FRUIT HARVESTING ROBOT

● PHAM VAN CHINH - PHAM MANH TUAN - BUI THANH LAM

ABSTRACT:

This paper introduces an automatic robot for harvesting fruits in agriculture using a 4-degree-of-freedom robotic arm and an automatic classification and detection system of fruit objects. The control system consists of an Arduino Mega microcontroller and a servo motor, NVIDIA Jetson Nano Developer Kit, 12.3MP IMX477 Camera. The color features are extracted in the HSV color space and then used as input to the OpenCV library-based processing algorithm which will automatically calculate the values for the classification threshold. The resulting objects extracted by this method are presented in binary images. The results after various test cases show that the robot performs well with the accuracy and response time satisfying the specifications in the design. It is also low-cost and highly scalable with great potential for applying in agriculture, especially in harvesting fruits. Details of how this automatic robot was implemented and the results are presented below.

Keywords: machine vision, OpenCV library, harvesting robot, image processing, robotic arm, fruit picking, position control, kinematics.

1. Introduction

The world is witnessing a huge development in technology, most notably the industrial revolution 4.0. Along with it is the presence of robotic arms that appear widely in different fields. Their abilities are advantageous: high quality, accuracy, efficiency and economy, able to work in hazardous environments, in demanding jobs with precision. Research on agricultural robots has been launched since the 1980s [1-3]. For example, Japan has developed a robot used to spray pesticides. In the early 1990s, Korea began to research the technology of coupling automation. At present, many harvesting robots have been developed, such as tomato picking robot, cucumber picking robot, etc. Fruit is an essential product in human life and the automation in agriculture to harvest fruit is more and more demanded. Therefore, robot picking of ripe fruit is becoming the focus of recent studies. In Vietnamese agriculture, one of the countries with the the strength of agricultural products, the application of robotic arms in farms is extremely necessary.

Robots with four degrees of freedom have been studied in many scientific articles and have high applicability [11-15]. It proposed an optimal design for a parallel 4-degree-of-freedom robot to perform pick-and-place actions at high speed and high acceleration. The design achieved good dynamic equilibrium and eliminated singularities during working. Du MengMeng et al. [16] studied the kinematics and parameterization model of the 4-DOF Scara robotic arm. The structure of the robot was developed to reduce inertia, smooth motion, and minimize torque for agricultural applications.

For the task of analyzing a large number of fruits, we propose a fruit object extraction system based on HSV color space and OpenCV library. HSV is a color space that helps images to be displayed visually and processed easily. Instead of

plotting small vectors on an image, in HSV the vectors are mapped to H, S, and V values. This makes it easy to select the desired range of values. This system is operated in practice with complex environment, so OpenCV-Python is a useful open-source image processing library to use. In this project, we used a robot with four degrees of freedom combined with an image processing system to automatically pick fruits. The robot includes 2 reciprocating joints and 2 rotating joints to ensure flexible operation to working positions. The picking element is designed to be firm and suitable for many types of fruit. The setpoint coordinates are sent to the microcontroller by the image processing system.

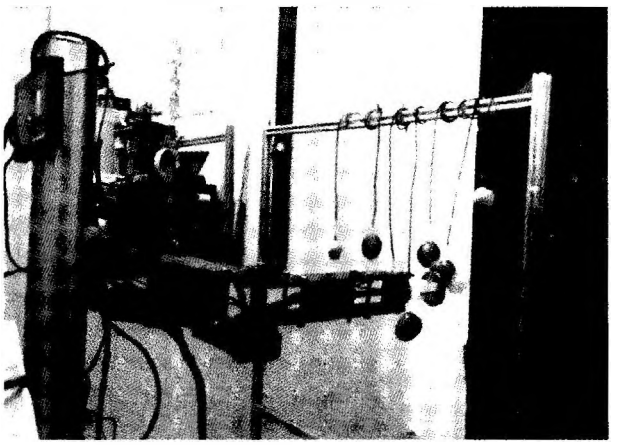
2. System overview

Ripe and unripe fruits have different color characteristics. Fruits are usually green before ripening, then they turn yellow, red or orange as the result of their pigments when ripe. The ripening period of the fruit is also divided into many different stages: the semi-ripe period (the top of the fruit is red-orange extending to the abdomen, the color is about 50%), the ripening period (the color of the fruit is characteristic full bloom, but the base is still green) and the final ripening period (full color and solid color). Fruit picking robot mainly handles at the stage when fruits needing to be harvested, which is the period of ripening. For the task of analyzing a large number of fruits, image-based classification is an accurate and optimal method. The images of fruits are recognized through the computer's vision sensors.

