2024 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
July 15-19, 2024. Boston, MA, USA

2024 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) | 979-8-3503-5536-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/AIM55361.2024.10636976

Automated Harvesting of Green Chile Peppers with a Deep Learning-based Vision-enabled Robotic Arm

Umme Kawsar Alam[1], Luke Garcia[2], Jeremy Grajeda[3], Mahdi Haghshenas-Jaryani[1] and Laura E. Boucheron[3]

***Abstract*— The green chile crop is entirely hand-harvested in New Mexico while the growing labor shortage has caused a significant reduction in production. This work presents the robotic harvesting of chile peppers in a lab setting, employing a 6-DOF robotic arm with a scissor-type cutting end-effector. The system utilizes a machine learning-based computer vision and a depth camera to detect and localize chile peppers in the camera frame. The locations are then transformed into the robot's operational frame. A motion planning algorithm was developed to minimize the robot's travel time for harvesting. A correction equation is derived to address inaccuracies in camera-based localization while eliminating chiles that are not reachable for the robot. From a dataset of 86 chile peppers, the study reports key harvesting metrics: a detection success rate of 62.8%, a localization success rate of 90.74%, a detachment success rate of 55.10%, a harvest success rate of 31.39%, and a damage rate of 6.97%.**

I. INTRODUCTION

Chile pepper, the signature crop of New Mexico (NM), holds immense significance for small-scale farmers in Hispanic and Native American areas. Despite the leading role of NM in chile production, challenges such as drought, climate change, and labor shortage have severely impacted this industry. In 2021, only 51,000 tons of chiles were harvested, which is a drastic decline from the 2000s [1]. One of the main reasons is that green chile peppers are currently only hand-harvested [2]. Nevertheless, manual harvesting poses significant challenges due to the intense heat in NM. Additionally, the repetitive and strenuous nature of the harvesting task makes it increasingly challenging to secure skilled labor, especially given the heightened uncertainty surrounding the availability of agricultural workers [3]. These facts emphasize the necessity for the mechanization and automated harvesting of this crop.

In the last few decades, there has been continuous development in autonomous farming and agricultural robotics to improve the situation of food scarcity due to the shortage of manual labor for harvesting. For example, a Cartesian manipulator robot was developed for harvesting melon-type fruits. In this system, the fruits are detected and localized through image processing applied to the fruits' grayscale images. The system achieved a detection success rate of 93.6% and harvesting success of 85.7% [4]. A robot manipulator with a stereo visual unit, a fruit collector, and a railed vehicle was designed for harvesting cherry tomatoes and achieved an 83% success rate [5]. A robotic manipulator with an end-effector consisting of a fruit-grasping mechanism, a size-judging mechanism, and a peduncle-cutting mechanism was developed for harvesting eggplants along with a machine vision algorithm to recognize the fruits [6]. Their system achieved a harvesting success rate of 62.5%. Goulart et al. [7] discussed the development and evaluation of gripper systems for soft tree-fruit harvesting, introducing metrics such as picking area and volume, grasp force, detachment success, harvest success, and post-harvest damage assessment, with an empirical evaluation of gripper designs for mango fruit harvesting. Their study reports a harvesting success rate of 65%. Another work focused on developing a visual servoing control system for a robotic manipulator designed for cherry tomato harvesting, incorporating an RGB-depth camera and a visual-based control method. The robotic manipulator, equipped with a novel cutting and clipping integrated mechanism, demonstrated efficient cherry tomato harvesting with an average time of 9.4 seconds per fruit and a success rate 96.25% in laboratory tests [8].

Although there have been several research efforts to develop harvesting robots for different kinds of fruits [9], [10], only a few studies have been conducted to develop a robotic harvesting system and fruit detection algorithms for the chile pepper. In our previous work [11], a 5-DoF serial manipulator is used to harvest chile peppers in a laboratory setting where a MATLAB program is developed for the localization of the fruits, and a human operator is needed to identify the fruits in the captured RGB image through a depth camera. In another study [12], advanced deep learning algorithms are utilized and compared to detect the chile pepper using a thermal camera and thermal reflections, facilitating the fruit detection part of the robotic harvesting.

This work presents an effort to make the harvesting process of green chile peppers fully automatic by utilizing a robotic arm. Green chile peppers are detected through a deep learning method and localized by utilizing depth and RGB color images of a camera. A motion planning algorithm for the manipulator is developed to make the harvesting process easier and faster. The system is examined for 86 green chile peppers and the key harvesting metrics are recorded.

This work was supported by the U.S. Department of Agriculture (USDA), National Institute of Food and Agriculture (NIFA), award number 2021-67021-34203.

[1]Umme Kawsar Alam and Mahdi Haghshenas-Jaryani are with the Department of Mechanical and Aerospace Engineering, New Mexico State University, Las Cruces, NM 88003, USA umme@nmsu.edu and mahdihj@nmsu.edu

[2]Luke Garcia was with the Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM 88003, USA when completing this research. lsgarcia556@gmail.com

[3]Jeremy Grajeda and Laura E. Boucheron are with the Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM 88003, USA jgra@nmsu.edu and lboucher@nmsu.edu

979-8-3503-5536-9/24/$31.00 ©2024 IEEE

Authorized licensed use limited to: UNIVERSIDAD DE GRANADA. Downloaded on January 07,2026 at 16:01:52 UTC from IEEE Xplore. Restrictions apply.

