ISSN 2007-9737

A Greedy Algorithm for Highlighting of Color Dominance in Tomato Leaves and Fruit Segmentation

Juan Pablo Guerra Ibarra*, Francisco Cuevas de la Rosa

Centro de Investigaciones en Óptica A.C.,
Mexico

{juangi, fjcuevas}@cio.mx

Abstract. The improvement of agricultural processes in recent years through the application of various technologies, including computational algorithms, has given rise to a field of research called precision agriculture. This field aims to provide the plant with the resources it needs for its development at the right time. Deficiencies of a nutrient element necessary for the development of a plant are mainly manifested in the leaves. In this paper, a greedy algorithm is proposed in order to optimize the segmentation method by color dominance that seeks to emphasize the dominance of the green color present in the leaves and the red of the ripe fruits of tomato plants existing naturally using the *RGB* color model. The algorithm searches a numerical range for the value that maximizes color dominance, the range is reduced until a stop condition is reached. The objective function to maximize is the average performance when segmenting the pixels of the leaves and fruits. The classification images can be used in the detection of pests, diseases or nutritional deficiencies.

Keywords. Optimization, greedy algorithm, segmentation, precision agriculture, computer vision.

1 Introduction

Resources optimization is a fundamental task for human activities since it maximizes the results and minimizes the use of the costs inherent to the process. Agriculture is a human activity that has always been optimized with the use of different technologies to maximize its production, for example the plow, and in recent years to make it sustainable in what is called Precision Agriculture (*PA*) [3]. For the correct development of plants, 15 chemical elements are necessary: Carbon, Oxygen, Hydrogen, Nitrogen, Phosphorus, Potassium, Calcium, Magnesium and Sulfur (Macros: 60-200 Kg/Ha). The other essential ones are Iron, Manganese, Copper, Zinc, Boron (micro: $\geq$ 50 Kg/Ha) [10]. The process of fertilization is fundamental in *PA*, providing crops with the necessary amounts of macro and micro nutrients to increase crop productivity. The lack or excess of any of the elements is mainly manifested in the color and shape of the leaves [1, 8, 15]. One of the first steps for the detection of nutrient deficiencies in crops using Computer Vision (*CV*) algorithms is to classify or segment the pixels of the images into a previously defined class [12, 16].

The performance of the segmentation method in the dominance of green color for leaves and red for fruits presented in [5], depends on the values assigned to the parameters named α_1 and α_2. Optimization problems related to *CV* have two fundamental aspects to be solved, the first one is how to pose a cost function that provides an optimal solution through one or more parameters and the second one is to pose an efficient numerical method to find those parameters [21]. In this paper, we present an adaptation of a Greedy Algorithm (*GA*) to find the optimal value of the parameters α_1 and α_2 that maximize the average of the yield metrics in the task of segmenting the leaves and fruits of tomato plants by means of the method of Segmentation by Color Dominance (*SCD*), which is an improvement over the method presented in [6]. The *GA* searches for the optimal value of the parameters α_1 and α_2 in a range bounded by a lower and an upper limit, a step value is established which takes values from the established range. The search space and the step value are reduced with each iteration, until the stop condition of the algorithm is reached.

ISSN 2007-9737

The rest of the paper is organized as follows: Section 2 contains a brief description of papers necessary to contextualize the presented method. The methods and necessary elements are presented in Section 3 including the greedy algorithm. Section 4 contains the results generated in the segmentation tasks of the leaves and fruits of the tomato plant. A comparison of results between the greedy algorithm and a CNN is made in Section 5. Conclusions are presented in Section 6.

2 Related Work

The objective of image segmentation is to partition or classify pixels into a particular class according to a priori information. The difficulty of an accurate segmentation increases proportionally to the number of classes. Segmentation methods, especially those related to thresholding, use parameters that can be optimized by applying some type of metaheuristic algorithm [11]. For example, Puranik [14] used a Particle Swarm Optimization (*PSO*) algorithm to determine the fuzzy rules for the segmentation of a color image, the goal is to produce a smaller number of fuzzy rules. Ghamisi [4] presented an improvement of the *PSO* method, called fractional-order Darwinian, with the goal of segmenting hyperspectral images. Additionaly, it is mentioned as the main advantage the use of several traditional *PSO* swarms running in parallel to improve the ability to escape local optima, and a fractional computation is added to control the convergence speed.

Tao [19] proposed a method to optimize 6 integer parameters using a Genetic Algorithm to segment a grayscale image into three different classes. The method presented by Liang [9] developed a combination of an Ant Colony Optimization (*ACO*) algorithm with the classical Otsu segmentation method, reported to be more efficient in terms of speed for segmenting images from 2 to 4 color levels. Vergues [20] presents a genetic algorithm for graph segmentation that overcomes the limitations of similar approaches, such as the inability of certain regions to expand after certain steps and the uniqueness of the solution. This is achieved without compromising the segmentation efficiency by introducing a generalized concept of non-over-segmentation. The method incorporates an energy function and eliminates the non-sub-segmentation criterion, replacing it with a probabilistic criterion similar to simulated annealing. Subr [17] presented a method based on a greedy iterative algorithm to enhance local contrast by posing it as an optimization problem, employing a scalar objective function for estimating the average local contrast of the image. The comparison of results of segmenting leaves of different types of plants presented by Patil [13] using the greedy snake algorithm is compared with the Kass M snake algorithm. From the comparison, it is observed that the *GA* is faster and more efficient than the Kass algorithm in terms of required iterations to obtain the desired contour of an image.

