https://dx.doi.org/10.22161/ijaers.678 ISSN: 2349-6495(P) | 2456-1908(O)

Citrus Fruit Quality Classification using Support Vector Machines

Jonatha Oliveira Reis Varjão[1], Glenda Michele Botelho[2], Tiago da Silva Almeida[3], Glêndara Aparecida de Souza Martins[4], Warley Gramacho da Silva[5]

[1,2,3,5] Department of Computer Science, Federal University of Tocantins, BRAZIL
[4]Department of Food Engineering, Federal University of Tocantins, BRAZIL

***Abstract*— *The large-scale fruit selection process is still manual or semi-automatic, mainly in small industries. This fact can lead to errors during the sorting of good fruits. Thus, this paper proposes an application using computer vision and machine learning to improve this task. The genus studied was the citrus, more specific the orange, one of the most produced fruit in Brazil. However, the methodology used can be applied on any fruit which quality can be measured by vision. The initial step was the construction of the learning space, consisting of image acquisition, pre-processing and features extraction. After the construction, the learning phase begins, consisted of the training of the support vector machine model, and then, statistical methods were used to valida te the model. As the final result, it achieved the accuracy of 97.3% in fruit classify.*

***Keywords*— Computer Vision, Fruit Quality, Support Vector Machines, Machine Learning.**

I. INTRODUCTION

Brazil is one of the largest fruit producers in the world, in accordance with FAO, Food and Agriculture Organization of the United Nations, it produced more than 40 million tons of fresh fruits in 2017, which 17 million tons were alone by orange fruit. In Brazil, the sorting process still is manual leading to errors in the quality inspection, due to the intensive, repetitive and tedious work routines, resulting in low-quality fruits that affect commercial acceptance [1].

With the increasing demand for the use of Artificial Intelligence, new areas had diverged like computer vision, machine learning, and the most recent area, Deep Learning (DL). Computer Vision (CV) aims to behave like human perception, using image processing and analysis to achieve this goal. Both Machine Learning (ML) and DL tend to minimize the intra-class variance along with the feature space for the given classes [2], the main difference is on the feature extraction phase. ML models often use feature extraction algorithms to find edges, corners, and descriptor like SIFT [3] and SURF [4], to create the feature vector as input for the training model. DL models use a hierarchical set of layers that produces learning representations from data, some layers can abstract the concept of edges, others contours, colors information, etc. In this approach, the model learns from the data, extracting features from the convolutions and pooling operations through the connected layers [4], [5]. The main idea for the feature extraction is to reduce the dimensionality, using obtained characteristics features from the signals, instead of the signal themselves [6].

The selection process induces the problem of the attribution of quality in the fruits, which, even according to legal standards, has a certain degree of subjectivity. Another aggravating factor is the possibility of a wrong classification by the person since human perception is easily deceived due to knowledge being inappropriate or being misapplied [7].

In this sense, a machine learning model using Support Vector Machine (SVM) in conjunction with a computer vision system to assist in a faster and more reliable sorting process is proposed. A computer vision system uses an optical device such as a sensor or a camera and a processing system. The image capture is followed by an analysis process and, in general, algorithms for segmentation are used to find regions of interest and feature extractors. Thus, to build the learning space and making it possible to classify the image according to previously adopted criteria. Also, it is possible to establish well-defined sample classes, according to the judgment of specialists and the characteristics to be identified.

