Towards an Automatic Classification of Persian Lime

Ellen Giacometti[‡], Gabriel Araujo[‡†], Amaro Lima[⋆], Thiago Prego[⋆], Cristiano Carvalho[†], Fabricio Silva[†].

***Abstract*— Persian Lime (*Citrus × latifolia*) is a citrus fruit commonly called Tahiti lime in Brazil, a major exporter of it. In this work, we propose a computer vision-based system to automatically classify Persian lime according to color, size, and defect. We designed a dataset composed of images from hundreds limes in several conditions to train and validate the proposed system. Data augmentation was employed to increase the data variability in the training stage. The results are competitive in terms of hit rate achieving over than 80% of accuracy in both maturity and defect classifications.**

***Keywords*— Persian lime, Random Forest, Classification, Data Augmentation, Artificial Neural Networks, Support Vector Machines**

I. Introduction

The Persian lime (*Citrus x latifolia*), also known as seedless lime or Tahiti lime, is a common type of lime in many countries. Brazil is a major exporter of limes, and the exportation amount is growing each year. Efficient and fast fruit classification can improve competitiveness in this sector. This paper describes a real-time method for autonomous Persian lime classification based on computer vision which was designed to be used under controlled conditions in industrial applications.

The usage of computer vision methods for fruit detection or classification is not a novelty and finds application in different purposes. For example, there are several methods for fruit detection on trees. There is a method in [10] that uses a fusion of images segmented based on an adaptive threshold in RGB and HSV color spaces. The method in [12] uses Angular Invariant Maximal (AIM) to detect key points, and then pairwise intensity comparisons (PIC) and gradients (based on Histogram of Oriented Gradients - HOG) are used to extract a feature map. The Random Forest (RF) technique performs the classification step.

There are many fruit classification methods as well. The work in [17] compared the color and shape with SIFT as features. They compared three methods in the classification step: RF, k-Nearest Neighbours (k-NN), and Support Vector Machine (SVM).

A method for tomato detection of defects was employed by [3] using a commercial structure for sorting with regular camera chamber and fluorescent illumination to acquire more than 40,000 images.

They employed several techniques for classification: SVM, k-NN, Classification and Regression Tree (CART), Linear Discriminant Analysis (LDA), and Fisher Discriminant Analysis (FDA) are the most common applied algorithms. An approach to localise Emperor citrus using RGB-D (RGB and depth) information with SVM is presented in [8]. There are some methods for detecting defects in citrus fruit in the literature. Although the work in [1] also deals with defects, it aims to classify among 11 different types of defects, differently from others whose object is to discriminate between healthy and defected fruits.

The proposed system in this paper can classify limes according to maturity (related to roughness on the lime skin), presence/absence of defects, and size. We employed three types of classifiers, Artificial Neural Network (ANN), SVMs, and RF. The classification occurs in a feature space that combines size, shape, color, and texture measures. Our work has two main contributions: i) A lime classification system; ii) A Persian lime dataset, exclusively designed for this task. The dataset is composed of images from hundreds of limes acquired in a controlled environment with homogeneous background, constant illumination. We employed data augmentation based on random rotation to increase the data variability and deal with class unbalance issues. The quantitative evaluation of the proposed approach was in terms of accuracy, and the results were promising. However, there is no quantitative comparison with previously published methods because there is no available dataset for Persian lime classification.

The remainder of this paper is organized as follows. Section II provides detailed information about the designed dataset with its implementation aspects. Each step of the proposed classification system is described in Section III. Results, comparisons, and discussion are in Section IV. Conclusions and perspectives for future work are in Section V.

II. Designed dataset

The designed dataset is composed of images from 1087 limes in several sizes and conditions. Each image has 793×793 pixels dimension. The acquisition occurred in a controlled environment with homogeneous background, constant illumination, and an image plane parallel to the background at a fixed distance, according to Figure 1.