In this paper, we propose a fruit object extraction system based on HSV color space and OpenCV library. We use a series of algorithms to process the input image from the camera, convert the color space, get the object profile, and calculate it to give the actual coordinates to the robot's microcontroller. Finally, the actual images of fruits are tested to check the performance of the system.

After that, we used a robot with four degrees of freedom combined with an image processing system to automatically pick fruits. The robot includes 2 reciprocating joints and 2 rotating joints to ensure flexible operation to working positions. The picking element is designed to be firm and suitable for many types of fruit. The setpoint coordinates are sent to the microcontroller by the image processing system. The Arduino Mega microcontroller is used to process the signal from the image processing system and give control commands to the servo motor, based on the inverse kinematics calculation of the robot. Experimental results show that the robot has good performance, small error, and low latency. The robot has high applicability in the purpose of picking fruit according to the specified coordinates.

Figure 1. Automatic fruit harvesting robot

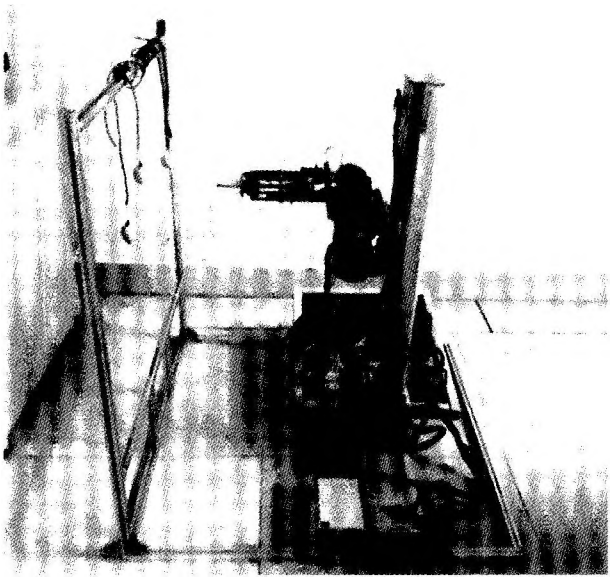


3. System design

3.1. Mechanical design

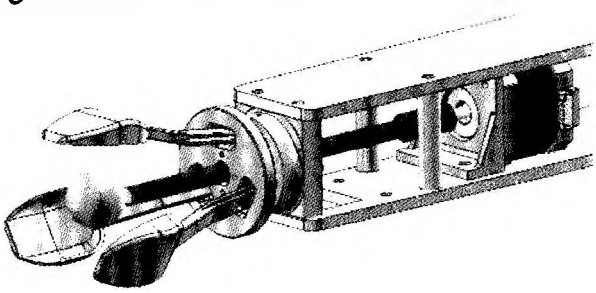
The system is a five degree of freedom robotic arm. Link 1,2,3,4 have lengths of 70 cm, 80 cm, 30 cm, and 10 cm, respectively. The joints of the robot work with rotation angles of (J1, 00), (J2, 00), (J3, 1800), and (J4, 1800). Each axis has a maximum operating angle, so the workspace must be determined for future testing. The maximum grip weight is 11.76 N and the maximum line speed is 100 mm/s. Fig. 2 shows the structure of the fruit picking robot. The end-effector is fixed for positioning, picking, and grasping. This design offers a simple and compact mechanical construction that not only provides sufficient DOF for pick and place tasks, but also facilitates highly efficient motion control.

Figure 2. A four degree of freedom robotic arm



The control signals of the transducer are transmitted from the a pre-programmed microcontroller. The end effector consists of three fingers as shown in Fig. 3. The nail is opened and closed by a moving device for opening and gripping. Each finger is attached with a soft coating material to reduce damage to fruit and vegetable during gripping. In addition, the end effector is equipped with a limit switch to detect the opening and closing range.

