

A Smart robotic arm for harvesting olive fruits

*Osama M. Al-Habahbeh¹, Salam Ayoub², Musa Al Yaman¹, Mohammad Matahen¹,
Mais Sarayra¹*

¹Mechatronics Engineering Department, School of Engineering, the University of Jordan, Amman, Jordan. Email: o.habahbeh@ju.edu.jo, m.alyaman@ju.edu.jo.

²National Agricultural Research Center, Amman, Jordan. Email: salam.ayoub@narc.gov.jo.

Abstract. This work introduces a smart robotic arm for harvesting olive fruits. The work started by planning the main steps. After deciding on the robot concept design, a kinematic and dynamic analysis is conducted. Based on the results of the analysis, a simulation is performed to determine the required specifications of the main components. Based on these results, an experimental model of the robot arm is designed, built, configured, and tested, where the model yielded good results. The next step is to configure and test different types of end effectors. In order to make the harvesting process effective for olive trees, picking a single fruit each time is not practical. Therefore, the procedure adopted in this work focuses on picking a group of fruit in each stroke. The harvesting system consists of a robotic arm guided by a stereovision camera to enable 3-D vision. Once the fruits location is detected, a reverse kinematics algorithm is initiated, yielding 3-points coordinates. These coordinates are commanded to the manipulator to move to the location and performs the picking process. Further developments will include building a larger manipulator that can reach out to all parts of a regular tree.

Keywords

Olive fruits harvesting, Smart picker, Agricultural robot, Fruit-picking automation, Agricultural labor, Agricultural engineering, Artificial intelligence, Robot design, Horticulture, Harvesting robot.

1 Introduction

Olive fruit picking represents more than half of the work required for the olive plantation [Hidalgo et al. [1]]. Olive harvesting is expensive since it is mostly done by hands [2]. On average, a ton of olives is sold for US\$550-900, while to harvest a ton, it costs US\$350 [3]. Therefore, there is a need for a robot to replace harvesters and reduce the cost of olives. A systematic review on robotics used in crop field operations has been conducted by Fountas et al. [4]; they underlined that the most explored robotic systems were related to harvesting and weeding, while the less studied were the disease detection and seeding robots. Numerous agrarian robots have been developed to perform precision agriculture operations and replace or augment humans in certain tasks [5]; these robots come in twotypes: a) Self-propelled

mobile robots, and b) Robotic “smart” implements that are carried by a vehicle. In recent years, many harvesting robots have been developed for various types of crops, such as the apple harvesting robot using the neural network object detection algorithm called Region-based Convolutional Neural Network (R-CNN) method [6]. Bachche [7] reviewed various design strategies in recognition and picking systems in the past 30 years. Tang et al. [8] reported the application and research progress of harvesting robots and vision technology in fruit picking. A conventional harvesting platform was transformed by Fei and Vougioukas [9] into a collaborative robot (co-robot) platform that matches the incoming fruit distribution with the worker's picking speed by controlling the heights of the hydraulic lifts that move workers up and down, resulting in an improvement of 9.5%. Dynamic predictive scheduling was modeled by Peng and Vougioukas [10] for teams of robots carrying trays during manual harvesting by picking crew. The feasibility of a picking robot design was determined by Wang et al. [11] using virtual simulation. A peduncle locking under-actuated harvester was simulated by Luo and Tan [12] using Adams software; the design is based on crank rocker mechanism, with fixed and floating plates used for fruit separation, and slider-crank mechanism used as handheld drive unit. A practical robotic fruit picking system was developed by Energid Technologies, as reported by Aloisio et al. [13]; the system employs flexible tubes with removal tools at one end that can be individually fired pneumatically and steered robotically. A synergistic robotic apple harvesting prototype is designed by Zhang et al. [14]; the prototype features deep learning-based fruit detection and localization using an RGB-D camera, a 3DOF manipulator with hybrid pneumatic/motor actuation, a vacuum-based end-effector, and a nonlinear velocity-based control scheme. A robot arm is proposed by Onishi et al. [15]; where Single Shot MultiBox detects the fruit position, and a stereo camera detects the 3D position. After conducting inverse kinematics, the robot harvests the fruit by twisting the hand axis. The experimental results showed that the robot could harvest a fruit in 16 s. Another research was conducted to study the vibration level of the harvesters by Cerruto and Manetto. [16]; a combination of six different harvesters head and four rod harvesters were studied. Kinematic, rod material and geometry were the main factors to examine, and the usage of the machines was set to 4 hours per day. The vibration exposure to the operator was much higher than the accepted level. Which could lead to a musculoskeletal disorders. A lower vibration level was obtained by changing the machine rod to carbon fiber, by increasing the rod diameter and most importantly changing the harvesters head so their kinematics inherently incorporates oscillation compensation. A novel low-cost pneumatic robot arm was designed by Aliff et al. [17]; it employs a vision-based control technique, a soft plastic arm, and air pumps with valves, two cameras, a control board, and a PC.

A well-known harvesting method is using shakers; however, canopy acceleration resulting when harvesting by shakers was correlated with fruit damage [Castro-Garcia et al. [18]]. Glavan et al. [19] discussed the application of Trunk shakers and Over-the-top harvesting in Europe; they concluded that Europe must produce good quality olive oil at around \$4.5/Liter, by harvesting efficiently at a cost of \$0.6/Liter. Solazzi et al. [20] described the complete design of an olive picking machine; comprising shaking rods and support structure, and driven by hydraulic plant. Different olive harvesting sites, both mechanized and manual, were assessed by Bernardi et al. [21]; considering the technical, economic, and environmental aspects, to develop a better version of the "olive harvesting database". A study was done to examine different types of olive harvesters by Calvo et al. [22]; one skilled operator was involved with four different electric harvesters. Observers noticed that the workers have tingling sensation in their fingers at the end of olive harvesting and all the machines in the study show a high acceleration value. Moreover, the study shows that the operator worked with the arms over the shoulders for most of the harvesting process,

together with the time and the vibration level exposed to the operator at a rate of 4 hour per day, may lead to extremity disorders.

A vision system is needed for fruit identification; Zhao et al. [23] presented a review on key vision control techniques and their potential applications in fruit or vegetable harvesting robots. An intelligent mobile picking robot was designed by Han et al. [24] based on STM32 development board, which is loaded with color sensor, infrared sensor, and ultrasonic sensor to identify maturity instead of CCD sensor. Binary large object (BLOB) analysis method was implemented by Dewi et al. [25] as the visual cue for a harvest handling robot; BLOB was used to detect fruit based on color and shape. The online images are captured by a PI Camera that supports Raspberry Pi, where the image plane is a 600×480 -pixel frame. The application of machine vision to agriculture, mainly for crop farming was reviewed by Mavridou et al. [26]; surveyed activities include fruit grading, fruit counting, and yield estimation.