Each image from the dataset was labeled as immature or mature and as with or without defects. As can be seen in Figure 2, immature Tahiti limes have a rough dark green

[‡] Program of Electrical Engineering, Federal Center of Technological Education of Rio de Janeiro (CEFET/RJ), Campus Maracanã, Rio de Janeiro, RJ, Brazil. E-mail: ellen.giacometti@aluno.cefet-rj.br, gabriel.araujo@cefet-rj.br; [†] Dept. of Automation and Control Engineering, Federal Center of Technological Education of Rio de Janeiro (CEFET/RJ), Campus Nova Iguaçu, Nova Iguaçu, RJ, Brazil. E-mail: {cristiano.carvalho, fabricio.silva}@cefet-rj.br; [⋆] Dept. of Telecommunications, Federal Center of Technological Education of Rio de Janeiro (CEFET/RJ), Campus Nova Iguaçu, Nova Iguaçu, RJ, Brazil. E-mail: {amaro.lima, thiago.prego}@cefet-rj.br.

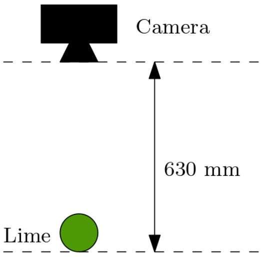

Fig. 1: Dataset image acquisition scheme.

aspect. Mature limes are smooth, green, and can be yellowish in some parts. A lime with defects has brown features in its skin and can be mature or immature.

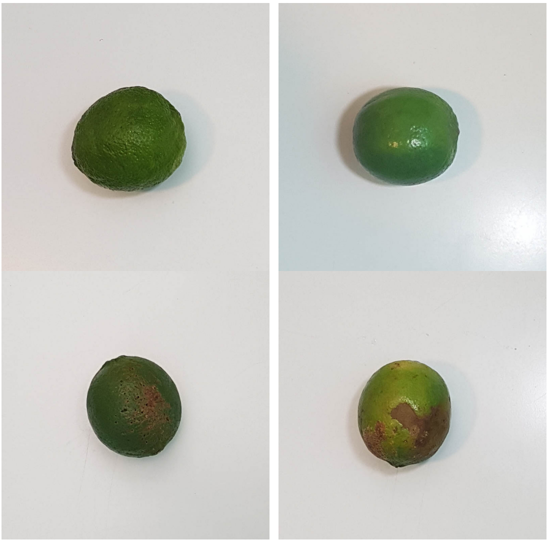

Fig. 2: Dataset samples. First row, from left to right: immature and mature limes without defects; second row: immature and mature limes with defects on their skin.

The dataset has two parts, the training, and test subsets. The training is composed of 959 and the test of 128 images. The training set contains 287 immature without defects, 72 immature with defects, 300 mature without defects, and 300 mature lime images with defects. The test set has 21 immature without defects, 27 immature with defects, 42 mature without defects, and 38 mature limes with defects. As one can note, the classes are highly unbalanced. We employ data augmentation to balance the classes in the training data. The data augmentation strategy consists of randomly rotating the images belonging to the classes with fewer samples in multiples of 90 degrees. After adding the artificial data, all four classes have 300 data samples, which changed the size of the training dataset to 1200 lime images.

III. A FRUIT CLASSIFICATION SYSTEM

A block diagram of the proposed system is in Figure 3. It is a straightforward three-step system composed of a pipeline of preprocessing, feature space mapping, and classification. A more detailed description of each step follows in the next sections.

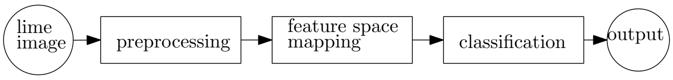

Fig. 3: Proposed lime classification system.

A. Feature space mapping

This section describes the preprocessing scheme used in the proposed system. The first step is the color conversion from the original Red, Green, and Blue (RGB) color space to Hue, Saturation, and Value (HSV) model. All images in this work are floating-point variables, so each pixel in each channel is in the range $[0, 1]$. This procedure is in Figure 4 which shows the original RGB image $\mathbf{I}$ of a lemon image and its respective $\mathbf{H}$, $\mathbf{S}$ and $\mathbf{V}$ channels.