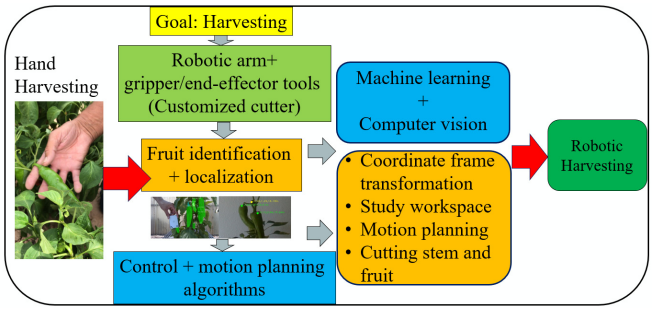

Fig. 1. Robotic harvesting concept

The paper structure includes an overview of the harvesting system in Section II, details on fruit detection and localization in Section III, and the introduction of coordinate frame transformation, end-effector reachability, and motion planning in Section IV. Section V concludes with the harvesting results and parameter discussions.

II. Robotic Harvesting System

The idea behind automated robotic harvesting is to convert the manual fruit-picking process into a fully mechanized system. To achieve this, a system must be developed to identify chiles on a plant, determine their locations, reach them, and perform the cutting operation. These steps encompass fruit detection using a machine learning algorithm, localization of the 3D distance of the fruit via robotic vision, motion planning for a robotic manipulator, and the actual automatic cutting operation as illustrated in Figure 1. To achieve this objective, this study employs and integrates a six-degree-of-freedom robotic manipulator (xArm6 by UFACTORY, Shenzhen, China), a depth camera (Intel RealSense D435i, Santa Clara, CA, USA), a customized cutter, and an Arduino Due (Arduino LLC, Piscataway, NJ, USA) that controls the cutter. The base of the manipulator is fixed to a table and a chile plant pot is placed in front of it. The camera, the RealSense D435i has enhanced depth detection capabilities by incorporating an inertial measurement unit (IMU) into the D435 version, enabling it to perceive depth information while in motion. Given that the minimum depth distance at the maximum camera resolution is approximately 28 cm, the chile plant is positioned beyond this distance in the camera's z-direction (forward). The camera is attached to the wrist, creating an "eye-in-hand" configuration, and a customized cutter is integrated into the manipulator's end-effector. This configuration is better suited for this application, as it reduces the complexity of coordinate frame transformation due to having a fixed relative pose between the camera and the end effector. The cutter is composed of two sharp blades, one of which is fixed and the other of which rotates around a pivot located inside the blade's body [11]. To regulate the motion of the cutter, a digital servo motor that is programmed by an Arduino is attached to it. The entire robotic harvesting setup is shown in Fig. 2.

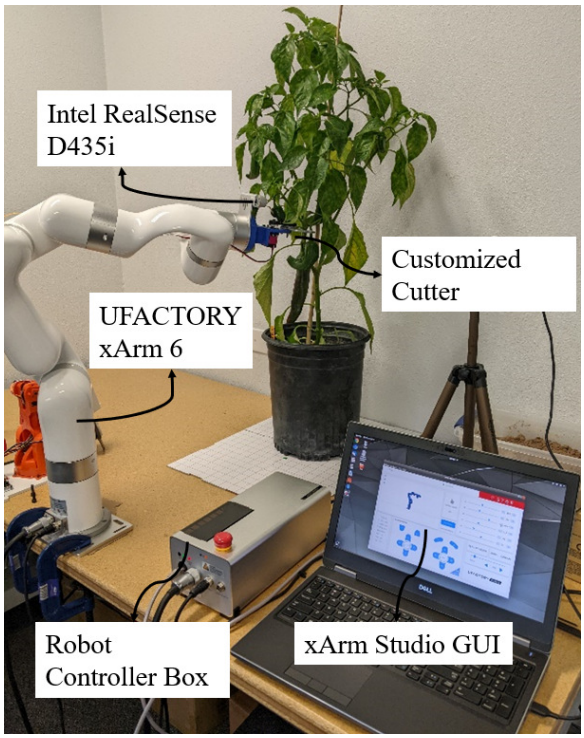

Fig. 2. Experimental setup for automated harvesting of green chile pepper: Eye-in-Hand configuration

III. Fruit Detection and Localization

A. Detection of Green Chile Peppers in the Plant

Detection of the chile is achieved using Mask R-CNN, which outputs a bounding box, mask, and label for each object detected by the network [14]. Mask R-CNN is an extension of the Faster R-CNN architecture of [15], retaining both the region proposal network (RPN) that identifies regions of interest (RoI) in the input image and the Faster R-CNN classifier that generates the bounding box and classification for each object identified by the network. The primary difference between Mask R-CNN and its predecessor is the addition of a fully convolutional network (FCN) that operates in parallel to the Faster R-CNN classifier, taking in the same RoI data generated by the first stage of Faster R-CNN and outputting candidate masks [14]. These candidate masks are used to determine which pixels within the image belong to a chile or peduncle (stem) that has been detected by the network in order to ensure accurate collection of depth data during the localization process.