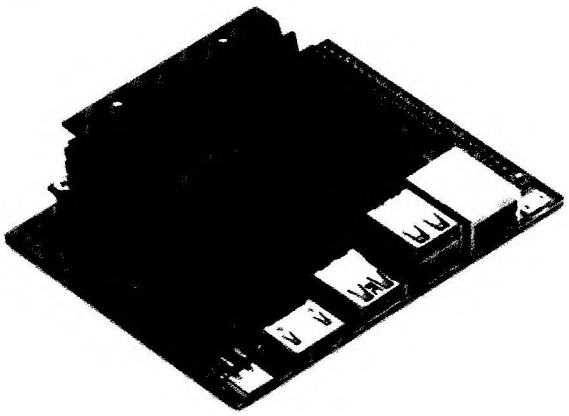
Figure 3. The end effector



3.2. Image processing unit

The NVIDIA Jetson Nano Deverloper Kit is a small but very powerful computer that allows you to run multiple neural networks simultaneously for applications such as image classification, object detection, segmentation, and speech processing, all in a single platform that is easy to use and

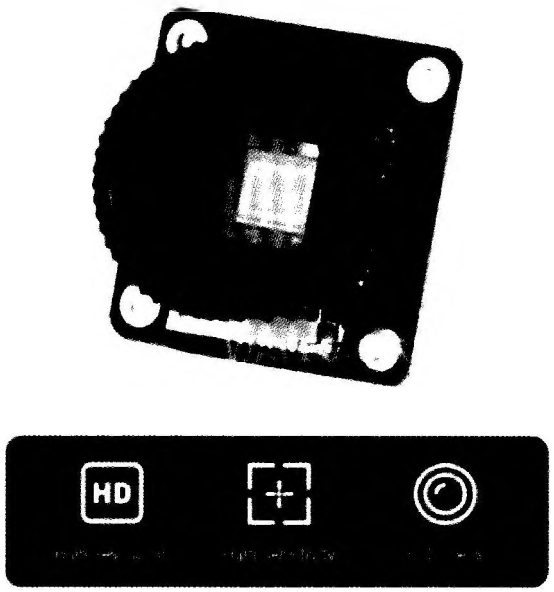
Figure 4. NVIDIA Jetson Nano Developer Kit B01



consumes less than 5 watts. The Jetson Nano also delivers 472 GFLOPS to run modern AI algorithms quickly, with a 64-bit ARM quad-core CPU, an onboard 128-core NVIDIA GPU, as well as 4GB of LPDDR4 memory. It is possible to run multiple neural networks and process several high-resolution sensors simultaneously.

The 12.3MP IMX477 camera is suitable for industrial use and other applications such as security or that require a higher level of image fidelity. This camera offers up to 12.3MP resolution, which is nearly 50% larger area per pixel compared to Raspberry Pi Camera 8MP, thus providing better visual effects.

Figure 5. Camera IMX 477 12.3MP



3.3. Controller

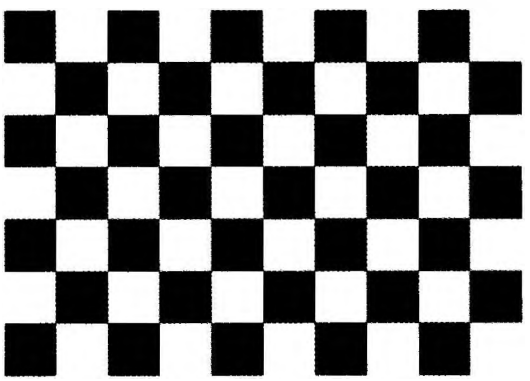
Movements of the links are driven by a NEMA 23 Teknic ClearPath Servo motor that operates at a maximum speed of 4000 RPM and a maximum torque of 2 Nm. Joint 1 (horizontal) is driven through a worm gearbox with a ratio of 20:1 and a holding torque of 10.17 N m. Vertical movement (y-axis) uses a gear box with a gear ratio of 10:1. These two joints are linked by a ??-shaped aluminum plate, so that the rotation axis of the two gearboxes is perpendicular to each other. The two swivel joints use a belt drive with a 1:2 gear ratio. The velocity of the rotary joints (i.e., servomotors) can be adjusted through frequency pulses varied from 0 to 500 kHz. The pulse signals are generated by the Arduino Uno microcontroller. The communication between the Arduino and the servo motor is established based on the transmission ports in the UART environment.