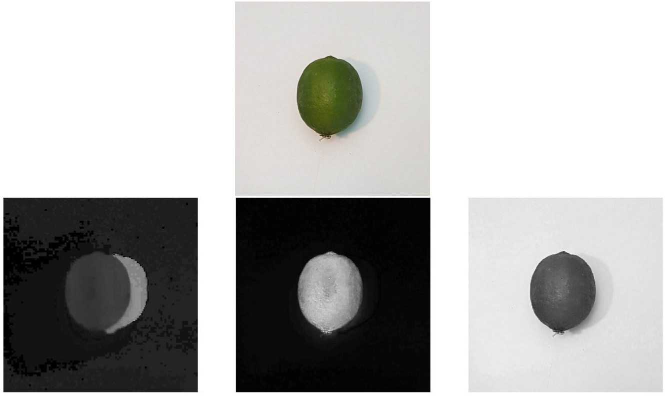

Fig. 4: *Above*: A sample from the dataset. *Below*: Its HSV color model components (respectively).

Figure 5 illustrates the masking operation. One can note that the $\mathbf{S}$ channel emphasizes the lime area from the background. The feature mapping occurs in this area, so we perform masking as follows. Let $\mathbf{G}$ be a Gaussian filter with standard deviation σ from Equation (1),

$$\sigma = 0.3(0.5(K-1)-1)+0.8, \tag{1}$$

with kernel size $K \times K$ pixels ($K = 5$ in this work). Equation (1) is described in the manual of *getGaussianKernel* OpenCV function. We can obtain the blurred channel $\mathbf{S_b}$ from a simple convolution. Then we binarize the blurred S channel, $\mathbf{S}_b$ by employing a threshold of 50% of the range. To fill the holes, gaps and remove saliences we use a morphological closing operation in the binarized image $\mathbf{S}_t$ with a disk-shaped structuring element $\boldsymbol{b}$ in a $B \times B$ pixels kernel ($B = 30$ in this work), resulting in the desired mask $\boldsymbol{\Delta}$. The closing operation can be obtained by employing a dilation followed by an erosion operation.

We obtain the region of interest RoI through AND logical operation, between the mask and the original image. The full process is shown in Figure 5.

We perform the feature space mapping within the Region of Interest (RoI) of each sample, as shown in Figure 6. The *findContours* OpenCV function compute the blue contour. Another OpenCV function employed was *minEnclosingCircle*. It was responsible for generating the red circle that circumscribes the contour. The first feature is the diameter D, which we use as

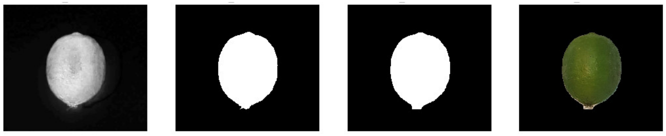

Fig. 5: The procedure used to generate the lime masks. From left to right, it consists of a blur, thresholding, closing, and AND operation between color image and binary mask.

the lime size measurement. We estimate the diameter through Equation (2),

$$D = 2(0.21)R, \tag{2}$$

where R is the radius of the circle given in [pixels] obtained by minEnclosingCircle, 0.21 is an experimentally obtained conversion constant given in $\left[\frac{\text{cm}}{\text{pixels}}\right]$ and D is the diameter in [cm].

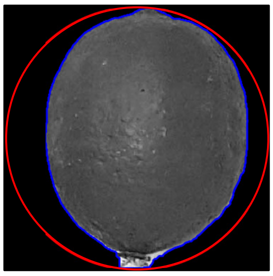

Fig. 6: A masked lime sample with its contour in blue and a red circle circumscribing the contour. The circle's radius represents the lime size.

We obtain the remaining features through the $\mathbf{H}$ and $\mathbf{V}$ channels of the RoI. The second feature is the histogram $\mathbf{h_H}$ of the $\mathbf{H}$ channel.

