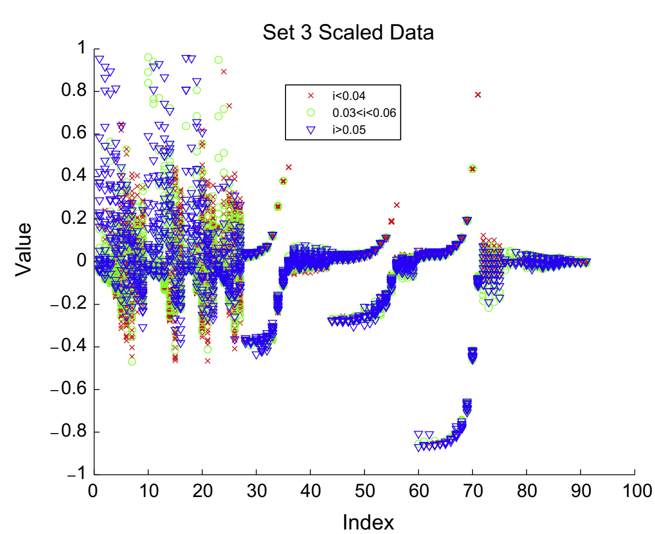

Fig. 6. Final input parameter vector used (set 3), formed by channels *a*, *b*, *H* and *S* transformed according to the algorithm defined in Fig. 5.

Analysis (PCA), Kernel PCA (KPCA) (Schölkopf et al., 1997), Linear Discriminant Analysis (LDA) and Kernel LDA (KLDA) (Yang et al., 2004).

The notation used throughout this work is as follows: let $\mathbf{x}_i \in R^n$ denote a row parameter vector, $i = 1, 2, \ldots, m$; being m the number of samples, and let X be matrix $n \times m$ formed by stacking $\mathbf{x}_i$ as rows. We will abuse the notation denoting $\mathbf{x}_i \in R^l$, $l < n$, as the feature vector constructed after the feature extraction method.

The basic idea of PCA is to find the directions of maximum variability of the data, so that the first k projections retain the maximum information possible allowing to reconstruct the data with minimal quadratic error (Muller et al., 2001). This fact allows the use of this technique as a feature extraction method with the objective of reducing the size of the input parameter. The method diagonalizes an estimate of the covariance matrix (C) of the centered data:

$$C = \frac{1}{l}\sum_{j=1}^{l}\mathbf{x}_j\mathbf{x}_j^T$$

The new features extracted from the data are the projections of the input data onto the eigenvectors of this covariance matrix. Since the algorithm for computing PCA may be expressed in terms of dot products, it is subject to the application of the kernel trick to construct a non-linear version of it. The idea behind Kernel PCA is the same as that of linear PCA, but making use of the kernel trick to map implicitly the input data to a high-order space and perform the search of directions in that space, thus allowing to consider non-linearities in the data. The matrix to diagonalize ($\widetilde{C}$), supposing that the data is centered in the feature space is:

$$\widetilde{C} = \frac{1}{l}\sum_{j=1}^{l}\Phi(\mathbf{x}_j)\Phi(\mathbf{x}_j)^T$$

If the data is not centered in the feature space, as is the general and our particular case, the matrix having to be diagonalized can be computed in terms of the matrix calculated from the uncentered data as described in (Schölkopf et al., 1997).

These methods are unsupervised and thus do not imply any risk of overfitting. However, it is not guaranteed that the projections of the data in the directions obtained by these methods are the most discriminative between classes. Furthermore, it is possible to lose the discriminative information if its variability is low and the number of components selected is not high enough.

LDA is a supervised method, and the idea behind it is to use the within-class variance to build a projection that maximizes the discriminability of the classes (Wang et al., 2004). This is achieved defining the between-class scatter matrix S_b, within-class scatter matrix S_w and the total scatter matrix $S_t = S_b + S_w$ and maximizing the following Rayleigh quotient:

$$J(X) = \frac{X^T S_b X}{X^T S_w X}$$

This may be recast as an eigenvalue problem, so a closed form solution of it is available. If the matrices are singular, PCA may be applied to reduce the dimension of the input space to the rank of S_t and then apply LDA, being guaranteed that no discriminative information is lost in the process (Yang and Yang, 2003).

