Algorithm for determining the characteristics of a tomato product

Rustam Baratov[1], Almardon Mustafoqulov[1, a)], Bekzod Akhatov[2], Said Usmonov[2] and Behruz Mallayev[2]

[1]*"Tashkent Institute of Irrigation and Agricultural Mechanization Engineers" National Research University, Tashkent, Uzbekistan*
[2]*Termez State University of Engineering and Agrotechnology, Termez, Uzbekistan*

[a)] *Corresponding author: mustafoali777@gmail.com*

Abstract. In this study, algorithms for determining the ripeness of tomatoes were analyzed and evaluated. The primary goal was to develop an accurate and efficient method to assess ripeness based on visual and sensor data. Image processing techniques using RGB and HSV color models were applied to detect color changes associated with ripening. Additionally, machine learning algorithms were employed to classify ripeness levels based on features extracted from images. The results showed that machine learning models, particularly convolutional neural networks (CNN), achieved a ripeness detection accuracy of over 90%. This research contributes to the development of automated tomato harvesting systems and can be applied in agricultural robotics to improve efficiency and reduce manual labor.

INTRODUCTION

It is known that the amount of income of a modern farm directly depends on the labor force. Even for harvesting ripe crops, a large number of workers are needed, but it is beneficial to use robotic manipulators instead of people to perform simple tasks and to improve the performance of existing ones. It is also known from previous published literature that the US agricultural labor force has decreased by 2% due to robotic manipulators in agricultural fields. For example, American professor Josh Lessing has been working on solving this problem since he founded the Root AI project on agricultural robotics in 2018 [1][2]. Furthermore, the apple industry relies heavily on manual labor. For example, in the United States alone, the seasonal labor required to pick apples accounts for more than 10 million man-hours each year, accounting for approximately 15% of total production costs [3].

In recent years, the role of robotic machines in harvesting agricultural products has also been increasing including sweet pepper[4], strawberry [5], apple[6] and and kiwifruit[7], [8]. Similar systems have also been used in agriculture to eradicate invasive plants [9]. The designs of automated tomato harvesting systems can be divided into two categories. The first category is shake and grab, where vibrations are applied to the tree trunk and/or branches to separate the tomato crop. While shake and grab systems are effective in separating the fruit from the tree, they often result in a rate of tomato bruising that is unacceptable for the fresh market. The other category is fruit-to-fruit harvesting, where manipulators are used to collect the fruit in a controlled manner, thereby significantly reducing tomato damage. However, the design of such systems with high harvesting efficiency and practical viability presents a great challenge. Previous scientific studies and published articles have also mentioned that one of the problems encountered in harvesting tomatoes is their harvest time [10][11]. In addition, the types of robotic manipulators and the main problems they face are also discussed separately [12].

MATERIALS AND METHODS

Today, robots and agricultural machinery (robotics) are developing rapidly in Uzbekistan. However, their existing problems are not being systematically studied. In our previous articles and analyses, we have often touched

upon the problems of angular displacements of their rotating parts. And we have proposed, at least partially, solutions [11][1][12][13, 14]. There are many recommended methods and algorithms for determining the characteristics and condition of agricultural products. In the following paper [14], due to the increasing labor shortage and labor costs in the apple industry, an algorithm for developing robotic systems for efficient and autonomous apple picking is presented. This paper presents a systematic overview and algorithm design of a recently developed prototype of a robotic apple picking machine. This robotic system is realized through the close integration of several core modules, including visual perception, planning, and control. This paper covers the main methods and achievements of deep learning-based multi-view fruit detection and localization, unified picking and unloading planning, and picking and unloading manipulation. This brings a significant improvement over other apple picking robots, with an apple picking speed in the range of 7-10 seconds. The prototype in Figure 1 shows promising performance for further development of efficient and automated apple picking technology. Figure 2 shows the basic algorithm flow of the software system in the apple picking cycle[15].