To train Mask R-CNN, a dataset consisting of 285 hand-annotated (mask and label for each green chile, red chile, and stem) images was produced. These images were selected from a larger dataset consisting of 757 1280×720 pixel RGB images that had been sharpened and received both white balance and gamma corrections. Of the larger dataset, 700 images were produced using an Intel RealSense Depth Camera D435i that photographed 16 different plants in both outdoor and greenhouse settings, varying the distance between the camera and the plant in 6-inch increments in the

Authorized licensed use limited to: UNIVERSIDAD DE GRANADA. Downloaded on January 07,2026 at 16:01:52 UTC from IEEE Xplore. Restrictions apply.

range of $[6, 24]$ inches, varying direction of approach (by the four cardinal directions), varying the angle of the camera relative to the plant in $30°$ increments through the range of $[0°, 60°]$, and varying the time of day (morning, noon, and evening). The remaining 57 images in the larger dataset did not follow the same acquisition methodology to increase variability. The 285 annotated images were divided into a training set (227 images) and validation set (58 images), and augmented using version 0.4.0 of the `imgaug` library [16], producing a total of 3178 training images and 812 validation images. A full list of augmentations implemented can be found in [17].

To determine the optimal training method (specifically the number of epochs, which layers should be frozen during training, and what the loss weight should be) for Mask R-CNN, training was initially performed using the non-augmented dataset. The training was then repeated with the full augmented dataset, retraining from scratch using the training method devised on the non-augmented dataset and altering the parameters and configuration of the network (such as the backbone of the RPN) to determine the optimal configuration. In all training cases, the images were resized to 1024×1024 pixels, using zero padding to maintain the same aspect ratio. For the final network, the RPN was modified to have an "anchor box" (reference boxes that allow the network to identify RoI of different scales and aspect ratios [14]) sizes of $[16, 32, 64, 256, 512]$ pixels, and built using ResNet-101 [18] as the backbone. Mask R-CNN was initialized using the COCO weights and trained using the default parameters of [14], except where otherwise noted. The training was divided into three stages, each lasting 10 epochs. In the first stage, all weights except those in the head (portion of the network responsible for the final classification) were frozen, and the loss weight was reduced to 1×10^{-4}. In the second stage, the ResNet-101 layers C4 and higher were unfrozen, and the loss weight was unchanged. For the final 10 epochs, the loss weight was reduced to 1×10^{-5} and all layers were trained. The final network had a precision of 76.09%, recall of 56.39%, a mean Average Precision (mAP) (evaluated for an intersection over union threshold of 0.5 in the output mask compared to ground truth) of 79.26%, and an F1 score (Dice coefficient) of 64.77% on the augmented validation set. An example of the output of Mask R-CNN, compared to the ground truth bounding box and mask, is presented in Fig. 3.

Fig. 3. Example of chile detection. In this image the green masks and bounding boxes represent the ground truth, the prediction is presented in red, and areas of agreement between the ground truth mask and the predicted masks are shown in orange.

B. Localization of Green Chile Peppers

The Intel RealSense depth camera creates two types of frames during the process of capturing a scene: the color frame and the depth frame. The color frame contains the visual information of the scene in the form of a regular RGB image. On the other hand, the depth frame contains information about the distance (depth) of each point in the scene from the camera.

The Mask R-CNN model is utilized to detect objects in the color frame. The model identifies objects of interest, such as the chile, and provides their specific locations (at the green dot, just after the stem's position represented by the yellow dot, see Fig. 4) in pixel coordinates expressed in the color frame. Now, the depth and color frames are aligned using the `rs.align` function, ensuring that depth values correspond accurately to the detected pixels in the color image.

The camera calibration process establishes the mapping between the captured image's pixel coordinate system and the real-world coordinate system, thereby obtaining the intrinsic parameters of the camera (i.e., focal length and principal points). The camera takes the depth information from the depth frame of each detected chile, along with its intrinsic parameters, to mathematically reverse the projection that occurs during the capture of the image. Thus, it provides the 3D position of the detected chile in the real-world system with respect to the camera.

After testing the localization process for the chiles, even though the z-coordinate of the detected chile was accurate, there was a deviation in the x and y-coordinates measured by the camera compared to the actual physical location of fruits. The dynamic calibration of the camera was performed to correct the lens distortion effect. However, as the issue persisted, an experiment was conducted to address it directly. In this experiment, a single chile was positioned at the center of the image frame and moved from left to right (along the camera's x-axis) and from top to bottom (along the camera's y-axis) as shown in Fig. 4. Throughout these movements, the camera's readings for the chile were recorded. 17 data points were collected for the left-to-right (x-axis) movement, while 9 points were collected for the chile's top-to-bottom (y-axis) movement. The resolution for the camera was set as 1280×720 pixels, which results in a larger number of collected data points in the x-direction than in the y-direction. The experimental procedure is shown in Fig. 4.

The collected data and trend lines for x-coordinate and y-coordinate correction are shown in Fig. 5 and Fig. 6. Analysis of these trend lines reveals a linear relationship between the camera measurement and the actual measurement. Consequently, mathematical correction equations were obtained by a simple linear regression based on the experimental data representing the linear trend between the actual position and

Authorized licensed use limited to: UNIVERSIDAD DE GRANADA. Downloaded on January 07,2026 at 16:01:52 UTC from IEEE Xplore. Restrictions apply.