3 Methodology and Materials

3.1 Color Dominance Algorithm

The numerical handling of images is usually represented by a bidimensional function $f(x,y)$.The algorithm utilizes the *RGB* color model and refers to each color band with the subscripts R, G, B for the colors **R**ed, **G**reen and **B**lue, employs (*CV*) technique, thresholding. The segmentation process is carried out in two stages [5]. The first one is based on the Equations 1 and 2 for the leaves and the fruits, respectively:

$$h(x,y) = \begin{cases} f(x,y) & if \quad (f_G(x,y) \geq f_R(x,y)) \\ & and \quad (f_G(x,y) \geq f_B(x,y)) \\ 0 & in \quad another \quad case \end{cases}, \tag{1}$$

$$j(x,y) = \begin{cases} f(x,y) & if \quad (f_R(x,y) \geq f_G(x,y)) \\ & and \quad (f_R(x,y) \geq f_B(x,y)) \\ 0 & in \quad another \quad case \end{cases}, \tag{2}$$

where $h(x,y)$ contains the pixels that are filtered from $f(x,y)$ with dominance of the green color channel over the other two and $j(x,y)$ contains the pixels with dominance of the red color channel $\forall x = 0,1,2,...,M$ and $\forall y = 0,1,2,...,N$, correspond to the width and height of the image respectively.

The second stage begins with the calculation of four differences, two for each dominant color channel. Equations 3 and 4 calculate the differences of the green dominant color channel over the other two of the function, while Equations 5 and 6 do the same for the red color channel in the function $j(x, y)$:

$$\Delta_G^R(x, y) = h_G(x, y) - h_R(x, y), \tag{3}$$

$$\Delta_G^B(x, y) = h_G(x, y) - h_B(x, y), \tag{4}$$

$$\Delta_R^G(x, y) = j_R(x, y) - j_G(x, y), \tag{5}$$

$$\Delta_R^B(x, y) = j_R(x, y) - j_B(x, y). \tag{6}$$

In the final stage of segmentation, the values of the four thresholds are determined based on Equations 7, 8 for the leaves, and Equations 9, 10 for the fruits:

$$U_G^R(x, y) = \frac{h_G(x,y)}{M_G} * \sigma_G^R * \alpha_1, \tag{7}$$

$$U_G^B(x, y) = \frac{h_G(x,y)}{M_G} * \sigma_G^B * \alpha_1, \tag{8}$$

$$U_R^G(x, y) = \frac{j_R(x,y)}{M_R} * \sigma_R^G * \alpha_2, \tag{9}$$

$$U_R^B(x, y) = \frac{j_R(x,y)}{M_R} * \sigma_R^B * \alpha_2. \tag{10}$$

where $U_G^R(x, y)$ and $U_G^B(x, y)$ are the thresholds used to detect leaves, and $U_R^G(x, y)$, $U_R^B(x, y)$ are utilized to find the fruit region. M_G and M_R are the highest values of the green and red color channels in $h(x, y)$ and $j(x, y)$, respectively. σ_G^R, σ_G^B, σ_R^G, and σ_R^B correspond to the standard deviations of the Δ_G^R, Δ_G^B, Δ_R^G, and Δ_R^B values, respectively. Finally, α_1 and α_2 are factors utilized to control the performance of thresholds, on which the final segmentation depends.

The image $h_F(x, y)$ with the pixels that make up the leaves filtered from $h(x, y)$ and the image $j_F(x, y)$ with the pixels that make up the fruits filtered from $j(x, y)$ $\forall x = 0, 1, 2, ..., M$ and $\forall y = 0, 1, 2, ..., N$ is obtained with Equation 11:

$$h_F(x,y) = \begin{cases} h(x,y) & if \quad (\Delta_G^R(x,y) > U_G^R(x,y)) \\ & and \quad (\Delta_G^B(x,y)) > U_G^B(x,y))) \\ 0 & in \quad another \quad case, \end{cases} \tag{11}$$

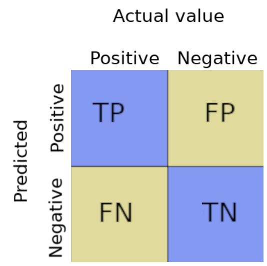


Fig. 1. Confusion matrix

On the other hand, the final segmentation of the pixels that make up the fruit is done with the Equation 12:

$$j_F(x,y) = \begin{cases} j(x,y) & if \quad ((\Delta_R^G(x,y) > U_R^G(x,y)) \\ & and \quad (\Delta_R^B(x,y) > U_R^B(x,y))), \\ 0 & in \quad another \quad case \end{cases} . \tag{12}$$

3.2 Metrics Performance

When evaluating outcomes, it is important to measure them quantitatively. This approach ensures that the results are objective, precise, and easily comparable, providing a clear and accurate assessment of performance. Quantitative measurement allows for data-driven analysis and informed decision-making. In this case, 5 metrics are used, which are: *Accuracy*, *Precision*, *Recall*, *F1-Score* [18] and *IoU* [22]. Four fundamental concepts must be defined: *True Positives* (*TP*), *True Negatives* (*TN*), *False Positives* (*FP*) and *False Negatives* (*FN*), as defined by the confusion matrix illustrated in Figure 1.

