



---

*Article*

## **Neural network training in SCILAB for classifying mango (*Mangifera indica*) according to maturity level using the RGB color model**

**Eduardo Castillo-Castaneda<sup>1,\*</sup>**

<sup>1</sup> Department of Mechatronics, Instituto Politecnico Nacional, CICATA Unidad Querétaro, CP 76090, México

\* **Correspondence:** [ecastilloca@ipn.mx](mailto:ecastilloca@ipn.mx); Tel: +52-442-229-0804

Academic Editor: Med Amine Laribi

**Abstract:** Industries that use fruits as raw materials must, at some point in the process, classify them to discard the unsuitable ones and thus ensure the quality of the final product. To produce mango nectar, it is necessary to ensure that the mango is mature enough to start the extraction of the nectar; however, sorting thousands of mangoes may require many people, who can easily lose attention and reduce the accuracy of the result. Such kind of decision can be supported by current Artificial Intelligence techniques. The theoretical details of the processing are presented, as well as the programming code of the neural network using SCILAB as a computer language; the code includes the color extraction from mango images. SCILAB programming is simple, efficient and does not require computers with large processing capacity. The classification was validated with 30 images (TIF format) of Manila variety mango; the mangoes were placed on a blue background to easily separate the background from the object of interest. Four and six mangoes were used to train the neural network. This application of neural networks is part of an undergraduate course on artificial intelligence, which shows the potential of these techniques for solving real and concrete problems.

**Keywords:** Artificial intelligence, neural network, mangoes classification, RGB color model

---

### **1. Introduction**

For many years, artificial intelligence has been presented as a complex field, understandable only by expert programmers and mathematicians. However, such thinking runs contrary to the basic principle of artificial intelligence: letting computers solve complex problems or repetitive tasks of human beings. The main purpose of the application presented in this paper is to demonstrate that a neural network can be easily developed (theoretically and practically) in a public access programming language such as SCILAB to solve a concrete problem. This application is part of an artificial intelligence course taught to second year students of undergraduate Engineering Systems programs.

Industries that use fruits as raw materials must, at some point in the process, classify them to discard the unsuitable ones and thus ensure the quality of the final product. To extract mango (*Mangifera indica*) nectar, it is necessary to classify it into three categories: mature, medium, or green. The reason is that only the mature mango (with a suitable level of sugar) can be used for nectar extraction; the green mango is stored for more days and the medium mango is refrigerated and returns to the sorting process two days later [1]. In many nectar factories, the classification is done manually, that is, the mangoes are unloaded from the truck and placed on a conveyor belt where a group of employees separate the mangoes manually, using touch and considering their appearance (particularly the color), see Fig. 1.



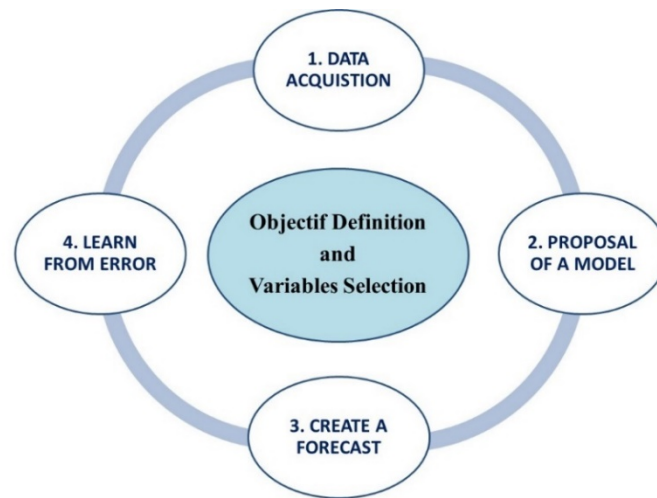
**Figure 1.** Manual mango classification [2].

Manual classification can be too subjective and therefore unreliable, employees can make mistakes during selection; it is a simple task but requires high levels of concentration over long periods of time. Also, a poor mango classification will reduce the quality of the extracted nectar. Another important situation for the employees, is that repetitive muscle movements can cause tendon inflammation, then injury and disability. Fortunately, automation and artificial intelligence (AI) can provide a solution to this problem and assist people in these types of repetitive tasks.

AI is the ability of computer programs to generate conclusions from the observation of some variables [3], it has been developed under that name since the 70's, its growth is linked to the evolution of computers, their processing and storage capacity. Today, the high performance of computers is used to assist people to perform complex, dangerous, and repetitive tasks, in environments not suitable for human beings. In the 70's, the progress of AI was limited by the low processing and storage capacity of computers, but today, AI applications are multiplying rapidly, and have become accessible to professionals in any field [4]. The computer performs the "observation of variables" through sensors, which have also become accessible due to their low cost. For example, digital cameras, which are photosensitive sensors that convert light into electrical signals, have achieved very high performance at very low cost.