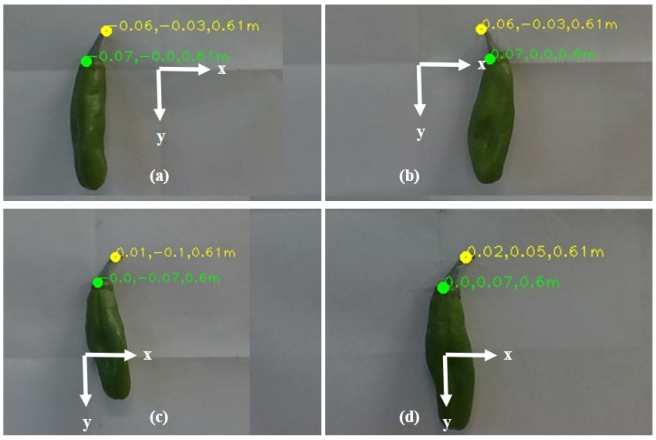

Fig. 4. Experiment for x and y coordinate correction. A single chile is moved 0.05 m from the image center to the (a) left, (b) right, (c) top, and (d) bottom, and the corresponding reading by the camera is shown.

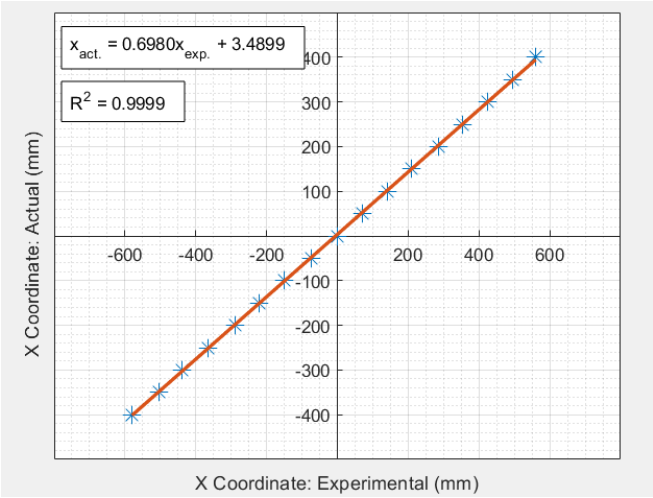

Fig. 5. Actual vs. experimental reading for x-coordinate of the detected chile. (1) is obtained from the trend line.

the camera's readings as follows,

$$x_{act} = 0.6980\, x_{exp} + 3.4899, \tag{1}$$

$$y_{act} = 0.6903\, y_{exp} - 2.3011. \tag{2}$$

where x_{exp} and x_{act} represent the camera reading and the actual reading of the distance for the x-coordinate correction, respectively, and similarly for the y-coordinate in (2).

One possible reason for this linear relationship could be attributed to the fact that the transformation of any point from the color frame to the depth frame is affine, which preserves a linear relation between the original and the transformed point. However, during the localization process, perfect alignment between the depth and color images may not have occurred. This creates a deviation between the actual (x_{act},y_{act}) location of the chile and their corresponding camera measurement (x_{exp}, y_{exp}), as well as the linear relationship between them. Fig. 7(a) and (b) show an example of a localized chile pepper obtained in this process before and after correction, respectively. Based on the results

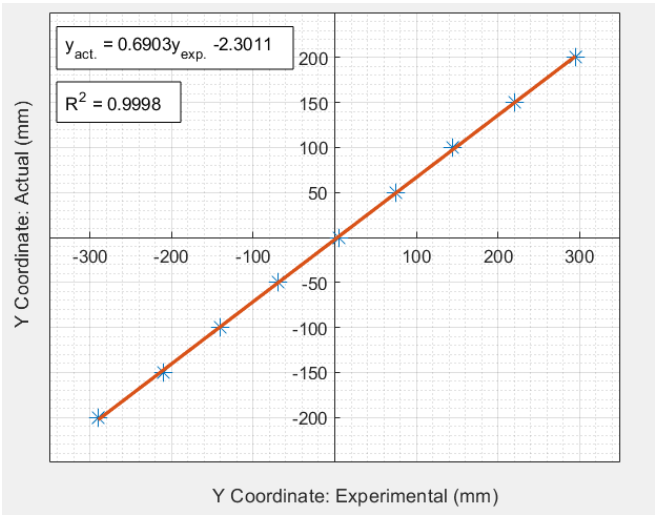

Fig. 6. Actual vs. experimental reading for y-coordinate of the detected chile. (2) is obtained from the trend line.

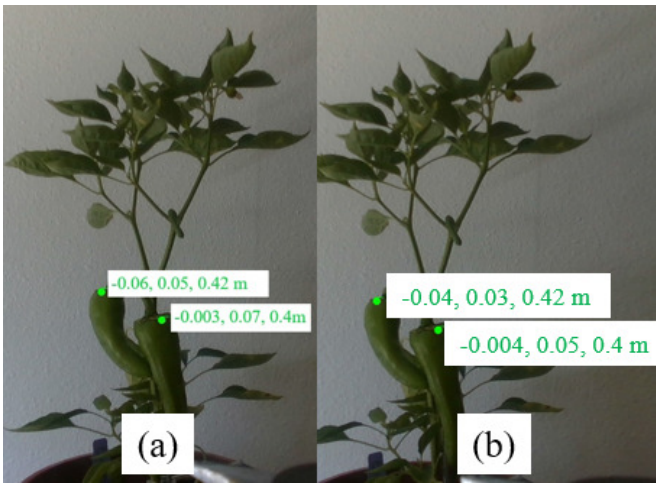

Fig. 7. Localized chile pepper (a) before and (b) after fixing x and y-coordinate error of the camera

of the localization success discussed in the results section, it can be said that the correction equations were consistent for the experiment.