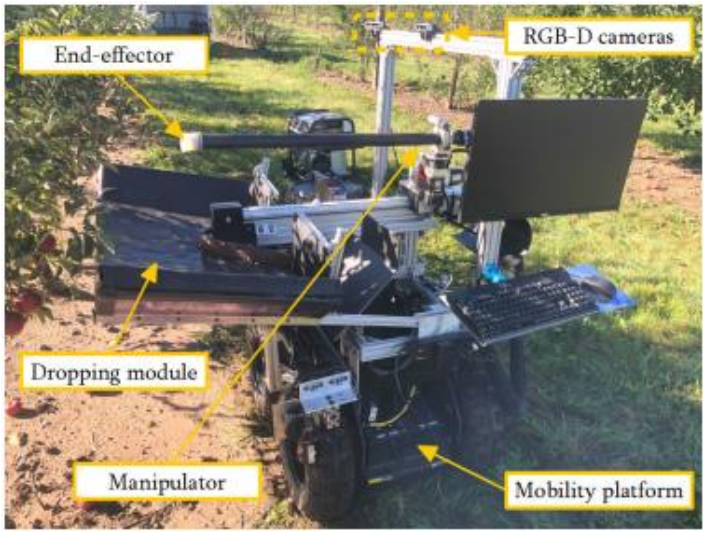

FIGURE 1. The developed robotic apple harvesting prototype

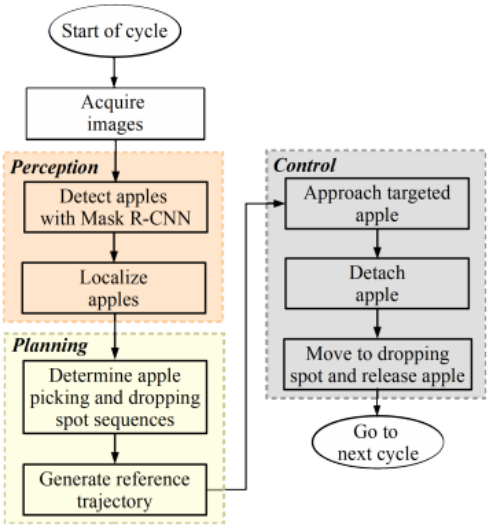

FIGURE 2. Algorithm flowchart in an apple harvesting cycle

It is evident that the software design of the robotic system requires multidisciplinary advances to ensure the coordination of various synergistic functions and achieve reliable automated apple harvesting. Also, the proposed algorithm is considered complex, which leads to errors in the identification of product characteristics.

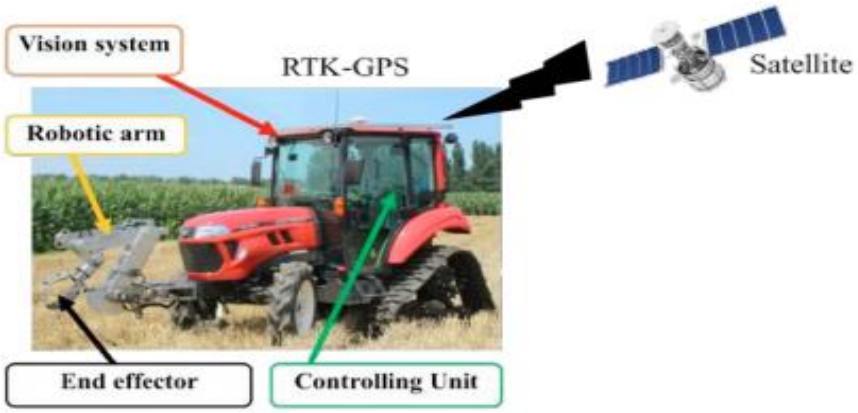

FIGURE 3. Harvesting robot for heavy-weight crops

Another reviewed literature includes scientific studies on determining the location of heavy agricultural products[16]. In Japan, the development of agricultural robot systems has also been promoted to prevent serious

problems such as labor shortages in agriculture. This paper presents the characteristics of a pumpkin picking robot. This robot system consists of a robotic arm, an end effector, and a control algorithm installed on a robotic tractor (Figure 3). The objectives of this study were to develop a control algorithm that can (1) accurately approach the position of the pumpkin, (2) accurately grasp the pumpkin without damaging it, and (3) transport the pumpkin to the truck. The Denewit-Hartenberg methodology was used for direct and inverse kinematics. Two types of experiments were conducted in the laboratory to determine the accuracy and repeatability of the system. The results showed that the positioning accuracy and repeatability of the developed system in the x, y, and z directions were 4.85, 4.15, and 5.23 mm. The positioning error of the end effector was almost 5 mm, respectively.

This agricultural harvester robot consists of a robot tractor, a robot arm, an end-effector, a positioning system, a vision system, and a control unit as shown in Figure 3. The robot arm and the end-effector are controlled by the control unit and powered by 200ACV.

