Automatic detection of ripe fruits with Python – an interactive learning activity with cross-disciplinary approach

Ștefan-Alfred MARIS[1], Alexandra SUTA[1],
Cristian Mircea MUSCAI[2], Simina MARIS[3]

[1] Polytechnic University of Timisoara
1 Mihai Viteazu Av., Timisoara

[2] ”Politehnica” University of Bucharest
313 Independenței Av., Bucharest

[3] ”Ioan Slavici” University of Timisoara
144 A. P. Podeanu street, Timisoara

siminamaris[at]gmail.com

Abstract: *Automatic fruit harvesting addresses several issues, which can be considered as an independent computer science project, among them being the correct detection of a ripe fruit. The main focus of this article is to describe a project which derives from the detection of a ripe fruit and which is used within an interdisciplinary learning activity. The learning activity is intended for but not restricted to coding courses (like Coding with Patience) at high school and bachelor degree. We describe the original algorithms used for detecting the fruit, its level of ripeness according to its colour, and its dimensions. As the image is taken in RGB format, the colour recognition algorithm deals with the levels of red, green and blue of the image, but these levels are dependent on an optimal lighting of the scene, which is achieved using a hardware solution. The algorithms are calibrated to identify red and yellow fruits and with proper adjustments they can be extended to other types of ripe fruits (e.g., oranges, lemons, red apples, peaches, etc.). The detection is performed using a camera module attached to a Raspberry Pi 3B+ system and the image analysis is performed with Python.*

Keywords: Colour recognition, Interdisciplinary learning, Computer vision, Python.

1. Introduction

In the framework of the 21st century competencies required from the students, scientific skills include “the ability to use and handle technological tools and machines as well as scientific data to achieve a goal” (Partnership for 21st Century, 2009). In addition, more and more of the modern occupations require IT or ICT skills.

https://doi.org/10.58503/icvl-v17y202213

But while IT is a very attractive domain on the labour market, the national school curricula do not always meet the requirements of the IT companies. Moreover, most of the concepts taught in school are presented purely from a theoretical point of view, lacking a connection to real-life situations. Among the consequences of this state-of-fact is the lack of interest of the students toward studying the curricula. If students fail to get a satisfactory answer to the question “Where will I use this knowledge?”, they are more likely to abandon study.

Interdisciplinary learning has been proven to be effective in improving students’ motivation, creativity and efficacy (Kuo et al., 2019). From this point of view, using robotics as a tool for computer science curricula is the logical sequel, as shown by Arlegui et al. (2008).

During the first decades of the 21st century, e-learning systems have seen an important advance, correlating the curricula with new technologies, such ICT (Maris et al, 2018). The COVID-19 pandemics further emphasised the importance of virtual learning environments as means of continuing education – either as synchronous or asynchronous learning.

The objective of this paper is to describe a learning activity connected with a real-life situation, which can be used either during physical or virtual classes by students and educators. For now, this activity is used locally, in Timisoara, during the Coding with Patience sessions, but it will be available online in the near future. The Coding with Patience sessions (Patrut et al., 2020), being a non-formal learning environment, are appropriate for interdisciplinary and blended learning and provide the circumstances for acquiring the ICT skills required by the society of the 21st century.

One of the emerging fields of computer science is computer vision. A phase of computer vision which has many applications in real-life is object recognition. Object recognition can be performed either by identifying the shape of the object, either by identifying its colour, or by identifying both characteristics. According to Xu et al. (2020), object recognition requires a low- and mid-level processing degree, which can be used appropriately in mid-level learning activities. But a correct identification of a ripe fruit (Halstead et al., 2018; Wan et al., 2018) is an essential tool for harvesting robots. Hence, computer vision is appropriate not only as a modern learning subject on its own, but also as a resource for interdisciplinary and even non-formal learning.

2. Planning the learning activity

2.1. Prerequisites

In order to attend this activity, a student must have basic knowledge of Python – either Python 2 or Python 3 (Python, 2020). If Python is not installed on the students’ computer, we recommend a portable version that can be downloaded from portablepython.com (Portable Python, 2022) and which can be used on older operating systems (like Windows XP and Windows 7).

For the final stages of the learning activity, a Raspberry Pi 3B+ system (Raspberry Pi, 2019) endowed with a camera module (Raspberry Pi Camera, 2019) is needed.

2.2. Objectives and competencies to be developed

The objectives of the learning activity are:

- Apply formal programming concepts (lists / arrays, matrices, loops, conditional statements, functions, data types) in real-life situations;
- Manipulate images (open, store, crop, apply colour filters);
- Identify appropriate characteristics for ripe fruits: colour, shape;
- Use a camera module in order to perform image recognition.