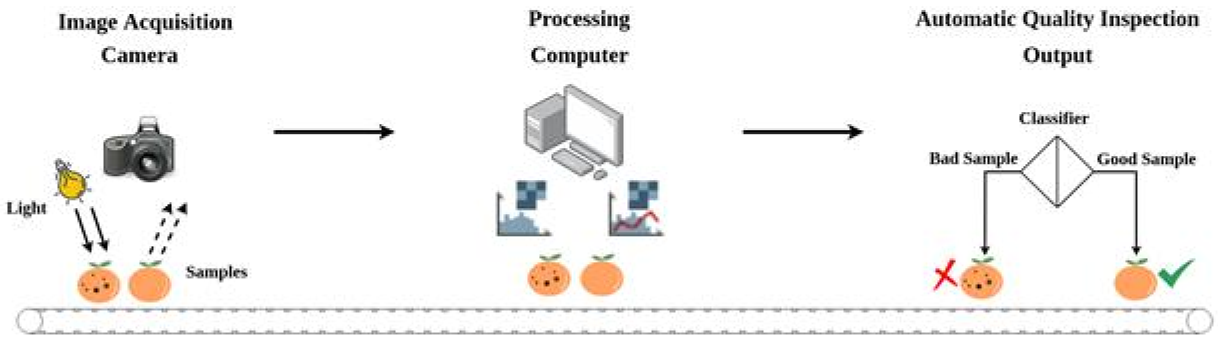

Fig. 1: Computer vision system architecture proposed in this paper.

Fig 1 shows the proposed architecture used to build the computer vision system, on the image acquisition stage the system captures the image. But on this paper, it doesn't discuss how the image acquisition system works or how it was implemented, the focus was the processing and automatic quality inspection stages.

The database was collected from COFILAB[1], which consist of two well-defined classes: citrus with stem, a collection of oranges in good maturity state and quality, and oranges infected with scale. The pre-processing stage is composed of three steps, background reduction, image filtering, and segmentation. After the segmentation, the learning space is built with the knowledge gathered from these steps.

The automatic quality inspection stage is composed by the use of the proposed classification model, which is SVM, a method based on machine learning theory. The feature vector built for the training was inspired by [8], using 64 colors features, 7 texture features, 8 shape features.

II. IMAGE ACQUISITION, PRE-PROCESSING, AND SEGMENTATION

The image acquisition was proposed using two datasets provided by COFILAB, Citrus with stem and Oranges infected with scale. Both datasets were created under the same circumstances [9], composed of a digital camera used to acquire high-quality images. At first, the images contained unnecessary information, like the background, as the research focused on the quality of the fruits, a background reduction was made. As the base is standardized by COFILAB, the reduction was a simple task, it starts with the use of Sobel filters to find contours and then a bounding-box is used to subtract the background.

Fig. 2 shows the steps to achieve the background subtraction, the Sobel filters are applied to find the contours and a routine to find the most significant contours is used, a threshold of 25% of the total area is used to dismiss the small contours leading only to match the fruit in the image. After the find of the most significant contour a mask using the min and max of each axis, width, and height of the image, is used to build the bounding-box to apply into the original image extracting the fruit and reducing the background.

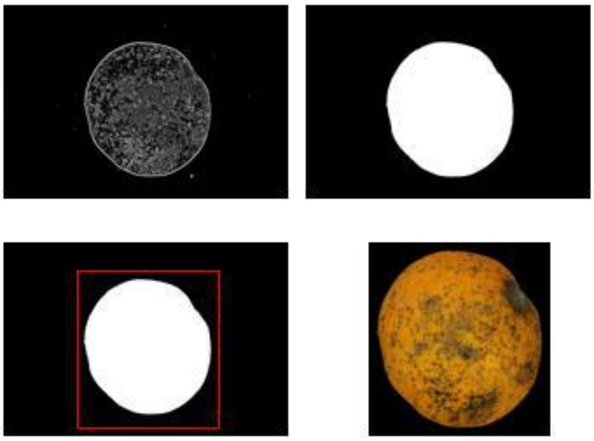

Fig. 2: The steps to achieve the background subtraction, in (a) the Sobel filter is applied, (b) apply mask inside the most significant contour region, (c) bounding box is formed by the min and max of each axis, (d) result image

After the background subtraction, image filtering is used to reduce the noise caused by sensors fault in the digital image acquisition. Vector median based filters or Gaussian filter are often used in this task, however, these classical methods tend to blur image edges and details

[1] COFILAB: Computers and Optics in Food Inspection - http://www.cofilab.com/

possibly losing crucial information about the image. In order to mitigate the blurriness caused by linear filters, Peer Group Filtering (PGF) is applied [10] without losing information about edges, making the segmentation robust.

