

PAPER • OPEN ACCESS

## Application of computer vision and low-cost artificial intelligence for the identification of phytopathogenic factors in the agro-industry sector

To cite this article: J J Castro-Maldonado *et al* 2018 *J. Phys.: Conf. Ser.* **1126** 012022

View the [article online](#) for updates and enhancements.



**IOP | ebooks™**

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

# Application of computer vision and low-cost artificial intelligence for the identification of phytopathogenic factors in the agro-industry sector

J J Castro-Maldonado<sup>1,2</sup>, J A Patiño-Murillo<sup>1,2</sup>, A E Florian-Villa<sup>1,2</sup> and O E Guadrón-Guerrero<sup>3</sup>

<sup>1</sup> Grupo de Investigación en Gerencia y Aplicación de la Ciencia y la Tecnología, Servicio Nacional de Aprendizaje (SENA), Medellín, Colombia

<sup>2</sup> Red Tecnoparque Colombia, Servicio Nacional de Aprendizaje (SENA), Medellín, Colombia

<sup>3</sup> Universidad de Pamplona, Pamplona, Colombia

E-mail: johnjacama@gmail.com

**Abstract.** This work presents a perspective of the processes of phytosanitary control in crops, starting with traditional methods and strategies applied for precision agriculture and the use of artificial intelligence and computer vision in this area. Then, the article describes the approach for developing the proposed algorithm based on artificial intelligence and computer vision for phytopathogenic detection. The methodology for the development and validation stages is specifically discussed. Several tests were carried out with the different image processing algorithms studied. Results show how the selected method with Haar filters and Gradient-oriented Histograms performs effectively for identification of phytopathogenic factors, from both qualitative and quantitative analysis.

## 1. Introduction

Traditional agriculture is carried out following a pattern of homogeneity, assuming that all the land to cultivate has the same characteristics. Therefore, this conventional approach ignores the changing properties that must be considered for agricultural production and that increase the efficiency and effectiveness of the industry value chain in this field [1]. In consequence, the level of production of the crop varies in different areas within the same lots, so there will be areas with higher productivity than others and therefore requiring a different dose of plague control substances (fungicides, pesticides) and fertilizers [2]. These changing soil properties may vary due to pedogenetic factors and the anthropic action [3].

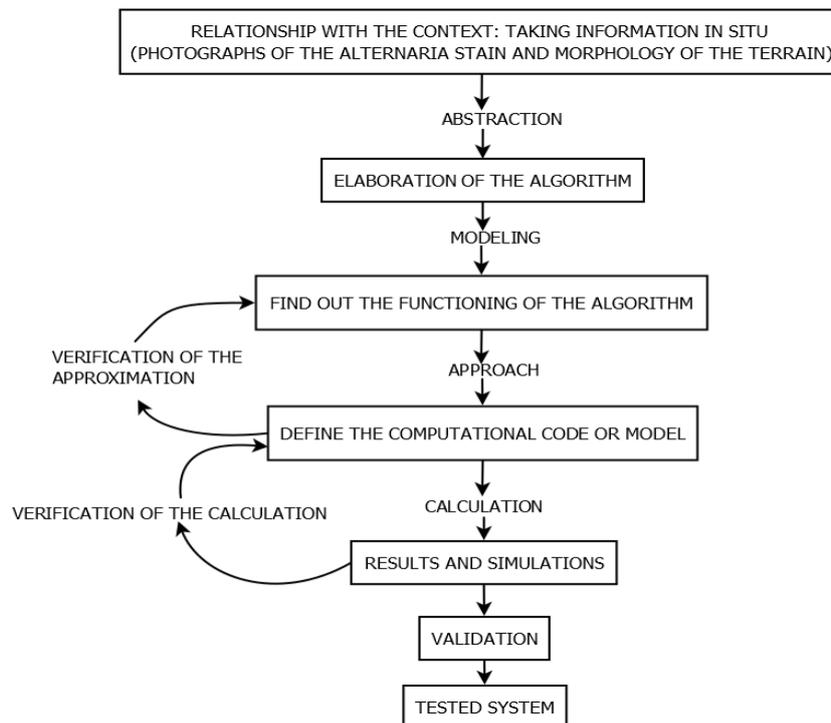
Currently, agriculture must satisfy the needs of the continuous increase in the world population. In this sense, it requires high investments in human capital, water, fertilizers and other supplies for the management of crops. All these factors contribute to the environmental deterioration, especially in soil and water [4,5]. This environmental degradation leads to the concept of sustainability in agriculture, which aims to maintain the quality of the soil, based on indicators for the continuous monitoring of the use and management of the land. In turn, this approach allows the implementation of different strategies for the conservation of the natural resources [2].