A computer program, which implements an AI application, must include the stages indicated in Fig. 2. To begin, we must define an objective by answering the question: What result do I want to obtain from the program? In addition, we must determine the best set of variables to achieve that goal. Considering the mango classification, the objective is "identify to which of the three categories, green,

medium, or mature, a mango belongs". The variables to achieve this objective can be sugar level, smell, texture, or color; in this work we chose the color since it allows to classify mango without contact, assuming a high correlation with the level of maturity [1].



**Figure 2.** Stages for the implementation of an AI computer program.

The Fig. 3 shows that there is an evident correlation between color and maturity level of a mango since mature ones turn yellow.



**Figure 3.** Levels of mango maturity.

From the objective and the variables, stage 1 is to acquire data (numerical values) of the selected variables, that is, the mango color. In stage 2, a model must be proposed to process the data, that is, a model that correlates the mango color with one of the three categories. In stage 3, the program must generate a forecast based on the data entered and the proposed model, that is, determine the category to which each mango belongs. At the end, in stage 4, the program must compare the result with the true mango category and learn from the error, that is, a person or a system evaluates the result obtained by the program and decides whether the classification is correct or not; the error obtained is combined with new data (to adjust the model) and the process starts again. The most complex stage is to propose a correct model; There are different ways to generate a model in AI, the best known are [5]: Expert Systems, Decision Trees, and Artificial Neural Networks (ANN). In this paper we present a solution based on ANN.

The proposed solution is based on the correlation that exists between the color of a fruit and its level of maturity, which has been widely used in other research works on fruit maturity [6,7]. In this way, it is possible to identify maturity level of mangoes from their color, without having to measure sugar levels or texture, which are the commonly used off-line methods. Furthermore, to make AI techniques accessible to a greater number of people, we use SCILAB as computer platform to extract color and to program the neural network as well as its training phase.

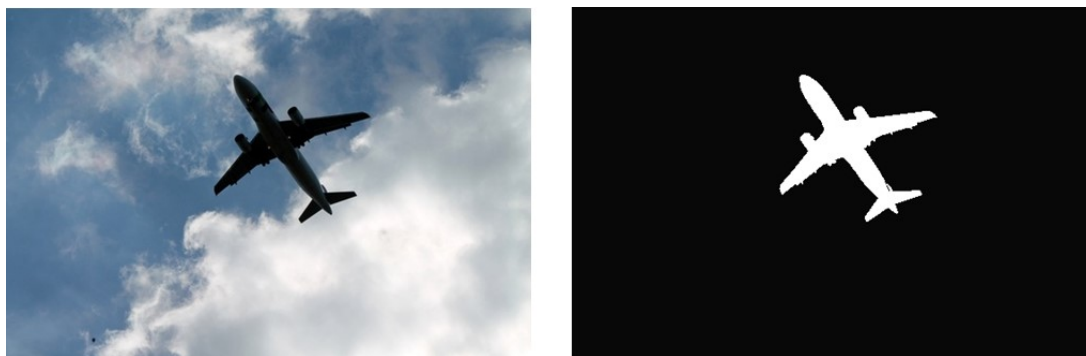
## 2. Mango color extraction

### 2.1. Image processing

For humans, identifying a green mango from a yellow (ripe) one is relatively simple since we were trained during childhood. However, humans can make mistakes when colors are similar, for example, to identify different shades of green. In contrast, a computer program can identify colors with extreme precision and repeatability, thousands of times. To achieve this goal, the computer only requires an image of the object acquired through a conventional camera and a computer program for processing.

The type of lighting selected to illuminate the object during image acquisition should not generate glare or shadows since both situations modify the true color of the object [8].






Before extracting the color, the object of interest (mango) must be separated from the background, this process is known as segmentation [9]. This separation is simplified by choosing a uniform background providing enough contrast between background and object. As an example, Fig. 4 shows an image segmentation to separate the airplane from the background (clouds and sky). The interest object appears on white (1 or true) and the rest on black (0 or false).








**Figure 4.** Segmentation applied to separate the airplane from the sky (including clouds).

### 2.2. RGB color model

Color cameras have an array of sensors, each sensor transduces the intensity (amplitude) of three wavelengths that correspond to the colors Red, Green and Blue, known as the RGB (Red, Green, Blue) model. The computer handles each element of the sensor under a digital form (a pixel), which is characterized by the numerical values corresponding to the intensity of each RGB component. Each component can take 256 different values, in the range of [0, 255]. Fig. 5 shows some examples of colors and their corresponding RGB values.