The segmentation routine was made using JSEG (JPEG image segmentation), an unsupervised segmentation of color-texture regions in images and videos [11]. The JSEG objective is to segment images and video into homogeneous color-texture regions, but to identify this homogeneity, three pre-set rules are necessary:

- Each image must contain homogeneous color-texture regions;
- Each region can be represented by quantized colors in it;
- Colors between two neighbor regions are distinguishable.

The JSEG segmentation is formed by two steps, a color quantization, which performs a color reduction using a clusterization algorithm replacing the pixel value by its cluster color, generating a class-map, and a spatial segmentation is applied into the texture composition on the class-map.

Initial the color quantization is proposed during the image filtering process using the PGF, resulting pixels receive assigned weights, textured areas weights less than smoothed areas. CIELUV color space is used because its perception is uniform, the human eye senses changes in color better in uniform regions [12], and a General Lloyd Algorithm (GLA) creates the vector quantization of the pixel colors. The cluster's initial position for GLA is estimated by the popular splitting initialization algorithm. The weighted distortion D is given by:

$$D = \sum_i D_i = \sum_i \sum_n v(n) \parallel x(n) - c_i \parallel^2, \quad x(n) \in C_i,$$

And the update rule is derived to be:

$$c_i = \frac{\sum v(n) x(n)}{\sum v(n)}, \quad x(n) \in C_i.$$

where c_i is the centroid of C_i, $x(n)$, and $v(n)$ are the color vector and the perceptual weight for pixel n, and D_i is the total distortion for cluster C_i.

At the completion of GLA, some pixels may have similar color values, causing the pixels to belong to different clusters, so an agglomerative clustering algorithm is used to merge clusters, minimizing the distance between them, parameterized by a threshold.

After color quantization, all necessary information for segmentation is saved into a class-map. The generated class-map, often called J-image, is the value of each pixel in its given class by its position in the image as a bi-dimensional vector (x,y), this value can be represented as J-value. Each point belongs to a class, using these spatial data the JSEG segmentation is proposed:

Let *Z* be the set of all *N* data points in a J-image. Let z = (x,y), z ∈ Z, and *m* be the mean,

$$m = \frac{1}{N} \sum_{z \in Z} z.$$

Suppose *Z* is classified into *C* classes, Z_i, *i*=1,...,*C*. Let m_i be the mean of the N_i data points of class Z_i,

$$m_i = \frac{1}{N_i} \sum_{z \in Z_i} z.$$

Let

$$S_T = \sum_{z \in Z} \parallel z - m \parallel^2$$

and

$$S_W = \sum_{i=1}^{C} S_i = \sum_{i=1}^{C} \sum_{z \in Z_i} \parallel z - m_i \parallel^2$$

Sw is the total variance of points belonging to the same class. Define the J-value as:

$$J = (S_T - S_W)/ S_W.$$

In the case of images containing homogeneous regions, the more separated the classes will be resulting on a high value of *J*. In opposition if classes are uniformly distributed on the image the value of *J* tends to be small.

Circular windows of various scales are used to determine possible regions in the image. The value J is calculated for each region obeying the window size and the mean of the values is given by:

$$\bar{J} = \frac{1}{N} \sum_k M_k J_k,$$

where *Jk* is *J* calculated in the region *k*, *Mk* is the number of points in region *k*, *N* is the total number of points in the class-map. Thus, the criteria for segmentation is to minimize $\underline{J}$ over all regions. The window size affects how much an image region can be detected. Small size windows are useful to locate intensity and color edges, while large windows detect texture boundaries. Therefore, a region growing using seeds is necessary, it is followed by a region merging to give the segmented image, this parameter is controlled by the user, named scale factor. It was empirically analyzed that scale factor below value 10 fewer areas were detected and above 10 had no effect in to improve detection. So, with scale factor 10 was able to detect more areas, being healthy or unhealthy.

A threshold T_J is used to establish how the seeds are created over the image, given by:

$$T_J = \mu_J + a\sigma_J.$$

where μ_J is the mean of the values that represent the homogeneity over the image and σ_J is the standard deviation, a is a constant chosen from preset values that result in the number of seeds. Pixels with local J values less than T_J are candidates to be a seed point, the connection used in the JSEG algorithm is the 4-connectivity, (x+1,y), (x-1,y), (x,y+1), (x,y-1), where (x,y) is the position of the pixel.