In this regard, precision agriculture has arisen as a farming alternative able to recognize and understand spatial variability, using this information to perform differential management strategies to improve efficiency in production [1]. Moreover, precision agriculture can be conceived as the development and application of technologies to establish the spatial behavior of soils for the optimization of agricultural resources [6].

This sustainable approach is supported by the development of computer and communication tools allowing the improvement of many processes in the industrial and productive areas [7]. However, the use of this technologies can be expensive, making difficult their implementation for small production units. This situation applies to the lulo culture in Colombia. Lulo (*Solanum Quitaense*) is a coveted fruit for juice and dessert production, and it is cultivated in non-extensive plantations at the south of Colombia. Given the need for the improvement of the monitoring techniques in lulo cultures, artificial vision techniques arise as useful alternatives for the development of a low-cost system for the phytosanitary control of lulo crops. This system will process and identify images that show the presence of phytosanitary problems in lulo crops, optimizing the growing conditions to increase the performance and productivity of the lulo farms.

In this context, this work proposes an algorithm for the processing of the images of lulo plant leaves to identify the presence of phytopathogens organisms. The algorithm was developed following the structure shown in Figure 1. Results show the effectivity of the method in an off-line simulation environment.



**Figure 1.** Methodological phases for the development of the algorithm. Adapted from the ASME standard V&V 10.1-2012 [8].

## 2. Experimental procedure

In this section, we mention the sequential steps that were carried out for the realization of this work. At first, we went through a research and analysis stage of the methodologies and applications related to artificial vision and image processing in different fields. As described in the next section, six methods for image detection with different characteristics such as color, geometric forms or contour

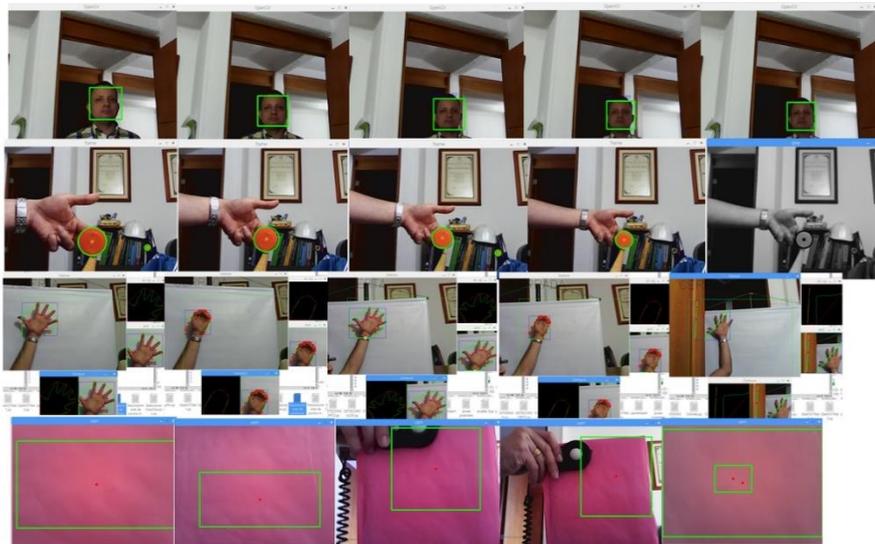
segmentation were explored. This exploration led to the selection of the most suitable strategy for algorithm development in our application, following the schematic procedure of Figure 1.