The demand for electric power sources is increasing over time, as electric actuators such as servo motors are more precise and it is easier to design a robotic system based on them. The robotic arm and end effector, as shown in Figure 4 and Figure 5, were designed and manufactured by Roshanianfard (2018a).

FIGURE 4. The robotic arm and end-effector

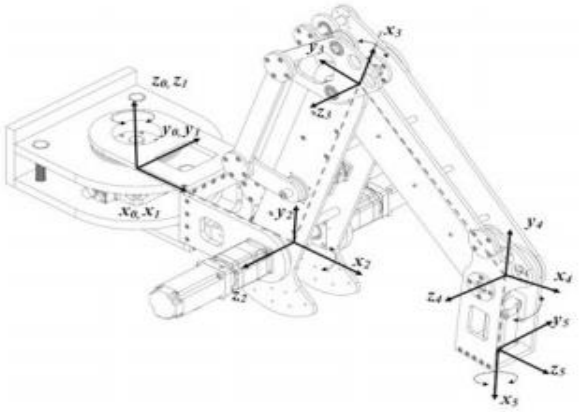

FIGURE 5. The open diagram of 5 joints robotic arm (Roshanianfard and Noguchi, 2018 b)

In this study, the robot arm control algorithm is designed to meet the following requirements:

1) Move to the pumpkin position accurately.
2) Hold the pumpkin correctly.
3) Transfer it to a box or container.
4) Move the robot arm with minimal vibration and maximum efficiency.

Designing an algorithm that can implement this requirement is the main objective of this study. Considering them and the harvesting situation, the point-to-point movement is suitable for the algorithm. Forward and inverse kinematics are used to obtain joint parameters such as joint angle, velocity, and acceleration time[16].

This text emphasizes the need to perform kinematic calculations during the development of the control algorithm. Initially, the coordinates of the object are specified in the form (P_x, P_y, P_z). T_{Trans} (x, y, z) is a translation matrix in three-dimensional space, used to move an object a certain distance in a coordinate system. R_{Rot} (k, θ) is a three-dimensional rotation matrix used to rotate an object about a given axis by an angle θ.

The direction and angle parameters of the axes are set as shown in Figure 5. Motors 1 to 4 are used to control the arm, and motor 5 is used to perform the final movement. The Denevit-Hartenberg (D-H) parameter is given in Table 1. The payload of this robotic arm is designed for almost 200 N with a safety factor of 2.

TABLE 1. The Denevit-Hartenberg (D-H) parameter.

Joint	Twist Angle, (α_{i-1})	Link Length, (a)	Link Offset, (d)	Joint Angle, θ_i
1	90^0	L_1	0	$-105^0 < \theta_1 < 105^0$
2	0^0	L_2	0	$0^0 < \theta_2 < 125^0$
3	0^0	L_3	0	$-130^0 < \theta_1 < -10^0$
4	-90^0	L_4	0	$-105^0 < \theta_1 < 0^0$
5	0^0	L_5	0	$0^0 < \theta_1 < 360^0$

This table describes the motion parameters of a robotic manipulator or mechanical system. The distance between $link_i$ and $link_{i+1}$ (1, 2, 3, 4) is expressed as L_i and the angle of $link_i$ is expressed as θ_i.

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = T_{Trans}(0, -L_4, 0) \bullet R_{Rot}(z, \theta_4) \bullet T_{Trans}(0, -L_3, 0) \bullet R_{Rot}(z, \theta_3) \bullet T_{Trans}(0, -L_2, 0) \bullet R_{Rot}(z, \theta_2) \bullet T_{Trans}(0, -L_1, 0) \bullet R_{Rot}(z, \theta_1) \bullet \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (1)$$

Here, the expressions for A, B, and C are equal to:

$$P_x = Cos\theta_1 \cdot [L_1 + L_2 \cdot Cos\theta_2 + L_3 \cdot Cos(\theta_2 + \theta_3) + L_4 \cdot Cos(\theta_2 + \theta_3 + \theta_4)] \quad (2)$$

$$P_y = Sin\theta_1 \cdot [L_1 + L_2 \cdot Cos\theta_2 + L_3 \cdot Cos(\theta_2 + \theta_3) + L_4 \cdot Cos(\theta_2 + \theta_3 + \theta_4)] \quad (3)$$

$$P_z = L_2 \cdot Sin\theta_2 + L_3 \cdot Sin(\theta_2 + \theta_3) + L_4 \cdot Sin(\theta_2 + \theta_3 + \theta_4) \quad (4)$$

However, the main drawback of this system is that it focuses on determining the location of agricultural products, meaning that attention is not paid to determining their characteristics (degree of ripeness), because ripe and unripe pumpkins are almost the same.