4. Theory and software library

4.1. Calibration solution for camera

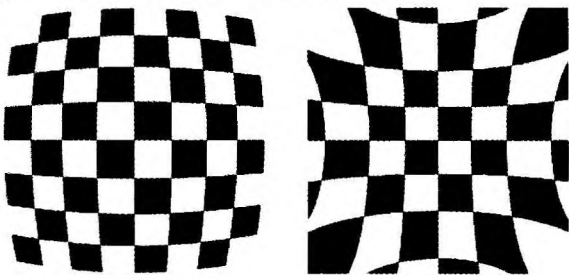
The process of converting images from zero to image space often gives non-preferred results because they are not maximized and is often distorted.

Figure 6. Camera calibration checkboard



When the camera is placed parallel and centered and has a standard position to ensure parameters of focus, the epipolar curves of distorted image will be corrected to become straight lines. The technique used to do this is called camera calibration. Camera calibration is done by taking checkerboard patterns at different angles to calculate the camera's internal parameters. Based on these parameters, it is possible to correct the distortion of the image by a suitable algorithm. The checker box shown in Fig. 6 is used to evaluate the curvature of the image obtained from the camera. Image distortion can be in the form of a convex or concave sphere as shown in Fig. 7.

Figure 7. Imagine distortion



Radial distortion occurs when light rays bend closer to the edges of the lens than to its optical center. The smaller the lens, the greater the distortion. The radial distortion coefficient models this type of strain. Distorted points are represented as (x_D, y_D).

$$x_D = x(1 + k^1r^2 + k^2r^4 + k^3r^6) \quad (1)$$

$$y_D = y(1 + k^1r^2 + k^2r^4 + k^3r^6)$$

Where:

- x, y: Pixel coordinates are not distorted.
- k_1, k_2, k_3: Radial distortion coefficient of the glazing system.
- $r^2 = x^2 + y^2$

4.2. Algorithm to convert jetson camera coordinates to real coordinates

The problem of determining real coordinates from image coordinates can be address using OpenCV.

Equation (2) show the relation of 2D coordinates on image with real 3D coordinates:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2)$$

Where:

- X, Y, Z: The actual coordinates of the object need to be determined.

• u,v: Coordinates of the projection point in pixels.

• A: Internal matrix of camera parameters.

• cx, cy: Coordinates of the reference point, always in the center of the image.

• f_x, f_y: Focal length

To simplify the calculation, the coordinates are converted to the following:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t \qquad (3)$$

Finally, the coefficient of real coordinates X, Y, Z in space is given as:

$$\left(s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} A^{-1} - t \right) R^{-1} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad (4)$$

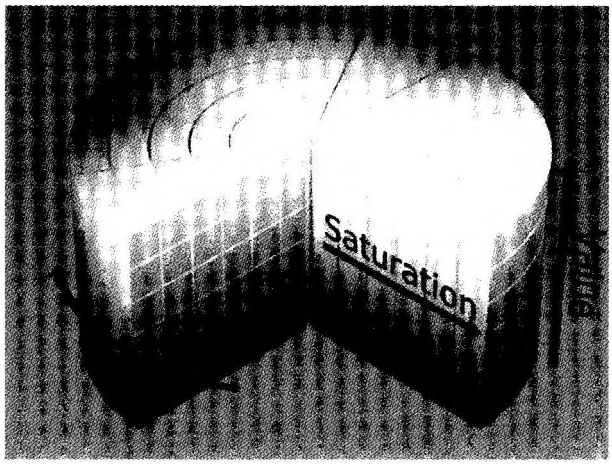
4.3. RGB to HSV color conversion algorithm

The RGB color space can be seen in Fig. 4(b). Colors that are often combined between red (R), green (G) and blue (B) are referred to the three primary colors [8]. This is also the most common color space. Each element can take a value from 0 to 255. Where (0,0,0) represents black and (255,255,255) represents white. From the three primary colors, different types of color spaces can be calculated using linear or non-linear transformations. Choosing the best color space is still one of the difficulties in color image segmentation. Here, we study the selection of HSV color space to process the classification of ripe fruits. The HSV color space is represented by three values of H (Hue), S (Saturation), and V (Value) as shown in Fig. 8.