In turn, analogously to Kernel PCA, Kernel LDA performs the algorithm on a higher dimensional space defined by the Kernel function (Yang et al., 2004), thus being able to deal with non-linear problems. The implementation of the algorithm we used was that proposed by Yang et al. (2004), that is, perform KPCA and then apply LDA to the transformed data.

Both LDA and KLDA have a tendency to overfitting if the number of samples is smaller than the dimensionality of the input parameter vector, as is the case for our problem. To handle the overfitting problem, an alternative is to use PCA or KPCA first to reduce the number of components and apply LDA to this transformed data set (Yang and Yang, 2003; Wang et al., 2004). Another drawback of these methods is that it extracts at most $l - 1$ features, being l the number of classes of the problem, which may not be enough to perform the classification.

For the non-linear methods (both KPCA and KLDA) a Gaussian kernel was used. The number of features extracted for each technique was considered as a variable to be selected according to the results obtained for each classifier. Section 3 presents the results obtained applying each feature extraction algorithm to the different input sets and classifiers tested.

2.4. Design of the classifier

Both Support Vector Machines (SVMs) (Burges, 1998) and Artificial Neural Networks (ANNs) (Duda et al., 2000) were considered for the design of the classifier.

2.4.1. Support vector machines

SVMs, in their simplest case, solve the problem of classifying between two classes which are linearly separable by finding the hyperplane that maximizes the margin between classes (Burges, 1998). Let the feature vectors be $\mathbf{x}_i \in R^l$, $i = 1, \ldots, m$ and the label be denoted by $y_i \in \{-1,+1\}$. The mathematical formulation of the problem is:

$$\begin{aligned} \text{minimize}: \quad & \frac{1}{2}\|w\|^2 \\ \text{subject to}: \quad & \mathbf{x}_i \cdot \mathbf{w} + b \geqslant +1 \text{ for } y_i = +1 \\ & \mathbf{x}_i \cdot \mathbf{w} + b \leqslant -1 \text{ for } y_i = -1 \end{aligned}$$

SVMs can be generalized to the linear non-separable case relaxing the constraints by means of slack variables and an extra cost term, and also allow the use of the kernel trick to map the data to a high-dimensional space and search the separating hyperplane there, enabling the construction of non-linear classifiers:

$$\begin{aligned} \text{minimize}: \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \\ \text{subject to}: \quad & \Phi(\mathbf{x}_i) \cdot \mathbf{w} + b \geqslant +1 - \xi_i \text{ for } y_i = +1 \\ & \Phi(\mathbf{x}_i) \cdot \mathbf{w} + b \leqslant -1 + \xi_i \text{ for } y_i = -1 \\ & \xi_i \geqslant 0 \quad \forall i \end{aligned}$$

This last formulation of the SVM was the option we chose to implement the classifier. SVMs can be extended to the multiclass classification problem using different approaches (Hsu and Lin, 2002), such as one-against-one, one-against-all and directed acyclic graph SVMs. The method used for the classifier was one-against-one, as implemented in LIBSVM, the library used for the experiments (Chang and Lin, 2011).

We tested different Kernels and different parameter values for the SVM. The approach for the design of the SVM classifier was common to all the different Kernels used. We first tested the performance of the classifier using a preliminary rough grid of values for the number of features and the parameters of the classifier. Then, a finer grid around the best values obtained in the previous tests was considered to determine the final parameters of the classifier. This search of parameters was made using the whole data set to extract the features and using leave-one-out cross validation on this data set for the classifier (phase 1).

Once the parameters of the classifier were found, a second step was taken to validate the results: we performed again a leave-one-out (LOO) cross validation on the classifier, but this time only the data used to train the classifier was used to construct the feature

Table 1
Input sets tested.

