TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics, Mechanics, and Engineering
Vol. 64, Issue I, March, 2021

AUTOMATED TOOL FOR OLIVE COLOR RECOGNITION IN SORTING SYSTEM DEVELOPMENT

Roxana Mihaela BABANATIS-MERCE, Theoharis BABANATSAS, Dan Ovidiu GLAVAN, Andreea Ioana GLAVAN, Adina BUCEVSCHI, Marcello Calvete GASPAR

Abstract: *In this paper we propose an olive color recognition software, and present its workflow and development. Our software is based on detection and classification of RGB color codes (Red, Green, and Blue) [1]. Based on the color we can determine the ripeness of the presented olives, which is an important step in the processing of olives in table olive or olive oil. The software has been developed in such a way that all spectrum colors can be recognized. Thus, giving many possibilities to use the software in the future, such as determining the level of ripeness in fruit and vegetables, the only condition being the change of the respective fruit or vegetable's color in various stages of ripeness [2]. The software was focused on the olive sorting process; thus, the software is finetuned to recognize two colors with the corresponding shades.*

Key words: *color, sorting, software, spectrum, RGB, olive.*

1. INTRODUCTION

Nowadays, more and more emphasis is placed on the quality of products, on products that are not chemically processed and come from sustainable, natural, or traditional sources.

Thus, the sorting of fruits and vegetables after they have been harvested must be done as efficiently as possible. Given the current appearance of the SARS-CoV-2 virus, on top of sorting optimization, new solutions are sought in order to reduce the direct human intervention on consumer products as much as possible in order to avoid product infestation.

New sorting solutions must be reliable and accessible to manufacturers [3, 4]. Thus, in an attempt to support olive oil and consumer oil companies which operate sustainably using traditional methods, we propose various color sorting solutions that, in addition to being highly efficient, are also financially accessible.

In an attempt to optimize sorting systems, we propose sorting the items based on their colors. In our current research, we try to sort the olives (between the unripe ones (green) and the ripe ones (black)) [5, 6]. In the future, we will expand our research to encompass other fruits and vegetables that show changes in color during their maturation process [7].

2. MATERIALS

To create the proposed color sorting tool, we use a Raspberry PI camcorder version 2 and a programmable Raspberry minicomputer (Figure 1), which are connected to a laptop. The software itself is developed using the Python and C++ programming languages, which are compatible with the Raspberry minicomputer [8]. We also use a selection of green and black olives as samples for color recognition.

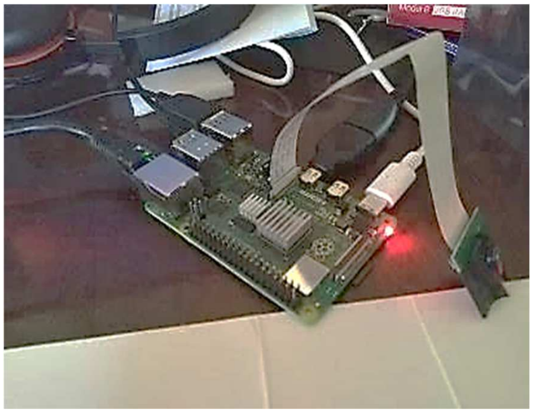

Fig. 1: Raspberry PI computer with the camcorder.

3. METHODOLOGY

The development of the proposed software tool was based on the needed components in the sorting process [9, 10], such as detecting the olives on the conveyor belt, photographing and determining the RGB code in the picture, and recognizing the color. This workflow is illustrated in Figure 2.

