Automatic classification of oranges using image processing and data mining techniques

Juan Pablo Mercol, Juliana Gambini, Juan Miguel Santos

Departamento de Computación, Facultad de Ciencias Exactas y Naturales, UBA, Buenos Aires, Argentina
juanpablomercol@yahoo.com.ar,{jgambini,jmsantos}@dc.uba.ar,

Abstract

Data mining is the discovery of patterns and regularities from large amounts of data using machine learning algorithms. This can be applied to object recognition using image processing techniques.

In fruits and vegetables production lines, the quality assurance is done by trained people who inspect the fruits while they move in a conveyor belt, and classify them in several categories based on visual features.

In this paper we present an automatic orange's classification system, which uses visual inspection to extract features from images captured with a digital camera. With these features train several data mining algorithms which should classify the fruits in one of the three pre-established categories.

The data mining algorithms used are five different decision trees (J48, Classification and Regression Tree (CART), Best First Tree, Logistic Model Tree (LMT) and Random Forest), three artificial neural networks (Multilayer Perceptron with Backpropagation, Radial Basis Function Network (RBF Network), Sequential Minimal Optimization for Support Vector Machine (SMO)) and a classification rule (1Rule).

The obtained results are encouraging because of the good accuracy achieved by the classifiers and the low computational costs.

Keywords: Image processing, Data Mining, Neural Networks, Decision trees, Fruit Quality Assurance, Visual inspection, Artificial Vision.

1 Introduction

During the last years, there has been an increase in the need to measure the quality of several products, in order to satisfy customers needs in the industry and services levels. In fruits and vegetables production lines, the quality assurance is the only step which is not done automatically. For oranges, quality assurance is done by trained people who inspect the fruits while they move in a conveyor belt, and classify them in several categories based on visual features.

In the industry, there are very few automatic classification machines principally because of the need of advanced image processing, and the price of the hardware needed to satisfy the speed requirements of the production lines [15].

The visual aspect is very important for fruits. An orange with an excellent peel is sold at a higher price than another orange with the same internal features but with superficial defects. This promoted the establishment of quality standards at many organizations (in [21] they show five categories for oranges, lemons and tangerines).

However, the differences in quality categories are diffuse and subjective, so two people can classify the same specimen into different categories. It is possible to reduce this subjectivity using an automatic classifier.

In the scientific community, there is significant interest in the development of artificial vision based fruit classification systems. [15] introduces an orange classifier which uses artificial neural networks and Zernike polynomials. [4] shows an apple classifier based on color and texture features, using principal components analysis and neural networks. [7] proposes a system to estimate the volume of a fruit from digital images.

One of the main complications faced by the authors is the detection of the calyx, because it can be wrongly classified as a defect [15]. Another difficulty is the speed needed to perform the classification, because it has to be done in the time imposed by the speed of the conveyor belt.

In this work, we present a method to classify oranges using images. The process consists of the extraction of relevant features to be able to classify the orange into three categories (good, intermediate and defective). One of the most relevant features used, is the fractal dimension (DF). This can be used to characterize the oranges' peel smoothness as a quality indicator.

In order to make this paper self contained, we briefly introduce the system that should be used to capture the images; but the method developed is exclusively focused in the classification step.

The paper is organized the following way: in the section 2, a general description of the system is made, introducing the image capture step. In the section 3 we focus in the processing subsystem, explaining how the features used by the data mining algorithms in the classification step are obtained. In the section 4 we briefly explain all the data mining algorithms used, and in the section 5 we present the results obtained with the experiments done. Finally, in section 6 we present the conclusions and future works.

2 System overview

The system consists of three subsystems. The first one captures the orange's picture, the second one processes the image and performs the classification, and the third one places the fruit already classified in the desired container. This paper focuses on the processing subsystem, which will be explained in detail in section 3.

Oranges move in the conveyor belt and enter one by one in the inspection chamber. In there, several mirrors capture images from many angles, except the bottom view which is blocked by the conveyor belt. Then the images are processed and the oranges are classified. A diagram of this mechanism is shown in Figure 1.

In our experiment, we capture the images manually using a digital camera.

Once a fruit is classified, the system has to take an action according to the obtained results. Therefore, the actuator consists of a series of gates placed at the end of the conveyor belt to divert the fruit according the classified quality level, and deposit it in the desired container.