The gripper is the part that does the picking task; various sensors are needed on the grippers to make them more flexible [Zhang et al. [27]]. Current grippers are unable to handle fruits properly because they are flexible and prone to damage. Besides, they have different shapes and textures. However, a promising approach for picking fruits and vegetables could be using contact grippers with under-actuated mechanisms and suction cups [Blanes et al. [28]. Ahmad and Ayoub [29] evaluated the effectiveness of pneumatic comb harvesting machine in comparison with traditional methods of olive harvesting. Castillo-Ruiz et al. [30] measured olive fruit detachment force (FDF) for different stalk twisting angles. Navas et al. [31] reviewed robotic harvesting methods and types of soft grippers. The superiority of soft robotic grippers for fruits and vegetables picking was reviewed by Peng et al. [32]; they summarized their characteristics such as variable stiffness, simple control, multi-sensing fusion, as well as types of actuators including pneumatic and electro-active polymers. A modifiable development platform for robotic fruit harvesting is presented by Brown and Sukkarieh [33]; it is used to test specific design choices on different fruit and growing conditions. The best harvest rate of 42% was observed when using soft gripper combined with persistent target tracking. A new picking way was introduced by Brown et al. [34]; using a granular material in a bag to act as a universal jamming gripper for picking arbitrary objects. Gao et al. [35]; developed a pneumatic finger-like end-effector for cherry tomato picking. The end-effector is pneumatically controlled and consists of finger clamped finger, telescoping cylinder and RGB-D camera. Although this end-effector has a good adaptability in fruit picking it has its failures due to localization failure and collisions with the branches and other fruits.

Based on the preceding literature review, it is evident that the cost of manual olive harvesting is high and increasing. A possible alternative is to use existing mechanical harvesters; however, these are only good for large and organized farms. In the previous works, the consensus was to take advantage of recent developments in robotics in order to reduce the cost of fruits harvesting in general. In addition, each of the previous works dealt with a specific type of fruits or vegetables. As far as the authors are aware, no olive picking robot has been presented in the literature. In this work, a new robot for olive harvesting was designed, and a prototype was built. Using the latest studies as guidance; the characteristics of the proposed robot include soft gripper, motorized 5DOF robotic arm, stereo vision system based on RGB camera and a nonlinear velocity control algorithm installed on Raspberry Pi processor. As part of the design, inverse kinematics and dynamic modeling have been performed.

The aim of this work is to develop a practical harvesting system for small farms and irregular trees that can reduce costs and provides superb service to farmers. An additional advantage of using harvesting robots, especially for olives, is that harvesting can be done

more than one time, at a relatively small additional cost. Currently, using hand picking means that due to the high cost, picking is done only one time. This means the farmers will wait for all the fruits to ripe, which means many of the fruits will fall down on the ground. On the other hand, using robots, harvesting can be done multiple times; the first round is done when some of the fruits are ripe (black) and the rest are still green. The second round is done when enough green fruits become black, and so on. Until all fruits are picked up and nothing falls down to the ground. The result will be more productivity. In this work, a robotic arm is designed, simulated, built, programmed, and tested to perform olive-harvesting task. However, at this stage, the olive-picking season is not yet in. Therefore, the proposed robot can handle a model tree that resembles olives. Later on, the robot will be tested for real olives. The presented robotic arm is planned to be part of a larger robotic system that will be used to harvest olive trees under variable conditions, such as inclined land and trees of irregular shape and size. At the end of the picking process, the fruits will be collected into an inverted umbrella.

2 Method

The materials used in this work are listed in Table 1. The stages followed in conducting this work include Design, Simulation, Building, Programming, and Testing.

Table 1. Main components and their specifications

Material	Specifications
Robotic arm	6 DOF Manipulator
End Effector	Mechanical Gripper
Camera	RGB-D
Controller	Raspberry PI

2.1 Design stage

A robotic arm and gripper for olive picking equipped with a camera for image processing was designed, simulated, built, programmed, and tested. The arm only constitutes phase-I of a future vision for a larger project to develop a complete robot for olive tree harvesting as shown in Fig. 1. The robot works essentially by revolving smart robotic arms around the tree to pick the fruits, and then gather them into an inverted umbrella. For the robot arm, a six degree of freedom (6-DOF) robot manipulator was selected with mechanical gripper end-effector. The working principle and components of the proposed robot are shown in Fig. 13.

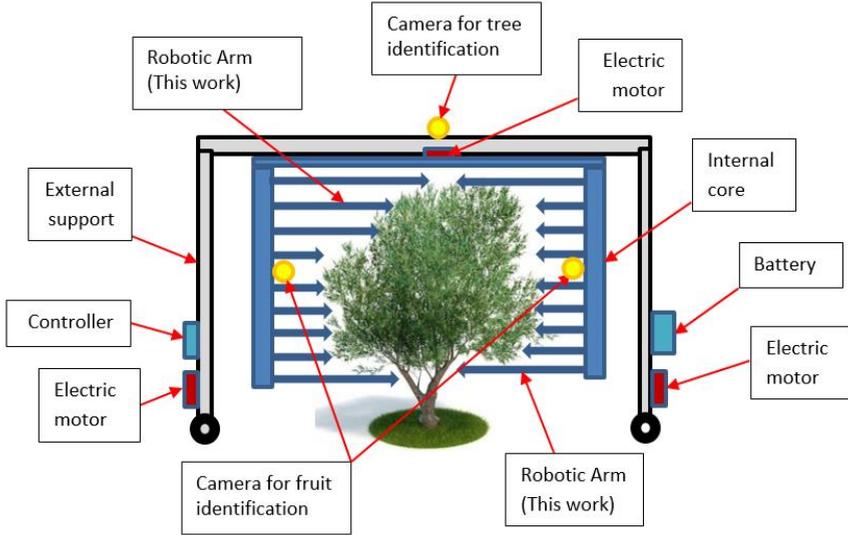


Fig. 1. Planned harvesting system

2.1.1 Picking system

This system performs grasping and cutting operations using a gripper. In order to achieve the highest performance, the most productive gripper must be chosen. In Table 2, a comparison is conducted between vacuum and mechanical grippers. Mechanical grippers are robotic arms with fingers known as jaws, attached to the end effector. They work the same way as human hand. They pick items by closing the jaws. These grippers are suitable for a variety of applications. However, the use of traditional mechanical gripper for olive picking was not proposed in the literature; most probably due to the fact that the process will be too long. However, gripping a group of fruits each time might be feasible. On the other hand, an existing commercial device that is used to harvest olive fruits is called shaker rake; in future work, an improvement to this device will be proposed. The gripper and the robot arm need actuation, where different types of actuators are listed in Table 3. For the traditional gripper, the detachment force was calculated using a digital force sensor by applying a force on the olive gradually during detachment, the study was done on multiple types of olive trees including Grossay, manzanillo, and shemali. The maximum force was found for Grossay olive trees, which was 6.8 N with branch length 17.3 cm and mass 5 g [36]. In future work, the picking process is planned such that the olive on the branch acts like a pendulum. It could be oscillated using the rotary harvester and then picked up. First the fruit is oscillated using direct contact then it is picked up using the detachment force produced by oscillation. The resultant force is represented by:

$$F = F_t - mg\cos(\theta) \quad (1)$$

The maximum force is found at the position of the fruit when theta = 0, hence:

$$F_c = F_t - mg\cos(0) \Rightarrow F_c = F_t - mg \quad (2)$$

Where,

F_c : Resultant centripetal force

F_t : Attachment force caused by the branch

- M: Mass of the olive fruit
- g: Acceleration constant (9.88 m/s²)
- θ: Angle between equilibrium and oscillated olive