IV. MOTION PLANNING OF THE MANIPULATOR

Task and motion planning are essential parts of robotic harvesting systems. Significant progress has been made regarding task and motion planning strategies for harvesting which are mainly based on coverage path planning to pick all available fruits in a scene or to minimize the required time for moving between fruits [20]. The direct displacement towards the desired position of the end-effector is the most common approach for the path planning of robotic manipulators. This approach was achieved using position-based control and visual feedback control.

A. Coordinate Frame Transformation

Coordinate frames are attached to the main components of the experimental robotic setup as shown in Fig. 8. Here,

Authorized licensed use limited to: UNIVERSIDAD DE GRANADA. Downloaded on January 07,2026 at 16:01:52 UTC from IEEE Xplore. Restrictions apply.

{B}, {C}, {W}, and {T} represent the frame attached to the manipulator's base, the camera frame, the wrist frame originating at the flange center of the manipulator, and the tool frame, respectively. The tool frame takes into account the dimensions of the attached cutter. The 3D location of the chile, obtained in the previous section, is with respect to the camera frame, {C}, defined as ${}^{C}\boldsymbol{p} = [x_{act}\, y_{act}\, z]^T$, where the x and y-coordinates are obtained based on the correcting equations in (1) and (2). The cutter's center point (the origin of the tool frame) needs to reach the chile to start the harvesting. For the manipulator's end-effector (here, the cutter's center position) to reach a certain position, all distances should be measured from the base, frame {B}. Therefore, the location of the chile in the frame {C} is transformed to the wrist frame, {W}, by multiplying the chile's locations in the frame {C} with the rotation matrix of the camera frame with respect to the wrist frame (${}^{W}R_C$) and adding the translation vector ($\boldsymbol{t}_C$). The wrist frame and the tool frame are the same. Because of having a Tool Center Point (TCP) offset (which considers the dimension of the cutter in the motion of the manipulator) in the settings of the robotic arm, by subtracting the distance of the cutter's center position ($\boldsymbol{d}_{\mathrm{T}}$) from the origin of the {W} frame, the chile's location is now obtained in the tool frame (${}^{\mathrm{T}}\boldsymbol{p}$):

$$ {}^{\mathrm{T}}\boldsymbol{p} = {}^{W}R_C\, {}^{C}\boldsymbol{p} + \boldsymbol{t}_C - \boldsymbol{d}_{\mathrm{T}}. \tag{3} $$

From the tool frame, the chile's location is transformed to the base frame, {B}, with a similar process. The Python function `arm_get_position(is_radian=True)` of the XArm6 provides the 3D position (x,y,z), and rotation (roll (α), pitch (β), and yaw (γ) angle) of the cutter's center position with respect to the base frame, {B}. Thus, the 3D rotation matrix ${}^{B}R_{\mathrm{T}}$, which is composed of the rotation by angles α, β, and γ around the X, Y, and Z axes respectively, is formed and the 3D location of the chile with respect to the base (${}^{B}\boldsymbol{p}$) can be obtained so that the cutter center position can reach them to perform the harvesting :

$$ {}^{B}\boldsymbol{p} = {}^{B}R_{\mathrm{T}}\, {}^{\mathrm{T}}\boldsymbol{p} + \boldsymbol{t}_{\mathrm{T}}, \tag{4} $$

where ${}^{B}R_{\mathrm{T}} = R_Z(\gamma)R_Y(\beta)R_X(\alpha)$ and $\boldsymbol{t}_{\mathrm{T}}$ ($[x\, y\, z]^T$) is the translation vector of the tool frame with respect to the base frame. Note that, the order of rotation is based on the X-Y-Z convention and the successive rotations are relative to the fixed frame, so pre-multiplication is done to form the rotation matrix, ${}^{B}R_{\mathrm{T}}$. The sequence of coordinate frame transformations is shown in Fig. 9.