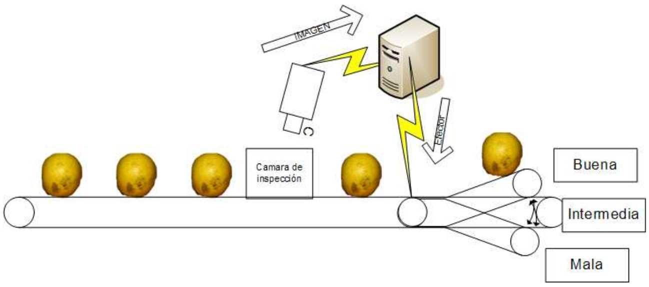


Figure 1: Diagram of the system where the oranges images are captured, analyzed and classified.

3 Processing

The processing system is divided into the following steps: 'Pre-processing and segmentation', 'Features' extraction' and 'Classification'. A diagram of this process can be seen in Figure 2.

3.1 Pre-processing and segmentation

Pre-processing consists of the quality improvement of the image, like noise reduction or contrast and brightness enhancement [16]. The goal is to improve the precision and speed of the feature extraction algorithms.

Segmentation resides in splitting up the image in regions in order to extract the objects of interest from the rest [16, 8]. In this paper, in the pre-processing step we improve the contrast of the image and extract the blue component from the RGB color space. The choice of the blue component is because it is the most discriminant component to remove the background of the image, due to the fact that for the orange color (made of red and some green), the value of the blue component is zero.

Being I_o and I_b the regions for the orange (foreground) and the background, we extract the region I_o from the background using a classical algorithm for background extraction [8].

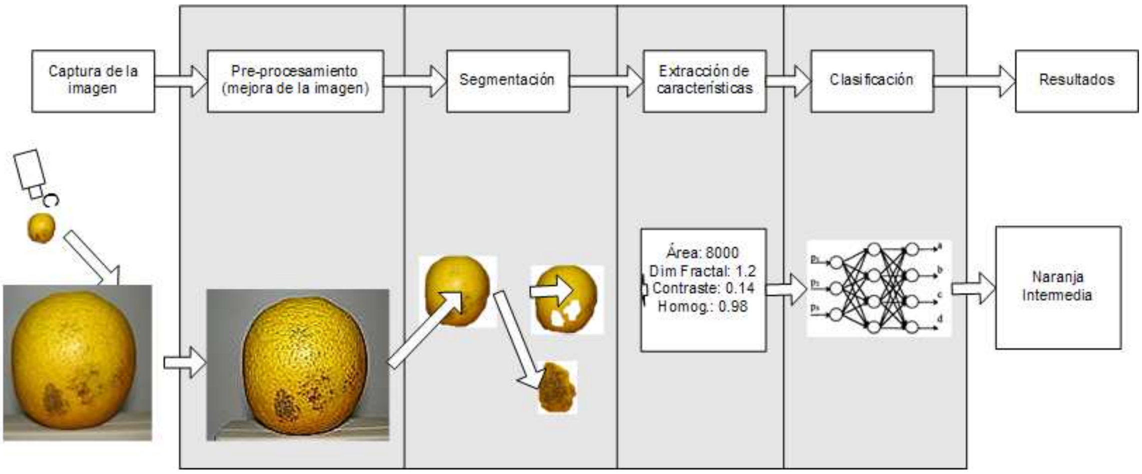


Figure 2: Image processing steps.

This process is done for every image captured.

3.2 Features' extraction

The objects in the image can be characterized by gray levels, color, texture, gradient, second derivative and by geometrical properties like area, perimeter, Fourier descriptors and invariant moments [18, 16].

In this paper, the features obtained are the area of the orange and the background, the fractal dimension of region I_o, the contrast, gray level uniformity, gray level correlation between neighbours, histogram, and the mean and median calculated in the HSV color space.

The data mining algorithms used in the classification step automatically detect the most relevant attributes (features) needed to perform the classification, discarding the rest. Therefore, in the features' extraction step we gather as most descriptors as we can in order to make classification algorithms more effective. Next we explain in detail each one of the features extracted.