Another study, as mentioned above, was analyzed to reduce physical labor and reduce damage to fruits and vegetables during harvesting. In this study, YOLOV5 was used to validate the position relationship between tomatoes and peduncles, taking into account the changing conditions such as light changes, interference from branches and leaves. The average single-frame image recognition time is 104 ms, which meets the real-time requirements of automatic picking. Compared with the SSD algorithm, it has obvious disadvantages.

This analysis compares the performance of the YOLOv5 and SSD target detection models. The YOLOv5 model was tested on tomato and stem-like parts, and the results are shown in Figure 6. Detection tests were also performed with the SSD model, but it did not perform well on peduncles (stems). According to formula (5), the box (x, y, w, h) is extracted into a one-dimensional vector corresponding to it[16].

$$\begin{cases} vector < ROI_{stem} > \leftarrow (x, y, w, h)\ obj_{id} = 0 \\ vector < ROI_{tomato} > \leftarrow (x, y, w, h)\ obj_{id} = 1 \end{cases} \quad (5)$$

FIGURE 6. YOLO v5 (left) and SSD (right) quickly recognizes tomatoes and peduncles.

It is clear that this method is also intended to determine the location of tomatoes and is not sufficiently illuminated to determine their degree of ripeness [17].

Now we will analyze the recommended algorithm and its operation process (Figure 7):

- The process begins (Start): The process begins, preparing for initial data entry.
- Enter initial data: The robot or system is given the necessary initial parameters, such as data from sensors (such as a camera) or predefined boundaries.
- Determining the coordinates of an object: The location of the tomato is determined. This is done using a camera or other sensors, i.e. its x, y position or its position relative to an obstacle (tomato stem, leaves, etc.).
- Perceiving an object and determining its characteristics: The ripeness, shape, size, and other characteristics of the tomato are determined.
- Determining the optimal trajectory: It is performed to find the optimal path (trajectory) to catch the tomato. Kinematic equations or artificial intelligence algorithms are used to plan the trajectory.
- Actuator (Angle displacement sensor): The robot manipulator or gripper performs the desired movement. The angular displacement sensor controls the precise movement of the robot.

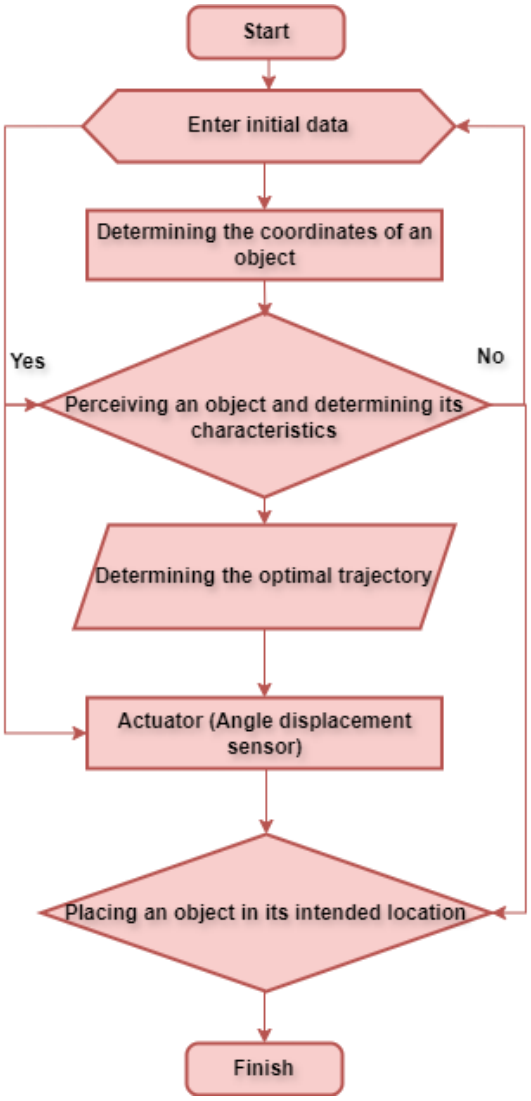

FIGURE 7. Algorithm flowchart in tomatoes harvesting

In this process, images captured by the camera are divided into matrices, such as the sequence shown in the image below (Figure 8). In the process of determining whether a tomato is ripe, each pixel in the image is represented as a matrix. In MATLAB, we use the color values of the pixels by analyzing the image in RGB channels. Below, we explain this mathematically in terms of x, y. Here, the system basically divides tomato products into 3 types (Well-ripened, medium-ripened, and un ripened (raw)).