Name	Description
Set 1	*RGB*, *Lab* and *HSV* channels histograms
Set 2	*a*, *b*, *H* and *S* channels histograms
Set 3	*a*, *b*, *H* and *S* transformed data according to algorithm defined in Fig. 5

extraction matrices. Then, the features of the test vector were extracted using the matrices obtained without it (phase 2). The reason for this two-step approach was to save computational time, since the library used for the SVM training was far more efficient performing the LOO cross-validation using a built-in function which requires the whole data set to be passed as an argument. Again, the results obtained and their discussion are presented in Section 3.

2.4.2. Artificial neural networks

The selected neural networks were both a Perceptron and a Multilayer Perceptron (MLP), trained with the backpropagation algorithm and using the Sigmoid as activation function for the input and hidden neurons. The input neurons used were the values of the feature vector for each case, and the output nodes were the three classes representing the different amount of impurities to be assigned to the samples.

For the MLP, we considered a single hidden layer and the number of neurons for this layer as the parameter to be adjusted by trial and error, since a network with only one hidden layer, depending on the number of neurons in this layer, is capable of learning any pattern and less prone to getting caught in local minima that networks with a higher number of hidden layers (Duda et al., 2000). We used early stopping for the training of the networks, splitting the training set into a new training (75%) and a validation sets (25% of the data).

The search for the best number of neurons and feature vector length was conducted as follows: we tested the performance of the network for the points of a coarse grid on the whole range of possible lengths of the feature vector combined with an increasing number of neurons for the hidden layer. We used 4-fold validation for this process and the feature vector was constructed as in phase 2 of the procedure carried out for the SVMs, since we had no computation time savings in using the phase 1 dataset.

Once we identified the best combinations of feature vector length and number of neurons, a finer search was made around this zone using 10-fold cross-validation. Finally, the final classification results were obtained using a leave-one-out cross validation for the best feature-length and neurone-number pair for each feature extraction technique .

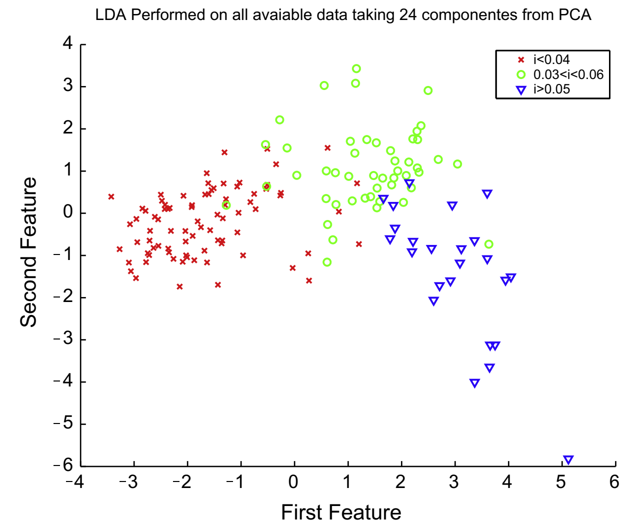

Fig. 7. Plot of the features extracted by LDA from set 3 input vector using all data. It can be seen that classes are fairly grouped and that a linear SVM is a good candidate for classifier.

3. Experimental results and discussion

Table 2 summarizes the best results obtained for the different sets with the corresponding feature extraction technique and the classifier. Considering the first two input vectors led to correct-classification rates below 70%. The results obtained using different feature extraction techniques for these input vectors were all of the same order of magnitude, so only those obtained using PCA are included (66.88% for set 1 and 64.29% for set 2 using SVMs, and 67.10% and 66.35% respectively using MLP).

Fig. 7 shows a plot of the features extracted using PCA plus LDA on the whole set of samples available, selecting 24 components in the PCA stage for the input set 3. It can be seen that the samples of each class are grouped and the different classes almost linearly separable. The existence of some samples in areas belonging to an adjacent class may be partially explained in the uncertainty of

Table 2
Summary of the results obtained.