The process continued with the selection of the required hardware accessories for the development of the system. As the focus was oriented towards a low-cost solution in both software and hardware, the choices were Python and the OpenCV libraries for programming the algorithm and the Raspberry Pi development card for data processing [9]. The open-access characteristics of the selected programming languages were a good match with the performance level and the open-source application of the Raspberry Pi.

For the validation process, the algorithm was evaluated through a simulated environment with the image acquisition hardware and with the parameters extracted from performance analysis in Matlab®. A series of photographs of plant leaves infected with the pathology of interest were acquired with the image capture system. Performance in this task was analyzed for two main features reported in several reference works, such as distance from the object to the image sensor [10,11] and the percentage of detection as a function of time [12,13]. At last, the algorithms were evaluated and compared through the calculation of parameters such as sensitivity, precision and the effectivity ratio.

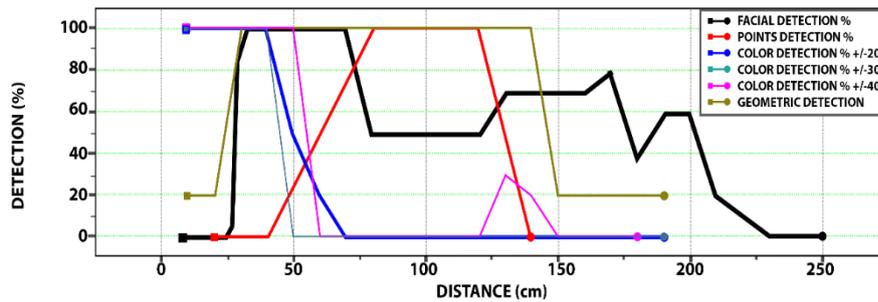
### 3. Results and discussion

Initially, we deployed and validated different algorithms found in the literature that employ low-cost software and hardware [14], to identify their properties and select the one that best addresses the application requirements. The studied algorithms were based on both segmentation through histograms or layers and inspired on artificial intelligence techniques. In this section, we mention the sequential steps that were carried out for the realization of this work.



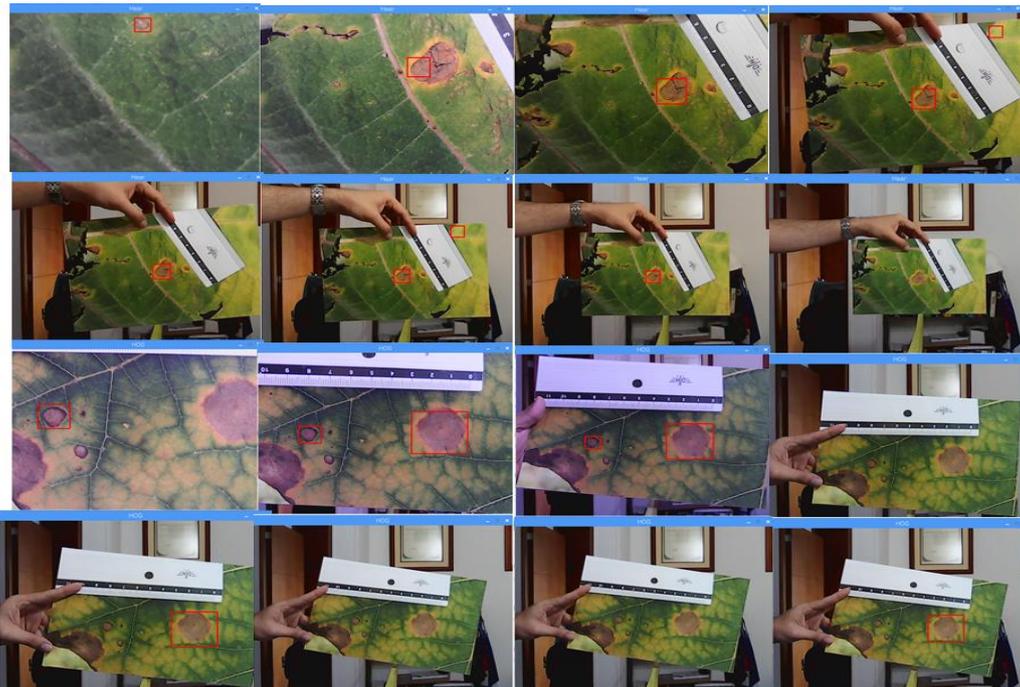
**Figure 2.** Some of the experimental works performed to verify and to validate the functioning of each of the analyzed algorithms