R	G	B	Color
202	189	006	
255	192	0	
255	0	0	
0	255	0	
255	255	255	

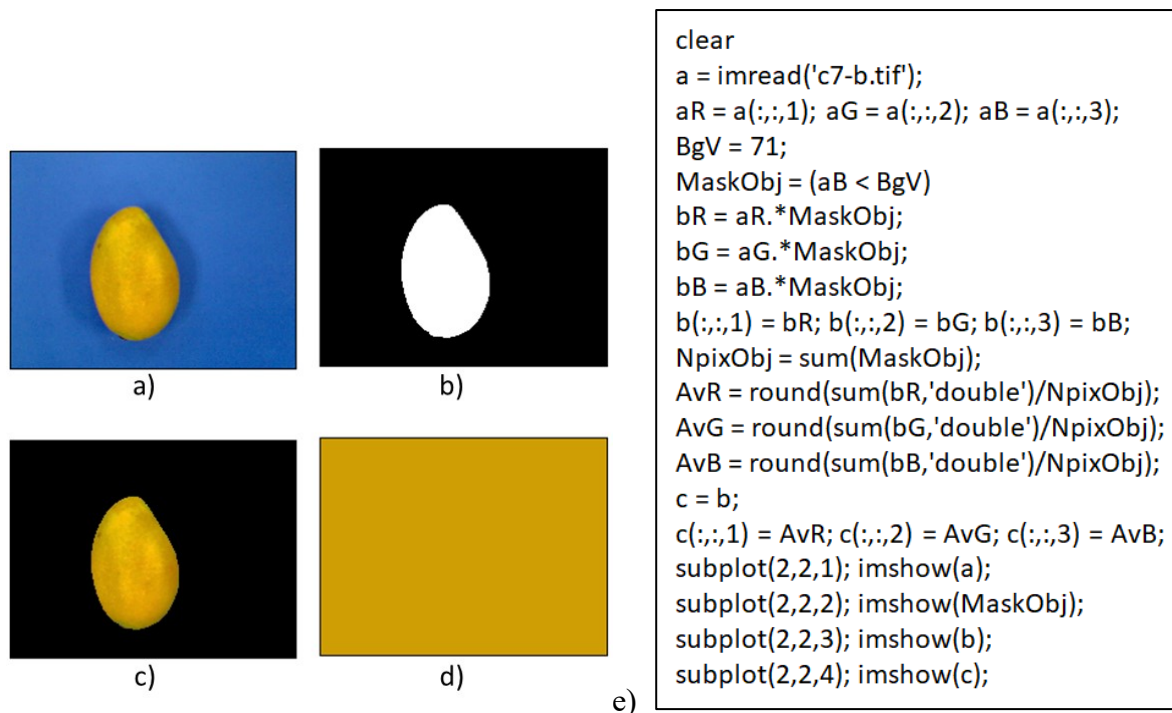
R	G	B	Color
0	0	255	
255	255	255	
255	255	0	
152	72	6	
141	179	226	

**Figure 5.** Some colors and the corresponding RGB values.

In this way, a computer is capable of handling  $256 \times 256 \times 256$  different shades of color, that is, 16,777,216 different colors in a fast, accurate and repeatable way. In a computer program, a color image is represented by three  $N \times M$  matrices (one for each RGB component), where  $N$  and  $M$  are the number of pixels, horizontal and vertical, respectively.

### 2.3. SCILAB program

To extract the color from a mango, the SCILAB program starts by reading an image and separating it from the background. The image 'c7-b.tif' is read and assigned to variable **a**, the original image is shown in Fig. 6a.



**Figure 6.** a), b), c), d) Color extraction images, e) SCILAB code.

The RGB components are then separated and assigned to the variables **aR**, **aG** and **aB**. The blue background corresponds to values greater than 71 for the blue component (**aB**), this value is assigned to the variable **BgV**. Values less than **BgV** for **aB** component correspond to mango color. With this value, a mask is created (logical variable **MaskObj**) that will be true (1) when the corresponding pixel is less than **BgV** or false (0) when it is greater than or equal to **BgV**, the image corresponding to the



mask is shown in the Fig. 6b. This mask is applied to each RGB component to generate a new image, named **b**, where the background will have been removed (RGB components equal to zero), Fig. 6c shows the object separated from the background.

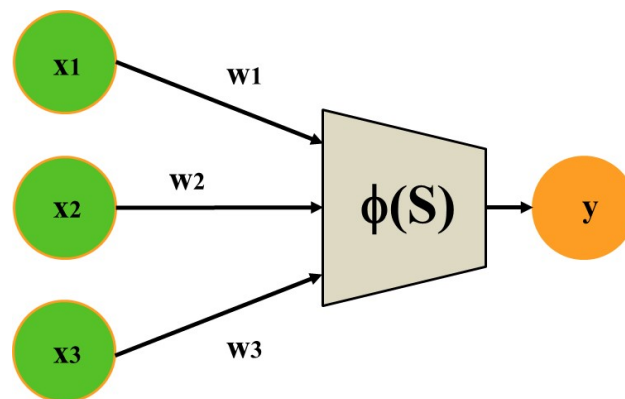
To extract the color of the object, the averages of the three RGB components (variables **AvR**, **AvG**, **AvB**) are calculated for all the pixels belonging to the object (variable **NpixObj**), Fig. 6d shows a color rectangle equivalent to the average values of the mango. The Fig. 6e presents the SCILAB code to extract the average color from an image of a mango; the corresponding computation time is 490 ms, under Windows 10, using a desk computer running at 1.6 GHz.