These objectives correspond to hard skills and competencies which are expected to be developed during computer sciences and robotics classes.

In addition, by completing this learning activity, the students will acquire the following soft competencies: curiosity, patience, problem solving, time management, accountability.

2.3. Materials and equipment

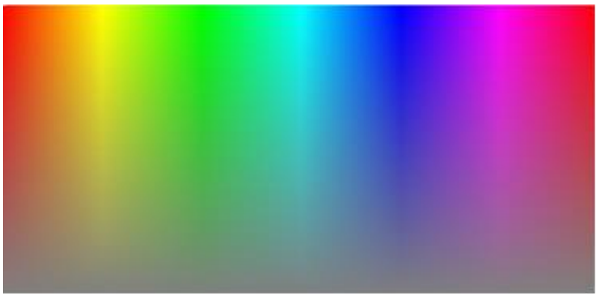

Figure 1. Colour spectrum (retrieved from standard Microsoft Office applications)

During the learning activity, the following materials will be used:

- Computers;
- Raspberry Pi endowed with a camera module;
- Python compiler (IDLE) and particularly the OpenCV (OpenCV- Python, 2022);
- Colour spectrum. A good colour spectrum is the one retrieved from the Microsoft Office applications (e.g., MS Paint, MS PowerPoint, as shown in Figure 1);
- Different images with ripe fruits (tomatoes, oranges, bell peppers).

2.4. Algorithms

In the following, we describe 3 of the algorithms used during the learning activity: an algorithm used to identify a ripe fruit on a plant (which is used to enhance critical thinking of the learners), a colour filter algorithm which is tested during the lessons for red and yellow fruits and which is actually implemented by the learners and an algorithm for identifying the geometrical limits of an object, which is implemented during the learning activity.

The algorithm for identifying ripe fruits, used by a harvesting robot, was presented by Maris et al. (2021) and it will be briefly described here:

- Start at the beginning of a lane between two rows of tomato plants
- Detect a plant
- Detect a ripe tomato:
    - Take a picture
    - Get picture parameters
    - Detect the colour of each pixel in the picture
    - Apply colour filter
    - Detect the parameters of the ripe tomato
- Grip the tomato
- Pluck the tomato from the plant
- Put the tomato in an adjoining basket
- Return to the plant and start over
- If no more ripe tomatoes are detected, find the following plant in the row and start over

The algorithm for identifying a colour (the colour filter algorithm) is:

- Open the picture
- Get image parameters (width, height)
- Initialize x=0, y=0
- While x<width and y<height
- Read the colour of the pixel (x,y) in (R,G,B) format
- If the colour condition is met
    - Colour the pixel in a certain colour (e.g., red, yellow, orange)
- Else
    - Colour the pixel in black
- Save the image

The algorithm for identifying the geometrical limits of a certain region of an image is briefly described here:

- Open the picture
- Get image parameters (width, height)
- While 0<x<width and 0<y<height
    - Determine the longest array L of pixels (horizontally) that has the colour C and its starting point x0

    - Determine the longest array H of pixels (vertically) that has the colour C and its starting point y0
- Return the geometrical limits of the object: (x0,y0) x (x0+L,y0+H)

3. Implementing the learning activity

3.1. Succession of lessons

The learning activity consists in a succession of 5 lessons, whose scenario will be presented briefly in the following. The total time needed to complete this learning activity is about 10 hours on average.

Lesson 1: How does a harvesting robot work. What does a harvesting robot need to know in order to harvest the appropriate ripe fruits.

The harvesting process is presented through movies and images of a real-life harvesting robot prototype. The students are encouraged to deduce the algorithm of identifying ripe fruits (as described above).

After figuring out the outlines of the algorithms used by the harvesting robot, the goals of the learning activity are stated, together with the succession of lessons, expected outcomes and assessment method.

Lesson 2: Images and colours. Storing an image. Cropping an image.

The students learn about the storage of an image as an array on the form [x,y,B,G,R], where x and y are the coordinates of a pixel (0<=x<=width, 0<=y<=height) and R (red), G (green), B (blue) define the colour of the pixel (0<=R,G,B<=255).

The students will create a simple program in Python which displays the width and height of an image and the colour of a certain pixel inside the image.

In order to crop an image, a new array is created, in which there will be stored only the values [x,y,R,G,B] for which x1<=x<=x2 and y1<=y<=y2 (possibly by replacing x with x-x1 and y with y-y1).