The R, G, B values are divided by 255 to change the range from 0 - 255 to 0 - 1 as shown in equation (5).

$$\begin{aligned} R' &= \frac{R}{255} \\ G' &= \frac{G}{255} \\ B' &= \frac{B}{255} \\ C\,max &= max(R', G', B') \\ C\,min &= min(R', G', B') \\ \Delta &= C\,max - C\,min \end{aligned} \qquad (5)$$

Figure 8. HSV color space



With $\Delta = 0$, C max = R, C max = G, C max = B, the formula for the relation between coefficients R, G, B when converted to color system H, S, V is:

$$H = \begin{cases} 0^0 \\ 60^0 \times \left(\frac{G' - B'}{\Delta} + 6 \right) \\ 60^0 \times \left(\frac{B' - R'}{\Delta} + 2 \right) \\ 60^0 \times \left(\frac{R' - G'}{\Delta} + 4 \right) \end{cases}$$

$$S = \begin{cases} 0 & C\,max = 0 \\ \frac{\Delta}{C\,max} & C\,max \neq 0 \end{cases}$$

$$V = C\,max$$

From the conversion formula, we get the image of the object captured by the camera in Fig. 9 and the conversion result shown in Fig. 10.

4.4. Image processing based on OpenCV library

Profile-based object extraction can track objects of any color. Besides, it is possible to track both static and dynamic objects. In the case of still images, we must load the image into the program and then use the predefined OpenCV functions to track the object. In the case of video, we can track both live video and recorded video. Live video can be also controlled using Python OpenCV code. When the system starts to receive the video, it decomposes the video into picture frames. All image processing will take place on these frames. Each frame will be processed

Figure 9. RGB color space

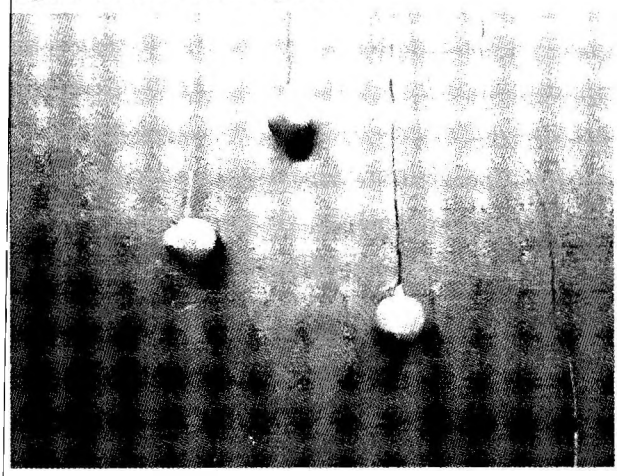
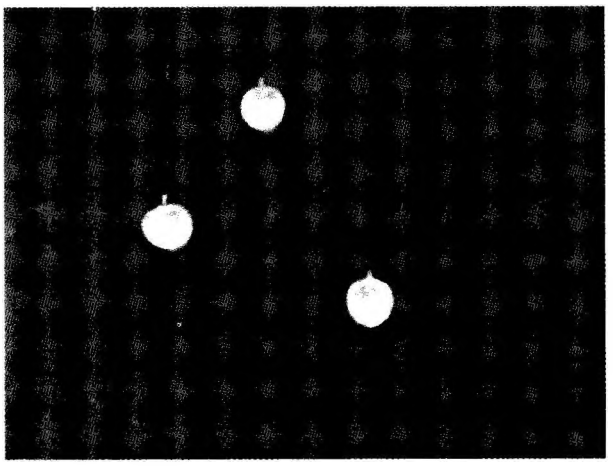


Figure 10. HSV color space



separately. Image representation is the first step of object tracking. When the object is present in the frame, it is represented. The image needs to be distinguished from the background. The color of the object is the main criterion for identifying the object. Using certain thresholding functions, each color object can be represented separately in the background. When the image is represented in the HSV color space, each value will have an upper amplitude band and a lower amplitude band. The upper and lower band ranges of some primary colors are shown in Table 1. After setting the ranges of values, the filtered object can be seen in Fig. 11.