### 3. Artificial Neural Networks

#### 3.1. Fundamentals

An ANN is a set of artificial neurons based on the mathematical model of the perceptron (artificial neuron) proposed in 1958 by F. Rosenblatt [10]. As Fig. 7 shows, the perceptron is composed of:

- $n$  input variables  $x_1, x_2, \dots, x_n$ ,
- 1 neuron with an associated activation function named  $\phi(S)$ ,
- $n$  synaptic weights  $w_1, w_2, \dots, w_n$  (one for each input variable), and
- 1 output variable  $y$ .



**Figure 7.** Model of an artificial neuron, named perceptron, with 3 input variables.

In this model, the output is given by:

$$y = \phi(S)$$

where  $S$  is the weighted sum of the input variables:

$$S = \sum_{i=1}^n w_i x_i$$

The most common form of the activation function is the sigmoid function whose output values are in the range  $[0, 1]$ :

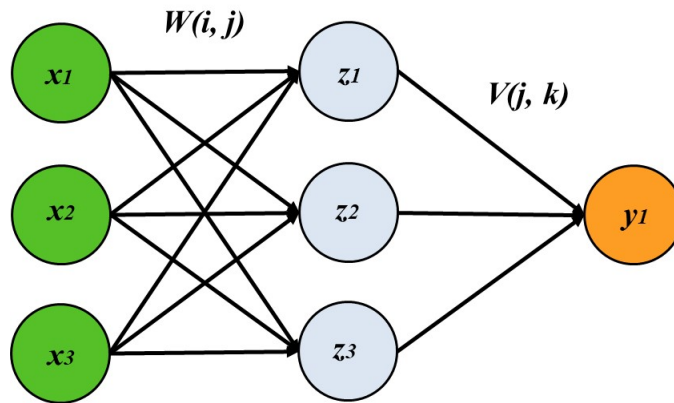
$$\phi(S) = \frac{1}{1 + e^{-S}}$$

In general, the input values must be normalized, within  $[0, 1]$  or  $[-1, 1]$ , to reduce inconsistencies

caused by variables with different units. Then, the output variable will also have values within  $[0, 1]$ . An ANN is composed of layers, for many applications only three layers are required: an input layer (with  $n$  input variables), a hidden layer and an output layer (with  $m$  output variables).

### 3.2. An ANN model for mango classification

To classify the mango according to color, 3 input variables (each RGB component) and 1 output variable are required. The output will have a value between 0 and 1, where 1 will mean green mango and 0 will mean mature mango. The Fig. 8 shows the three-layer ANN used to solve the mango classification problem using a hidden layer of 3 neurons.



**Figure 8.** An ANN with 3 layers.

In this case, the weights are represented by two matrices  $W(i, j)$  and  $V(j, k)$  where  $i$  is input variables  $x_i$  index,  $j$  is the neuron index  $z_j$  for the hidden layer, and  $k$  is the output variables  $y_k$  index. For the ANN in Fig. 8, the vectors  $X(i)$ ,  $Y(k)$  and the matrices  $W(i, j)$  and  $V(j, k)$  have the following form an order:

$$X_{1 \times 3} = [x_1 \quad x_2 \quad x_3],$$

$$Y_{1 \times 1} = y_1,$$

$$W_{3 \times 3} = \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \\ w_{3,1} & w_{3,2} & w_{3,3} \end{bmatrix} \quad \text{and} \quad V_{3 \times 1} = \begin{bmatrix} v_{1,1} \\ v_{2,1} \\ v_{3,1} \end{bmatrix}$$

In this way, the values of the neurons in the hidden layer are easily calculated by multiplying the vector of input variables  $X$  and the weight matrix  $W$  as follows:

$$Z_{1 \times 3} = [z_1 \quad z_2 \quad z_3] = \phi(X \cdot W)$$

It should be noted that the weighted sum  $S$  is calculated implicitly with the matrix product  $X \cdot W$ . In the same way, the vector of output variables can be calculated as follows:

$$Y_{1 \times 1} = y_1 = \phi(Z \cdot V)$$

For example, let us consider the mango of figure 6a, where the average RGB components are  $(AvR, AvG, AvB) = (207, 158, 5)$ . To normalize the values, they must be divided by 255, which is the greatest value they can take. These normalized values form the vector of input variables of the ANN:

$$X_{1 \times 3} = [207/255 \quad 158/255 \quad 5/255] = [0.8118 \quad 0.6196 \quad 0.0196]$$

Furthermore, for a first approximation, the values of the matrices  $W$  and  $V$  can be randomly assigned (in section 3.4 an iterative algorithm is detailed to estimate them accurately); the SCILAB code to calculate the output variable is shown in Fig. 9; the corresponding computation time is 55 ms.

```

clear
function [act] = phi(S)
    act = ones(S)./(1+%e^-S)
endfunction

X = [0.8118, 0.6196, 0.0196];
W = [0.1,0.2,0.3;0.2,0.3,0.4;0.3,0.4,0.5];
V = [0.5;0.6;0.7];
Z = phi(X*W);
Y = phi(Z*V);

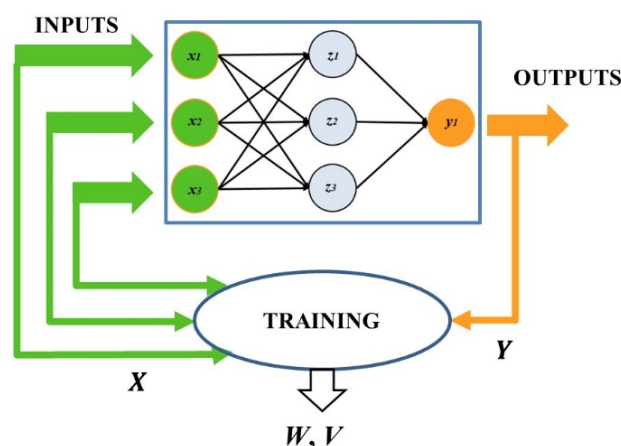
```