B. Workspace Reachability of the End-Effector

The workspace of a robotic manipulator is the entire volume it can cover by the end-effector while executing every movement that each of its joints can carry out within its limits. When all of the links are in the vertical position, the workspace of the robotic arm measures 1029 mm from the base to the top and 416 mm from the base to the bottom. A circle with a radius of 762 mm is covered by the manipulator when joint-1 moves and all of the links are in the horizontal position [13]. As shown in Fig. 10, a 3D filter is created

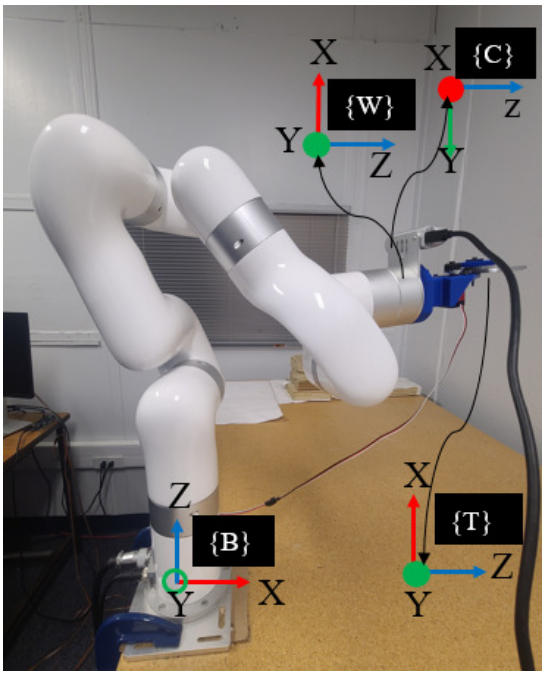

Fig. 8. Various coordinate frames in the experimental setup

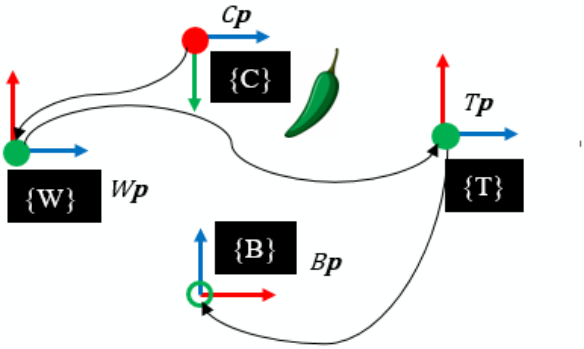

Fig. 9. Sequence of transformation from one coordinate frame to another

based on the arm's workspace to remove any coordinate position of the chile that the manipulator cannot reach. After converting each location of the chile pepper to the base coordinates of the robot, the square root of the total squared x and y coordinates of the pepper should fall within (762+150) mm, and the x-coordinate should always be positive. The z-coordinate should also fall inside the height restriction (1029+150 mm for a positive z). Adjustments are made for the height of the cutter (150 mm), which is connected as the end-effector.

C. Motion Planning

A motion planning algorithm in Cartesian space has been developed to guide the cutter in transitioning from its starting point to each detected chile's location for the harvesting task. The algorithm is based on finding the minimum path between the starting point of the cutter and all detected/localized chile peppers in the plant. We kept the orientation of the tool to be fixed as the roll, pitch, and yaw angle are set

Authorized licensed use limited to: UNIVERSIDAD DE GRANADA. Downloaded on January 07,2026 at 16:01:52 UTC from IEEE Xplore. Restrictions apply.

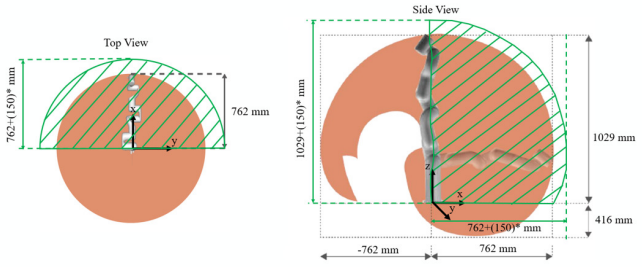

Fig. 10. A 3D filter (green color) to filter out any coordinate outside of its boundary (left: top view, right: side view). 150 mm length is added to the workspace of the robot to accommodate the height of the cutter.

to be $\alpha = 0°$, $\beta = -90°$, and $\gamma = -180°$ during the end-effector movement. The robotic arm end-effector moves with a constant velocity of v along a straight line between any two points in Cartesian space at each step, given n localized green chile pepper, the total time of harvesting, t_{tot}, can be obtained as follows,

$$t_{tot} = \sum_{i=1}^{n} \Delta t_i = \frac{1}{v} \sum_{i=1}^{n} \| {}^{B}\boldsymbol{p}_{To} - {}^{B}\boldsymbol{p}_i \| \tag{5}$$

where, Δt_i is the required time interval at each step, ${}^{B}\boldsymbol{p}_{To}$ is the position vector of the current location of the tool, and ${}^{B}\boldsymbol{p}_i$ is the position vector indicating the location of i^{th} green chile pepper, respectively. A 3D graph comprised of vertices at the green chile peppers' location connecting by straight lines (edges) was established. The sequences of the end-effector's path between chiles was determined based on searching the graph where for any vertex i find the next vertex $j \in \{1, 2, \cdots, n\}$ such that $j \neq i$ and $\min \| {}^{B}\boldsymbol{p}_i - {}^{B}\boldsymbol{p}_j \|$. The cutter subsequently proceeds to the next closest chile and repeats this process until reaching the final one. Finally, the cutter returns to its initial position. This approach reduces the harvesting time of the robot and ensures the shortest travel time. To achieve this, all the detected chile's locations are stored in a list called `chile_position`, and an empty list called `sorted_chile` is created. The cutter's initial position is set as the first element of the `sorted_chile` list. The Euclidean distance between the last-added element of the `sorted_chile` list and all the elements of the `chile_position` is calculated. The element in the `chile_position` that creates the minimum distance from the last-added element in the `sorted_chile` list is named as the `closest_tuple`. This `closest_tuple` is then added to the `sorted_chile` list and removed from the `chile_position` list. The end effector of the manipulator moves based on the sequence of the chile's coordinates sorted in the `sorted_chile` list. In Algorithm 1, the process of motion planning is described. The movement of the cutter (the opening and closing of the blades) is controlled by an Arduino Due. In the Python code of the XArm6, communication occurs with the Arduino, and the end effector waits for 1 second with an open blade position after reaching the targeted chile. Then, it closes the blade to perform the harvesting and moves backward (100 mm) before going to the next one.