The *Accuracy* metric is defined as in Equation13:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}. \tag{13}$$

The *Precision* metric, are calculated using Equation 14:

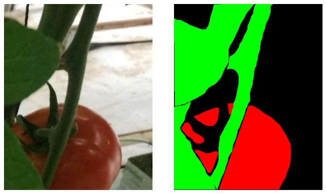
$$Precision = \frac{TP}{TP+FP}. \tag{14}$$

The *Recall* metric is defined as in Equation15

$$Recall = \frac{TP}{TP+FN}. \tag{15}$$

F1-Score combines *Precision* and *Recall*, defined in Equation 16:

doi: 10.13053/CyS-29-1-5500



(a) (b)

Fig. 2. a) Original image and its mask b)

$$F1 - Score = 2 * \frac{precision*recall}{precision+recall}. \tag{16}$$

Lastly, *IoU* metroc is calculated using Equation 17:

$$IoU = \frac{TP}{TP+FP+FN}. \tag{17}$$

Equations 13 to 17 and their average are used to measure the performance of the method proposed in this paper.

3.3 Dataset

The dataset named "Tomato Detection" consists of 850 images with shots of tomato plantations grown inside greenhouses, which are accessible from the link `https://www.kaggle.com/datasets/andrewmvd/tomato-detection`.

In the experimental section 300 images were used, representing a sample of 33% of the dataset. The selected images are in PNG file format, and have $500x400$ pixels in size resolution, the Figure 2 a) show an example of the selected images.

Labeling images. In order to measure the efficiency of the segmentation made by tany method, it is required to mask the 300 selected images to perform the optimization process. Image labeling was performed using the Computer Vision Annotation Tool (*CVAT*), available at the link `https://www.cvat.ai/`. Figure 2 b) show the result of labeling the pixels of the leaves in green and in red those of fruits.

3.4 Adaptation of the Greedy Algorithm

Greedy Algorithm (*GA*) are used to solve general optimization problems. This type of approach makes the optimal decision at each stage of the problem, with the goal of finding the global optimum at the end [2, 7, 17]. *GA* are often used in problems where the solution can be constructed sequentially, and each stage chooses values that maximize some objective function. Although the greedy approach does not guarantee the optimal solution, it is a simple and efficient way to solve many problems.

Figure 3 shows the general flow chart proposed to find the α_1 and α_2 parameters values that maximize the segmentation of leaves and fruits made by the *SCD*. In general terms, the algorithm determines in each iteration the optimal value of the results of the segmentation of the leaves and fruits in a given numerical range. The *GA* tests various values for the parameters α_1 and α_2, which are within a previously defined numerical range. At each iteration, their values increase by the specified Step size. The algorithm converges when the Step size is smaller than the stopping condition.

The *GA* starts by reading the five control parameters. The first one is the *Lower Limit Leaves* (*LLL*) which is the initial value where the search will start. *Upper Limit Leaves* (*ULL*) is the final value where the search is terminated. The third parameter is the *Step* size which is the feed rate that will have the α values in the range set by *LLL* and *ULL*. The second last control value is the *Stop* condition that ends the execution of the algorithm. The fifth parameter *Reduction* is the percentage of the size adjustment of those that are processed to reduce the execution time. The initial value of the *Lower Limit Fruit* (*LLF*) and the *Upper Limit Fruit* (*ULF*) are assigned the same values as the range of the leaves.

The *Read_dataset* function is responsible for reading the numerical data of the selected images. The function *Reduction_size_imag* has the purpose of resizing the images to be segmented according to the value of the *Reduction* parameter. The *SCD* described in the Section 3.1 is encoded in the functions *Segmenting_leaves* and *Segmenting_fruits*, which receive as parameter