- **Orange area:** The area of the orange A_o is calculated as the sum of the pixels belonging to the orange: $A_o = \sum_{i=1}^{144} \sum_{j=1}^{192} f(i,j)$, where $f(i,j) = \begin{cases} 1 & \text{if } (i,j) \in I_n \\ 0 & \text{in other case} \end{cases}$

  We also calculate the complement descriptor (background area) A_b.

- **Fractal dimension analysis:**

  The fractal dimension FD of a set in $\mathbb{R}^n$, is a real number which characterizes its geometrical complexity, and can be used as an irregularity indicator of a set [5, 3]. The FD is defined for self-similar sets, and in the case of sets which do not have this property, the FD has to be estimated [14].

  One of the methods proposed in [5] to estimate FD and characterize the smoothness level in a section of an image is the *box-counting* dimension. This method is commonly used because it exhibits a good balance between computation time and accuracy. However, this estimation has the inconvenient that can only be applied to binary images.

The *box-counting* dimension of a planar set A consists of estimating the quantity of cells in which the set has not null measure, in function of the size of those cells.

Being $N_l(A)$ the quantity of cells of side l where the set has not null measure, the *box-counting* dimension DB of A is

$$DB = \lim_{l \to 0} \frac{\log(N_l(A))}{\log(\frac{1}{l})}, \tag{1}$$

if the limit exists. In practice, for finite resolution images, l has superior and inferior limits, and DB can be estimated with the slope of the minimum square regression that approximates the logarithmic diagram $\log(N_l(A)) vs. \log(\frac{1}{l})$.

Given a binary image, it is partitioned in cells of side l, and for different values of l, the quantity $N_l(A)$ of cells in which the object of interest (foreground) has not null measure is calculated. Except the case in which $l = 1$, for all l, it is necessary to make many partitions of the image and calculate $N_l(A)$ as the average. Then, DB is estimated as the minimum square regression slope previously mentioned.

To be able to apply this method it is necessary to transform the image to binary, so a thresholding process has to be applied for this purpose. In this work, in order to obtain the texture of the peel of the fruit to estimate the fractal dimension, we start from a gray level image of the orange with the background removed, and apply a border detection procedure with the Canny [2] algorithm, getting the image I_{can}. Then, the box counting dimension DB is calculated over the image I_{can}. A result of 1 means that the texture of the orange's peel is smooth, which means that it is a good quality orange. In the opposite side, for greater imperfections, the value of the estimator DB tends to increase.

Table 1 shows the results of the border detection with the Canny algorithm, and the fractal dimension obtained for a good quality orange and a defective quality one, where it can be seen that the value of DB is lower (tending to 1) than the defective one.

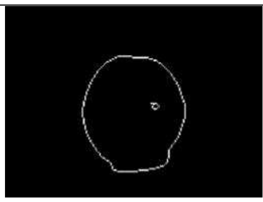
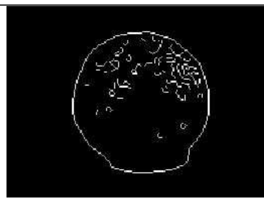
	Good orange	Defective orange
Original image		
Border detection		
Fractal Dimension	1.0494	1.2371

Table 1: Steps in the estimation of the fractal dimension

- **Texture analysis using statistical descriptors:** For the texture analysis we use six statistical descriptors, which use a co-occurrences gray level matrix. This is made calculating the number of adjacent pixels repetitions with the same gray level in the whole image.

The statistical descriptors used are:

- **Contrast:** The global contrast of the image (also known as variance or inertia) measures the contrast intensity between a pixel and its neighbour. Its calculation is based on the corresponding co-occurrences matrix.
- **Correlation:** Measures the relation of a pixel and its neighbour. The degree in which if the gray level of a pixel increases, its neighbour also increases.
- **Energy:** Also known as 'uniformity', 'energy uniformity' and 'second angular moment', consists of the sum of the squared elements in the co-occurrences matrix taken by pairs.
- **Homogeneity:** It is a value that measures how much the distribution of the elements of the co-occurrences matrix closeness the main diagonal of that matrix.
- **Skewness:** It is a measure of the degree of asymmetry of a distribution around the mean. It is obtained by calculating the third standardized central moment of the distribution. If the obtained value is zero, it means that it is centered (like the normal distribution). If it is positive, it is asymmetrical to the right, and if it is negative, to the left.
- **Kurtosis:** Measures how distant is the distribution of the data to the normal distribution. Is the result of calculating the fourth standardized central moment of a distribution. The kurtosis of the normal distribution is 3. A value greater than 3 (platykurtic) means that the distribution is flat (with thicker tails), and a distribution with kurtosis less than 3 (leptokurtic) means the opposite (thiner tails and a sharp peak).