Algorithm 1 Motion Planning Algorithm

```
Get the list chile_position
Create an empty list sorted_chile
InitialPosition ← getInitialPositionOfCutter()
sorted_chile.append(InitialPosition)
LengthChile ← length of chile_position
counter ← 0
if counter < LengthChile then
    Create an empty list distances
    for tuple in chile_position do
        distance ← 3Ddistance(sorted_chile[−1],
        tuple)
        distances.append(distance)
    end for
    minDistance ← min(distances)
    closestIndex ← distances.index(minDistance)
    closestTuple ← chile_position[closestIndex]
    sorted_chile.append(closestTuple)
    chile_position.remove(closestTuple)
    counter ← counter + 1
else
    Stop
end if
sorted_chile.remove(InitialPosition)
```

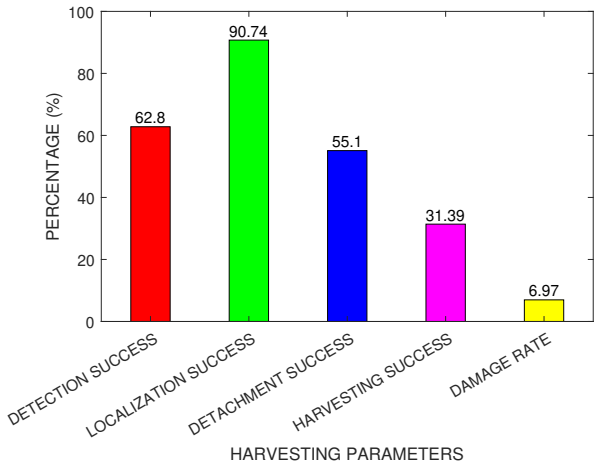

Fig. 11. Harvesting Parameters

V. RESULTS

Of 86 green chiles, 54 were correctly detected in the laboratory test. Then, out of the 54 correctly detected chiles, 49 were correctly localized, and 27 were successfully harvested without damage. A total of 6 fruits were damaged during the harvesting process. The harvesting performance metrics are shown in Fig. 11. The definitions of these metrics that have been adopted from [11], [19] and they are as follows, 1) Fruit detection success (62.8%): The number of successfully detected green fruit per total number of

Authorized licensed use limited to: UNIVERSIDAD DE GRANADA. Downloaded on January 07,2026 at 16:01:52 UTC from IEEE Xplore. Restrictions apply.

green fruit visible from the direction of the camera; 2) Fruit localization success (90.74%): The number of successfully localized fruit per total number of correctly detected fruit; 3) Detachment success (55.1%): The number of successfully harvested fruit per total number of correctly localized fruit; 4) Harvesting success (31.39%): The number of successfully harvested fruit per total number of fruit; 5) Damage rate (6.97%): The number of damaged fruit, caused by the cutter during the harvesting, per total number of fruit; 6) Cycle time (20s): The average time of entire harvest operation, including detection, localization, detachment, and robot transport to the next fruit. While all other performance metrics show improvement, the detachment success rate (55.1%) is slightly lower than the one in our earlier work (65.5%) [11]. Note that in contrast to this work, the fruit detection was entirely done by a human in our previous work, resulting in a 100% accurate detection, whereas, currently, a 62.8% fruit detection success rate was achieved using the Mask R-CNN algorithm developed in this work. Additionally, any error in the camera calibration process can contribute to inaccuracies in the localization success rate. Moreover, the movement of the cutter in the crowded cluster of chiles causes the chile's location to change. Since the localization process is completed before the manipulator moves, even the slightest movement can lead to an inaccurate chile location. This inaccuracy affects the detachment success rate by damaging the fruit if the cutter cuts the chile at any place other than the stem or causes the cutter to miss its precise position entirely. Moreover, due to the detection carried out by a machine learning-based system, which constitutes an additional step compared to the previous study [11], the cycle time exceeds that of the earlier work. Compared to the state-of-the-art harvesting robots [19], the localization success rate exceeds the average of 85%, and the cycle time is shorter than the average of 33 seconds. Although the detachment success rate is below the average of 75%, it still falls within the range of 42% to 93%. However, harvesting success rates fall below the average of 66% and remain outside the range (42% to 93%). Additionally, the damage rate slightly exceeds the average of 5%.

VI. CONCLUSIONS

This paper presents the automatic harvesting of green chile pepper to alleviate the problem of manpower shortage. With the help of machine learning, robotic vision, and a customized cutter, the harvesting process is transformed from manual to mechanized without damaging the fruit. The success rate of detachment is 55%, which shows the promising nature of this research. Nevertheless, this study's drawback is that, from one direction of the plant, all the fruits are detected and localized at one time, and the sequence of the harvesting is determined. However, when the actual harvesting occurs, there can be a small shift in the following chile's placement because of the cutter's action or a small movement in the branches as a result of harvesting the preceding chile. It causes the cutter to miss any chile or damage the fruits while harvesting. If the chile's detection and localization could be completed while the end effector travels rather than all at once, this might be avoided.