The picking system design is based on the centripetal force caused by oscillation, which equals;

$$F_c = m \cdot r \cdot \omega^2 = \frac{mv^2}{r} \quad (3)$$

Where,

- v: Linear velocity of the olive (same as the harvester)
- r: Branch length.
- Ω: Angular velocity

Solving for the harvester velocity leads to the following equation:

$$v^2 = \frac{rF}{m} \quad (4)$$

Therefore, in order for the end effector to harvest the fruit, the linear velocity should be greater than the right side of the equation. A factor of 2 is used to guarantee success in detaching the fruit. This lead to the equation:

$$v^2 > \frac{r \cdot 2F}{m} \quad (5)$$

The force in equation (5) is the centripetal force caused by oscillation, and in this situation, we need the centripetal force to detach the olive fruit from the tree. Therefore, the centripetal force will be the minimum force needed to detach the olive from the branch. If traditional mechanical picker is not selected, another option is to design a vacuum gripper to create enough suction force to detach the fruit from the tree (FDF), Castillo-Ruiz [30]; found by experiments that the force needed to rip the fruit can be up to 1000 gram and the average diameter of the fruit is 18 to 20 mm. Thus, the force needed to detach the fruit in Newton is [37]:

$$F = P_{atm}A_o - P_{suction}A_i \quad (6)$$

Where,

- P_{atm} : Atmospheric pressure
- A_o : Outside area
- P_{suction} : Required suction pressure
- A_i : Inside area of the suction cup

Since the outer diameter is 2 cm and the inner diameter 1.5 cm, then,

$$P_{suction} = \frac{-2 \times 9.81 + 101325 \times \frac{\pi}{4} (0.02)^2}{\frac{\pi}{4} (0.015)^2} = 69.1 \text{ kPa} \quad (7)$$

Using a factor of safety of 2, an air pump with 69 kPa vacuum air pressure must be selected.

A third type of pickers is considered, which is the Rotary harvester; this harvester contains two effective parts connected to electric motors, where the motors move in the same direction or in opposite directions, depending on the applied control design. Each part contains several mini rackets that could be made of Rubber, Steel, Aluminum, Plastic, or reinforced plastic [38]. In this work, a design of this picker is proposed; it includes 8 sticks on every head, where every stick is 15 cm long. Since the average olive fruit diameter is 2.5, the number of olives the picker can harvest in one rotation is 48 olives for every head. Therefore, the total is 96 olive fruit. Further study of this type is planned in future work.

Table 2. Comparison between mechanical and vacuum grippers

Property/Gripper type	Mechanical	Vacuum
Speed	Lower because of the actuators response	Quick movement since the vacuum is fast in response
Material Handling	Harsh when gripping the object	Softer and gentle
Accuracy	More Accurate	May grip another object rather than the wanted.
Payload	High payload	For light loads
Noise	Limited noise source	High noise production
Cost	High maintenance cost	Expensive, but maintenance cost is low
Size	Applicable with medium to large size objects	Usable for small size object

Table 3. Comparison between actuators

Actuator	Advantages	Disadvantages
Hydraulic	High power, constant torque	Expensive, oil leakage
Pneumatic	High speed, cheap	Limited power, oil leakage, noisy
Electrical	Precise control, cleaner, less noisy, one component	Less power, overheat

To support the picking system, a robotic arm must be used; for this purpose, a 6-DOF manipulator is utilized. Since the robotic manipulator must be tall enough to reach the tree, a platform (base) is needed to support the robot. We have converted an off-the-shelf trolley into a platform.

2.1.2 Kinematic and dynamic analyses

In order to pick the fruits, the gripper must reach the desired position. Therefore, inverse kinematics of the manipulator must be performed to solve for the position. Since olive tree is complicated, the first 3 joints control the position of the end effector and the last 3 joints control the orientation of the end effector **Error! Reference source not found.** to harvest inner olives. The DH Parameters of the manipulator are shown in Table 4.

Table 4. DH Parameters of the manipulator

Link	a(m)	$\alpha(^{\circ})$	d(m)	θ
1	0	90	0.105	θ_1
2	0.105	0	0	θ_2
3	0.10	0	0	θ_3
4	0	90	0	θ_4
5	0	0	0.150	θ_5
6	0	0	0	θ_6

The transformation matrix for every joint is determined using the following matrix:

$$T_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

The transformation matrix of Frame 6 with reference to Frame 0 should be found in order to solve the forward kinematics of the robot, such that;

$$T_6^0 = T_1^0 * T_2^1 * T_3^2 * T_4^3 * T_5^4 * T_6^5 \quad (9)$$

Equation 9 expresses the end effector's position with reference to the base frame 0, which can be written as:

$$\begin{aligned} X &= 0.01 * \cos(t1) * (0.7\cos(t2 + t3) - 0.04 * \sin(t2 + t3) + 0.7 * \cos(t2) \\ &\quad + 0.25) \\ Y &= -(\sin(t1) * (70 * \cos(t2 + t3) - 4 * \sin(t2 + t3) + 70 * \cos(t2) + 25)) / 100 \\ Z &= -(7 * \sin(t2)) / 10 - (1229^{(1/2)} * \cos(t2 + t3 - \text{atan}(35/2))) / 50 \end{aligned} \quad (10)$$

Where t_1, t_2 , etc. are elements of the transformation matrix. The orientation of the end effector is controlled by all joints. As for the Inverse kinematics, it cannot be solved symbolically since the equations are nonlinear. Therefore, numerical methods are used to solve this problem. The function used is MATLAB Fsolve (fun, x0) which solves nonlinear equations with multiple variables. Considering the loads on the manipulator, the static torque is defined as:

$$\text{Static torque} = m * g * d * \cos(\theta) \quad (11)$$

On the other hand, the dynamic torque is found by multiplying the angular acceleration by the moment of inertia. However, before calculating the dynamic torque, we need to specify the speed of every joint to find the angular acceleration. The acceleration is equal to the slope of the speed time curve. Therefore, the slope should be small so that dynamic torque can be decreased. Assuming that the robot is designed to move from 0 degree to 90 degrees in 5 seconds, the distance will be $\pi/2$. Angular displacement is equal to the area under the speed time curve, which is found by integration. The speed and acceleration are calculated as:

$$\begin{aligned} \omega_{max} &= \frac{\pi}{5} \text{ rad/s} = 0.63 \text{ rad/s} = 22.9 \text{ degree/s} \\ \alpha_{max} &= \frac{\pi}{12.5} \text{ rad/s}^2 = 0.25 \text{ rad/s}^2 = 14.4 \text{ degree/s}^2 \end{aligned} \quad (12)$$

The maximum dynamic torque for first 3 joints is:

$$\text{Max Torque} = I * \alpha_{max} \quad (13)$$

For joint 1, the dynamic torque will be 2.5 N.m. however, the maximum torque equals the sum of dynamic and static torques, which equals 44 N.m for joint 1. Based on the above, the needed linear speed of the end effector must be greater than 21.5 m/s. The linear speed is converted to angular speed that is needed to detach the fruit, using the following equation:

$$N = \frac{v * 60}{2 * \pi * r} \quad (14)$$

The angular velocity (N) needed to detach the fruit is calculated as 1,187 rpm. This means the olive fruit should oscillate in angular speed more than 1187 rpm to detach from the tree. The next task is to calculate the required torque for the motor to handle maximum olive numbers in the sticks of the end effector. The number of sticks is 12 on every side and the length of each stick is 15 cm. The average width of olive fruit is 2.5 cm. Therefore, every stick could handle 6 olives. We can calculate the sum of torques for all olives or take the average distance of all fruits from the center of the motor. The distance from the sticks to the motor's rotor is 5 cm. The design is shown in Fig. 2.