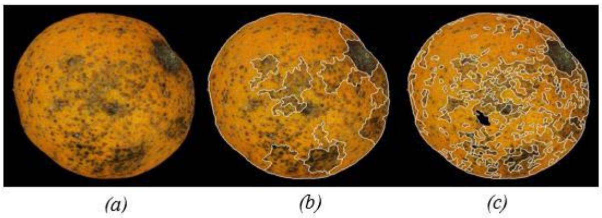

Fig. 3: JSEG step, (a) image filtering is applied generating the class-map, (b) default scale parameters on JSEG, (c) using 10 as the scale parameter

Fig. 3 illustrate the JSEG resulting image with the segmented areas overlayed with a white line, (b) result shows the default configuration for the JSEG, (c) uses 10 as the factor scale, values above 10 results on similar images, but the computational cost and time is increased, the proposed method uses 10 as a factor scale.

As for post-processing, a color reduction is applied to the segmented areas to reduce the color information, the objective in this phase is to improve the color disparity between areas, enhancing possibles rotten areas and preserving healthy areas, the Fig. 4 shows a better visualization of the color reduction.

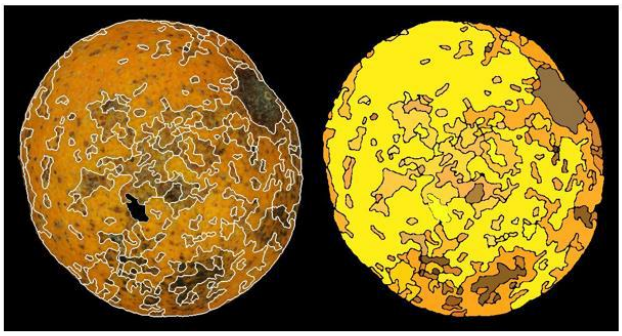

Fig. 4: Color reduction procedure output

III. SUPPORT VECTOR MACHINES

The learning paradigm chosen in this research was supervised learning, in this approach the machine, at the training process, know what class the inputs belongs to. SVM model, proposed by Vapnik in [13], is a representation of examples as points in space, mapped so that the series examples are divided by a clear gap that is as large as possible. New examples are then mapped, shared and defined for a category based on which side of the gap they fall into.

SVM constructs a hyperplane, or a set of hyperplanes, in a space of high or infinite dimension, which can be used for classification or regression. A good separation is achieved by the hyperplane that has the largest distance to the trained know points closest to classes, Fig. 6 exemplify the problem to find the largest distance between the separable classes, this distance is called functional margin. In general, the larger the margin, the smaller the generalization error is obtained.

The margin can be determined by calculating the distance between any two points, one of each translational hyperplane, both located in the normal vector w. Denoted by x_1 and x_2 the points in the vector w belonging to the upper and lower hyperplanes, respectively, the margin is computed simply as the length of the line segment connecting x_1 and x_2, that is, $||x_1 - x_2||_2$.

The margin can be written much more conveniently, taking the difference evaluated at x_1 and x_2 respectively.

$$(b + x_1^t w) - (b + x_2^t w) = (x_1 - x_2)^t w = 2$$

Given that the two vectors $x_1 - x_2$ and w are parallel to each other, we can solve for the margin directly in terms

$$\|x_1 - x_2\|_2 = \frac{2}{\|w\|_2}$$

of w, as:

The margin problem is extensively discussed in the theory of statistical learning. This discussion addressed the use of Kernels Machines where it explains the margin problem. The functions chosen were the most used in the literature, such as:

- Linear Function;
- Polynomial Function;
- Radial Basis Function;
- Sigmoid Function.