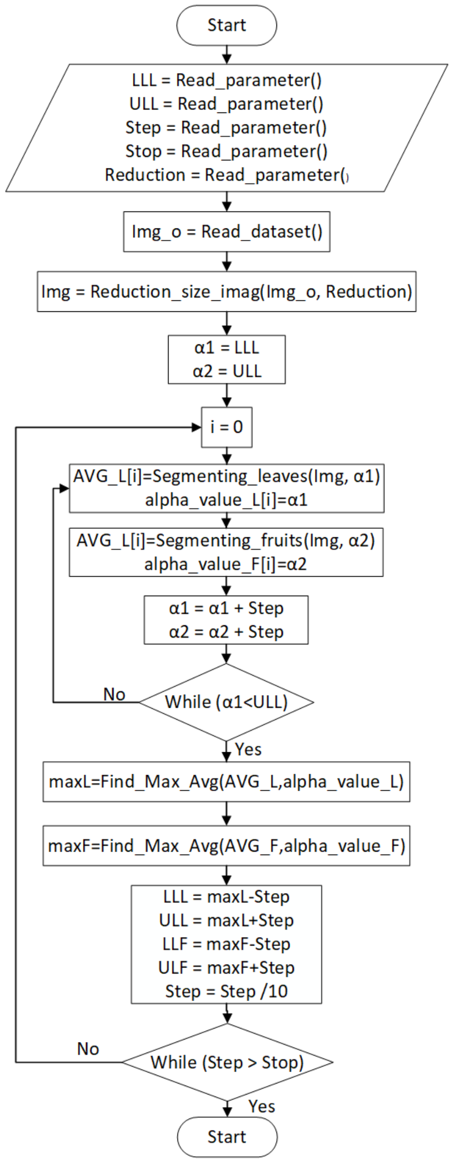


Fig. 3. Flow chart of the *GA* for optimization of the values of α_1 and α_2

the values of α_1 or α_2 and return the average of the segmentation of the leaves and fruits, which are stored in *AVG_L* and *AVG_F*, as well as the different values of α_1 and α_2 in the search range of an iteration.

The function *Find_Max_Avg receives* as parameters the segmentation averages and the different values of α, its objective is to determine which

Table 1. Initial values of the parameters of the *GA*

Parameter	Value
Lower limit leaves (LLL)	1
Upper limit leaves (ULL)	10
Step	1
Stop	0.0001
Reduction	0.3

value of α generates the highest average in the search range.

With the respective optimal values of α_1 and α_2 in a search range, new search values and step size values are assigned for the next interaction. The lower bounds are calculated by subtracting the optimal value of α_1 or α_2 from the Step size while the values of the upper bounds are determined by adding the optimal value of α_1 or α_2 to the *Step* size. The value of the *Step* Size for the next iteration is calculated by dividing the current value by 10.

4 Experimentation and Results

The initial values for the experimentation of the *GA* are shown in Table 1, this consisted in the execution of the algorithm described in Figure 3. Tables 2 to 7 show the averages of the segmentation performed with the values within the search range. The yellow color highlights the best result of the corresponding search space.

The first iteration generated the highest averages in the segmentation of leaves and fruits with the same value for α_1 and α_2 being 3, with an average of 0.86180 and 0.78676 respectively, see Table 2.

Table 3 contains the values of the second interaction, the search range was 2 to 4 with a *Step* size of 0.1. The value for the α_1 was 3.40000 with an avg. of 0.86371, and in the case of the fruits the value was 2.70000 for α_2 with an avg. of 0.79236.

The third iteration is performed with a *Step* size of 0.01. The value of $\alpha_1 = 3.38000$ which will generate a avg. of 0.86374 and the value of $\alpha_2 = 2.63000$ which will generate a avg. of 0.79243 which are the highest segmentation averages respectively. The data with the different values within the ranges are shown in Table 4.

Table 2. Leaves, fruits segmentation avgs of iteration 1

Leaves		Fruits	
α_1 **value**	**Avg.**	α_2 **value**	**Avg.**
Range 1 to 10		Range 1 to 10	
1.00	0.77776	1.00	0.72454
2.00	0.84028	2.00	0.77952
3.00	0.86180	3.00	0.78676
4.00	0.85903	4.00	0.73260
5.00	0.82375	5.00	0.63686
6.00	0.75497	6.00	0.53668
7.00	0.65773	7.00	0.45490
8.00	0.54717	8.00	0.38488
9.00	0.45178	9.00	0.33122
10.00	0.37025	10.00	0.28320

Table 3. Leaves, fruits segmentation avgs of iteration 2

Leaves		Fruits	
α_1 **value**	**Avg.**	α_2 **value**	**Avg.**
Range 2 to 3.9		Range 2 to 3.9	
2.00000	0.84028	2.00000	0.77952
2.10000	0.84373	2.10000	0.78272
2.20000	0.84695	2.20000	0.78543
2.30000	0.84966	2.30000	0.78760
2.40000	0.85213	2.40000	0.78951
2.50000	0.85434	2.50000	0.79103
2.60000	0.85634	2.60000	0.79204
2.70000	0.85812	2.70000	0.79236
2.80000	0.85959	2.80000	0.79151
2.90000	0.86083	2.90000	0.78971
3.00000	0.86180	3.00000	0.78676
3.10000	0.86259	3.10000	0.78429
3.20000	0.86318	3.20000	0.78127
3.30000	0.86356	3.30000	0.77727
3.40000	0.86371	3.40000	0.77274
3.50000	0.86362	3.50000	0.76684
3.60000	0.86332	3.60000	0.76084
3.70000	0.86262	3.70000	0.75324
3.80000	0.86172	3.80000	0.74597
3.90000	0.86050	3.90000	0.73875

Table 5 shows the data of the fourth iteration, the segmentation avgs. are of 0.86375 and 0.79246 for leaves and fruits respectively, these were generated with the values of $\alpha_1 = 3.38600$ and $\alpha_2 = 2.63300$. The *Step* size is 0.001.

The third iteration is performed with a *Step* size of 0.00001. The value of $\alpha_1 = 3.38660$ which will generate a avg. of 0.86375 and the value of $\alpha_2 = 2.63390$ which will generate a avg. of 0.79246 which are the highest segmentation averages respectively.

The data with the different values within the ranges are shown in Table 6.