The texture features are measurements from the $\mathbf{V}$ channel. They are Skewness s_k, Kurtosis k_u and the following Gray-Level Co-occurrence Matrix $\mathbf{P}$ (GLCM) features: contrast c_n, dissimilarity d_i, homogeneity h_o, Angular Second Moment (ASM) a, energy e_n and correlation c_r. All the features were obtained through *scikit* library [11], which are based on [5].

The next step of the system is the classification, which is performed in a feature space that combines size, color and texture measurements.

B. Classification

This work addresses three different classification techniques to evaluate their capability of dealing with the tasks under analysis. The classifiers were ANN, SVM, and RF. There are two scenarios for each type of classifier, one to distinguish between a mature and an immature lime (experiment 1), and the other to classify a lime with from others without defects (experiment 2). All algorithms and parameters applied for each classifier used are available on *scikit-learn* python library [11]. At the end of every classifier explanation a table is presented for each classifier containing the best parameters configuration performed by grid search in each experiment (Exp).

1) Artificial Neural Network: An ANN is an information processing system inspired by biological neural networks. However, this method is not identical (as far as the operational process is concerned) to the brain neural networks [4].

In general, three types of architectures stand out in ANN studies: single-layer feedforward networks, multiple layer feedforward networks, and recurrent networks as described in [6]. In this work, we use the multiple-layer feedforward network (known as multi-layer perceptron network - MLP) and backpropagation as it can recognize and separate linearly non-separable patterns and according to [16], backpropagation minimizes the error obtained by the network. Results using this technique could be seen in Table I.

TABLE I: Grid search performed in the parameter space of the training set using ANN. The last two columns are the best parameters for each experiment.

Parameter	Grid Search	Exp 1	Exp 2
Neurons in hidden layer	1, 2, 5, 10, 15, 25, 50, 100, 150, 200, 225, 250 and 300	300	250
Activation function	('tanh', 'relu')	relu	relu
Optimization algorigm	('sgd', 'adam', 'lbfgs')	adam	adam
L2 penalty (regularization term α)	0.0001, 0.001, 0.01, 0.05, 0.1, 10^{-5}, 10^{-6} and 1	0.001	10^{-6}
Learning Rate	('constant', 'adaptive')	*adaptive*	*adaptive*

2) Support Vector Machines: SVMs were developed by Vapnik [15] and his theory establishes a series of principles that one must follow to obtain classifiers with good generalization, defined as their ability to correctly predict the class of new data from the same domain in which the learning occurred. Some of the features of SVMs that make their use attractive are in [14], as follows: good generalization capability; robustness in large dimensions; and convexity of the objective function. The performance of SVM approach is shown in Table II.

TABLE II: Grid search performed in the parameter space of the training set using SVM. The last two columns are the best parameters for each experiment.

Parameter	Grid Search	Exp 1	Exp 2
C	1, 0.25, 0.5, 0.75, 0.05, 0.001, 0.01, 0.1, 5, 10, 50, 100, 500, 1000	1	1
γ	0.1, 0.5, 1, 3, 5, 7, 10, 100 and *auto*	0.1	3
Kernel	*poly* and *rbf*	*poly*	*poly*

3) Random Forest: RF is a decision-tree-based machine learning algorithm used in this case to solve a classification. This method uses bagging for creating multiple models (decision trees) from a single training dataset, resulting in a very powerful classifier developed and described by [2] and [7]. Table III shows the results of applying RF.

In general, forests with trees maximum depth above 5 can lead to overfitting (a model too specialized in the training set with no generalization capability). However, it did not occur in the results, as can be seen in Section IV.

TABLE III: Grid search performed in the parameter space of the training set using RF. The last two columns are the best parameters for each experiment.