**Figure 9.** SCILAB code to calculate de ANN output.

The result obtained from the previous code is  $y_1 = 0.7436$ , which corresponds to a green mango (value close to 1); however, the result is not yet reliable since the matrices  $W$  and  $V$  must be adjusted to consider the error estimation of  $y_1$ .

### 3.3. ANN training

As shown in the diagram of Fig. 2, the ANN must learn from error, this is known as training, which consists of calculating the best values for the elements of the matrices  $W$  and  $V$ . The ANN is trained from the error considering a finite number of training cases, where the output  $Y$  and input variables  $X$  are both known, see Fig. 10.



**Figure 10.** Training of an ANN from known values of  $X$  and  $Y$  vectors.

The most common algorithm for ANN training is known as backpropagation [11] which consists of calculating the error of the  $Y$  output as follows:



$$E = \frac{1}{2}(Y - Yc)^2$$

where  $Yc$  is a vector of known values of the output variable. Considering the perceptron model of Fig. 7, the previous equation can be rewritten by:

$$E(W) = \frac{1}{2}\phi((Xc \cdot W) - Yc)^2$$

where  $Xc$  is a vector of known values of the input variable, corresponding to  $Yc$  values. Thus, the problem is reduced to find the value of  $W$  that minimizes the function  $E(W)$ . The gradient descent algorithm is an iterative method to find a minimum of a function by taking steps proportional to the negative of the function's gradient at the current point. In this case, it can be shown [12] that the gradient is given by:

$$\frac{dE}{dW} = (Y - Yc)\phi'Xc = ADW$$

Where  $\phi'$  is the first derivative of the activation function and is given by:

$$\phi' = \frac{d\left(\frac{1}{1+e^{-s}}\right)}{ds} = \phi(1-\phi)$$

The previous equation can be separated into two parts, the layer error  $EY$  and the weight matrix adjustment value  $ADW$ :

$$EY = (Y - Yc)\phi' \quad \text{and} \quad ADW = EY \cdot Xc$$

Then, for each iteration of the algorithm, the current value of matrix  $W_k$  is updated by:

$$W_{k+1} = W_k - \eta ADW,$$

where  $\eta$  is named the learning rate, its purpose is to accelerate the search. The current speed of computers allows an ANN to be trained in a matter of seconds, even when thousands of iterations are required to reach error values close to zero.

When the ANN also has a hidden layer, the best value must also be calculated for the matrix of weights  $V$ . In this case, the errors of the  $EY$  and  $EZ$  layers are:

$$EY = (Y - Yc)\phi' \quad \text{and} \quad EZ = (EY \cdot V)\phi'$$

Furthermore, the adjustments for each matrix are given by:

$$ADW = Xc \cdot EZ \quad \text{and} \quad ADV = Z \cdot EY$$

The SCILAB code corresponding to the ANN training is presented in Fig. 11. For instance, the values of the input and output variables are not yet related to the mango classification, they are shown only as an example. The initial values for the matrices  $W$  and  $V$  are random and vary in the range  $[-1, 1]$ . The method performs 20,000 iterations; at the end, the output variable  $Y$  should be approximately equal to  $Yc$ ; the corresponding computation time (20,000 iterations) is 895 ms.

```

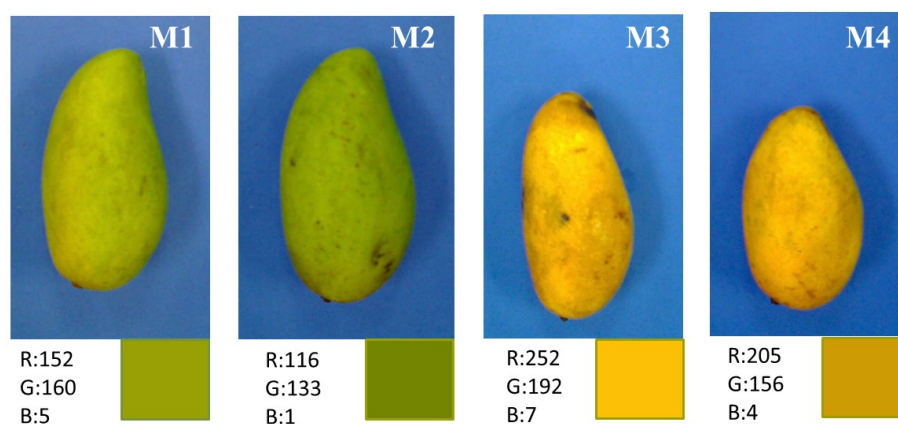
clear
function [act]=phi(S), act=ones(S)./(1+exp(-S)), endfunction
function [Dact]=dphi(S), Dact=phi(S).*(1-phi(S)), endfunction
Xc = [0,0,1;1,1,1;1,0,1;0,1,1];
Yc = [0,1,1,0]';
W = 2*rand(3,4)-1;
V = 2*rand(4,1)-1;
eta=1;
for i=1:20000
    Sz = Xc*W; Z = phi(Sz);
    Sy = Z*V; Y = phi(Sy);
    EY = (Y-Yc).*dphi(Sy);
    EZ = (EY*V').*dphi(Sz);
    ADV = Z'*EY;
    ADW = Xc'*EZ;
    W = W-eta*ADW;
    V = V-eta*ADV;
end