Input parameter vector set	Feature extraction method	No. features	Classifier	% Classif.
Set 1	PCA	36	Gr. 3 Polynomial SVM	66.88
Set 2	PCA	16	Gr. 3 Polynomial SVM	64.29
Set 1	PCA	20	ANN MLP 80 neurons hidden layer	67.10
Set 2	PCA	35	ANN MLP 23 neurons hidden layer	66.35
Set 3	LDA	24	Linear SVM	85.71
Set 3	KLDA	85	Linear SVM	83.77
Set 3	PCA	23	Gaussian SVM	84.42
Set 3	PCA	23	Gr. 3 Polynomial SVM	85.71
Set 3	KPCA	25	Gaussian SVM	85.71
Set 3	KPCA	25	Gr. 3 Polynomial SVM	87.66
Set 3	LDA	26	ANN MLP 39 neurons hidden layer	80.40
Set 3	LDA	34	ANN Perceptron	80.73
Set 3	PCA	45	ANN MLP 51 neurons hidden layer	77.15
Set 3	PCA	38	ANN Perceptron	82.38
Set 3	KLDA	34	ANN MLP 70 neurons hidden layer	78.63
Set 3	KLDA	34	ANN Perceptron	76.51
Set 3	KPCA	51	ANN MLP	72.43
Set 3	KPCA	39	ANN Perceptron	77.15