Table 4. Leaves, fruits segmentation avgs of iteration 3

Leaves		Fruits	
α_1 **value**	**Avg.**	α_2 **value**	**Avg.**
Range 3.3 to 3.5		Range 2.6 to 2.8	
3.30000	0.86356	2.60000	0.79204
3.31000	0.86361	2.61000	0.79230
3.32000	0.86363	2.62000	0.79237
3.33000	0.86363	2.63000	0.79243
3.34000	0.86365	2.64000	0.79243
3.35000	0.86369	2.65000	0.79233
3.36000	0.86371	2.66000	0.79231
3.37000	0.86373	2.67000	0.79233
3.38000	0.86374	2.68000	0.79237
3.39000	0.86372	2.69000	0.79241
3.40000	0.86371	2.70000	0.79236
3.41000	0.86372	2.71000	0.79232
3.42000	0.86372	2.72000	0.79221
3.43000	0.86369	2.73000	0.79224
3.44000	0.86366	2.74000	0.79212
3.45000	0.86369	2.75000	0.79212
3.46000	0.86367	2.76000	0.79200
3.47000	0.86367	2.77000	0.79188
3.48000	0.86366	2.78000	0.79175
3.49000	0.86363	2.79000	0.79167
3.50000	0.86362	2.80000	0.79151

Table 5. Leaves, fruits segmentation avgs of iteration 4

Leaves		Fruits	
α_1 **value**	**Avg.**	α_2 **value**	**Avg.**
Range 3.37 to 3.39		Range 2.62 to 2.64	
3.37000	0.86373	2.62000	0.79237
3.37100	0.86373	2.62100	0.79237
3.37200	0.86373	2.62200	0.79239
3.37300	0.86372	2.62300	0.79239
3.37400	0.86373	2.62400	0.79241
3.37500	0.86372	2.62500	0.79240
3.37600	0.86373	2.62600	0.79240
3.37700	0.86373	2.62700	0.79239
3.37800	0.86374	2.62800	0.79238
3.37900	0.86374	2.62900	0.79241
3.38000	0.86374	2.63000	0.79243
3.38100	0.86373	2.63100	0.79241
3.38200	0.86373	2.63200	0.79245
3.38300	0.86375	2.63300	0.79245
3.38400	0.86374	2.63400	0.79244
3.38500	0.86374	2.63500	0.79244
3.38600	0.86375	2.63600	0.79240
3.38700	0.86374	2.63700	0.79239
3.38800	0.86373	2.63800	0.79240
3.38900	0.86373	2.63900	0.79243
3.39000	0.86372	2.64000	0.79243

In the sixth iteration the *Step* size is 0.00001 to explore the ranges of 3.38650 to 3.38669 for leaves and 2.63270 to 2.63289 for fruits. The highest avg. for leaves is 0.86375 with a $\alpha_1 = 3.38654$, and for fruits the avg. is 0.79246 with a $\alpha_2 = 2.63275$, observe the Table 7.

ISSN 2007-9737

Table 6. Leaves, fruits segmentation avgs of iteration 5

Leaves		**Fruits**	
α_1 **value**	**Avg.**	α_2 **value**	**Avg.**
Range 3.385 to 3.3869		Range 2.632 to 2.6339	
3.38500	0.86374	2.63200	0.79245
3.38510	0.86374	2.63210	0.79245
3.38520	0.86374	2.63220	0.79246
3.38530	0.86374	2.63230	0.79245
3.38540	0.86374	2.63240	0.79244
3.38550	0.86374	2.63250	0.79245
3.38560	0.86374	2.63260	0.79244
3.38570	0.86374	2.63270	0.79245
3.38580	0.86374	2.63280	0.79246
3.38590	0.86374	2.63290	0.79245
3.38600	0.86375	2.63300	0.79245
3.38610	0.86375	2.63310	0.79244
3.38620	0.86375	2.63320	0.79243
3.38630	0.86375	2.63330	0.79243
3.38640	0.86374	2.63340	0.79244
3.38650	0.86375	2.63350	0.79243
3.38660	0.86375	2.63360	0.79243
3.38670	0.86375	2.63370	0.79244
3.38680	0.86374	2.63380	0.79243
3.38690	0.86374	2.63390	0.79244

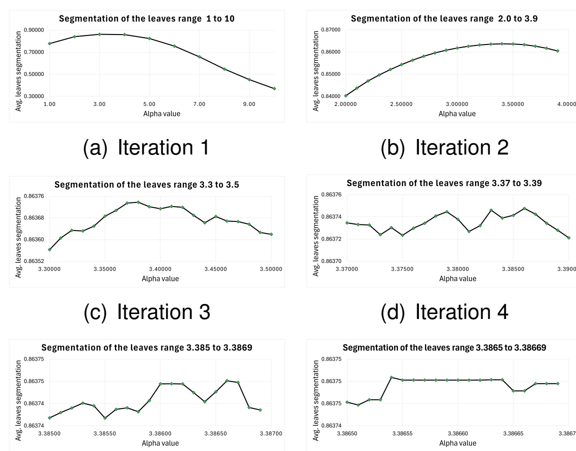
Table 7. Leaves, fruits segmentation avgs of iteration 6