- **Histograms analysis:** We analyze the histograms of the red H_r, green H_g, blue H_b, and the gray levels H_{gray} histograms. To simplify the analysis, we divide the histograms in six bins. For example, for the red component, the first bin is the amount of pixels in the image which values belong to the interval $[0, \frac{255}{6})$, the second bin $[\frac{255}{6}, \frac{255}{6} \times 2)$ and so on until the six intervals are covered.

- **Mean and median analysis in the HSV color space:** We take color values in the region of a circumscribed rectangle inside the orange region. This rectangle is divided into smaller regions forming a grid, and for each box several measures are taken.

  For this experiment, we use a grid of 3 rows and 3 columns, and for each row the mean and median of each of the 3 components of the HSV color space are calculated, getting a total of 54 attributes. This process can be seen in Figure 3.

  The HSV color space is used based on the good results reported in [6].

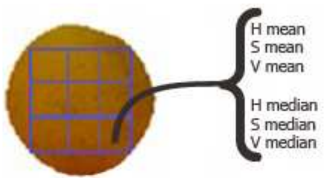


Figure 3: Region of the image used to extract the mean and median features of the HSV color space.

4 Classification using machine learning algorithms

In order to associate the features of the image with the corresponding class (good, intermediate or defective), we use the following data mining algorithms: five different decision trees (J48, Classification and Regression Tree (CART), Best First Tree, Logistic Model Tree (LMT) and Random Forest), three artificial neural networks (Multilayer Perceptron with Backpropagation, Radial Basis Function Network (RBF Network), Sequential Minimal Optimization for Support Vector Machine (SMO)) and a decision rule (1Rule). All of them are executed in the Waikato Environment for Knowledge Analysis software (WEKA).

- **J48 decision Tree:** Jan & Kamber [9] define a decision tree as a tree structure like a flow diagram, in which each node indicates a test on an attribute, each branch represents the result of that test and the leaves node represent classes.

  Mitchell [17] argues that a decision tree is a method that is used to perform approximations when the objective functions are discrete. An advantage of the decision trees is that they can represent the knowledge like IF-THEN rules.

- **Classification And Regression Trees (CART):** CART is a method to produce decision trees from categorical or continuous variables. If the variables are continuous, it makes a regression tree, and if they are categorical, it makes a classification tree. The splitting criteria is the Gini index.

- **Best First Tree:** Unlike traditional decision trees (i.e. C4.5, CART) which expand in depth, Best First trees expand selecting the node which maximizes the impurity reduction among all the available nodes to split. The impurity measure used by this algorithm is the Gini index and information gain [20].

- **Logistic Model Tree (LMT):** Logistic Model Trees (LMT) are a combination of logistic regression and decision trees, because it is a tree with the peculiarity that each leave is a logistic regression model [13]. Logistic regression only captures lineal patterns, while decision trees generate non linear models. One of the disadvantages of this method is the increased computational complexity [13].

- **Random Forest:** Random Forest classifiers generate a series of decision trees, where each tree is made using a vector, which is generated randomly for each tree, but using the same distribution for all trees. After generating a considerable amount of trees, each one votes for the most popular class, and the final model classifies with the class voted by the majority [1]. One interesting aspect of this classifier is that because of the large numbers law, overfitting is not produced [1].

- **Multilayer Perceptron Neural Network:** A neural network can be seen as a massively parallel distributed processor, made of simple processing units, which are capable of storing experimental knowledge and have it ready to be used later [10]. It resembles the human brain in which the knowledge is obtained from the environment through a learning process, and the neural interconnection strengths, known as synaptic weights, are used to store the acquired knowledge [10].