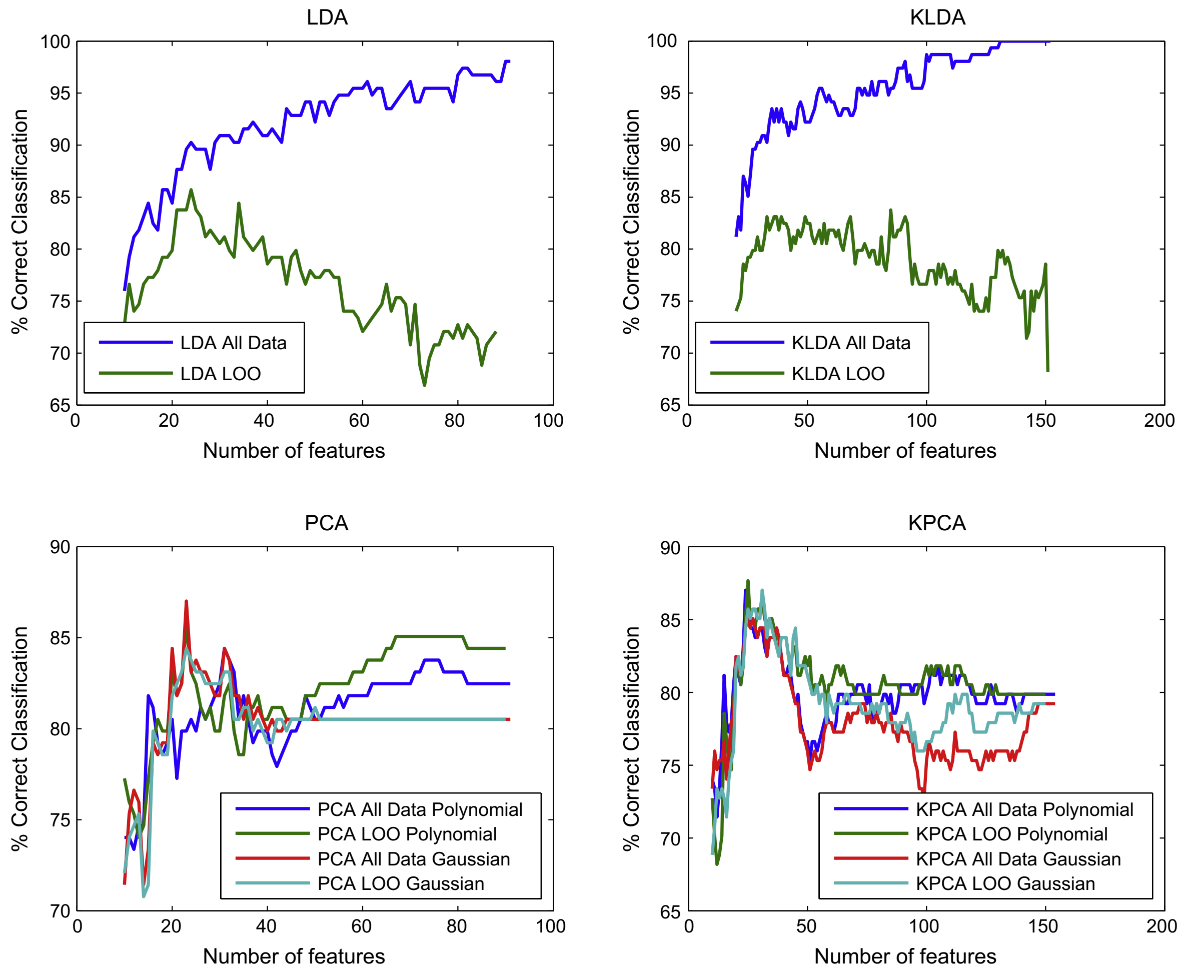

Fig. 8. Evolution of the correct-classification rate for the four feature extraction techniques applied to input set 3 for the SVM classifier.

the reference data (±0.01%) for samples lying in the edge values between classes.

Fig. 8 shows the evolution of the correct-classification rate for the four feature extraction techniques applied to input set 3 using the best parameters for the SVMs obtained during phase 1. The overfitting effect of LDA and KLDA can be clearly appreciated; during phase 1, the curve is always increasing with the number of features taken from PCA or KPCA; while for phase 2, there is a decay of the rate after an optimum is reached. It is remarkable also the gap between both curves, imputable to the overfit introduced by the supervised feature extraction method. It can be seen that PCA and KPCA do not show this behavior, as there is little difference between the curves obtained in phase 1 and phase 2.

The difference between using linear and non-linear feature extraction methods was small, although the best classification rate was obtained using KPCA and a grade-3-polynomial SVM (87.66%). The use of a linear kernel SVM with PCA and KPCA gave poor results, unlike those achieved with LDA and KLDA, where linear kernel was the one offering the best classification results.

Fig. 9 shows the evolution of the test error with the number of neurons included in the network for different lengths of the input vector and for each of the feature extraction techniques evaluated using 4-fold cross validation. The overfitting effect of the supervised feature extraction methods is visible in these plots as well, revealed by the maximum in the correct classification and the subsequent decay. It is also noticeable the small dispersion of the curves for LDA and KLDA with the number of neurons in the MLP.

The behavior of both the PCA and KPCA curves was much flatter with the number of features, again, due to the lack of overfitting effect of the feature extraction techniques. The performance is roughly grouped within an spread of 8 points and centered in 76% for PCA, and an spread of 10 points and center at 73% for KPCA.

The best result obtained applying ANN was achieved using PCA combined with a Perceptron, yielding a 82.38% correct classification rate, followed by LDA plus Perceptron with a 80.73%. The result obtained using the Perceptron with PCA was somewhat unexpected, since the linear kernel did not behave well when used in the SVMs with PCA data. In turn, the result obtained with LDA was anticipated due to the SVMs results.

The final classification rates obtained using the ANNs showed a significant gap with those obtained using 4-fold, being the gap greater for the non-linear feature extraction techniques.

Finally, it is worth noting that the average gap of 10 points of correct classification rate between input sets 1 and 2 and input set 3 for the ANN classifier, and the 20 points average gap for the SVM show the importance of proposing a good input set prior to the feature extraction and classification phases. Encoding the available information in such a way that was robust to displacements of the histograms was the key to obtain acceptable results.