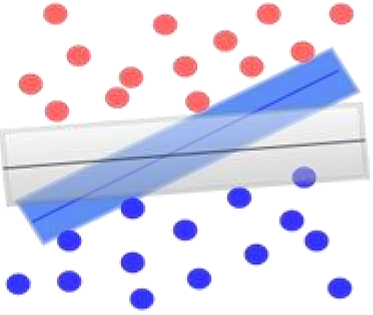

Fig. 5: Different separators trying to fit a larger margin

IV. PROPOSED METHOD

The proposed method uses an image processing routine described in Section II to process the input, and a feature space composed of 64 color features, 7 texture features, and 8 shapes features to create the feature vector. The initial dataset configuration was unbalanced, 125 images from the orange infected with scale, and 210 images from the citrus with stem, so a data augmentation procedure was used to balance the data sets.

The final configuration for the dataset was 300 images for each class. Operations like rotation, random noise, random crop, perspective-skewing, and elastic distortions were applied during the augmentation.

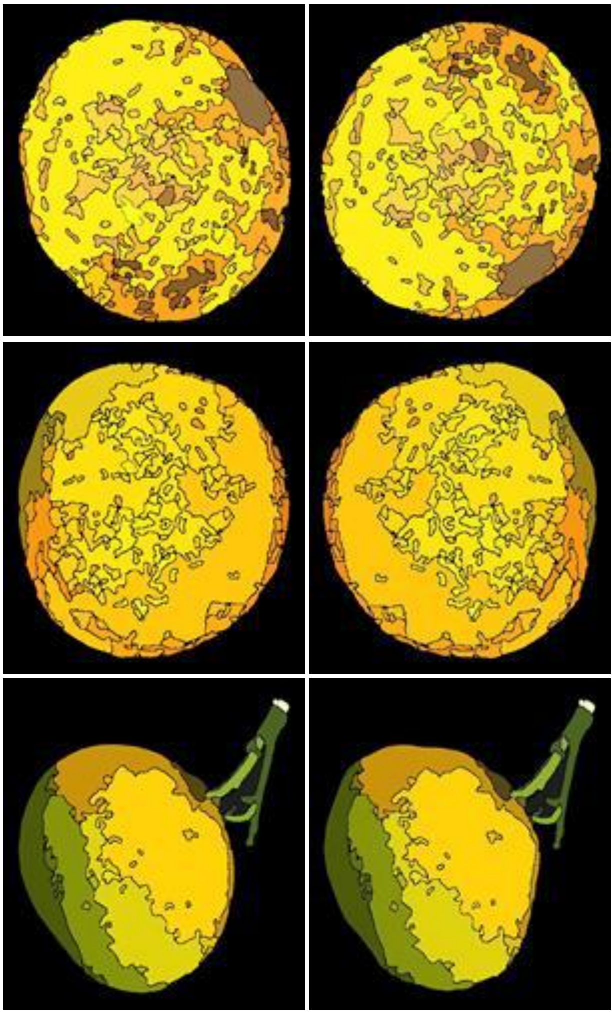

Fig. 6: Left side - Original Images, Right side - Augmented Images.

Fig. 6 exemplifies the operations, the left side is the original images, the right side is the augmented images results, the images might be similar, but the features generated is completely different.

Since the images do not have the same size, to create the 64 color feature, a dynamic filter was created to output the 64 color characteristics, also the color features used the RGB color space and HSV in its construction. As part of the texture features, it uses the mean, the contrast, the homogeneity, the energy, the variance, the correlation, and entropy, based on sum and difference histogram measures proposed by Unser in [14]. The shape features or morphology based measures, the features used as the area, perimeter, Euler number of the object, convex, solidity, minor length, major length, and eccentricity. In total the feature vector is built with 79 dimensions.

https://dx.doi.org/10.22161/ijaers.678 ISSN: 2349-6495(P) | 2456-1908(O)

Normalization is applied to the feature vector to preserve the learning abstraction within all the features, the main objective in normalization is to change the dimension values in a uniform common scale.

Within the features vector built, the training process uses 70% of the data set and 30% for tests, both classes uniformly distributed in each process. The metrics chosen to evaluate the model was f1-score, accuracy and confusion matrix one of the most used metrics to evaluate pattern recognition models [1],[2],[8],[15].