Parameter	Grid Search	Exp 1	Exp 2
Number of trees	$1, 3, 5, \cdots, 1500$	101	55
Maximum number of features at every split	*auto*, *sqrt* and *log*	*sqrt*	*sqrt*
Maximum tree depth	$1, 2, 3, \cdots, 30$	14	23
Minimum number of samples to split	$2, 3, 4, 5$	3	2
Minimum number of samples at each node leaf	$2, 3, 4, 5$	2	2
Bootstrap	*true*, *false*	*false*	*false*

IV. Results and discussion

In this section, we introduce the metrics used in this work. Then we present and discuss the results obtained in each experiment. Later on, we compare the features of the proposed system with the ones in the literature.

It is usual to evaluate classification problems by metrics derived from the confusion matrix. This work used *sklearn* [13] to generate them for each experiment. Another evaluate measure used is balanced accuracy, $\rho = \frac{1}{2}\left(\frac{T_A}{T_A+F_B} + \frac{T_B}{T_B+F_A}\right)$, applied because classes are not balanced regarding the number of samples for each class. At the balanced accuracy (ρ) formula T_A means true positive, T_B true negative, F_A false positive and F_B false negative, and it was applied because classes are not balanced regarding the number of samples for each class.

A. Accuracy

The accuracy of the classification systems concerning the maturity and defects of the limes are in Tables IV and V, respectively.

Table IV presents the performance for the maturity classification. The classes mature and immature limes present 80 and 48 samples, respectively. The classifiers SVM, ANN, and RF achieved 72.9%, 73.8%, and 81.9% of hit rate on the test set, respectively, showing the performance superiority of the RF for the problem under analysis. It is worth noticing that the sensitivity for mature lime class $\left(\frac{T_A}{T_A+F_B}\right)$ was 88.8% using RF.

TABLE IV: Mature × immature limes classification.

Classifier	T_A	F_A	T_B	F_B	ρ
SVMs	55	11	37	25	72.9%
ANN	58	12	36	22	73.8%
RF	71	12	36	9	81.9%

TABLE V: With × without defects limes classification.

Classifier	T_A	F_A	T_B	F_B	ρ
SVMs	49	12	51	16	78.2%
ANN	44	13	50	21	73.5%
RF	50	10	53	15	80.5%

Table V presents the performance for the defect classification in the limes. Different from the maturity classification, where the classes differ by almost twice the number of images, the defect classification presents 65 images with and 63 images without defects classes, which represent practically the same amount. The classifiers SVM, ANN, and RF achieved 78.2%, 73.5%, and 80.5% of hit rate on the test set, respectively, showing a slight improvement in using the RF classifier. It is worth noticing that the highest specificity, $\left(\frac{T_B}{T_B+F_A}\right)$, was 84.1%, also using RF.

In all cases, the average time to extract the features is 76.6 ms per image. This time is at least 25 times higher than the time necessary to classify the features vectors using the RF, which is the slowest classifier. One can consider the proposed system as real-time if the acquisition rate is less than 13 frames per second. The computer used in the experiments has a Processor Intel(R) Core™ i5-6200U CPU @ 2.30GHz, 2400 Mhz, 2 Core(s), 4 Logical Processor(s) and 8 GB of RAM memory. The system was implemented in python.

B. Comparison with the related works

Table VI has a comparison between the proposed method and the ones briefly described in Section I. The proposed method is the only one in the table addressing both problems, maturity, and defects. Besides, one can note that the proposed method is the only one to compose a citrus fruit dataset large enough for modern machine learning methods considering the tasks under analysis. Accuracies reported by the authors are in the last column. Since the datasets are different, it is crucial to point out that the values are merely illustrative. Yet, we consider that our results are satisfactory and promising.

V. Conclusions

This paper describes a real-time method for Persian lime classification using computer vision. It classifies limes according to the maturity and presence/absence of defects employing three classifiers, ANN, SVM, and RF. The proposed method performs classification in a feature space that combines size, shape, color, and texture measures. Also, we produced a dataset composed of images from hundreds of limes in several conditions. To increase the dataset size and solve an unbalance issue, we employed data augmentation based on random rotations. The proposed method achieves an accuracy of 81.9% of hit rate in maturity classification and 80.5% in defect classification using RF.