REFERENCES

[1] Mountain Regional Field Office of the National Agricultural Statistics Service, USDA; New Mexico Department of Agriculture, "2021 New Mexico Chile Production," 2021. U.S. Department of Agriculture, National Agricultural Statistics Service, Las Cruces, NM, USA.

[2] S. J. Walker and P. A. Funk, "Mechanizing Chile Peppers: Challenges and Advances in Transitioning Harvest of New Mexico's Signature Crop," HortTechnology, vol. 24, no. 3, pp. 281-284, 2014. [Online]. Available: https://doi.org/10.21273/HORTTECH.24.3.281. Accessed: Jan. 9, 2024.

[3] L. Calvin and P. Martin, "The U.S. produce industry and labor: Facing the future in a global economy," U.S. Dept. Agriculture, Econ. Res. Service, Washington, DC, USA, Econ. Res. Rep. No. ERR-106, Nov.2010.

[4] Y. Edan, D. Rogozin, T. Flash, and G. E. Miles, "Robotic melon harvesting," in IEEE Transactions on Robotics and Automation, vol. 16, no. 6, pp. 831-835, Dec. 2000, doi: 10.1109/70.897793.

[5] Q. Feng, W. Zou, P. Fan, C. Zhang, and X. Wang, "Design and test of robotic harvesting system for cherry tomato," International Journal of Agricultural and Biological Engineering, vol. 11, no. 1, pp. 96-100, 2018.

[6] S. Hayashi, K. Ganno, Y. Ishii, and I. Tanaka, "Robotic harvesting system for eggplants," Japan Agricultural Research Quarterly: JARQ, vol. 36, no. 3, pp. 163-168, 2002.

[7] Goulart, R., Jarvis, D., and Walsh, K. B. (2023). "Evaluation of End Effectors for Robotic Harvesting of Mango Fruit," Sustainability, 15(8), 6769.

[8] Li, Y.-R., Lien, W.-Y., Huang, Z.-H., and Chen, C.-T. (2023). "Hybrid Visual Servo Control of a Robotic Manipulator for Cherry Tomato Harvesting," Actuators, 12(6), 253.

[9] Mail, M. F., Maja, J. M., Marshall, M., Cutulle, M., Miller, G., and Barnes, E. (2023). "Agricultural Harvesting Robot Concept Design and System Components: A Review." AgriEngineering, 5(2), 777-800.

[10] J. F. Elfferich, D. Dodou, and C. D. Santina, "Soft Robotic Grippers for Crop Handling or Harvesting: A Review," IEEE Access, vol. 10, pp. 75428-75443, 2022, doi: 10.1109/ACCESS.2022.3190863.

[11] M. U. Masood and M. Haghshenas-Jaryani, "A Study on the Feasibility of Robotic Harvesting for Chile Pepper," Robotics, vol. 10, no. 3, p. 94, 2021. [Online]. Available: https://doi.org/10.3390/robotics10030094.

[12] S. C. Hespeler, H. Nemati, and E. Dehghan-Niri, "Non-destructive thermal imaging for object detection via advanced deep learning for robotic inspection and harvesting of chili peppers," Artificial Intelligence in Agriculture, vol. 5, pp. 102-117, 2021.

[13] UFACTORY XARM USER MANUAL, Shenzhen UFACTORY CO. Ltd, Shenzhen, China, pp. 29–30.

[14] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322.

[15] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017, doi: 10.1109/TPAMI.2016.2577031.

[16] A. B. Jung, K Wada, J. Crall, S. Tanaka, J. Graving, C. Reinders, S. Yadav, J. Banerjee, G. Vecsei, A. Kraft, Z. Rui, J. Borovec, C. Vallentin, S. Zhydenko, K. Pfeiffer, B. Cook, I. Fernández, F. De Rainville, C. Weng, A. Ayala-Acevedo, R. Meudec, M. Laporte, and others, "imgaug", [Online]. Available: `https://github.com/aleju/imgaug`, Accessed 01-Feb-2020

[17] L. Garcia, "Automated detection of green chile from RGB images using deep learning," Master's thesis, New Mexico State University, 2022.

[18] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition,"Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.

[19] Bac, C. Wouter, Eldert J. Van Henten, Jochen Hemming, and Yael Edan, "Harvesting robots for high-value crops: State-of-the-art review and challenges ahead," Journal of Field Robotics, vol. 31, no. 6, pp. 888-911, 2014.

[20] Rajendran, V., Debnath, B., Mghames, S., Mandil, W., Parsa, S., Parsons, S. et al. (2023) Towards autonomous selective harvesting: a review of robot perception, robot design, motion planning and control. Journal of Field Robotics, 1–33. https://doi.org/10.1002/rob.22230

Authorized licensed use limited to: UNIVERSIDAD DE GRANADA. Downloaded on January 07,2026 at 16:01:52 UTC from IEEE Xplore. Restrictions apply.