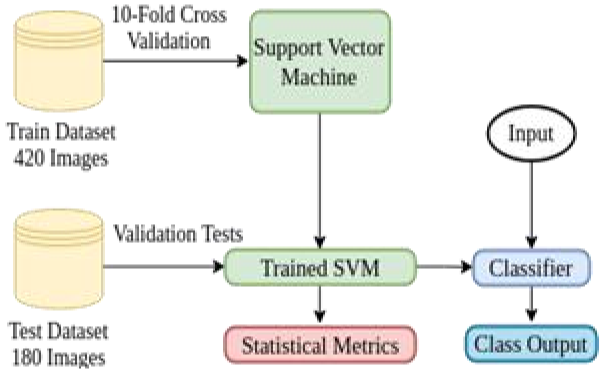

Fig. 7: Flow chart of the proposed method

A cross-fold validation using 10 folds were applied in the training process. Fig. 7 illustrate the proposed method using a flow chart.

V. RESULTS AND DISCUSSION

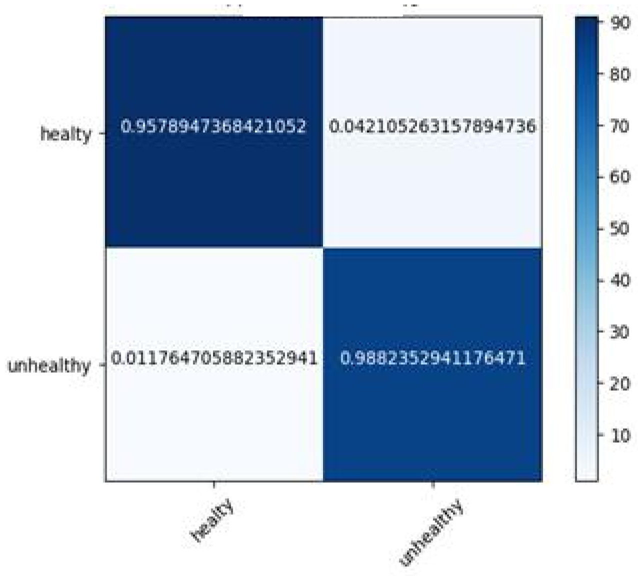

Fig. 8: The best generated classifiers in RGB color space using a Linear Kernel.

Evaluating the model, a cross-validation methodology was applied using 10 folds. Cross-validation results in a less biased model because it ensures that every observation from the dataset has the chance of appearing in the training and test set [15]. It split the data into 10 sets of 60 images, in each iteration it uses 70% for training and 30% for the test, and each class is balanced among the folds. At the completion of each fold iteration, a set of metrics is proposed, using f1-score and accuracy to evaluate each fold, additionally at the end of the iterations a confusion matrix is created. This paper analyzes two color spaces in the creation of the color feature, the additive color space, RGB, and perceptually-uniform color space, HSV.

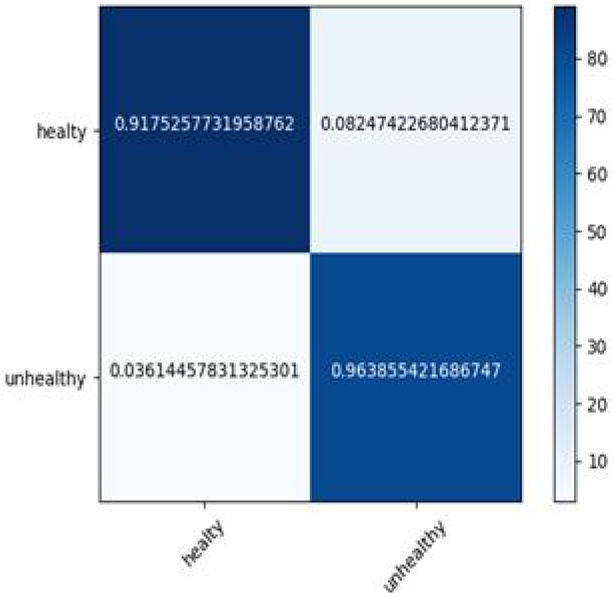

Fig. 9: The best classifier generated among the HSV color space using Radial Basis Function Kernerl.