```

**Figure 11.** SCILAB program for ANN training.

#### 4. Mango classification





To validate the SCILAB programs, a database with 30 images of Manila-type mangoes was used. The images are in TIF format and have a size of 1016 x 870 pixels. Using the SCILAB code of Fig. 6e, the averages of the three RGB components for each mango were extracted and stored in an EXCEL file. From these 30 mangoes, 2 mangoes were selected that classified unambiguously (visually) as Green (M1, M2), 2 other mangoes as Mature (M3, M4), see Fig. 12. These mangoes were used to train the ANN.



**Figure 12.** Mangoes selected for ANN training.

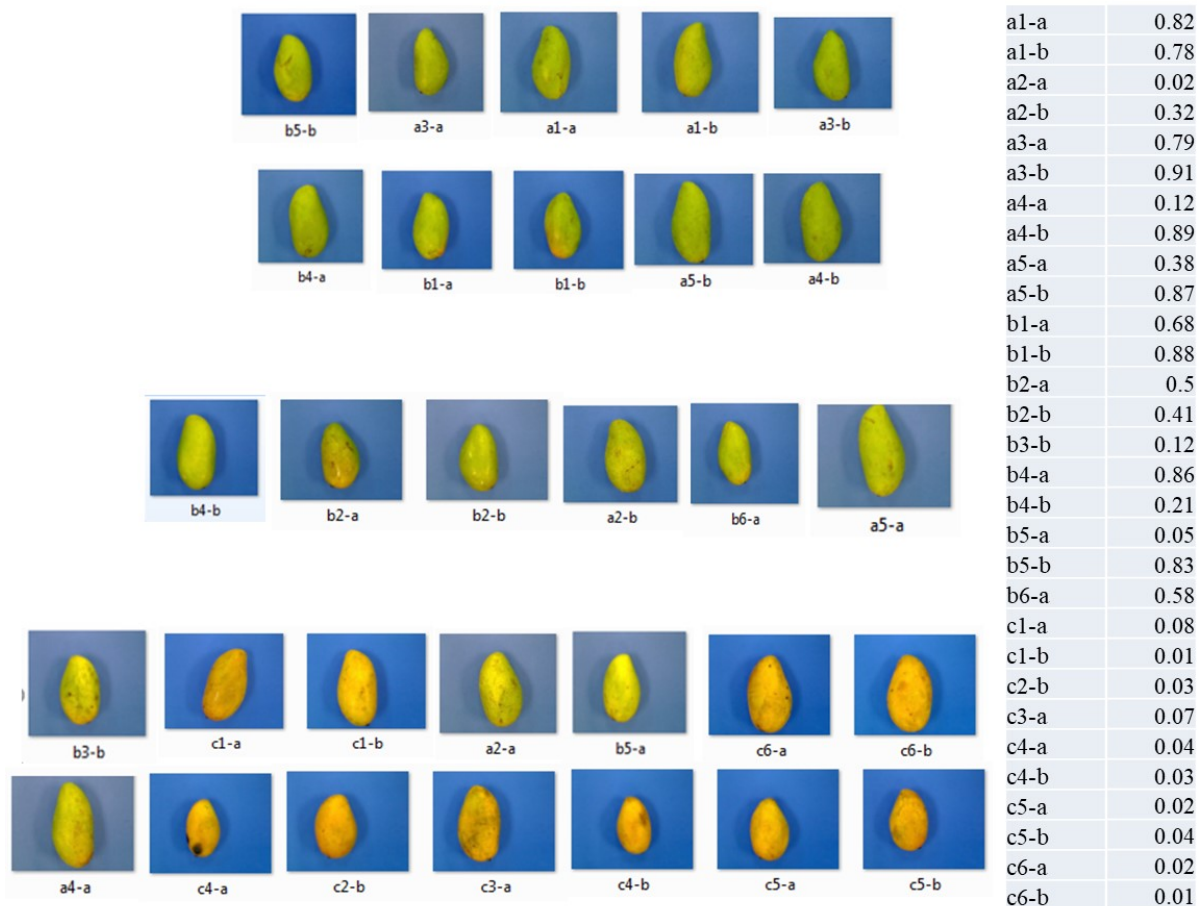
Using the SCILAB code in Fig. 11, the input variables  $Xc$  and the output variables  $Yc$  corresponding to the mangoes M1, M2, M3 and M4 were set as shown in Fig. 13. The green mangoes correspond to an output value equal 1, while mature ones have a value equal to 0. The input values

were divided by 255 for normalization.