- Representing an Image in Matrix Form: The RGB channels of an image are represented in matrices:

R(x, y): The value of the red channel in the image.
G(x, y): The value of the green channel in the image.
B(x, y): The value of the blue channel in the image.
Each pixel in the image is specified using coordinates (x, y).
Color conditions:
To determine the ripeness of a tomato, we use the following condition:

$$\text{Mask}(x, y)=\begin{cases}\text{If } R(x,y) > G(x,y) \text{ and } R(x,y) > B(x,y) \text{ and } R(x,y) > T, 1 \\ \text{otherwise}, 0\end{cases} \quad (6)$$

Here: T is the threshold value of red.

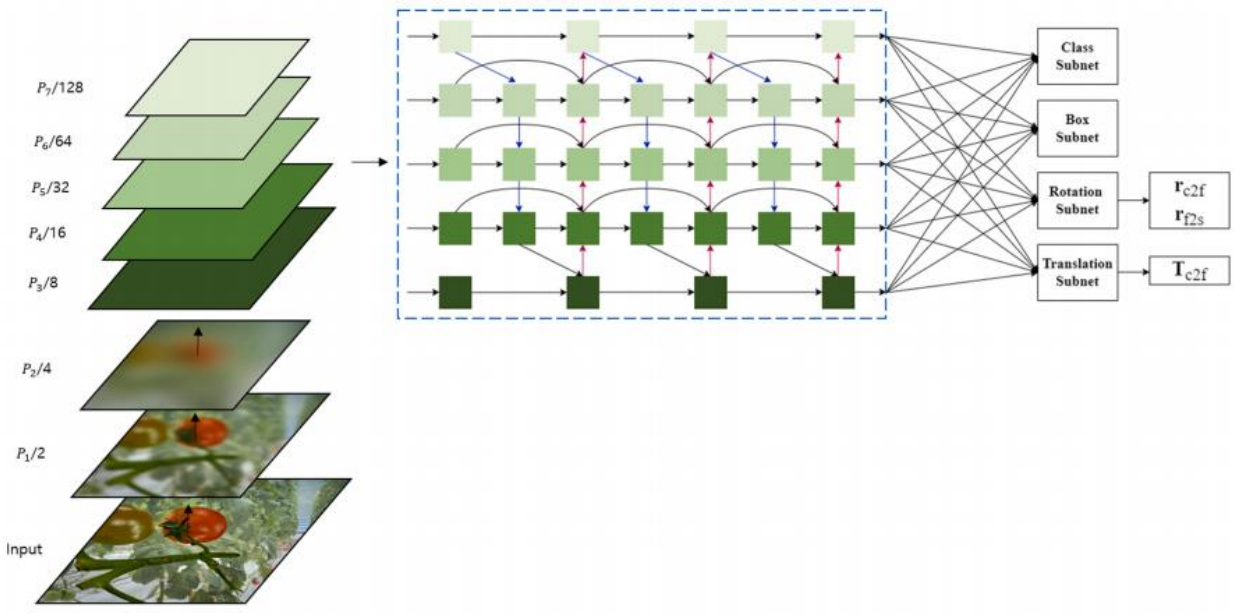

FIGURE 8. Sequence of dividing the resulting image into matrices

Separating Ripe Tomatoes

Separating ripe tomato parts using a separating mask:

$$Result\ (x, y)=Image(x, y)Mask\ (x, y) \quad (7)$$

where: Image (x, y) is the original image matrix, Mask (x, y) is the logical matrix obtained as a result of the conditions defined for the ripe parts.

EXPERIMENTAL RESEARCH

To determine the ripeness of tomato products, they are divided into matrices in the "MATLAB Editor" and the percentage of ripeness is determined using the following expression.

First, an original image is taken from a greenhouse (or possibly a field) where the tomato crop is grown.