In each color space, the SVM trains and generate the chosen metrics, in Fig. 8 the best classifier among the RGB color space utilizes a Linear Kernel generating a 97,3% accuracy. In Fig. 9 the classifier generated in the HSV color space uses a Radial Basis Function achieving a 94% accuracy.

Table 1: Classifier Generated at RGB color space

Classifier	***Accuracy***
Linear	***97,3%***
Radial Basis	96,6%
Sigmoid	*97,1%*
Polynomial	*86,2%*

Table 2: Classifier Generated at HSV color space

Classifier	***Accuracy***
Linear	*90,6%*
Radial Basis	**94%**
Sigmoid	*91,2%*
Polynomial	*84%*

VI. CONCLUSION

This paper proposes an image processing method, composed of image filtering, segmentation, and feature extraction, also presented an analysis of variations of the SVM for citrus fruit quality classification in which a very good result was observed with the color information feature represented in the RGB color space and with its linear kernel, obtaining a rate of 97,3% shown in Table 1, 3,3% higher than HSV color space radial basis classifier, seen in Table 2.

ACKNOWLEDGMENTS

I would like to thank the Federal University of Tocantins for the fomented funds in the execution of this research.

REFERENCES

[1] Y. A. Ohali and Y. Al Ohali, “Computer vision based date fruit grading system: Design and implementation,” *Journal of King Saud University - Computer and Information Sciences*, vol. 23, no. 1. pp. 29–36, 2011.

[2] P. Perera and V. M. Patel, “Learning Deep Features for One-Class Classification,” *IEEE Trans. Image Process.*, May 2019.

[3] D. G. Lowe, “Object recognition from local scale-invariant features,” *Proceedings of the Seventh IEEE International Conference on Computer Vision*. 1999.

[4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-Up Robust Features (SURF),” *Computer Vision and Image Understanding*, vol. 110, no. 3. pp. 346–359, 2008.

[5] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553. pp. 436–444, 2015.

[6] T. Wiatowski and H. Bolcskei, “A Mathematical Theory of Deep Convolutional Neural Networks for Feature Extraction,” *IEEE Transactions on Information Theory*, vol. 64, no. 3. pp. 1845–1866, 2018.

[7] R. L. Gregory, “Knowledge in perception and illusion,” *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, vol. 352, no. 1358, pp. 1121–1127, Aug. 1997.

[8] Y. Zhang and L. Wu, “Classification of Fruits Using Computer Vision and a Multiclass Support Vector Machine,” *Sensors*, vol. 12, no. 9. pp. 12489–12505, 2012.

[9] A. Vidal, P. Talens, J. M. Prats-Montalbán, S. Cubero, F. Albert, and J. Blasco, “In-Line Estimation of the Standard Colour Index of Citrus Fruits Using a Computer Vision System Developed For a Mobile Platform,” *Food and Bioprocess Technology*, vol. 6, no. 12. pp. 3412–3419, 2013.

[10] Y. Deng, C. Kenney, M. S. Moore, and B. S. Manjunath, “Peer group filtering and perceptual color image quantization,” *ISCAS’99. Proceedings of the 1999 IEEE International Symposium on Circuits and Systems VLSI (Cat. No.99CH36349)*. .

[11] Y. Deng and B. S. Manjunath, “Unsupervised segmentation of color-texture regions in images and video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8. pp. 800–810, 2001.

[12] N. Chaddha, W.-C. Tan, and T. H. Y. Meng, “Color quantization of images based on human vision perception,” *Proceedings of ICASSP ’94. IEEE International Conference on Acoustics, Speech and Signal Processing*. .

[13] C. Cortes and V. Vapnik, “Machine Learning,” vol. 20, no. 3. pp. 273–297, 1995.

[14] M. Unser, “Texture classification and segmentation using wavelet frames,” *IEEE Trans. Image Process.*, vol. 4, no. 11, pp. 1549–1560, 1995.

[15] J. S. Urban. Hjorth, *Computer Intensive Statistical Methods: Validation, Model Selection, and Bootstrap*. Routledge, 2017.