M1:		input : $\{x_1 : 152, x_2 : 160, x_3 : 5\}$ → output: $\{y_1 = 1\}$
M2:		input : $\{x_1 : 116, x_2 : 133, x_3 : 1\}$ → output: $\{y_1 = 1\}$
M3:		input : $\{x_1 : 252, x_2 : 192, x_3 : 7\}$ → output: $\{y_1 = 0\}$
M4:		input : $\{x_1 : 205, x_2 : 156, x_3 : 4\}$ → output: $\{y_1 = 0\}$

**Figure 13.** Input and output variables for ANN training.

As expected, the result of the training are the matrices  $W$  and  $V$ . To verify the accuracy of the result, the output vector  $Y$  should be, ideally, equal to  $Yc = [1 \ 1 \ 0 \ 0]$ . For this case, after 20,000 iterations, the vector value is  $Y = [0.996778 \ 0.9963279 \ 0.0037164 \ 0.0028814]$ , that guarantees very good accuracy.



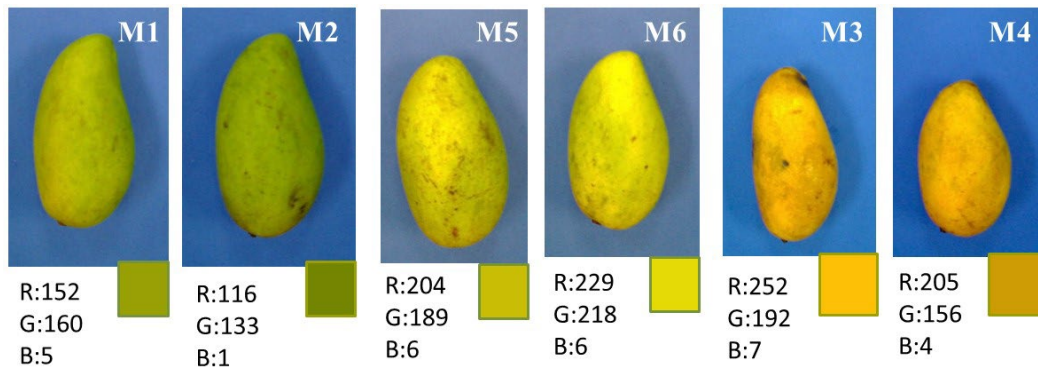
**Figure 14.** Preliminary result of the classification of 30 mangoes.

To classify the 26 remaining mangoes, the program in Fig. 9 is used, one mango at a time, where the vector of input values  $X$  will be the averages of the 3 RGB components. In this case, the expected result is the output variable  $Y$  that will indicate which of the three types of the mango belongs to. Obviously, a value of  $Y$  close to 0.5 will mean a medium mango. Fig. 14 shows the classification of

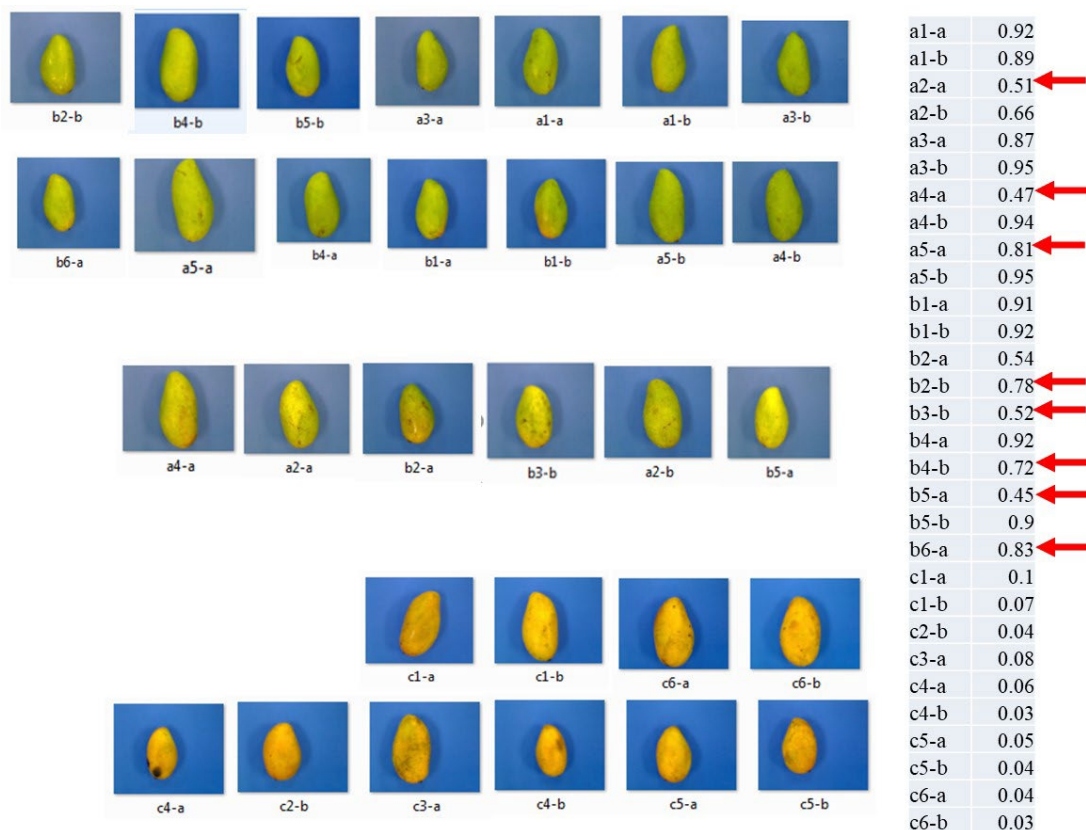
the 30 mangoes, the green mangoes are located at the top, the medium at the middle and the matures at the bottom. The  $Y$  values for each mango are shown in the table on the right.