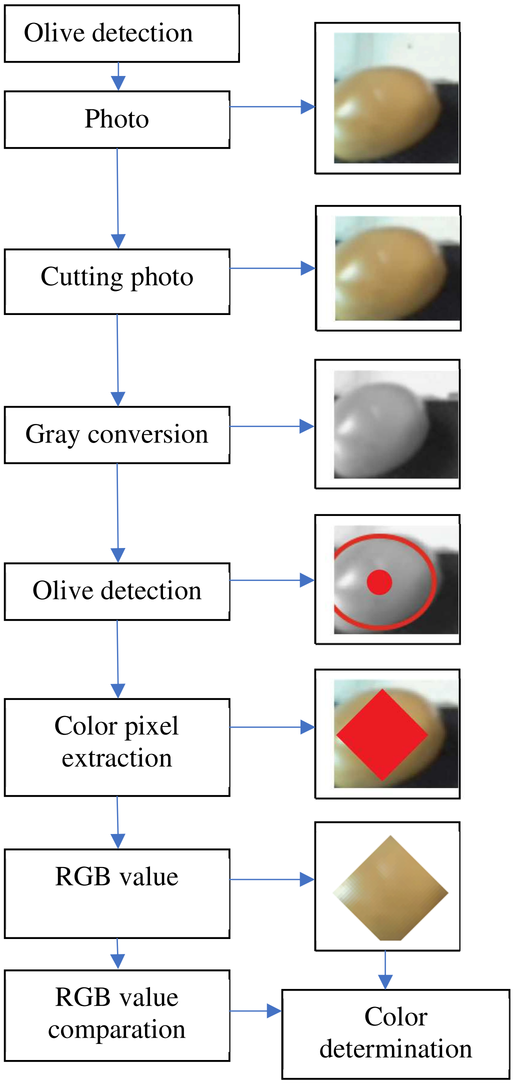

Fig. 2: Logical diagram showcasing the main sequence workflow.

The color recognition software was developed using the Python [11] programming language. This software was chosen for the compatibility with the minicomputer used (Raspberry) [8, 12].

The software was developed based on the logical workflow scheme presented above. Below, we explain representative sequences from the software, in terms of color recognition and determination of RGB code [13, 14].

Thus, the key aspects are:

- The function that analyzes the image

  def analyse_image;
- Reading the image

  img = cv2.imread;
- Cutting out the unnecessary parts of the image

  img = img[15:60, 0:64];
- Converting the image to grayscale

  Img = cv2.cvtColor (img, cv2. COLOR_BGR2GRAY);
- Framing the olive within a circular shape, the area from which the color will be extracted;

  circles = cv2.HoughCircles(image=img);
- Determining the radius of the circular area of the olive in order to extract the image needed to determine the color,

  radius = int(circle[2] - 3)

  center = (int(circle[1]), int(circle[0]));

Using this method, a 3 mm radius value was set. This value represents the shape that frames the olive within a circle while excluding the other elements, such as background or conveyor belt, from the image.

This value can be adjusted by the user, depending on the size of the detected olive. The area of the image within the circle is extracted, and afterwards it is analyzed in order to determine the average RGB value of the pixels inside the circular area.

The sequence of steps within the software that determine the average RGB value of the extracted pixels is the following:

- Scroll through the pixels around the center

  for i in range(center[0] - radius, center[0] + radius + 1);
- Take the RGB value of the pixels and create a cumulative sum

  R, G, B = imgc[i, j] ;
- Replace the pixel with a red one

  imgc[i, j] = (0, 0, 255);
- Compute the average value of the collected pixels

  r = int(r/x)

  g = int(g/x)

```
b = int(b/x)
print("average red ", r)
print("average green ", g)
print("average blue ", b)
print("radius ", radius) ;
```

- Save the image locally

```
cv2.imwrite("/home/pi/olive/detected.png",
imgc) ;
```

Thus, the values obtained from the above software sequence will be compared with color values stored in the database, in order to determine ripeness.

An important feature of the software is the ability to extract the RGB value in real time and compare it to the known color values stored in the database. Thus, we define two distinct databases, one for the olives which are considered unripe, green, and another for the ripe olives, of a darker brown color.

The values are stored in the database according to two different methods, both manually and automatically.

The manual method involves entering RGB values by hand, according to the shades of green and black that serve as our recognition targets. Table 1 indicates the black RGB codes and Table 2 the green RGB codes. The software has the ability to recognize any color, but we focus on our target values, specific for the olive color recognition task.

Table 1

Black RGB codes.

Color	Value code (R, G, B)
■	0,0,0
■	104,104,104
■	126,126,126
■	170,170,170
■	194,194,194

Table 2

Green RGB codes.

Color	Value code (R, G, B)
🟩	124,252,0
🟩	127,255,0
🟩	50,205,50
🟩	0.255.0
🟩	34,139,34
🟩	0,128,0
🟩	0,100,0
🟩	173,255,47
🟩	154,205,50
🟩	0,255,127
🟩	0,250,154
🟩	144,238,144
🟩	152,251,152
🟩	143,188,143
🟩	60,179,113
🟩	32,178,170
🟩	46,139,87
🟩	128,128,0
🟩	85,107,47
🟩	107,142,35

The automatic method of storing values is based on automated machine learning methods. Using this method, the software has the option of selecting the learning color (green or black).