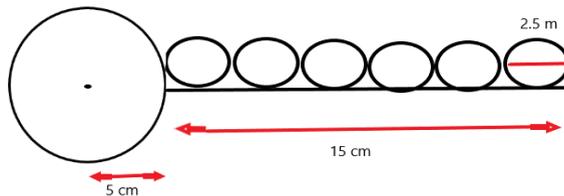


Fig. 2. Maximum number of olives on the stick

The maximum torque needed to oscillate all the olives is calculated as:

$$\text{Torque} = F_{oscillation} * d_{average} * \text{No. of olives} \quad (15)$$

Where,

$F_{oscillation}$: Force required to oscillate one olive fruit, equal to 0.05 N

$d_{average}$: Average distance from the fruit to the center of rotation

No. of olives: The number of olives one stick can handle

Since Torque for one stick is 0.0375 N.m, Torque of the motor is:

$$T_m = \text{torque for one stick} \times \text{number of sticks} \quad (16)$$

For this design, T_m equals 0.45 N.m.

2.2 Simulation Stage

To simulate the manipulator, a GUI was implemented using MATLAB GUI inventor. The GUI solves both the inverse and forward kinematics and implements the position of the end effector according to the inserted parameters for either the joint angles or the position of the end effector. The code generates a CAD model based on the input parameters as well as the associated workspace. A solidworks design of the assembled manipulator is shown in Fig. 3. Torque for every joint need to be calculated in order to size the motor to move the robot links. The worst case should be considered for calculations. The torque is divided into two categories, the first one is to calculate the maximum static torque which is exerted on the links when the angle is equal to zero, as shown in Fig. 4. The second one is the dynamic torque. The design of the rotary harvester placed on the end effector is modelled using Solid Works, and contains 3 pieces. The harvester sticks are shown in Fig. 5. The properties of the links are shown in Table 5.

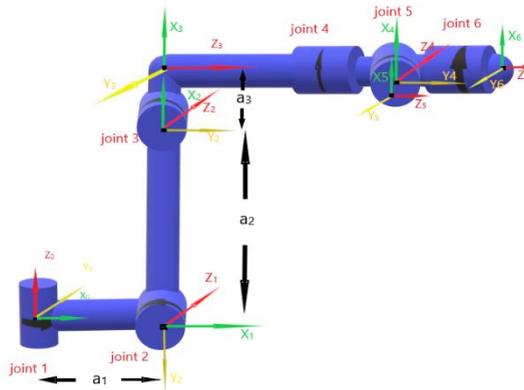


Fig. 3. Model of the manipulator

Table 5. Properties of the links

	Mass (kg)	Moment of inertia (kg.m ²)
Link 1	1.32	2.47
Link 2	1.57	3.95
Link 3	1.68	3.34
End effector	1.05	0.2

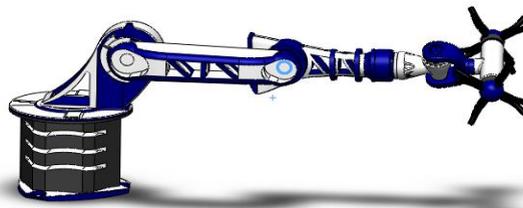


Fig. 4. Maximum torque position

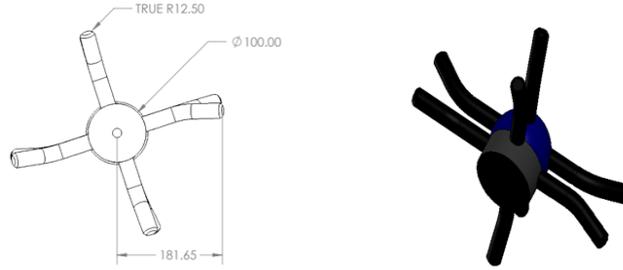


Fig. 5. Harvesting sticks

The manipulator model must be simulated to verify its operation. Assuming that there is an olive fruit at position $x=0.5$ m, $y=0.7$ m, and $z=0.9$ m, measured from the base frame of the robot. The designed GUI is used to calculate the angles of the joints with the final trajectory of the robot. As seen in Fig. 6, the robot joints need to make the following movements in order to reach the position of the olive; Joint-1: 54.5° , Joint-2: 70.2° , Joint-3: 24.2° , Joint-4: 31.7° , Joint-5: 11.8° , and Joint-6: 18.4° . The robot trajectory can be calculated and drawn at any position in the workspace after the links and joints are identified. Fig. 7 shows the robot initial position when all angles are zeros and the robot links are straitened. After the robot reaches the assumed position of the olive fruit, the trajectory of the robot is shown Fig. 8. As shown in Fig. 9, the first joint needed almost 2.5 seconds to reach its final position which is 54 degrees. In Fig. 10, Joint-2 final position was reached in 3.1 seconds. Joint-1 reached the final position faster than joint 2 since the angle of the second joint is bigger than the first one. As shown in Fig. 11, joint 3 needed only 1.1 seconds to reach the final position. This is the fastest joint, and that is because joint 3 moved only 24 degree.