FIGURE 9. The original appearance of the image. The image is separated into RGB channels and its size is determined.

$$P = \frac{\sum_{x,y} Mask(x,y)}{M \cdot N} \cdot 100 \quad (8)$$

here M and N are the dimensions of the image matrix.

The matrix extracts the characteristics of the tomato product from the original image (divides it into Well-Ripe, Not-so-Ripe, and Ripe (Raw) tomatoes). In this case, the intensity value is determined based on the value of the matrix, depending on the size of the image and the color of the tomato.

FIGURE 10. Processing tomato images according to ripeness levels

In this case, the intensity value is determined based on the value of the matrix, depending on the size of the image and the color of the tomato. A histogram is constructed based on the intensity values of the specified image.

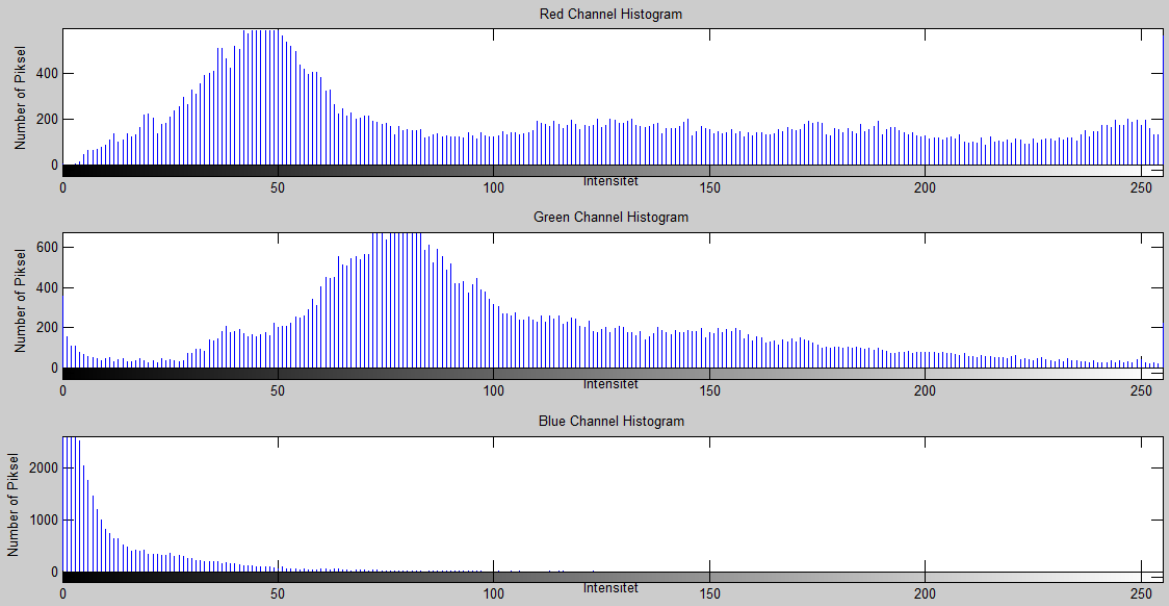

FIGURE 11. RGB channel histograms of a tomato image

Based on the constructed matrix and histogram, the number of pixels of red tomatoes is the largest around 50 pixels, the pixels of green tomatoes are the largest around 70 pixels, and the pixels of intensity are the highest around zero. We will test the above theoretical analyses and algorithm using a special program and determine their errors. The performance of the developed algorithm was tested in the teachable machine program. To build the model's database, images of three different types of tomatoes are loaded into different files. Here, a continuous stream of images is captured from the camera and compared with existing images in the database.

This table shows the Confusion Matrix, which describes the results of model testing to determine the characteristics of a tomato product (ripe, raw, and damaged) using a Teachable Machine and a built-in algorithm.

TABLE 2. Model testing to determine the characteristics.

	Type of Tomatos	Predection		
Class	Unriped Tomatos	45	0	1
	Unriped Tomatos	0	48	2
	Dameged Tomatos	0	1	43
		Unriped Tomatos	Unriped Tomatos	Dameged Tomatos

The results show that out of 46 unripe tomatoes taken during the re-sampling process of the same image, it correctly identified 45 and mistakenly identified 1 as damaged. Out of 50 ripe tomatoes, it correctly identified 48 and mistakenly identified 2 as damaged, and out of 44 damaged images, it correctly identified 43 and mistakenly identified 1 as ripe.