The first selected method involved a facial detection algorithm programmed under the cascade detector strategy or AdaBoost [15–17]. This code can *learn* to detect the desired object from a set of test images. Then we worked with a “point algorithm” based on routines oriented to geometric calculation, filtering and segmenting the image to identify contours and points of interest. Later, we analyzed three variations of color detection algorithms through RGB layers, which work with some filters and masks for the recognition of the desired color to detect. Finally, we use an algorithm to detect geometric figures based in "HoughCircles", an OpenCV library designed to find circular patterns within the image. This last code was selected because of the pathogen of study forms a circular spot in the crop. Figure 2 illustrates some of the validation images of the employed methods.



**Figure 3.** Detection percentage as a function of time and the distance between the object to be detected and the camera sensor.

For the performance comparison of the different algorithms, the detection percentage was calculated as a function of the distance between the object to be detected and the camera sensor. Figure 3 presents the detection efficiency of the studied algorithms, in the context of our application. The data was acquired with unstable levels of illuminance, to simulate the real operating conditions of the actual implementation in exterior locations. The detection was calculated in a time-lapse under two minutes, after that time the process was declared as failed. As seen in Figure 3, the facial detection and the geometric algorithms were the most successful regarding detection percentage. Nevertheless, more tests are required for a more in-depth evaluation of the sensitivity and precision of the algorithms.



**Figure 4.** Experimental work developed for the validation of the proposed algorithm. The detected spot can be appreciated by red boxes in the pictures.

Based on the previous analysis, we adopted three strategies looking to exploit the advantages of each studied method. In this way, one generated code mixed the elements of geometric figures and color detection, other used Adaboost detectors with Haar filters and the third one employed Adaboost with Histograms of Oriented Gradients (HOG) [18]. These techniques were selected to identify the features of the pathology of interests, represented by the spot on the lulo cultivation leaf caused by the *Alternaria* fungus. Figure 4 presents the experimental work for the validation of the proposed algorithms. The detected spot can be appreciated by red boxes in the pictures.

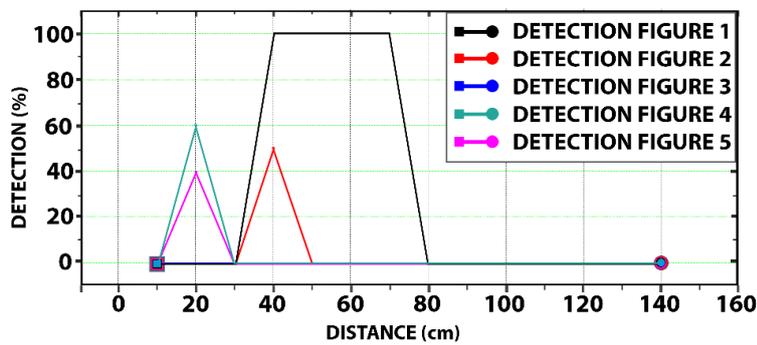


Figure 5. Detection percentage for the algorithm detecting geometric figures and RGB color masks.

In the same way as the previously executed comparison of algorithms, the performance of the proposed codes was evaluated with the calculation of the detection percentage for several test images. Figure 5 shows the results obtained for the algorithm based on the detection of geometric figures and RGB color masks. The effectivity of this method is almost negligible, only presenting a satisfactory result test image 1 for a range between 40 and 80 centimeters.