Whenever objects are detected in the frame, a square outline is drawn for the object. That contour

Table 1. Upper and lower band ranges of some primary colors

Color	Lower range	Upper range
Red	[160,170,50]	[179,250,220]
Green	[53,74,160]	[90,147,255]
Yellow	[110,50,150]	[130,260,255]

will represent the object's boundaries. The next step is to find the center of the object. Here, the object focus is considered based on the area inside the contour.

If the image moment inside the contour is set to M, then the coordinates of the center can be calculated as follows:

cx = int(M['m10']/M[m00'])

cy = int(M['m01']/M[m00'])

Contour properties of an object can be obtained by using the cv2.boundingRect(contour) function in the OpenCV library.

5. Result

5.1. Image processing

Some of the object classification results are shown in Fig. 17. Ripe lemons in the left side of Fig. 17, tomatoes in the middle of Fig. 17 and oranges in the right side of Fig. 17 were selected as the subjects. They are processed and graded on each frame of the video. The coordinates of the object are determined and displayed on the screen with high accuracy (over 90%).

5.2. Robotic arm

The workspace is the total volume that the end effector can be achieved. The workspace size depends on the extent of the x, y, and z axis coordinates. The test is carried out by comparing the reference trajectory and the measured position of the link after being moved based on the inverse kinematics calculation. The effector of x-axis can reach a maximum length of 74 cm from the position of the base and has a good response from 40 to 65 cm.The end-effector cannot touch the object at those ranges because the link is too long while the object is too close to the robotic arm. Unknown state also occurs for points further than

74 cm as the length of the link is too short to reach to that point. The same pattern occurs for the z-axis coordinates. The z-axis travel range is limited from 0 cm to 70 cm. In addition, the average errors of the position of the end effector at the x, y, and z coordinates are 3.7%, 8.89% and 9.84%, respectively. Errors are acceptable in a range between 40 cm to 65 cm for x and y coordinates, and between 25 and 65 cm for the z coordinate.

6. Acknowledgments

This work has been supported by Vietnam National University, Hanoi (VNU), under project No. QG.20.80. The authors would like to thank VNU and all the other partners of the project ■

REFERENCES:

1. Metta, Giorgio. (2001). An attentional system for a humanoid robot exploiting space variant vision. *IEEE-RAS International Conference on Humanoid Robots 2001,* Tokyo, Japan, Nov. 22-24.

2. TQ Vinh, PM Thang, PM Duong (2010). Mạng thông tin điều khiển trong hệ thống tự động hóa tòa nhà. *Tạp chí Khoa học Tự nhiên và Công nghệ, Đại học Quốc gia Hà Nội,* tập 26, số 2, 129-140.

3. Elina, Olga. (2019). Development of Innovative Technologies in Russia Agro-Industry. *International Conference on Enginering Technologies and Computer Science(EnT).* IEEE. DOI:10.1109/ENT.2019.00027.

4. Buhl, Jens F., et al. (2019). A dual-arm collaborative robot system for the smart factories of the future. *Procedia manufacturing* 38, 333-340.

5. Byambasuren, Bat-Erdene, et al. (2020). *Apllication of image processing and industrial robot arm for quality assurance process of production.* 2020 IEEE Region 10 Symposium (TENSYMP).

6. Fu, Longsheng, et al. (2020). Application of consumer RGB-D cameras for fruits detection and localization in field: A critical review. *Computers and Electronics in Agriculture,* 177.

7. Santos, Luis, et al. (2019). Deep learning applications in agriculture: A short review. In book: Robot 2019: *Fourth Iberian Robotics Conference,* 139-151.

8. Lottes, Philipp, et al. (2017). Effective vision - based classification for seperating sugar beets and weeds for precision farming. *Journals of Field Robotics,* 34-36.