After this setting is selected, olives are passed in from of the tool's video camera and the software records the images and computes the color values. These values are store in the database of the respective selected color.

The program was tested for both methods of storing RGB values (manual and automated learning method) using a dataset containing a number of 100 black olives and 100 green olives. The test was repeated three times for each method. It should be mentioned that the tests were performed in a room where the only light source was a 100W lightbulb.

4. RESULTS AND DISCUSSION

Three tests were performed for each color for each of the two methods, with average values (T_{A1}, T_{A2} and, T_{A3}). We compute the total average test value (T_{Atv}) based on the experimental results. The results obtained from these tests are showcased in Table 3. These values indicate a large difference between the two methods.

Table 3

Comparative results.

Introducing the RGB values in the database	Accuracy of olive color detection [%]							
	Green olives				Black olives			
	Test							
	T_{A1}	T_{A2}	T_{A3}	T_{Atv}	T_{A1}	T_{A2}	T_{A3}	T_{Atv}
Manual	77	86	80	81	81	67	71	73
Learning	100	100	100	100	100	100	100	100

For the manual RGB value collection method, we notice reading errors which can be seen in Figure 3. These errors are likely due to the fact that often in a digital processing the color can differ from reality.

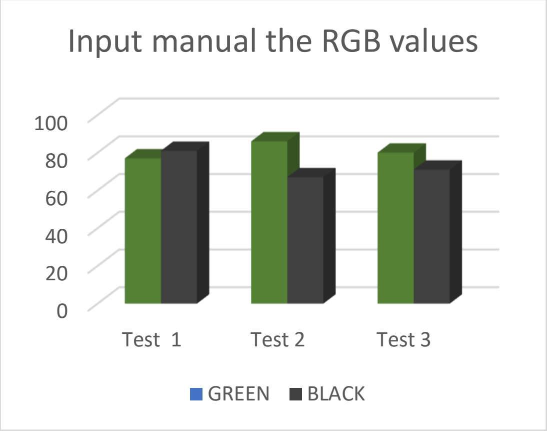

Fig. 3: Input manual method.

For the learning collection method, the performance increased significantly, as seen in Figure 4, with 100% accuracy. Given that the same video camera reads the color both for learning and for real-time RGB color code extraction.

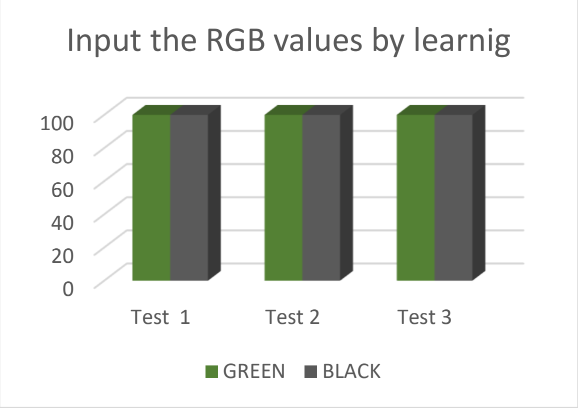

Fig. 4: Input by learning method.

5. REFERENCES

[1]. Fu, LS., Gao, FF., Wu, JZ., Li, R., Karkee, M ., Zhang, Q. *Application of consumer RGB-D cameras for fruit detection and localization in field: A critical review.* Computers and Electronics in Agriculture Volume 177, 2020.

[2]. Wittstruck, L., Kuhling, I., Trautz, D., Kohlbrecher, M., Jarmer, T. *UAV-Based RGB Imagery for Hokkaido Pumpkin (Cucurbita max.) Detection and Yield Estimation*. Jurnal Sensors, 2020

[3]. Mnerie, D., Tucu, D., Anghel, G. V. , Slavici, T.. *Study about integration capacity of systems for agro-food production*. Actual tasks on agricultural engineering, proceedings. Book Series: Actual Tasks on Agricultural Engineering, vol. 36, Zagreb, 617-622, 2008.

[4]. Tucu, D., Golimba, A.-G., Slavici, T.. *Fuzzy methods in renewable energy optimization investments*. Actual tasks

on agricultural engineering, proceedings. Book Series: Actual Tasks on Agricultural Engineering, vol 38, Zagreb, 455-462, 2010.