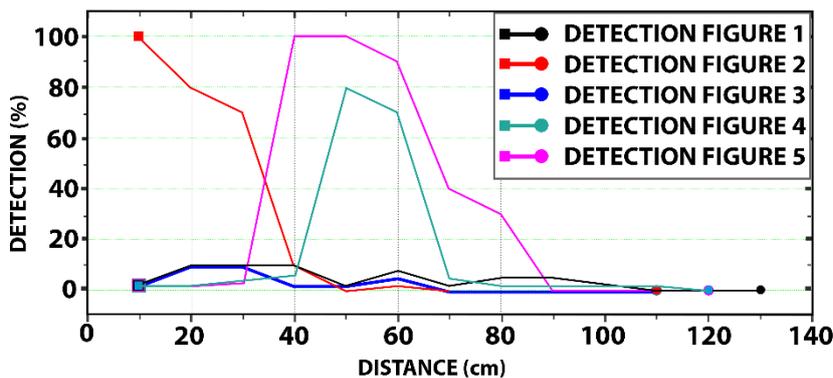


Figure 6. Detection percentage for the algorithm of cascade detectors with Haar filters.

Figure 6 presents the results from the validation of the algorithm with the Adaboost technique using Haar type filters. Although the detection percentage increased with this strategy when compared with the color detection code, the accuracy of the method remains pretty low. The greater detection range was reached for the test image 4, in a region from 30 to 110 centimeters.

Figure 7 depicts the detection percentage for the algorithm using cascade detectors with HOG. This alternative is more effective than the others, presenting some degree of target detection for all of the test images. The largest detection range occurred for test image 2, extending from 10 to 110 centimeters.

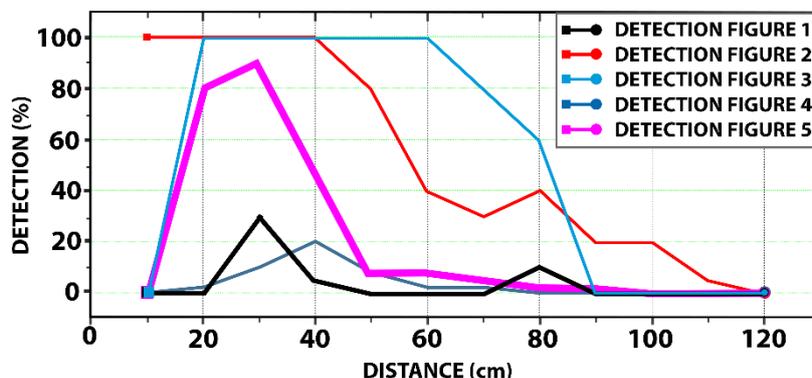
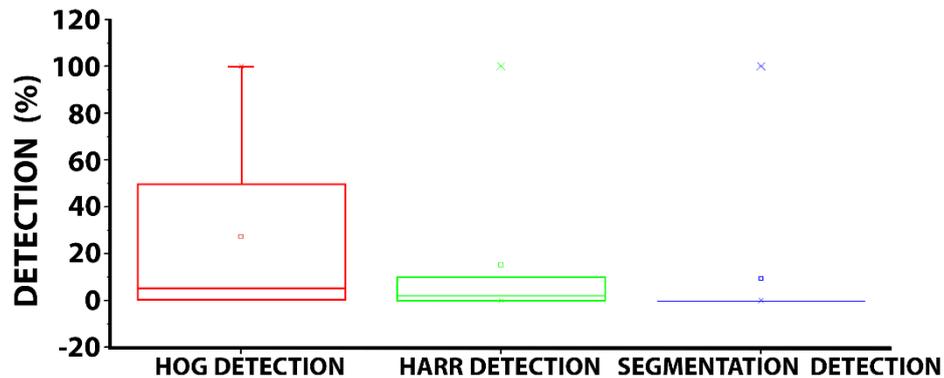


Figure 7. Detection percentage for the algorithm of cascade detectors with HOG method.

Detection performance for the three contrasted alternatives is pretty clear from Figures 5 to 7. This is verified in Figure 8, showing the boxplots [19] of the distribution of detection percentages for the three techniques. Despite the small mean values obtained, the superiority of the AdaBoost method with HOG is evident. Consequently, this is the selected alternative for project implementation.