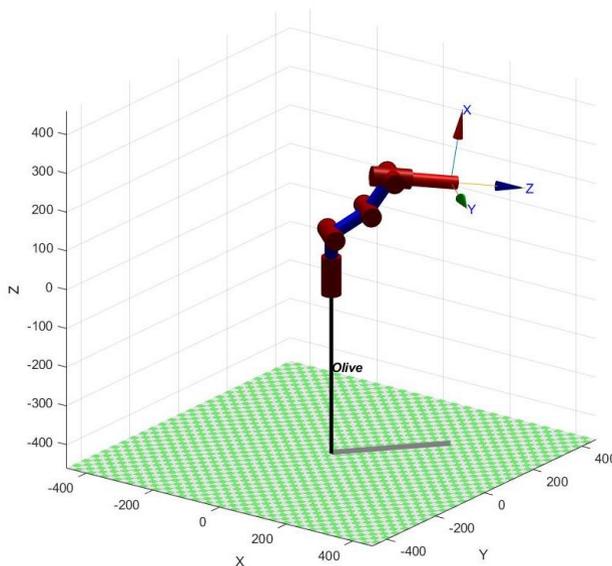


Fig. 6. Robotic arm model

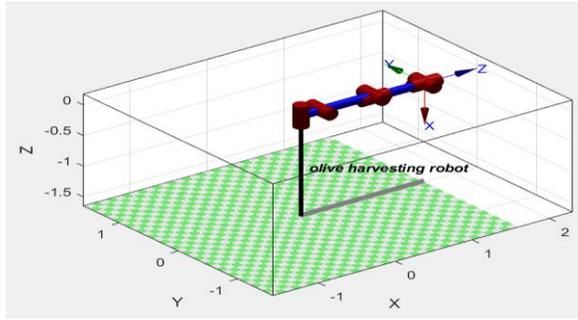


Fig. 7. Zero position trajectory

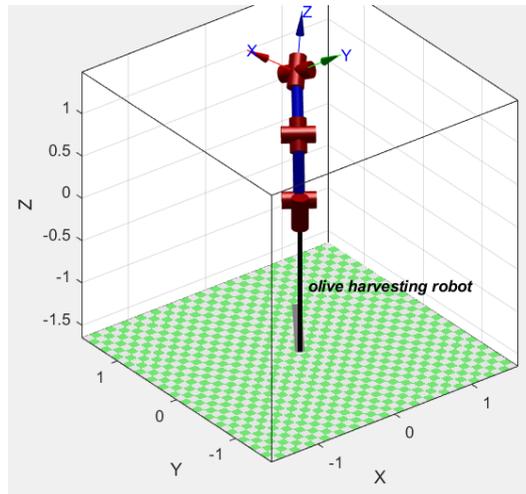


Fig. 8. Specified trajectory of the robot

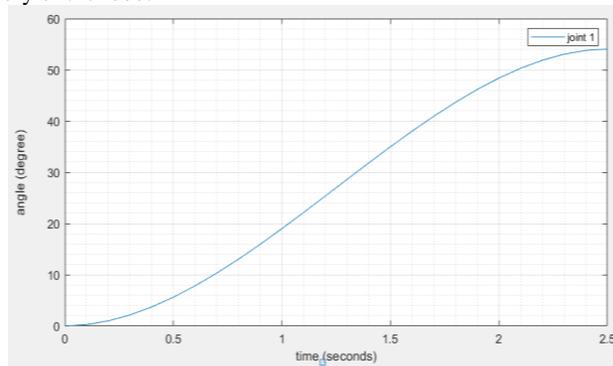


Fig. 9. Joint 1 angle with time

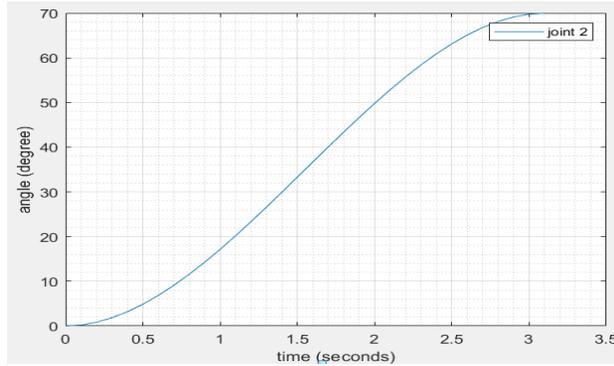


Fig. 10. Joint 2 angle with time

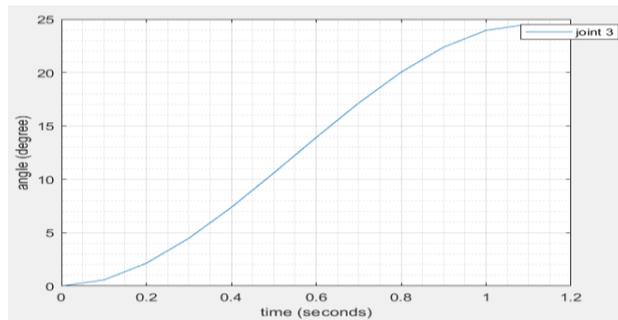


Fig. 11. Joint 3 angle with time

The loads imposed on the robot must be simulated so as to verify its fitness for service. The simulation was done by moving the robot with a speed of 10 rad/s while the robot was on the maximum torque position. The simulation helped us to calculate the stress load on the manipulator. In Fig. 12, the contour map shows that as we go away from the base, the displacement increases, as expected.

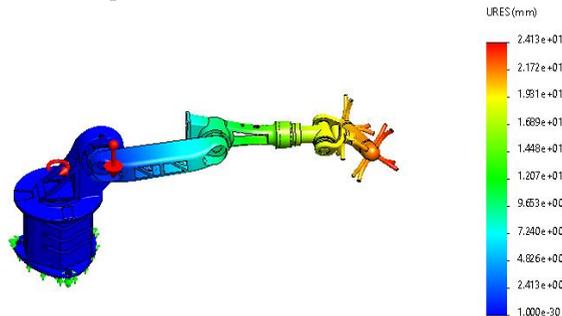


Fig. 12. Displacement contour map

2.3 Building Stage

This stage involves the actual building of the experimental robot which will be used for actual picking tests. Fig. 13 shows the harvesting robot diagram. It includes all the main components that are used to build the robot, as well as the connections between them. For the controller, Raspberry PI is used. A summarized list of the components is shown in Table 1, while a more detailed list is shown in Sub-section 2.5.1.

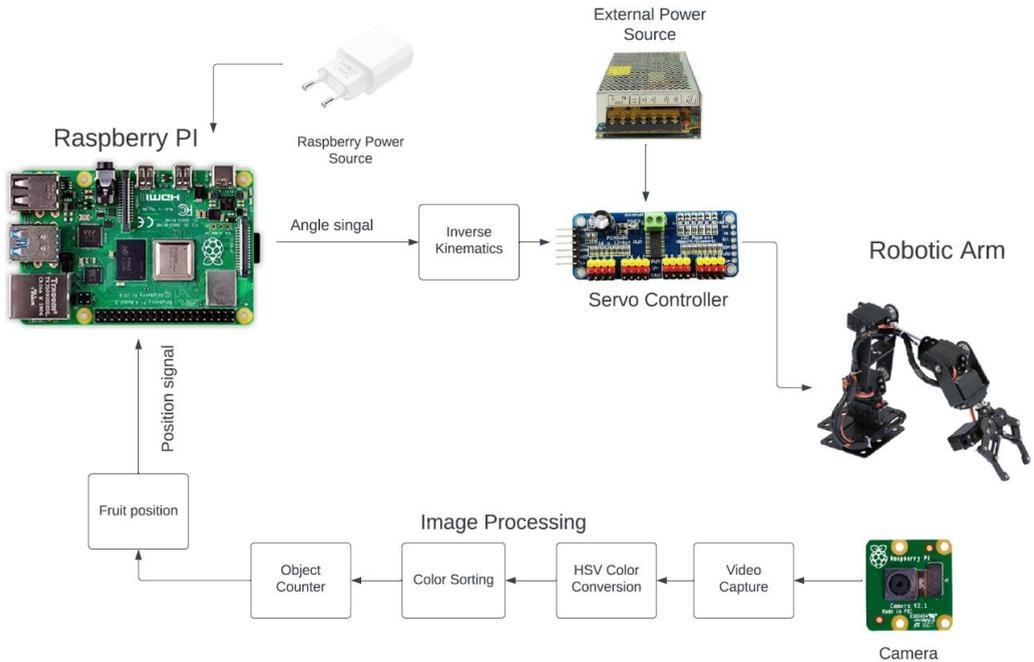


Fig. 13. Harvesting robot diagram

2.4 Programming Stage

2.4.1 Recognition system

A vision system is needed to identify and locate the olive fruits. The decision to move the picker will be done based on an algorithm which is trained to detect the olive fruits [39]. For this purpose, image processing algorithms are employed. TensorFlow object detection Application Programming Interface (API) is used with Python for olive detection. Based on that, the desired end effector position is calculated. The reason for using TensorFlow object detection API is that it has several capabilities that make it easy to build object detection models. The code starts by associating a virtual environment with all the object detection items. The next step is to import dependencies and define images for collection that will be used for practicing the model. In this case, these images will be for olives. Examples of these images are shown in Fig. 14. The next step is to set up folders, starting by creating a folder called workspace, and inside it another folder called images, and inside the images folder another folder called collected images. After the practicing images are collected, they must

be labeled. In the labeling step, the label image package from the GitHub site is used [40], which is a good package for labeling multiple objects. A code is used to download and launch the label image package then create a file to save the package. After executing the code, the image-labeling package will be launched directly. The labeling process will start after importing the practicing images. The process starts by selecting the first point and then drag it down to create the object. Figure 3 shows the process of labeling olives.