[5]. Therios, I.. *Olives*. Athens, Ed. Colums Design, 2014

[6]. Camposeo, Vivaldi, G. A. and Gattullo, C. E.. *Ripening indices and harvesting times of different olive cultivars for continuous harvest*. Sci Hortic (Amsterdam) vol. 151, 2013.

[7]. Serrano, M., Diaz-Mula, H.M. and Valera, D. *Antioxidant compounds in fruits and vegetables and changes during postharvest storage and processing*. Stewart Postharvest Review, v. 7, n. 1, 2011

[8]. . Nusairat, J.E. *Rust for the IoT. Building Internet of Things Apps with Rust and Raspberry Pi*. Apress, Berkeley, CA,pp 391-427, 2020.

[9]. Babanatis Merce, R.M., Babanatsas, T., Maris, S., Tucu, D., Ghergan, O. C.. *Study of an automatic olives sorting system. Actual tasks on agricultural engineering, proceedings*. Book Series: Actual Tasks on Agricultural Engineering, vol 46, Zagreb, 485-490, 2018.

[10]. Babanatis-Merce, M. R., Babanatsas, T., Maris, S. A., et al. *Experimental study on the automatic selection of olives*. 9th International Conference on Manufacturing Science and Education – MSE 2019 "Trends in New Industrial Revolution. vol. 290, 2019.

[11]. Hunt J. *A First Python Program. In: A Beginners Guide to Python 3 Programming*. Undergraduate Topics in Computer Science. Springe, pp 23-31, 2019.

[12]. https://www.raspberrypi.org/products/raspberry-pi-4-model-b/, accesing 24.11.2020.

[13]. Hyun Mun Kim, Woo-Shik Kim and Dae-Sung Cho. *A new color transform for RGB coding*. 2004 International Conference on Image Processing, pp. 107-110, vol. 1, Singapore, 2004.

[14]. https://www.rapidtables.com/web/color/RGB_Color.html, accessing 25.11.2020.

Realizarea unui program de recunoaștere a culorii maslinelor in dezvoltarea unui sistem de sortare

Rezumat: In aceasta lucrare prezentam principiul de dezvoltare a programlui nostru de recunoastrere a culorii maslinelor, program care se bazeaza pe codurile de culoare RGB (Red, Green and Blue). Pe baza culorii se poate determina daca maslinele suny coapte sau nu, o etapa important in procesarea maslinelor. Programul a fost dezvoltat in asa fel incat se recunoasca toate culori din spectrul de culori. Dand astfel numeroase posibilitati de utilizare a programului in viitor cum ar fi la determinarea nivelului de maturitate la fructi dari si la legume singura conditie fiind schimbarea culorii in diverse etape de maturitate. Programul a fost axat pe necesitatile de sortare a maslinelor din acest motiv fiind prevazut sa recunoasca doua culori.

Roxana Mihaela BABANATIS-MERCE, As. PhD Student Eng., “Aurel Vlaicu” University Arad, Faculty of Engineering, Department of Automation, Industrial engineering, Textile production and Transport, E-mail: elamerce@yahoo.com, Phone: 0257-283010.

Theoharis BABANATSAS, As. Dr. Eng., “Aurel Vlaicu” University Arad, Faculty of Engineering, Department of Automation, Industrial engineering, Textile production and Transport, E-mail: babanatsas@outlook.com, Phone: 0257-283010.

Dan Ovidiu GLAVAN, Conf. Dr. Eng., “Aurel Vlaicu” University Arad, Faculty of Engineering, Department of Automation, Industrial engineering, Textile production and Transport, E-mail: glavan@fortuna.com.ro, Phone: 0257-283010.

Andreea Ioana GLAVAN, Faculty of Science and Engineering Computing Science, University of Groningen, Netherlands, E-mail: andreeaglavan1@gmail.com

Adina BUCEVSCHI, Lect. Dr. Eng., “Aurel Vlaicu” University Arad, Faculty of Engineering, Department of Automation, Industrial engineering, Textile production and Transport, E-mail: glavan@fortuna.com.ro, Phone: 0257-283010.

Marcello Calvete GASPAR, Prof. dr. Eng., Escola Superioar de Tehnologia, Instituto Politehnico de Castelo Branco, Castelo Branco, Portugal.