**Figure 8.** Boxplots of the distribution of detection percentages for the three explored techniques.

The base code with Adaboost and HOG detection is validated using Matlab®, to establish the sensitivity and accuracy of the algorithm. Figure 9 shows those results, with the implemented solution generating many false positives. However, the detection in some cases is very efficient despite the lack of definition of the object to be identified. This result is evidence of the robustness of the solution; nevertheless, aspects such as the number of positive images in the training data of the implemented code should be studied in future works.



**Figure 9.** A sample of the Matlab validation results of the selected Adaboost algorithm with HOG using real test images of the lulo plant leaves.

#### 4. Conclusions

This work described the selection procedure of an algorithm intended for the development of a low-cost detection system for pathologies in the lulo plant. Tests determined that algorithms for the detection of geometric figures and those developed from cascade detectors are more robust than the methods based on color segmentation. This difference in performance is due to the illuminance factor, which is an effect generating a lot of noise in detection despite the high RGB tolerance.

Although the artificial intelligence algorithm of cascade detectors using Adaboost with Haar filters is widely used in face detection, effectivity was lower when compared to the Adaboost algorithm developed with oriented gradient histogram (HOG) descriptors for the intended application.

The training process and the database generation for this purpose are fundamental aspects of the development of an artificial intelligence algorithm. Therefore, the number of negative images should double the number of positive images to obtain an acceptable efficiency score of detection. Also, an appropriate process of image scaling and selection of the region of interest (ROI) is mandatory. The procedure described in this work shows the feasibility of the development of a system to detect phytopathogenic factors in the lulo plant leaves.

## References

- [1] Jaramillo DF, Sadeghian S and Lince LA 2013 *Bol. Cienc. Tierra* **33** 111
- [2] Muñoz JD, Martínez LJ and Giraldo R 2006 *Agron. Colomb.* **24** 355
- [3] Rodríguez J, González AM, Leiva FR and Guerrero, L 2008 *Agron. Colomb.* **26** 308
- [4] Piscitelli M, Ibáñez G and Pérez J 2016 *Rev. CINTEX* **21** 17
- [5] Valencia F, Patino J, Lopez JD and Espinosa J 2013 *13th IFAC Symposium on large scale complex systems: Theory and applications* (Shanghai: IFAC) **538**
- [6] Garzón CA, Cortés C, and Camacho JH 2010 *Rev. UDCA Actual. Divulg. Científica* **13** 87
- [7] Capraro F, Tosetti S, and Vita F 2010 *Rev. Iberoam. Automática E Informática Ind. RIAI* **7** 73
- [8] ASME 2012 *ASME V & V 10.1-2012: an American national standard* (New York: The American Society of Mechanical Engineers)
- [9] Rodriguez AA, Figueredo J, and Chica JA 2018 *Rev. Espac* **39** 30
- [10] Arnal JG 2013 *SpringerPlus* **2** 1
- [11] Yamamoto K, Guo W, Yoshioka Y and Ninomiya S 2014 *Sensors* **14** 191
- [12] Bhange M and Hingoliwala HA 2015 *Procedia Comput. Sci.* **58** 280
- [13] González FA, Gómez JJ and Amaya DF 2017 *Rev. CINTEX* **22** 51
- [14] Mohapatra AG and Lenka SK 2016 *Procedia Comput. Sci.* **78** 499
- [15] Lienhart R, Kuranov A and Pisarevsky V 2003 Empirical analysis of detection cascades of boosted classifiers for rapid object detection *Pattern Recognition Lecture Notes in Computer Science* vol 2781 ed B Michaelis and G Krell (Berlin, Heidelberg: Springer) pp 297-304
- [16] Guevara ML, Echeverry JD and Ardila W 2008 *Sci. Tech.* **14** 1
- [17] Rajesh K and Dhuli R 2018 *Biomed. Signal Process. Control* **41** 242
- [18] Taquíá Gutiérrez JA 2017 *Interfases* **10** 11
- [19] Castro-Maldonado J J 2018 *Diseño de un sistema de visión artificial de bajo costo para el control fitosanitario de cultivos de lulo* (Pamplona: Universidad de Pamplona)