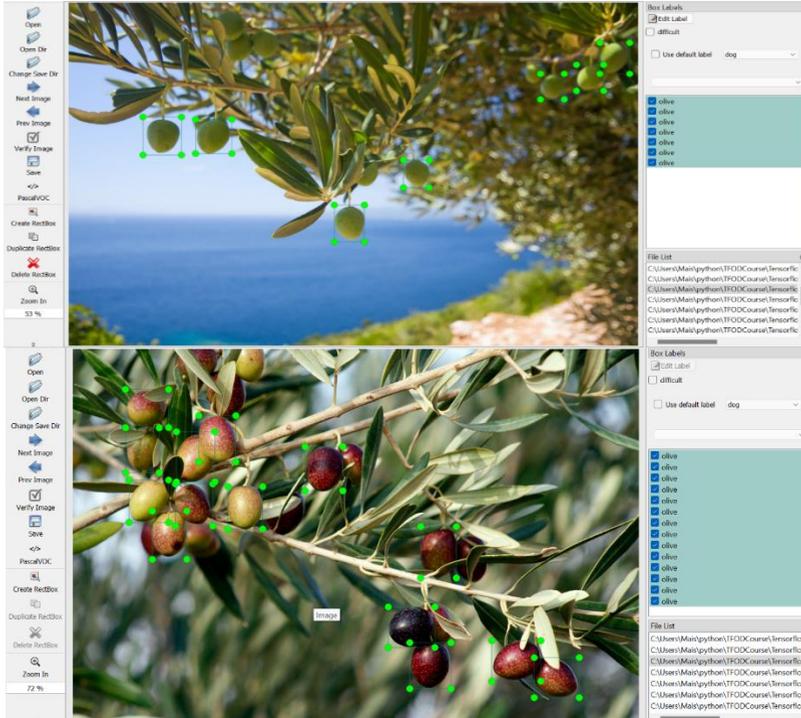


Fig. 14. Olive labeling using image labeling package

Another approach to olive detection is to use two specifications for the olive fruit; roundness and green color. This is in the case when olive is used to make pickles and not squeezed to get oil. The metric that will be used to measure the roundness of the object if the circle. When it is perfect circle the roundness will equal 1. Any other object that is not circle will have a metric less than one. The more the shape is round, the more the metric approaches 1. The metric is defined as:

$$metric = \frac{4\pi \cdot area}{perimeter^2} \quad (17)$$

After several iterations, we concluded that most olives have metric above 0.85. Using MATLAB image processing feature, we wrote a code that subtracts any color that is not green from the image, then identifies the roundness of any object in the image. It identifies any olive with roundness above 0.85. However, the main problem of this code is that the image background needs to be white. GUI was done to make the usage of the code easier. It should be noted that at this time of the year, olive fruits are not yet grown. Therefore, the harvesting system will be developed using a model plant with red fruits. Later on, when olive fruits are ready, the system will be modified and tested for olive harvesting. The detection

code used in the harvesting system depends on OpenCV library. This library implements an image processing and computer vision algorithm. The code starts with importing OpenCV library along with other libraries. The next step is to load a webcam image. A While loop is set in the code to play the video then convert the frame from Blue Green Red (BGR) to Hue Saturation and brightness Value (HSV) as shown in the Fig. 15. The following step is to define the color that needs to be extracted from the image (assuming it is red). This is done by defining the lower and upper bounds of the color. Then the mask of the image is obtained, which will return a new image that has only the pixels in that color range. Meanwhile, the other color pixels will be blacked out, as shown in Fig. 16. Fig. 17 shows the result of applying the mask on the original image, while Fig. 18 shows the contours applied to the result image.



Fig. 15. BGR to HSV conversion



Fig. 16. Mask of the image.



Fig. 17. Mask of the original object

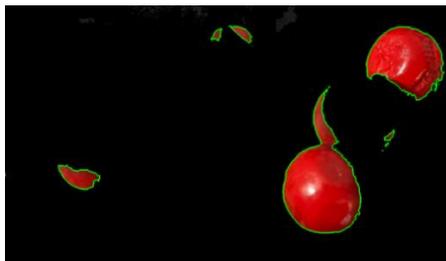


Fig. 18. Contoured mask of image

Green fruits can be detected as shown in Fig. 19, where all colors but green have been subtracted. Next, the picture is converted to binary in order for the code to read it. Only the needed pixels are kept. As seen in the figure, there are gaps in the olives due to the camera light, where parts of the olive seemed white. However, small gaps were filled after using a code to fill gaps in white pixels. Masking other colors and filled gaps is shown in Fig. 20. Fig. 21 shows the fruits after executing the code taken from the designed GUI, where olive fruits were detected and parameters were calculated using the image. Furthermore, the positions of the centers were identified. The performance of the algorithm was tested by adding other elements to the photo as shown in Fig. 22. These elements include non-green color, carrot, and other green color objects such as leaves. The first iteration did not recognize the green leaves as olives, and it masked the leaves since it is not a fruit. Therefore, the algorithm was successful in detecting the fruits.

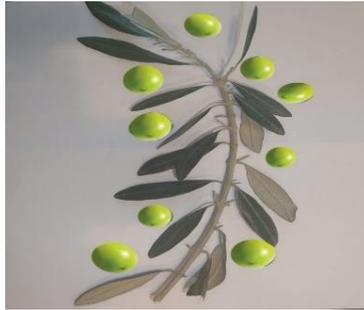


Fig. 19. Green fruits sample

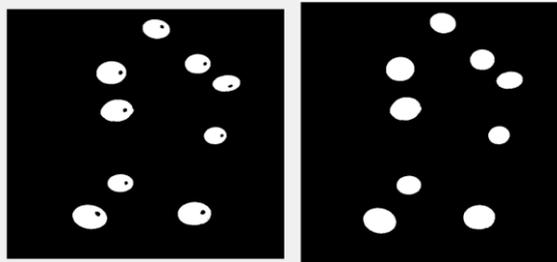


Fig. 20. Masking other colors and filled gaps

- 3- Servo Controller to control multiple servo motors. This type of controller is receiving the signals from main controller (Raspberry Pi) and can control up to 16 servos. Also, it has a separate power source that will provide the servos with enough power.
- 4- 5V, 5A Adapter to supply the power to the servos.
- 5- Mockup tree, shown in Fig. 24.