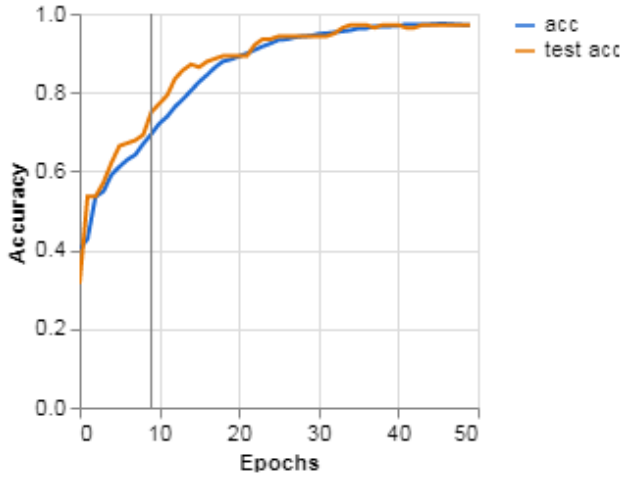

FIGURE 12. Evaluating the accuracy of an algorithm in a "teachable machine" program

This graph illustrates how the accuracy indicator changed during the training process of a model based on the algorithm developed for processing the captured images. Here, the blue line is theoretical data, while the red lines represent experimental results based on the algorithm. Y-axis (Accuracy): The probability of the model making a correct prediction (from 0 to 1 or 0% to 100%). X-axis (Epochs): The number of training cycles (50 epochs are shown here).

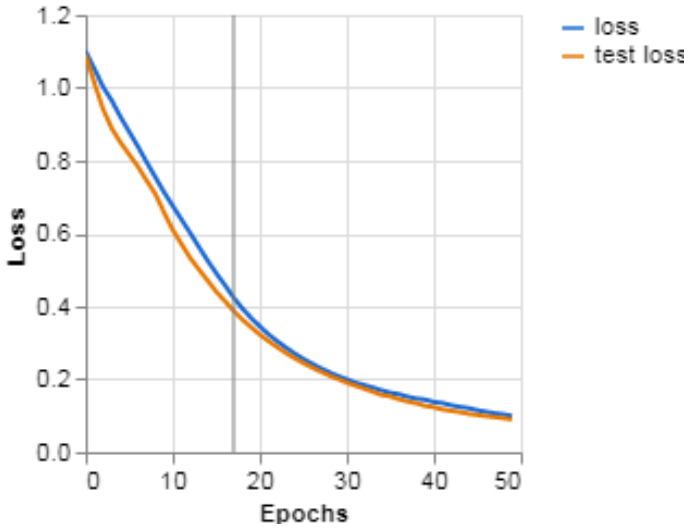

FIGURE 13. Error assessment of the developed algorithm (model)

The loss function is very high (about 1.2) and is decreasing rapidly. The difference between the training and test losses is small, which indicates that the model fits well. The loss value continues to decrease more slowly but steadily. The training and test lines are very close to each other, which indicates that there is no overfitting problem. The loss value is almost approaching 0, which indicates that the model is performing with high accuracy. The training and test losses are almost equal, which indicates that the model has good generalization ability.

CONCLUSIONS

The following conclusions were drawn from the theoretical and experimental analyses conducted:

- The model is highly accurate, as the numbers in the diagonal cells (those correctly identified) are much larger. Only in some cases did the model misidentify ripe and damaged tomatoes (Table 2).
- The model was well trained and did not suffer from overfitting.
- Stable and high accuracy was achieved.

- The model can be considered reliable enough to be used in a real-world application environment (Figure 10).
- As can be seen from the graph, the model achieved stable and reliable results during the training process.
- No signs of overfitting or underfitting were encountered.
- The model also performs well on test samples, meaning that the model can be used on real data (Figure 11).