Leaves		**Fruits**	
α_1 **value**	**Avg.**	α_2 **value**	**Avg.**
Range 3.3865 to 3.38669		Range 2.6327 to 2.63289	
3.38650	0.86375	2.63270	0.79245
3.38651	0.86375	2.63271	0.79245
3.38652	0.86375	2.63272	0.79245
3.38653	0.86375	2.63273	0.79245
3.38654	0.86375	2.63274	0.79245
3.38655	0.86375	2.63275	0.79246
3.38656	0.86375	2.63276	0.79246
3.38657	0.86375	2.63277	0.79246
3.38658	0.86375	2.63278	0.79246
3.38659	0.86375	2.63279	0.79246
3.38660	0.86375	2.63280	0.79246
3.38661	0.86375	2.63281	0.79246
3.38662	0.86375	2.63282	0.79246
3.38663	0.86375	2.63283	0.79246
3.38664	0.86375	2.63284	0.79246
3.38665	0.86375	2.63285	0.79246
3.38666	0.86375	2.63286	0.79245
3.38667	0.86375	2.63287	0.79245
3.38668	0.86375	2.63288	0.79245
3.38669	0.86375	2.63289	0.79245

The *GA* reached the convergence point specified in Table 1 using the *Stop* parameter in 6 iterations, averaging 25 minutes per iteration, for a total convergence time of 150 minutes.

Figures 4 and 5 show the plots of the leaves and fruits segmentation avgs. respectively of the images of the experimental set during *GA* iterations. In the first two iterations in both leaves and fruits, the segmentation averages have a behavior that is coupled to a polynomial of degree 2 or 3. On the other hand, the graphs from the third to the sixth iteration show a series of local maxima and minima. It should be noted that the proposed *GA* finds the global maximum in each search space as the basis for the next iteration.

Figure 6 shows an example of the segmentation performed on leaves and fruits with the highlighted values from the Table 7.



(a) Iteration 1
(b) Iteration 2
(c) Iteration 3
(d) Iteration 4
(e) Iteration 5
(f) Iteration 6

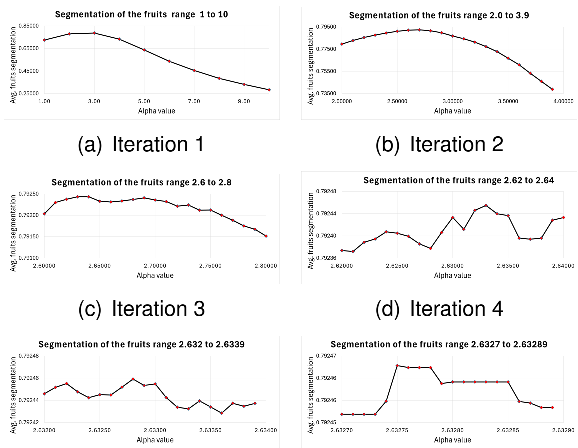
Fig. 4. Search for the optimal value of α_1

5 Comparison of *GA* Results with *CNN*

In the current state of the art regarding segmentation algorithms, those based on deep learning exhibit a large number of implementations. The results obtained with the *CNN PSPNet* [23] model when segmenting the leaves and fruits of tomato plants are presented and compared. The CNN model was created using the TensorFlow libraries. For the training process, two sets of images were created, the Training one with 180 items and the Validation one with 80, where it was necessary to perform the same labeling as described in the corresponding section for both sets of images. The

doi: 10.13053/CyS-29-1-5500

ISSN 2007-9737



(a) Iteration 1 (b) Iteration 2

(c) Iteration 3 (d) Iteration 4

(e) Iteration 5 (f) Iteration 6

Fig. 5. Search for the optimal value of α_2

(a) (b) (c)

Fig. 6. Segmentation performed with *SCD* and the optimized values of the Table 7

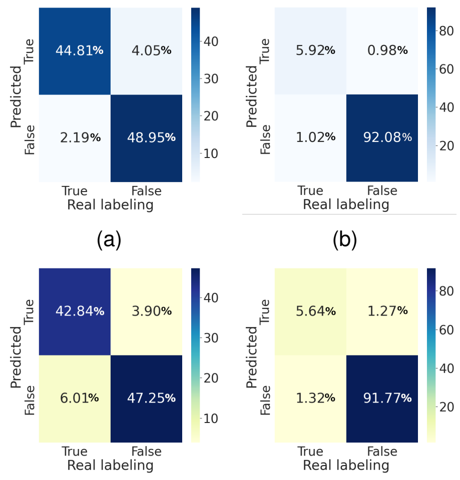
learning process was carried out for 50 epochs with a learning rate of 0.05, Adam optimizer is utilized and batch size is set to 16. The time required for the coating process is about 6 hrs.

Table 8 shows the values obtained by segmenting the leaves and fruits of the 300 images with the optimized values with the *GA* of $\alpha_1 = 3.38654$, $\alpha_2 = 2.63275$ and without reduction, as well as the values of the five metrics and the average of the semantic segmentation performed with the *PSPNet*, marking in bold the superiority of the *GA* averages over those of the *PSPNet*.