References

[1] J. Blasco, N. Aleixos, J. Gómez-Sanchís, and E. Moltó. Recognition and classification of external skin damage in citrus fruits using multispectral data and morphological features. *Biosystems Engineering*, 103(2):137–145, June 2009.

[2] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct. 2001.

[3] Arthur Z. da Costa, Hugo E.H. Figueroa, and Juliana A. Fracarolli. Computer vision based detection of external defects on tomatoes using deep learning. *Biosystems Engineering*, 190(1):131–144, Feb. 2020.

[4] Laurene Fausett. *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*. Pearson, 1993.

[5] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, Nov. 1973.

TABLE VI: Summary of the references presented in Section I associated with related works. The abbreviations are for Ref: Reference, M: Maturity, D: Defect, F: Fruit, Dsize: Dataset size (number of images), AT:Applied techniques (CAM: Chromatic aberration map, RF: Random forest, SHP: Shape, C: Color, NIR: Near infrared, UVFL: Ultraviolet induced fluorescence and BDA: Bayesian discriminant analysis) and Acc: Accuracy (R: Ratio between detected and labelled number of pixels).

Ref	**M**	**D**	**F**	**Dsize**	**AT**	**Acc**
[1]	No	No	Citrus (orange & mandarin)	2,132	NIR+UVFL+RGB, BDA & 11 classes	86% (11 defects classification)
[10]	No	No	Citrus (orange)	13	CAM+HM	92.6% (R)
[12]	No	No	Round fruits	10	HoG, PIC, RF	82% (F_1-score)
[17]	No	No	Fruits (orange, apple & strawberry)	178	SHP, C, SIFT, k-NN, SVM, RF	79∼96%
[8]	No	No	Citrus (Emperor citrus)	506	RGB+depth, SVM	92% (F_1-score)
[3]	No	Yes	Tomato	43,843	ResNet50	94.6% (precision)
[9]	Yes	No	Citrus (sweet orange)	70	LPB, CHT, RusBoost	82.3% (precision)
Proposed	**Yes**	**Yes**	**Citrus (Persian lime)**	**1,087**	**Histogram+GLCM SVM, ANN, RF**	**81.9% (maturity) 80.5% (defects)**

[6] Simon Haykin. *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice Hall, 1998.

[7] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*. Springer-Verlag New York, 2013.

[8] Guichao Lin, Yunchao Tang, Xiangjun Zou, Jinhui Li, and Juntao Xiong. In-field citrus detection and localisation based on rgb-d image analysis. *Biosystems Engineering*, 186(1):34–44, Oct. 2019.

[9] Jun Lu, Won Suk Lee, Hao Gan, and Xiuwen Hu. Immature citrus fruit detection based on local binary pattern feature and hierarchical contour analysis. *Biosystems Engineering*, 171(1):78–90, July 2018.

[10] Jun Lu, Nong Sang, Yangyan Ou, Zhuo Huang, and Pujuan Shi. Detecting citrus fruits with shadow within tree canopy by a fusing method. In *2012 5th International Congress on Image and Signal Processing*, pages 1229–1232, Chongqing, China, Oct. 2012.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. *Scikit-learn*: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[12] Zania S Pothen and Stephen Nuske. Texture-based fruit detection via images using the smooth patterns on the fruit. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5171–5176, Stockholm, Sweden, May 2016.

[13] *sklearn*. *sklearn*. https://scikit-learn.org/stable/index.html, 2022. (Date accessed: 2022-01-06).

[14] Alexander J. Smola, Peter Bartlett, Bernhard Schölkopf, and Dale Schuurmans. *Advances in Large-Margin Classifiers*. MIT Press, 1999.

[15] Vladmir Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, 1995.

[16] Paul John Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Harvard University, 1975.

[17] Hossam M. Zawbaa, Maryam Hazman, Mona Abbass, and Aboul Ella Hassanien. Automatic fruit classification using random forest algorithm. In *2014 14th International Conference on Hybrid Intelligent Systems*, pages 164–168, Kuwait, Kuwait, Dec. 2014.