4. Conclusion

In this work we have proposed a system based on computer vision to perform a fast approximate classification of the impurity

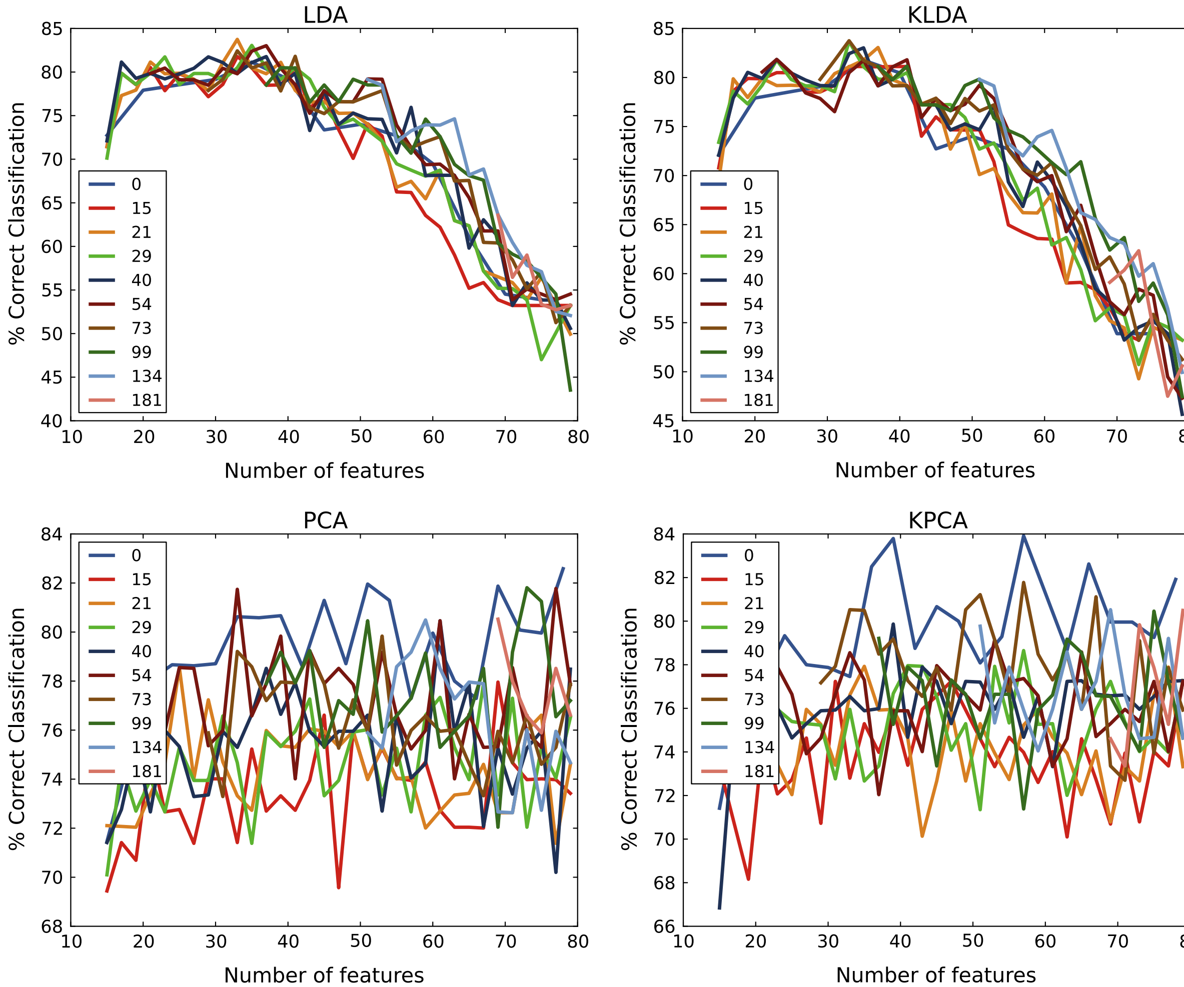

Fig. 9. Evolution of the correct-classification rate for the four feature extraction techniques applied to input set 3 as a function of the length of the feature vector and the number of neurons in the hidden layer, denoting 0 the Perceptron.