Lesson 3: Images and their colour. Applying a colour filter.

Different ripe fruits have different colours. Usually, the colour of a ripe fruit is red, orange or yellow. "Pure" reds are coded as [B,G,R]=[0,0,255], "pure" yellows are coded as [B,G,R]=[0,255,255] and "pure" oranges are coded as [B,G,R]=[0,165,255].

However, the colour of a ripe fruit is not always a "pure" colour. Hence, instead of "pure" colours, the colour filter should retain colour shades. Moreover, different varieties of the same species may have different colours for the ripe fruit (e.g., red tomatoes, orange tomatoes, yellow tomatoes, red peppers, orange peppers, yellow peppers, green peppers).

Hence, one cannot identify ripe fruits using a general colour-detection algorithm and should always specify the colours and shades that apply for a certain variety of fruits.

The colour condition for red shades we used during the learning activity is:

red>120 and red-green > 30 and red-blue > 30

By applying the red shade colour filter to the colour spectrum, the result presented in Figure 2 is achieved.

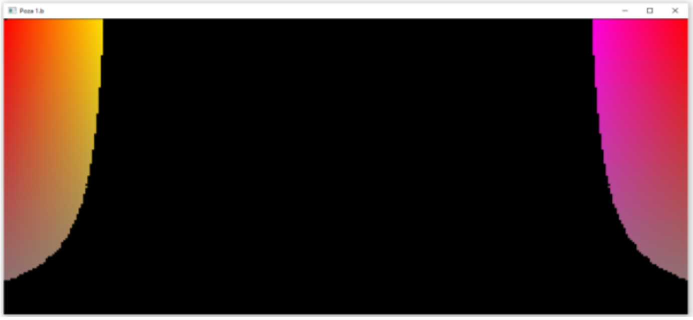

Figure 2. Red shades obtained from the red filter condition

For yellow shades, the colour condition used during the learning activity is:

(blue<110 and red > 130 and green > 130) or (blue<90 and red <200 and red-green>20)

By applying the yellow colour filter to the colour spectrum, the following results are achieved (Figure 3).

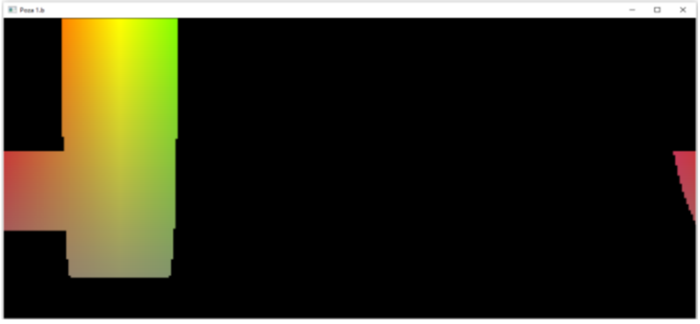

Figure 3. Yellow shades obtained from the yellow filter condition

Lesson 4: Identifying a ripe tomato and a ripe bell pepper. Characteristics of a ripe fruit.

For picking ripe red tomatoes, an efficient colour filter should retain different red shades. For picking ripe yellow peppers, the colour filter should retain different yellow shades.

By applying the red colour filter to various tomato fruits, results similar to Figure 4 are obtained. The quality of the filtered image depends on the sharpness of the initial image. Further, the sharpness of the initial image depends on the illumination of the scene, which, on the real-life prototype of a harvesting robot, is enhanced by the headlights mounted on the robot.

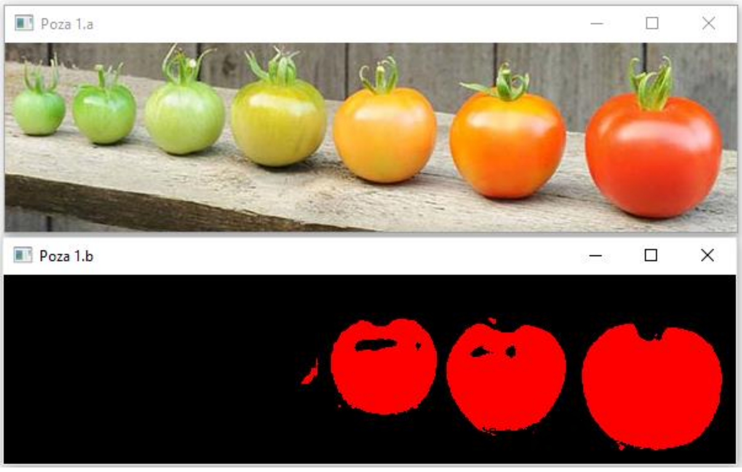

Figure 4. Ripe tomato detection (applied on an image retrieved from the internet)

In real-life situations (e.g., greenhouses), the colour filters we use during the learning activity identify correctly the ripe fruits from the background (Figure 5, Figure 6). However, students should experiment and adapt the colour filters according to their needs.