  The 'Multilayer Perceptron' neural network has an input layer made of input nodes or 'sensory units', one or many hidden layers and an output layer. During the training step, the input signal spreads forward from the input layer to the output layer, producing a result. This result is compared to the desired value and errors are calculated in the opposite direction while the synaptic weights are adjusted. Due to this error propagation process from the output layer to the input layer, this algorithm is known as 'Backpropagation'

- **Radial Basis Function Network (RBF Network):** Unlike the multilayer perceptron backpropagation algorithm which uses a recursive approximation technique known as stochastic approximation, the RBF network can be seen as a curve fitting problem in a highly dimensional space, where it has to find the best surface to fit the training data [10]. The network has three layers. The first layer is the input from the outside, the second is a hidden layer that makes a non linear transformation from the input space to the highly dimensional hidden space. The third layer is the output layer and shows the response of the neural network to the input data [10].

- **Sequential Minimal Optimization for Support Vector Machine (SMO):** During the training of a SVM it is required the resolution of a big quadratic programming (QP) problem. The SMO algorithm divides this problem in many smaller QP problems, and they are solved analytically requiring fewer computational cost [19].

- **Classification Rule (1R):** The One Rule (1R) algorithm makes a classification rule applying only a single attribute, producing a result similar to a single level decision tree [11]. This method makes very simple models and has been proved that with several data sets, it shows results as good as the ones achieved with more complex methods like C4.5 decision trees [11].

5 Results

For this experiment we use a data set obtained after processing 32 high quality oranges' images, 30 intermediate quality oranges' images and 31 defective quality oranges' images. For each specimen, we extract the 95 previously described features, which all of them are numerical attributes. The class assignment was done manually by the authors, based on visual features. We faced some subjectivity in the intermediate quality discrimination, because there are some oranges with higher quality and others with more defects. Therefore, in a near future it will be needed an expert's assistance to take his classification as the correct one to compare to.

Due to the few available examples, we use ten fold cross validation to validate the algorithms. This consists of partitioning the data set in k subsets, using $k-1$ subsets for training and model generation, and the other subset to validate the obtained model. This process is repeated k times using always a different subset for the validation, and finally all the results are averaged to produce a single estimation [12].

After applying different models, we analyze the confusion matrix and compare the accuracy of the classifiers (percentage of cases correctly classified over the total of cases classified).

- **Results obtained with a J48 decision tree:** After training a J48 decision tree with the described dataset, the decision tree shown in Figure 4 a) is generated. It can be seen that the attribute *contrastV* (the contrast of the $Value$ component of the HSV color space) is in the root of the tree. This means that this is the best attribute to differentiate between the classes.

As it can be seen in the confusion matrix shown in Table 2, the accuracy achieved by the classifier is 76.3%. From all of the classification errors produced, there is only one case in which a good orange is missclassified as a deffective one. The rest of the missclassifications are between the 'intermediate' class and the others.

- **Results obtained with a Best First decision tree:** Figure 4 b) shows the model generated by the Best First algorithm. In the root node it has the same attribute as the J48 tree (the contrast of the $Value$ component of the HSV color space), and other discriminant attributes are the histograms of the green and blue components of the RGB color space. The accuracy of this model is 83.9% (see Table 2).
- **Results obtained with a Logistic Model Tree (LMT):** In the resulting model, the most discriminant attributes are the fractal dimension, the contrast of the value, the kurtosis, the mean of the Hue, $Saturation$ and $Value$, the gray level histogram and the red component histogram. The accuracy achieved is 86%.
- **Results obtained with a Random Forest decision tree:** The accuracy obtained with this tree is 81.7%. Analyzing the confusion matrix, we notice that there are no errors between the 'good' and 'deffective' classes.
- **Results obtained with a Simple CART decision tree:** This decision tree achieves an accuracy of 83.9% (see Tabla 2), creating a model very similar to the one obtained with the Best First algorithm (see Figure 4 b)).

a) J48

```
contrastV <= 0.079261
|   VMean3 <= 0.121511: good (32.0)
|   VMean3 > 0.121511: intermediate (5.0)
contrastV > 0.079261
|   red1 <= 1266
|   |   blue5 <= 608
|   |   |   energy <= 0.63939
|   |   |   |   SMean9 <= 0.068687: mala (2.0)
|   |   |   |   SMean9 > 0.068687: intermediate (26.0/1.0)
|   |   |   energy > 0.63939: mala (3.0)
|   |   blue5 > 608: mala (7.0)
|   red1 > 1266: mala (18.0)
```