REFERENCES

1. R. Baratov and A. Mustafoqulov. “Smart angular displacement sensor for agricultural field robot manipulators,” E3S Web Conf., vol. 386, May 2023, doi: 10.1051/E3SCONF/202338603008.
2. “Root AI raises seed funding for Virgo robot to harvest multiple crops.” https://www.therobotreport.com/root-ai-raises-seed-funding-for-virgo-robot-designed-to-harvest-multiple-crops/ (accessed May 02, 2022).
3. K. Zhang, K. Lammers, P. Chu, N. Dickinson, Z. Li, and R. Lu, “Algorithm Design and Integration for a Robotic Apple Harvesting System,” IEEE Int. Conf. Intell. Robot. Syst., vol. 2022-October, pp. 9217–9224, 2022, doi: 10.1109/IROS47612.2022.9981417.
4. Z. Tian, X. Guo, W. Ma, and X. Xue, “Research on kiwifruit harvesting robot worldwide: A solution for sustainable development of kiwifruit industry,” Smart Agric. Technol., vol. 10, p. 100792, Mar. 2025, doi: 10.1016/J.ATECH.2025.100792.
5. Y. Xiong, Y. Ge, L. Grimstad, and P. J. From, “An autonomous strawberry-harvesting robot: Design, development, integration, and field evaluation,” J. F. Robot., vol. 37, no. 2, pp. 202–224, Mar. 2020, doi: 10.1002/ROB.21889.
6. S. P. Galinato, “2014 COST ESTIMATES OF ESTABLISHING, PRODUCING, AND PACKING RED DELICIOUS APPLES IN WASHINGTON”.
7. C. Lehnert, A. English, C. McCool, A. W. Tow, and T. Perez, “Autonomous Sweet Pepper Harvesting for Protected Cropping Systems,” IEEE Robot. Autom. Lett., vol. 2, no. 2, pp. 872–879, Apr. 2017, doi: 10.1109/LRA.2017.2655622.
8. “Research on kiwifruit harvesting robot worldwide: A solution for sustainable development of kiwifruit industry - ScienceDirect.” https://www.sciencedirect.com/science/article/pii/S2772375525000267 (accessed Feb. 09, 2025).
9. A. M. Denmukhammadiev, O. Q. Matchanov, E. E. Sobirov, and S. S. Azamov, “Energy-efficient environmentally friendly electric device for the destruction of weeds by heating them,” IOP Conf. Ser. Earth Environ. Sci., vol. 1076, no. 1, 2022, doi: 10.1088/1755-1315/1076/1/012036.
10. R. Baratov, M. Begmatov, A. Mustafoqulov, F. Kucharov, and E. Sabirov, “Study on the methods of measuring power of the rotating mechanisms,” E3S Web Conf., vol. 434, p. 01014, Oct. 2023, doi: 10.1051/E3SCONF/202343401014.
11. R. Baratov and A. Mustafoqulov, “Model of field robot manipulators and sensor for measuring angular displacement of its rotating parts”, doi: 10.1051/e3sconf/202340104006.
12. A. Mustafoqulov, R. Baratov, Z. Radjapov, S. Kadirov, and B. Urinov, “Angular displacement measurement and control sensors of agricultural robot-manipulators,” BIO Web Conf., vol. 105, Apr. 2024, doi: 10.1051/BIOCONF/202410503003.
13. R. Baratov, H. Sunnatillayeva, and A. M. Mustafoqulov, “SMART SYSTEM FOR WHEAT DISEASES EARLY DETECTION,” Chem. Technol. Control Manag., vol. 2023, no. 6, pp. 38–43, Dec. 2023, doi: 10.59048/2181-1105.1509.
14. R. Baratov, H. Valixonova and A. M. Mustafoqulov. “A smart system for de stem for detecting an ting anatomic and omic and histological sympt ogical symptoms of diseased a oms of diseased agricultural plants”. Chemical Technology, Control and Management, Tashkent. Accepted 30 June 2025; Available online 30 July 2025. https://ijctcm.researchcommons.org/journal/.
15. K. Zhang, K. Lammers, P. Chu, N. Dickinson, Z. Li, and R. Lu, “Algorithm Design and Integration for a Robotic Apple Harvesting System,” IEEE Int. Conf. Intell. Robot. Syst., vol. 2022-Octob, pp. 9217–9224, 2022, doi: 10.1109/IROS47612.2022.9981417.
16. T. Kamata, A. Roshanianfard, and N. Noguchi, “Heavy-weight Crop Harvesting Robot - Controlling Algorithm,” IFAC-PapersOnLine, vol. 51, no. 17, pp. 244–249, 2018, doi: 10.1016/j.ifacol.2018.08.165.
17. G. Zhaoxin, L. Han, Z. Zhijiang, and P. Libo, “Design a Robot System for Tomato Picking Based on YOLO v5,” IFAC-PapersOnLine, vol. 55, no. 3, pp. 166–171, 2022, doi: 10.1016/j.ifacol.2022.05.029.