In the metrics used to measure the efficiency of the *GA* and the *CNN PSPNet model*, the performance of the former is superior to that of the latter. The segmentation of the leaves and fruits of tomato plants in the images with the values of

Table 8. Comparison of results between the *GA* and *PSPNet*

Algoritmth	Class	Acuracy	Precision	Recall	F1-Score	IoU	Avg.
GA	Leaves	0.92271	0.91845	0.94032	0.91555	0.86027	0.90946
PspNet		0.91468	0.85097	0.93192	0.87591	0.80195	0.87504
GA	Fruits	0.98025	0.82441	0.84449	0.81143	0.71328	0.83477
PspNet		0.95951	0.79217	0.82841	0.80134	0.68568	0.81338



(a) (b)

(c) (d)

Fig. 7. Leaves and fruits segmentation heat maps. a) *GA* leaves, b) *GA* fruits c) *CNN* leaves, and d) *CNN* fruits

$\alpha_1 = 3.2$ and $\alpha_2 = 2.6$ reported in [6] with the 300 test images generated an average of 0.90308 for leaves and 0.82309 for fruits which are lower than those generated by the *GA*. Figure 7 shows the heat plots of leaves and fruits segmentation performed by *GA* and *CNN PSPNet model*.

A statistical analysis of the 300 averages of the test images was performed, selecting 14 of them randomly. Table 9 shows the results of F-test for comparing variances of two samples, from which it is concluded that the variances of leaves and fruit segmentation are equal for any of the two segmentation methods.

After concluding that both segmentation methods have the same variance, we proceeded to perform the Student's t-test for equal variances to conclude if there are differences in the means of the methods compared, Table 10 shows the Student's t-test data.

Table 9. F-Test analysis of variance for two samples

F-test for two-sample variance				
	Leaves		Fruits	
	GA	CNN	GA	CNN
Mean	0.93831	0.85376	0.86445	0.83482
Variance	0.00110	0.01495	0.01160	0.00637
Observations	13	13	13	13
Degrees of freedom	12	12	12	12
F	0.07380		1.81944	
Critical value for F (one-tailed)	0.37221		2.68663	

Table 10. Student's t-test for two samples assuming equal variances

Student T-test for equal variances				
	Leaves		Fruits	
	GA	CNN	GA	CNN
Mean	0.93831	0.85376	0.86445	0.83482
Variance	0.00110	0.01495	0.01160	0.00637
Observations	13		13	
Pooled variance	0.00803		0.008991	
Degrees of freedom	24		24	
T-statistic	2.40619		0.79662	
Critical t-value (two-tailed)	2.06389		2.06389	

From the information in Table 10 it can be concluded that there are differences in the means calculated when segmenting the leaves of tomato plants by *GA* and *CNN*, but there is no difference in the means of fruit segmentation.

6 Conclusion

The segmentation results derived from the optimization process of the parameter values α_1 and α_2 performed by means of the *GA* proposed in this research generates an increase in the average of the five metrics in each of the six iterations that the algorithm was executed until reaching the convergence value, which demonstrates that the algorithm increases the segmentation averages in each iteration. This can be seen in the data highlighted in Tables 2 to 7. An interesting aspect to mention is shown in Figures 4 and 5 in items c) to f) several local minima are observed, which are not ignored when choosing the optimal segmentation value, which demonstrates the robustness of the proposed algorithm.

A negative aspect of the developed method is the slow optimization process, which is affected by issues such as the number of images in the data set and their dimensions, because when processing a larger number of images or being a set of images of larger dimensions, the time needed to converge will be longer. Similarly, the quality of the labeling performed to calculate performance metrics directly affects the optimization result, so poor labeling directly affects performance measurement.

In the direct comparison of results the *GA* delivers better results than the *PSPNet model*, requiring less than half the time and without the need to use specialized hardware.

7 Future Work

The future line of work is to reduce the total time of the optimization process by means of an interpolation process that reduces the execution time of the algorithm and guarantees average similar to those generated with the *GA*. In the same way, experiments are conducted by processing a percentage of the total images to reduce the time per iteration.

Acknowledgements

— Consejo Nacional de Humanidades Ciencia y Tecnologías (CONAHCYT).

— Centro de Investigaciones en Óptica A.C.

References

1. **Chen, L., Lin, L., Cai, G., Sun, Y., Huang, T., Wang, K., Deng, J. (2014).** Identification of nitrogen, phosphorus, and potassium deficiencies in rice based on static scanning technology and hierarchical identification method. PLoS ONE, Vol. 9, No. 11, pp. 1–17. DOI: `10.1371/journal.pone.0113200`.

2. **García, A. (2024).** Greedy algorithms: a review and open problems. Vol. 00, pp. 1–20.

3. **Gebbers, R., Adamchuk, V. I. (2010).** Precision agriculture and food security. Science, Vol. 327, No. 5967, pp. 828–831. DOI: `10.1126/science.1183899`.

4. **Ghamisi, P., Couceiro, M. S., Martins, F. M., Benediktsson, J. A. (2014).** Multilevel image segmentation based on fractional-order darwinian particle swarm optimization. IEEE Transactions on Geoscience and Remote Sensing, Vol. 52, No. 5, pp. 2382–2394. DOI: `10.1109/TGRS.2013.2260552`.