2.5.2 Results

The experimental manipulator was used to pick the fruits from the mock tree. Preliminary results have been obtained. A trial was done using predefined angles in order for the robotic arm to reach the fruit. Different joint speed profiles are shown in Fig. 23. From the data presented in Table 6, the picking success rate is 60%. However, the image processing success rate was higher by 10%. This is due to the fact that the servo motors used in the robotic arm do not have any feedback and they work in an open-loop setup. Image processing and detection of the fruits is shown in Fig. 24. An example of the robotic arm reaching for the fruits is shown in Fig. 25, where the gripper caught the fruit in a wrong way, but managed to pick the fruit successfully. Another case where the robot did not actually grab the fruit, but managed to detach it by lateral force. An example of the accuracy of the robot picking is shown in Fig. 26, where it actually gripped the fruit.

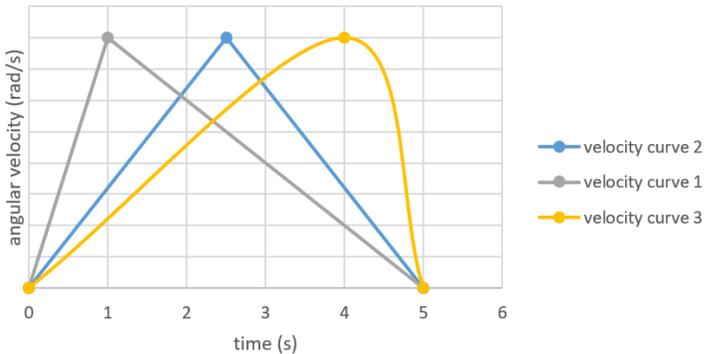


Fig. 23. Different speed profiles



Fig. 24. The Image processing and detection of the fruits



Fig. 25. Example of the Robotic arm reaching for the fruits



Fig. 26. An example of the accuracy of the robot picking

Table 6. Results gathered from multiple trials

Trail	Location Detection	Picking Success	Time (s)
1	Success	Yes	29
2	Success	No	35
3	Success	No	40
4	Failed	No	35
5	Failed	Yes	32
6	Success	Yes	34
7	Success	Yes	45
8	Failed	No	41
9	Success	Yes	36
10	Success	Yes	34

3 Conclusion and Recommendations

We have designed, simulated, built, programmed, and tested a robotic manipulator that can pick olive fruits. In order to compete with existing harvesting technologies such as shakers, future work will introduce further improvements to the experimental model, such as installing rakes to pick a group of fruits instead of a single one in each stroke. In addition, the harvesting robot runs on electricity and does not produce pollutants. The design was prepared using robot arm theory, combined with numerical matrix methods. The simulation was done using Solidworks™ and MATLAB™. The experimental model was built using a 6-DOF manipulator equipped with a mechanical gripper. Programming was done using Python and MATLAB. Programming is needed for two purposes; first is to analyze the pictures taken by the camera and find the position of the fruits, second is to calculate inverse kinematics for the position in order to extend the robot gripper to the fruit location. Testing of the experimental robot was done only using the mechanical gripper, as the other picking mechanisms such as the rake are not yet completed. The obtained results have been promising since a good percentage of the fruits were picked up successfully. It is worth noting that since the olive season has not arrived yet, the experiments were done using a mock tree.

During the design process, a mathematical analysis was conducted to find the critical parameters such as the loads, including static and dynamic torques. This was followed by robot simulation, where the 3-D model was built using SolidWorks™. This model allows performing stress and deformation analysis for the robot. Simulation of the robot trajectory was performed using MATLAB, where a Graphical User Interface (GUI) was designed to calculate the forward kinematics, inverse kinematics, and trajectory. Further simulations were done to evaluate the performance of the system. For the end effector, more is done with the simulation than with the actual model. In the simulation, rackets were installed to a cylinder connected to a motor. The speed and torque needed to harvest the olive fruits were calculated. On the other hand, digital olive detection was conducted and tested to identify olive fruits. The operation of the experimental robot was demonstrated using a mock tree. The obtained results have been reasonable. However, further improvements are needed. The robot will be redesigned so that it becomes faster in picking fruits. This can be done by minimizing the number of joints. Also, a totally different design can be tested in order to improve the performance. The control mechanism will be modified to closed-loop instead of open-loop in order to improve the performance. In the next step, a larger sized robot will be built and tested on a real tree. The future work involves installing shaker rake and static rake at the end effector of a larger robot with higher power. These are the lessons learned and hoping for better results in the future.

This work was supported by The National Center for Research and Development/The Higher Council for Science and Technology [Grant No. 3/4/4/486] and The University of Jordan [Grant No. 62/2021-2022].

References

1. Rafael E. Hidalgo, Fernández, Pilar Carranza-Cañadas, Francisco J. García-Salcedo, Paula Triviño-Tarradas, "Parameterisation and Optimisation of a Hand-Rake Sweeper: Application in Olive Picking", *Agriculture*, (2020), Vol.10, pp.379. DOI:10.3390/agriculture10090379.

2. "Low price electric 12v electric battery olive harvest machine|machine|machine machinemachine battery - AliExpress", *Aliexpress.com*, (2022). [Online]. Available: <https://www.aliexpress.com/i/32510729481.html>.
3. "Sample Costs | The Olive Oil Source", *Oliveoilsource.com*, 2022. [Online]. Available: <https://www.oliveoilsource.com/page/sample-costs>. [Accessed: 20- Feb-2022].
4. Fountas S, Mylonas N, Malounas I, Rodias E, Hellmann Santos C, Pekkeriet E. Agricultural Robotics for Field Operations. *Sensors*. (2020); 20(9):2672. <https://doi.org/10.3390/s20092672>.
5. Vougioukas, SG. (2019). Agricultural Robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 2(1), 365-392. <http://dx.doi.org/10.1146/annurev-control-053018-023617> Retrieved from <https://escholarship.org/uc/item/7b24n14c>.
6. K. Zhang, K. Lammers, P. Chu, Z. Li and R. Lu, "System design and control of an apple harvesting robot", *Mechatronics*, vol. 79, p. 102644, (2021). Available: [10.1016/j.mechatronics.2021.102644](https://doi.org/10.1016/j.mechatronics.2021.102644).
7. Shivaji Bachche, *Deliberation on Design Strategies of Automatic Harvesting Systems: A Survey, Robotics*, (2015), Vol. 4, pp.194-222. DOI:10.3390/robotics4020194.
8. Yunchao Tang, Mingyou Chen, Chenglin Wang, Lufeng Luo, Jinhui Li, Guoping Lian, Xiangjun Zou, *Recognition and Localization Methods for Vision-Based Fruit Picking Robots: A Review, Frontiers in Plant Science*, (2020), Vol.11, 510. DOI: 10.3389/fpls.2020.00510.
9. Fei, Z., Vougioukas, S.G., (2021), Co-robotic harvest-aid platforms: Real-time control of picker lift heights to maximize harvesting efficiency, *Computers and Electronics in Agriculture*, 180, 105894.
10. Peng, C., Vougioukas, S.G., (2020), Deterministic predictive dynamic scheduling for crop-transport co-robots acting as harvesting aids, *Computers and Electronics in Agriculture*, 178,105702.
11. Wang, C., Liu, S., Zhao, L., Luo, T., (2020), Virtual Simulation of Fruit Picking Robot Based on Unity3D, *Journal of Physics: Conference Series*, 1631(1),012033.
12. Luo, H., Tan, Y., (2018), Research on an under actuated dual separation plate harvester, *Beijing Linye Daxue Xuebao/Journal of Beijing Forestry University*, 40(12), pp. 110-116.
13. Aloisio, C., Mishra, R.K., Chang, C.-Y., English, J., (2012), Next generation image guided citrus fruit picker, *2012 IEEE Conference on Technologies for Practical Robot Applications, TePRA 2012*, 6215651, pp. 37-41.
14. Zhang, K., Lammers, K., Chu, P., Li, Z., Lu, R., (2021), System design and control of an apple harvesting robot, *Mechatronics*, 79,102644.
15. Onishi, Y., Yoshida, T., Kurita, H. et al. An automated fruit harvesting robot by using deep learning. *Robomech J* 6, 13 (2019). <https://doi.org/10.1186/s40648-019-0141-2>.
16. E. Cerruto and G. Manetto, "Vibration from Electric Hand-Held Harvesters for Olives," *Appl. Sci.*, vol. 12, no. 4, (2022), doi: 10.3390/app12041768.
17. M. Aliff, S. Dohita, T. Akagi and H. Li, "Development of a Simple-structured Pneumatic Robot Arm and its Control Using Low-cost Embedded Controller", *Procedia Engineering*, vol. 41, pp. 134-142, (2012). Available: [10.1016/j.proeng.2012.07.153](https://doi.org/10.1016/j.proeng.2012.07.153).
18. Sergio Castro-Garcia, Uriel A. Rosa, Christopher J. Gliever, David Smith, Jacqueline K. Burns, William H. Krueger, Louise Ferguson, and Kitren Glozer, "Video Evaluation of Table Olive Damage during Harvest with a Canopy Shaker",