b) BFTree

```
contrastV < 0.07
contrastV < 0.07163
|  blue2 < 6.5: good(31.0/0.0)
|  blue2 >= 6.5: intermediate(2.0/0.0)
contrastV >= 0.07163
|  contrastV < 0.12227
|  |  Hmean3 < 0.11296
|  |  |  VMean3 < 0.10921: intermediate(3.0/1.0)
|  |  |  VMean3 >= 0.10921: mala(4.0/0.0)
|  |  Hmean3 >= 0.11296: intermediate(21.0/1.0)
|  contrastV >= 0.12227
|  |  green4 < 7155.0: mala(26.0/0.0)
|  |  green4 >= 7155.0: intermediate(4.0/0.0)
```

c) Simple CART

```
contrastV < 0.07163
|  blue2 < 6.5: good(31.0/0.0)
|  blue2 >= 6.5: intermediate(2.0/0.0)
contrastV >= 0.07163
|  contrastV < 0.12227: intermediate(24.0/6.0)
|  contrastV >= 0.12227
|  |  green4 < 7155.0: mala(26.0/0.0)
|  |  green4 >= 7155.0: intermediate(4.0/0.0)
```

d) 1 Rule

```
IF contrastV < 0.061663758 -> good
IF contrastV < 0.122273124 -> intermediate
IF contrastV >= 0.122273124 -> mala
```

Figure 4: Decision trees and classification rule models.

- **Results obtained with a Multilayer Perceptron Network:** It consists of 95 nodes in the input layer(one for each attribute), 3 in the output layer (3 classes), and 48 in the hidden layer. The accuracy obtained (85%) is shown in Table 2.
- **Results obtained with a Radial Basis Function Network:** Despite that the accuracy of the RBFNetwork is 83.9%, analysing the confussion matrix it can be seen that the model wrongly classifies a good orange as defective, and two defective oranges as good.
- **Results obtained with a Sequential Minimal Optimization SVM Network:** This network produces very good results, because it reaches an accuracy of 86% without any classification error between the good and defective classes.
- **Results obtained with a One Rule classification rule:** The model obtained with this algorithm is in Figure 4 d). Here it can be seen that using only one attribute (the contrast of the $Value$ component of the HSV color space) an accuracy of 83.9% is achieved. This means that it is only necessary to make a color space transformation and compute the contrast of the $Value$ to be able to classify with this algorithm.

6 Conclusions and future works

In this work we present the application of ten data mining algorithms for orange quality classification through visual features. The first group of algorithms are decision trees (J48, Best First, Logistic Model, Random Forest and Simple CART), the second group is made of neural networks (Multilayer Perceptron, Radial Basis Function and Sequential Minimal Optimization) and then a classification rule (1Rule) is analyzed.

The main advantage of decision trees and classification rules over neural networks are their simplicity and interpretation of the obtained classification rules.

Although most of the algorithms produce good results (higher than 80%), the SMO neural network and the LMT decision tree are the ones which, in the experiments done, produce the models with the highest accuracy (86%). Most of the errors produced by the SMO algorithm are intermediate oranges missclassified as good ones (4 errors) while the LMT missclassifies 4 deffective oranges as intermediate. Also, both misclassify 6 intermediate oranges as defective.

However, based on the parsimony principle, the best algorithm is the classification rule $1R$, because it achieves rather good accuracy (83.9%), and only generates three classification rules which use only one attribute (the contrast of the $Value$ component of the HSV color space). Because of this, the processing speed is higher than if all the features are extracted and processed by the classifier.

In the opposite side, the algorithm with the worst accuracy is the J48 classification tree (76.3%), which missclassifies 1 good orange as defective. Despite that the classification accuracy of the RBF network is 83.9%, it incorrectly classifies 1 good orange as defective, and 2 deffective as good ones, making this algorithm a bad choice.

In a future work, we will optimize the processing speed of the algorithms. To do this, it will be necessary to measure the amount of oranges classified in a certain amount of time, like for example the amount of oranges classified per second. This will be done taking into account the speed requirements of real production lines.