content of olive oil samples in three different levels. We have tested three different initial input parameter vectors derived from the histogram of the channels of the *RGB*, *CIELAB* and *HSV* color spaces, and four different feature extraction methods. The chosen classifiers were Support Vector Machines and Artificial Neural Networks, and several kernels and parameters were evaluated. The best classification result was achieved using KPCA jointly with a grade-3-polynomial SVM on a input parameter vector derived from the histogram of the channels *a*, *b*, *H* and *S* (87.66%). The best result for the ANN was obtained using PCA with the Perceptron (82.38%). The results also show the importance of selecting a good initial input parameter vector, as the use of input set 3 supposed a boost of 10 points in average correct classification rate for the ANN classifier, and 20 points for the SVM.

The proposed system simplifies greatly the operations needed to perform the impurity content analysis for situations where the required precision is not high, such as when the results of the analysis are used exclusively for process supervision purposes. In these situations, the proposed method constitutes a low-cost approximate alternative to the official method.

Acknowledgments

This work was partially supported by the Projects DPI2011-27284, TEP2009-5363 and AGR-6429. We also would like to extent a special thank to CM Europa (Martos, Spain), for their support in the lab measures and samples. Pablo Cano Marchal is in receipt of a FPU Grant from the Spanish Ministry of Education.

References

Brosnan, T., Sun, D.-W., 2002. Inspection and grading of agricultural and food products by computer vision systems—a review. Computers and Electronics in Agriculture 36 (2-3), 193–213, <http://www.sciencedirect.com/science/article/pii/S0168169902001011>.

Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2 (2), 121–167, http://dx.doi.org/10.1023/A:1009715923555.

Cano Marchal, P., Gómez Ortega, J., Aguilera Puerto, D., Gámez García, J., 2011. Situación actual y perspectivas futuras del control del proceso de elaboración del aceite de oliva virgen. Revista Iberoamericana de Automática e Informática Industrial RIAI 8 (3), 258–269, <http://www.sciencedirect.com/science/article/pii/S1697791211000148>.

Chang, C.-C., Lin, C.-J., 2011. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27, <http://www.csie.ntu.edu.tw/cjlin/libsvm>.

Diaz, R., Gil, L., Serrano, C., Blasco, M., Moltó, E., Blasco, J., 2004. Comparison of three algorithms in the classification of table olives by means of computer vision. Journal of Food Engineering 61 (1), 101–107, <http://www.sciencedirect.com/science/article/pii/S0260877403001912>.

Du, C.J., Sun, D.W., 2006. Learning techniques used in computer vision for food quality evaluation: a review. Journal of Food Engineering 72 (1), 39–55, WOS:000231844700006.

Duda, R.O., Hart, P.E., Stork, D.G., 2000. Pattern Classification. second ed. Wiley-Interscience.

Furferi, R., Carfagni, M., Daou, M., 2007. Artificial neural network software for real-time estimation of olive oil qualitative parameters during continuous extraction. Computers and Electronics in Agriculture 55 (2), 115–131, <http://www.sciencedirect.com/science/article/pii/S0168169906001190>.

Furferi, R., Governi, L., Volpe, Y., 2010. ANN-based method for olive ripening index automatic prediction. Journal of Food Engineering 101 (3), 318–328, <http://www.sciencedirect.com/science/article/pii/S0260877410003559>.

Gatica, C., Best, S., Ceroni, J., Lefranc, G., 2011. A new method for olive fruits recognition. In: San Martin, C., Kim, S.-W. (Eds.), Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Lecture Notes in Computer Science, vol. 7042. Springer, Berlin/Heidelberg, pp. 646–653, <http://www.springerlink.com/content/d3x6110533453872/abstract/>.

Gordillo, B., Ciaccheri, L., Mignani, A., Gonzalez-Miret, M., Heredia, F., 2011. Influence of turbidity grade on color and appearance of virgin olive oil. JAOCS, Journal of the American Oil Chemists' Society 88 (9), 1317–1327.