Some mangoes classified as mature look better as medium, and some mediums appear green. To improve the result, two mangoes were included for training. Two mangoes, that classify unambiguously as medium, were chosen; thus, 6 mangoes were used for training, as shown in Fig. 15.

An improved result is shown in Fig. 16. Some mangoes were moved from mature to medium (b3-b, a2-a, a4-a, and b5-a). Red arrows indicate mangoes that changed their classification.



**Figure 15.** Green, medium, and mature mangoes for ANN training.



**Figure 16.** Result with improved training using 6 mangoes.

## 5. Conclusions

This work presents both, the theoretical details and the programming code using SCILAB, to

classify mangoes according with their maturity level using RGC color model. The code includes the color extraction from mango images. SCILAB programming is simple, efficient and does not require computers with large processing capacity. The code extracts the color of the mangoes from images in TIF format; and classifies, through an ANN, each object in the image into mature, medium, or green mango. It seems that the number of mangoes used for ANN training is fundamental to increase the classification performance, but it is also important that the selected mangoes must be representative enough of the three categories. 30 Manila-type mangoes were classified, 6 of them were used for the training; within the ANN, the level of maturity is in the range [0, 1], where 0 corresponds to mature mango, 0.5 to medium mango and 1 to green mango. The method implemented to train the ANN was that of the descent of the gradient, which converges after 20,000 iterations for this size of ANN (3 layers); the computation time for training is 895 ms, which is very low, even when the code runs on conventional desktop computers. Once the ANN has been trained, the SCILAB code takes 55 ms to run the code that classify one mango. Evidently, the presented code can be used to classify other types of fruits based on their color, a 3-layer ANN, with a hidden layer of 3 neurons is sufficient.

In addition, from the didactic point of view, the students who followed the Artificial Intelligence course using SCILAB as a programming language, were able to apply the same code for other useful solutions with artificial intelligence and satellite images, such as: identifying objects in the sea (boats, whales, islands, ...); identify objects in the sky (planes, birds, clouds, ...); and detecting houses in the desert or in the middle of large rainforests.

## References

1. Jha, S.N., Chopra, S. and Kingsly, A.R.P., Modeling of color values for nondestructive evaluation of maturity of mango. *Journal of Food Engineering*, 2007, 78: 22–26.
2. In: <https://www.magmar.com.mx/>, available 4th July 2021.
3. Russell, S.J. and Norvig, P., *Artificial Intelligence: A modern Approach*. 1995, New Jersey, USA: Prentice Hall.
4. Bughin, J., Seong, J., Manyika, J., Chui, M. and Joshi, R., *Notes from AI frontier: Modeling the impact of AI on the world economy*. Discussion paper, 2018, McKinsey Global Institute.
5. Lucci, S. and Kopec, D., *Artificial Intelligence in the 21st Century*. 2015, VA, USA: Mercury Learning and Information.
6. Prabha, D.S. and Kumar, J.S., Assessment of banana fruit maturity by image processing technique. *J Food Sci Technol*, 2015, 52(3): 1216-1327.
7. Torres, R., Montes, E.J., Pérez, O.A. and Andrade, R.D., Relación del Color y del Estado de Madurez, con las Propiedades Fisicoquímica de Frutas Tropicales. *Información Tecnológica*, 2013, 24(3): 51-56.
8. Russ, J.C., *The Image Processing Handbook*. 1999, Prentice Hall.
9. González, R.C. and Woods, R.E., *Digital Image Processing*. 2002, Prentice Hall.
10. Rosenblatt, F., The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 1958, 65(6).
11. Rojas, R., The Backpropagation Algorithm. *Neural Networks*, 1996, pp. 149-182. Springer-Verlag.
12. Mandic, D.P., A generalized normalized gradient descent algorithm. *IEEE Signal Processing Letters*, 2004, 11(2): 115-118.