Algorithm	Real / Obtained	Good	Inter.	Defective	Accuracy	% Average.
J48	Good	29	5	0	85.3%	76.3%
	Intermediate	2	17	6	68%	
	Defective	1	8	25	73.5%	
BFTree	Good	30	2	0	93.8%	83.9%
	Intermediate	2	23	5	76.7%	
	Defective	0	6	25	80.6%	
LMT	Good	30	2	0	93.8%	86.0%
	Intermediate	1	25	4	83.3%	
	Defective	0	6	25	80.6%	
Random Forest	Good	30	2	0	93.8%	81.7%
	Intermediate	2	20	8	66.7%	
	Defective	0	5	26	83.9%	
Simple Cart	Good	30	2	0	93.8%	83.9%
	Intermediate	2	23	5	76.7%	
	Defective	0	6	25	80.6%	
Multilayer Perceptron	Good	31	3	0	91.2%	84.9%
	Intermediate	1	21	4	80.8%	
	Defective	0	6	27	81.8%	
RBF Network	Good	28	2	2	87.5%	83.9%
	Intermediate	1	25	4	83.3%	
	Defective	1	5	25	80.6%	
SMO	Good	28	4	0	87.5%	86.0%
	Intermediate	2	27	1	90.0%	
	Defective	0	6	25	80.6%	
1R	Good	30	0	2	93.8%	86.0%
	Intermediate	1	25	4	83.3%	
	Defective	0	6	25	80.6%	

Table 2: Classification results.

References

[1] Leo Breiman. Random forests. In *Machine Learning*, pages 5–32, 2001.

[2] F. John Canny. A Computational Approach to Edge Detection. *j-IEEE-PAMI*, 8(6):679–698, 1986.

[3] S.S. Chen, J.M. Keller, and R.M. Crownover. On the calculation of fractal features from images. *Pattern Analysis and Machine Intelligence*, 15(10):1087–1090, 1993.

[4] Unay D. and Gosselin B. Apple defect detection and quality classification with mlp-neural networks. In *International workshop ProRISC 2002*, 2002.

[5] K. Falconer. *Fractal Geometry: Mathematical Foundations and Applications*. John Wiley & Sons, Chichester, England, 1990.

[6] M. Fernández Ribot. *Selector de fruta y simulación de una aplicación real.* PhD thesis, Universidad Politécnica de Catalunya, España, 2006.

[7] K. Fobes. *Volume estimation of fruits from digital profile images.* PhD thesis, Department of Electrical Engineering, University of Cape Town, Cape Town, 2000.

[8] R. González, R.C.; Woods and Eddins S. *Digital (I)mage (P)rocessing (U)sing MATLAB.* Pearson Prentice-Hall, 2004.

[9] Jan H. and Kamber M. *Data Mining: Concepts and Techniques.* Morgan Kaufmann, 2000.

[10] S. Haykin. *Neural Networks. A Comprehensive Foundation Second Edition.* Prentice Hall International Inc, 1999.

[11] Robert C. Holte. Very simple classification rules perform well on most commonly used datasets. *Mach. Learn.*, 11(1):63–90, 1993.

[12] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, pages 1137–1145, 1995.

[13] Niels Landwehr, Mark Hall, and Eibe Frank. Logistic model trees. In *Machine Learning*, pages 241–252. Springer-Verlag, 2003.

[14] Y. Liu and Y. Li. Image feature extraction and segmentation using fractal dimension. In *International Conference on Information and Signal Processing*, pages 975–979. IEEE, Sept. 1997.

[15] Recce M., Taylor J., Plebe A., and Tropiano G. High speed vision-based quality grading of oranges. In *International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing (NICROSP '96)*, pages 136–144, 1996.

[16] D. Mery. Inspección visual automática. In *Actas del 1er Congreso Internacional de Ingeniería Mecatrónica, Lima*, Lima, Perú, 2002.

[17] Tom M. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

[18] Marques N. and Chen N. Border detection on remote sensing satellite data using self-organizing maps. In *En Fernando Moura Pires, Salvador Abreu (Eds.): EPIA 2003, LNAI 2902. Spinger-Verlag Berlin Heidelberg.*, pages 294–307, 2003.

[19] John C. Platt. Fast training of support vector machines using sequential minimal optimization. pages 185–208, 1999.

[20] Haijian Shi. Best-first decision tree learning. Master's thesis, University of Waikato, Hamilton, NZ, 2007. COMP594.

[21] H. Zubrzycki and Molina. Factibilidad comercial de cítricos entre argentina y brasil. In *Publicación de la EEA Bella Vista Serie Técnica N° 17 2005*, 2005.