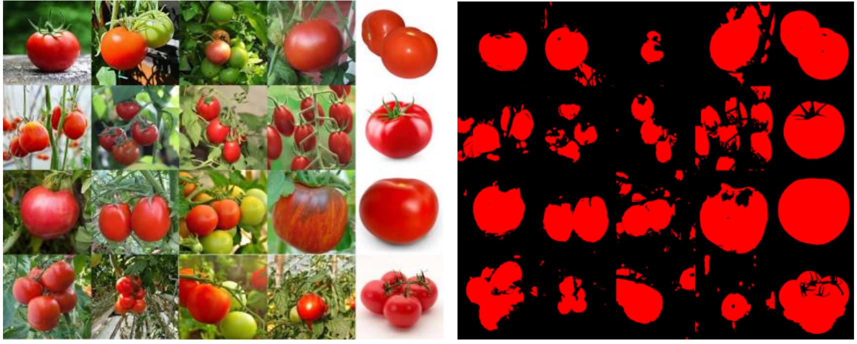

Figure 5. Red tomato detection (applied on images retrieved from the internet)

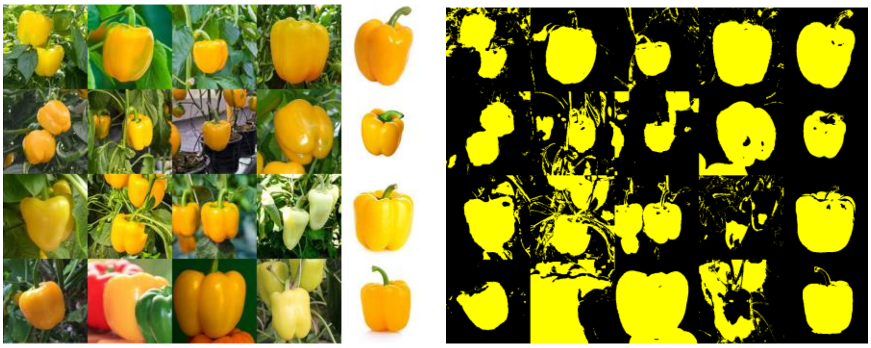

Figure 6. Yellow pepper detection (applied on images retrieved from the internet)

In order to detect a ripe fruit, first a rectangle is draw around the filtered image of the fruit. This permits to evaluate the dimensions of the fruit, compared with the average dimensions of a ripe fruit from the harvested variety.

Lesson 5: Identify a ripe fruit from a real-time picture.

The real-time picture uses images taken by a camera attached to the Raspberry Pi system.

For the real-life prototype of a harvesting system, the fruits are detected by analysing the images taken by a camera attached to the robotic arm. For learning purposes, it is sufficient to use a system which detects pictures of fruits (Figure 7).

The settings used by the camera are: ISO 200, image brightness 60, contrast 30, image dimension (resolution) 320x240 pixels. ISO 200 corresponds to a well-lit environment, which is provided by the headlights attached to the robot. The value for brightness should be an integer between 0 and 100, with the default 50. The value for contrast should be an integer between -100 and +100, with the default value 0. By modifying the brightness and the contrast of the camera sensor, the images become clearer.

The students should experiment with different conditions for brightness and contrast in order to calibrate the camera module and to obtain accurate information about the objects in the image.

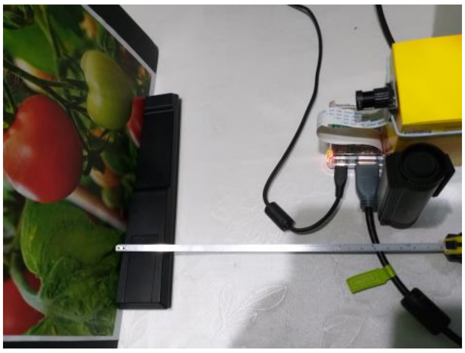

Figure 7. Experimental stand for detecting ripe fruits

3.2. Outcomes of the learning activity

The outcomes of the learning activity are:

- Deduction of the operating algorithm for a harvesting robot
- Image manipulation
- Applying a colour filter
- Deduction of algorithms for identifying a ripe fruit (tomato, orange, bell pepper, etc.)
- Integrating a camera module with a Raspberry Pi in order to produce a simple functional image recognition system

3.3. Assessment of learning

For now, the learning process is assessed by using a checklist. Students get help in order to perform the tasks and each of them advances in his/her own time.