9. My, Chu A., et al. (2019). Novel robot arm design and implementation for hot forging press automation. *International Journal of Production Research* 57(14).

10. Sandoval, J., et al. (2018). Collaborative framwork for robot-assisted minimally invasive surgery using 7-DOF anthropomorphic robot. *Robotics and Automations Systems* 106, 95-106.

11. Wang, Yiming, et al. (2019). *Autonomous 3-D reconstruction, mappping, and exploration of indoor environment with a robotic arm.* IEEE Robotics and Automation Letter 4(4).

12. Oyekan, john, et al. (2020). Applying a 6 DOF Robotic Arm and Digital Twin to Automate Fan-Blade Reconditioning for Aerospace Maintance, Repair and Overhaul. *Sensor,* 20(16).

13. Pierrot, Francois, et al. (2009). Optimal design of a 4-DOF parallell manipulator: From academia to industry. *IEEE Transactions on Robotics* 25(2).

14. Rahul, K. Hifur Raheman amd Vikas Paradkar. (2020). Design of a 4-DOF parallel robot arm and the firmware implementation on embedded system to transplant pot seddlings. *Artificial Intelligence in Agriculture* 4.

15. Muladi, Siti Sendari and Ihlam Ari Elbaoth Zaini. (2019). Color-based PBEJCT Sorting in a Wide Range and Dense Target Points using Arm Robot. *2nd International Confrerence on Vocational Education and Training (ICOVET 2018).* Netherlands: Atlantis Press.

16. Roshanianfard, Ali, Du Mengmeng and Samira Nematzadeh. (2021). A 4-DOF SCARA robotic åm for various farm applications: Designing, kinematic modelling and parameterization. *Acta Technologica Agriculture* 24(2), 61-66.

Received date: January 2, 2022
Reviewed date: January 21, 2022
Accepted date: February 15, 2022

Authors information:
1. PHAM VAN CHINH[1]
2. PHAM MANH TUAN[2]
3. BUI THANH LAM[3]
[1]Vietnam Centre for Science and Technology Evaluation, Ministry of Science and Technology
[2]Faculty of Mechanics and Automation,
Vietnam National University - University of Engineering and Technology
[3]Faculty of Mechanical Engineering Hanoi University of Industry (HaUI)

MÔ HÌNH ROBOT THU HÁI HOA QUẢ

● **PHẠM VĂN CHỈNH[1]**
● **PHẠM MẠNH TUẤN[2]**
● **BÙI THANH LÂM[3]**
[1]Viện Đánh giá Khoa học và Định giá Công nghệ
Bộ Khoa học và Công nghệ
[2]Trường Đại học Công nghệ - Đại học Quốc gia Hà Nội
[3]Trường Đại học Công nghiệp, Bộ Công Thương

TÓM TẮT:

Bài báo giới thiệu một mô hình robot tự động thu hoạch hoa quả trong nông nghiệp sử dụng cánh tay robot 4 bậc tự do và hệ thống phân loại, phát hiện tự động độ chín của hoa quả. Hệ thống điều khiển bao gồm bo mạch NVIDIA Jetson Nano cùng vi điều khiển Arduino Mega 2560 kết nối với động cơ servo và camera12.3MP IMX477. Các đặc trưng màu được trích xuất trong không gian màu HSV và sau đó được sử dụng làm đầu vào cho thuật toán xử lý nhận dạng độ chín của hoa quả dựa trên thư viện OpenCV. Kết quả sau nhiều trường hợp thử nghiệm khác nhau cho thấy robot hoạt động tốt với độ chính xác và thời gian phản hồi đáp ứng các thông số kỹ thuật trong thiết kế. Với việc làm chủ công nghệ và sản xuất với chi phí hợp lý và khả năng mở rộng cao, sản phẩm của nhóm nghiên cứu có nhiều tiềm năng ứng dụng trong nông nghiệp, đặc biệt là thu hoạch trái cây trong các nhà kính.

Từ khóa: thị giác máy, OpenCV, robot thu hoạch, xử lý hình ảnh, điều khiển vị trí, chuyển động học.