Hepworth, N., Hammond, J., Varley, J., 2004. Novel application of computer vision to determine bubble size distributions in beer. Journal of Food Engineering 61 (1), 119–124, <http://www.sciencedirect.com/science/article/pii/S0260877403001936>.

Hsu, C., Lin, C., 2002. A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks 13 (2), 415–425.

Jin, S., Yin, Y., 2010. Research on rapid detection of total bacteria in juice based on biomimetic pattern recognition and machine vision. In: 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT). vol. 6. pp. 395–399.

Malamas, E.N., Petrakis, E.G., Zervakis, M., Petit, L., Legat, J., 2003. A survey on industrial vision systems, applications and tools. Image and Vision Computing 21 (2), 171–188, <http://www.sciencedirect.com/science/article/pii/S026288560200152X>.

Moyano, M., Meléndez-Martínez, A.J., Alba, J., Heredia, F.J., 2008a. A comprehensive study on the colour of virgin olive oils and its relationship with their chlorophylls and carotenoids indexes (I): CIEXYZ non-uniform colour space. Food Research International 41 (5), 505–512, <http://www.sciencedirect.com/science/article/pii/S0963996908000550>.

Moyano, M., Meléndez-Martínez, A.J., Alba, J., Heredia, F.J., 2008b. A comprehensive study on the colour of virgin olive oils and its relationship with their chlorophylls and carotenoids indexes (II): CIELUV and CIELAB uniform colour spaces. Food Research International 41 (5), 513–521, <http://www.sciencedirect.com/science/article/pii/S0963996908000562>.

Muller, K., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B., 2001. An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks 12 (2), 181–201.

Ram, T., Wiesman, Z., Parmet, I., Edan, Y., 2010. Olive oil content prediction models based on image processing. Biosystems Engineering 105 (2), 221–232, <http://www.sciencedirect.com/science/article/pii/S1537511009003328>.

Riquelme, M., Barreiro, P., Ruiz-Altisent, M., Valero, C., 2008. Olive classification according to external damage using image analysis. Journal of Food Engineering 87 (3), 371–379, <http://www.sciencedirect.com/science/article/pii/S0260877407006413>.

Rosell, J.R., Sanz, R., 2012. A review of methods and applications of the geometric characterization of tree crops in agricultural activities. Computers and Electronics in Agriculture 81, 124–141.

Ruiz-Altisent, M., Ruiz-Garcia, L., Moreda, G., Lu, R., Hernandez-Sanchez, N., Correa, E., Diezma, B., Nicolaï, B., García-Ramos, J., 2010. Sensors for product characterization and quality of specialty crops—a review. Computers and Electronics in Agriculture 74 (2), 176–194, <http://www.sciencedirect.com/science/article/pii/S0168169910001377>.

Sangwine, S.J., Horne, R.E.N., 1998. The Colour Image Processing Handbook. Springer.

Schölkopf, B., Smola, A., Müller, K., 1997. Kernel principal component analysis. In: Gerstner, W., Germond, A., Hasler, M., Nicoud, J. (Eds.), Artificial Neural Networks—ICANN'97, Lecture Notes in Computer Science, vol. 1327. Springer, Berlin/Heidelberg, pp. 583–588, <http://www.springerlink.com/content/w0t1756772h41872/abstract/>.

Wang, M., Perera, A., Gutierrez-Osuna, R., 2004. Principal discriminants analysis for small-sample-size problems: application to chemical sensing. Proceedings of IEEE Sensors, 2004, vol. 2. IEEE, pp. 591–594.

Yang, J., Yang, J.-y., 2003. Why can LDA be performed in PCA transformed space? Pattern Recognition 36 (2), 563–566, <http://www.sciencedirect.com/science/article/pii/S0031320302000481>.

Yang, J., Jin, Z., Yang, J.-y., Zhang, D., Frangi, A.F., 2004. Essence of kernel fisher discriminant: KPCA plus LDA. Pattern Recognition 37 (10), 2097–2100, <http://www.sciencedirect.com/science/article/pii/S0031320303004205>.