The progress of students is recorded on a scale from 1 to 5 (where 1 = minimal competencies gained, 3 = average competencies gained and 5 = maximal competencies gained) from 3 points of view:

- progress in algorithmics
- progress in coding (Python)
- progress in using a microcomputer and camera system (Raspberry Pi + camera module)

In order to pass the learning activities, the students should get at least 2 average marks.

In addition to the checklist, a **self-assessment of learning** is performed at the end of each lesson. The questions that students are requested to answer are:

- What have I learned today?
- What went well?
- What went wrong?
- What do I need to pay attention to in the future?
- What is my overall impression on the activities performed today?

4. Conclusions

Starting from the need to improve the quality of the learning process, by adapting it to the realities of the 21st century, various solutions were identified, such as: designing interdisciplinary learning activities, using real-life outcomes to describe the learning outputs, using e-learning as a more efficient learning environment, using robotics as a tool for increasing the students' learning motivation.

This paper presents an interdisciplinary learning activity based on the vision system of a harvesting robot. During 5 lessons (on average, 10 hours of studying), students learn to code with Python, to manipulate images, to think critically, to devise their own algorithms for ripe fruit recognition and to use the camera module of a Raspberry Pi system.

The skills gained by the students are both hard and soft and contribute to build a sound background to 21st century life.

The learning is assessed both from the point of view of the educator and the student. While the educator can issue a certificate for the completion of the learning activity, the self-assessment process refers to the way in which the student relates to the activities.

Further work in this domain involves the creation of a fully virtual environment for this learning activity and also the use of robotics for more interdisciplinary learning activities related to mathematics, physics, computer science, earth sciences.

References

Arlegui, J., Menegatti, E., Moro, M. & Pina, A. (2008). Robotics, computer science curricula and interdisciplinary activities. In *Proceedings of the TERECop workshop teaching with robotics, conference SIMPAR* (pp. 10-21).

Halstead, M., McCool, C., Denman, S., Perez, T. & Fookes, C. (2018). Fruit quantity and ripeness estimation using a robotic vision system. *IEEE robotics and automation LETTERS*, 3(4), 2995-3002.

Kuo, H. C., Tseng, Y. C. & Yang, Y. T. C. (2019). Promoting college student's learning motivation and creativity through a STEM interdisciplinary PBL human-computer interaction system design and development course. *Thinking Skills and Creativity*, 31, 1-10.

Maris, S., Maris, S. A., Dungan, L. & Slavici, T. (2018). Re-shaping Education: Role of Internet Resources in Increasing the Efficiency of the Educational Process. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, 9(1), 132-137.

Maris, S.A., Ghergan, O., Maris, S., Draghicescu, D., & Popescu, E. (2021). The use of robots for environment protection in greenhouses. *Scientific Conference "Integrated Solutions regarding Environment Protection by Preventing and Reducing agro-industrial pollution"*. Zrenjanin, Online, 31.05-01.06.2021.

OpenCV – Python. Retrieved on 20.09.2022 from https://pypi.org/project/opencv-python/.

Partnership for 21st Century Skills, P21 framework definitions (2009, May, 25). https://eric.ed.gov/?id=ED519462.

Patrut, B., Maris, S. A. & Maris, S. (2020). Coding with Patience (Programare cu Rabdare)–a Non-Formal Educational Initiative to Increase the ICT Skills. *Postmodern Openings*, 11(3), 290-302, 2020.

Portable Python – Download (2022, September, 16) https://portablepython.com/download/.

Python. (2020, November, 14). *Python documentation*. https://www.python.org/doc/ .

Raspberry Pi Camera. (2019, December, 16). *Raspberry Pi documentation - Camera*. https://www.raspberrypi.com/documentation/accessories/camera.html

Raspberry Pi. (2019, December, 19). *Raspberry Pi 3 Model B+*. https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/.

Wan, P., Toudeshki, A., Tan, H. & Ehsani, R. (2018). A methodology for fresh tomato maturity detection using computer vision. *Computers and electronics in agriculture*, 146, 43-50.

Xu, S., Wang, J., Shou, W., Ngo, T., Sadick, A. M., & Wang, X. (2021). Computer vision techniques in construction: a critical review. *Archives of Computational Methods in Engineering*, 28(5), 3383-3397.