5. **Guerra Ibarra, J. P., Cuevas, F. J. (2023).** Segmentation of Leaves and Fruits of Tomato Plants by Color Dominance. AgriEngineering, pp. 1846–1864. DOI: `https://doi.org/10.3390/agriengineering5040113`.

6. **Guerra Ibarra, J. P., Cuevas de la Rosa, F. (2024).** Optimization of Color Dominance Factor by Greedy Algorithm for Leaves and Fruit Segmentation of Tomato Plants. Pattern Recognition, Springer Nature Switzerland, Cham, pp. 200–209.

7. **Ji, L., Yan, H. (2002).** Attractable snakes based on the greedy algorithm for contour extraction. Pattern Recognition, Vol. 35, No. 4, pp. 791–806. DOI: `10.1016/S0031-3203(01)00085-1`.

8. **Lee, K. J., Lee, B. W. (2013).** Estimation of rice growth and nitrogen nutrition status using color digital camera image analysis. European Journal of Agronomy, Vol. 48, pp. 57–65. DOI: `10.1016/j.eja.2013.02.011`.

9. **Liang, Y. C., Chen, A. H. L., Chyu, C. C. (2006).** Application of a hybrid ant colony optimization for the multilevel thresholding in image processing. Lecture Notes in Computer Science, Vol. 4233 LNCS, pp. 1183–1192. DOI: `10.1007/11893257{\_}129`.

10. **MENGEL, K., KIRKBY, E. (1978).** Plant nutrition principles, Vol. 15B,A.

11. **Nakane, T., Bold, N., Sun, H., Lu, X., Akashi, T., Zhang, C. (2020).** Application of evolutionary and swarm optimization in computer vision: a literature survey. IPSJ Transactions on Computer Vision and Applications, Vol. 12, No. 1. DOI: `10.1186/s41074-020-00065-9`.

12. **Nanehkaran, Y. A., Zhang, D., Chen, J., Tian, Y., Al-Nabhan, N. (2020).** Recognition of plant leaf diseases based on computer vision. Journal of Ambient Intelligence and Humanized Computing, Vol. 1, No. 0123456789, pp. 1–18. DOI: `10.1007/s12652-020-02505-x`.

13. **Patil, B. M., Amarapur, B. (2018).** Segmentation of leaf images using greedy algorithm. 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS 2017, pp. 2137–2141. DOI: `10.1109/ICECDS.2017.8389830`.

14. **Puranik, P., Bajaj, P., Abraham, A., Palsodkar, P., Deshmukh, A. (2009).** Human perception-based color image segmentation using comprehensive learning particle swarm optimization. 2009 2nd International Conference on Eme Trends in Engg and Tech, ICETET 2009, pp. 630–635. DOI: `10.1109/ICETET.2009.116`.

15. **Romualdo, L. M., Luz, P. H., Devechio, F. F., Marin, M. A., Zúñiga, A. M., Bruno, O. M., Herling, V. R. (2014).** Use of artificial vision techniques for diagnostic of nitrogen nutritional status in maize plants. Computers and Electronics in Agriculture, Vol. 104, pp. 63–70. DOI: `10.1016/j.compag.2014.03.009`.

16. **Singh, V., Misra, A. K. (2017).** Detection of plant leaf diseases using image segmentation and soft computing techniques. Information Processing in Agriculture, Vol. 4, No. 1, pp. 41–49. DOI: `10.1016/j.inpa.2016.10.005`.

17. **Subr, K., Majumder, A., Irani, S. (2005).** Greedy algorithm for local contrast enhancement of images. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 3617 LNCS, pp. 171–179. DOI: `10.1007/11553595{\_}21`.

18. **Taheri, M., Lim, N., Lederer, J. (2016).** Balancing Statistical and Computational Precision and Applications to Penalized Linear Regression with Group Sparsity. Department of Computer Sciences and Department of Biostatistics and Medical Informatics., pp. 233–240.

19. **Tao, W. B., Tian, J. W., Liu, J. (2003).** Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm. Pattern Recognition Letters, Vol. 24, No. 16, pp. 3069–3078. DOI: `10.1016/S0167-8655(03)00166-1`.

20. **Vergés-Llahí, J., Climent, J., Sanfeliu, A. (2000).** Colour image segmentation solving hard-constraints on graph partitioning greedy algorithms. Proceedings-International Conference on Pattern Recognition, Vol. 15, No. 3, pp. 625–628. DOI: `10.1109/ICPR.2000.903623`.

21. **Vongkulbhisal, J., De La Torre, F., Costeira, J. P. (2019).** Discriminative Optimization: Theory and Applications to Computer Vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 41, No. 4, pp. 829–843. DOI: `10.1109/TPAMI.2018.2826536`.

22. **Yu, J., Xu, J., Chen, Y., Li, W., Wang, Q., Yoo, B., Han, J.-J. (2021).** Learning Generalized Intersection Over Union for Dense Pixelwise Prediction. Proceedings of the 38th International Conference on Machine Learning, PMLR, Vol. 139, pp. 12198–12207.

23. **Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J. (2017).** Pyramid scene parsing network. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Vol. 2017-Janua, pp. 6230–6239. DOI: `10.1109/CVPR.2017.660`.

Article received on 06/06/2024; accepted on 16/10/2024.
Corresponding author is Juan Pablo Guerra Ibarra.