- HortTechnology Journal, (2009), Vol. 19, No. 2, pp.260-266. DOI: <https://doi.org/10.21273/HORTSCI.19.2.260>.
19. Dan Glăvan, Theoharis Babanatsas, Roxana Mihaela, Babanatis Merce, “Study of Harvesting Methods and Necessity of Olive Harvesting Robot”, Annals of Faculty Engineering Hunedoara– International Journal of Engineering, (2016), Vol. 14, No. 3, pp.143-146.
 20. Luigi Solazzi, Roberto Scalmana, Riccardo Adamini, Rodolfo Faglia, Alberto Borboni, “Design of an innovative olive picking machine”, Agric Eng Int: CIGR Journal, (2014), Vol. 16, No.3, pp.102-112.
 21. Bernardi, B., Falcone, G., Stillitano, T., (...), Bacenetti, J., De Luca, A.I., (2021), Harvesting system sustainability in Mediterranean olive cultivation: Other principal cultivar, Science of the Total Environment, 766,142508.
 22. A. Calvo, R. Deboli, C. Preti, and A. de Maria, “Daily exposure to hand arm vibration by different electric olive beaters,” J. Agric. Eng., vol. 45, no. 3, pp. 103–110, (2014), doi: 10.4081/jae.2014.424.
 23. Yuanshen Zhao, Liang Gong, Yixiang Huang, Chengliang Liu, A review of key techniques of vision-based control for harvesting robot, Computers and Electronics in Agriculture, (2016), Vol.127, pp.311–323.
 24. Han, J., Liu, L., Zeng, H., Design and Implementation of Intelligent Agricultural Picking Mobile Robot Based on Color Sensor, Journal of Physics: Conference Series, (2021), 1757(1),012157.
 25. Dewi, T., Mulya, Z., Risma, P., Oktarina, Y., (2021), BLOB analysis of an automatic vision guided system for a fruit picking and placing robot, International Journal of Computational Vision and Robotics, 11(3), pp. 315-327.
 26. Mavridou E, Vrochidou E, Papakostas GA, Pachidis T, Kaburlasos VG. Machine Vision Systems in Precision Agriculture for Crop Farming. Journal of Imaging. (2019); 5(12):89.
<https://doi.org/10.3390/jimaging5120089>.
 27. Baohua Zhang, Yuanxin Xie, Jun Zhou, Kai Wang, Zhen Zhang, “State-of-the-art robotic grippers, grasping and control strategies, as well as their applications in agricultural robots: A review”, Computers and Electronics in Agriculture, (2020), Vol. 177.
 28. Carlos Blanes, Martin Mellado, Coral Ortiz, Ángel Valera, “Review. Technologies for robot grippers in pick and place operations for fresh fruits and vegetables”, Spanish Journal of Agricultural Research, (2011), Vol.9, No.4, pp.1130-1141.
DOI: 10.5424/sjar/20110904-501-10.
 29. Ahmad R., Ayoub S., “A comparative study of hand-held harvesting machine with traditional methods used for olive harvesting in Jordan, Proceedings of the 5th Int. Conf. Olivebioteq, (2014), pp.206-211.
 30. Francisco J. Castillo-Ruiz, Sergio Tombesi, Daniela Farinelli, “Olive fruit detachment force against pulling and torsional stress”, Spanish Journal of Agricultural Research, (2018), Vol.16, No.1, <https://doi.org/10.5424/sjar/2018161-12269>.
 31. Navas, E.; Fernández, R.; Sepúlveda, D.; Armada, M.; Gonzalez-de-Santos, P. Soft Grippers for Automatic Crop Harvesting: A Review. Sensors (2021), 21, 2689.
<https://doi.org/10.3390/s21082689>
 32. Peng Y, Liu Y., Yang Y., Yang Yi, Liu N., Sun Y., (2018), Research progress on application of soft robotic gripper in fruit and vegetable picking, Nongye Gongcheng Xuebao/Transactions of the Chinese Society of Agricultural Engineering, 34(9), pp. 11-20.

33. Brown, J., Sukkarieh, S., (2021), Design and evaluation of a modular robotic plum harvesting system utilizing soft components, *Journal of Field Robotics*, 38(2), pp.289-306.
34. E. Brown et al., "Universal robotic gripper based on the jamming of granular material," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 107, no. 44, pp. 18809–18814, (2010), doi: 10.1073/pnas.1003250107.
35. J. Gao et al., "Development and evaluation of a pneumatic finger-like end-effector for cherry tomato harvesting robot in greenhouse," *Comput. Electron. Agric.*, vol. 197, no. February, p. 106879, 2022, doi: 10.1016/j.compag.2022.106879.
36. M. Ibrahim, "Development of an electrical hand-held olive harvester", *Misr Journal of Agricultural Engineering*, vol. 35, no. 3, pp. 827-846, (2018). Available: 10.21608/mjae.2018.95541.
37. Ahmed, Shaheen. "The Open Educator." *The Open Educator*, (2017).
38. Z. Littlefield et al., "Evaluating end-effector modalities for warehouse picking: A vacuum gripper vs. a 3-finger under actuated hand," 2016 IEEE International Conference on Automation Science and Engineering (CASE), (2016), pp. 1190-1195, doi: 10.1109/COASE.2016.7743540.
39. P. Patel and B. Bhavsar, "Object Detection and Identification", *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 10, no. 3, pp. 1611-1618, (2021). Available: 10.30534/ijatcse/2021/181032021.
40. "GitHub," [Online]. Available: <https://github.com/tzutalin/labelimg>.