

Real-time Implementation of Image Based PLC Control for a Robotic Platform

K. K. AYTEN and O. KURNAZ

Abstract—In this study, image based real-time control of a linear robotic platform was performed. This robotic platform is used to determine the location of the mushroom and to direct the linear platform to the detected position in real time with PLC control. Haar-Cascade classifier was used to detect mushroom position and Visual Studio C # .NET platform was used to test the Cascade classifier and write other evaluation codes. One of the most important outputs of this work is to determine the actual position in the global coordinate from the pixel-based location of the object in the image using an ordinary USB camera or built-in camera. Calibration technique was used for this determination.

Index Terms—Image processing, Object tracking, Trajectory control, Automation, Haar-Cascade Classifier, Camera calibration, PLC control.

I. INTRODUCTION

AUTONOMOUS SYSTEMS, which have emerged as a result of the developments in technology, have become applications that are used in many areas like production, agriculture, construction and health. The concept of Industry 1.0 has taken its place in the literature with the invention of steam machines and then the concept of Industry 2.0 has emerged with the use of electricity in production. With the combination of electronic devices and software technologies, production systems have become autonomous day by day and at the end of this period, Industry 3.0 has begun. After the concept of Industry 3.0, the technology has progressed very rapidly and concepts such as cloud technologies, internet of objects, cyber physical systems have started to be discussed more. As a result of this, Industry 4.0 concept which has started to enter into every area of our life in a fast way has emerged. Hence, the production quality is steadily increasing,

KAĞAN KORAY AYTEN, is with Department of Electrical Engineering University of Erzurum Technical University, Erzurum, Turkey, (e-mail: kagan.koray@erzurum.edu.tr).

 <https://orcid.org/0000-0002-8933-612X>

OĞUZHAN KURNAZ, is with Department of Mechatronics Engineering University of Bursa Technical University, (e-mail: oguzhan.kurnaz@btu.edu.tr).

 <https://orcid.org/0000-0002-4595-8031>

Manuscript received November 24, 2018; accepted January 30, 2019.
DOI: [10.17694/bajece.487212](https://doi.org/10.17694/bajece.487212)

and many faulty production details that people cannot realize in production stages are easily defined by autonomous systems. The agricultural sector is one of the areas where the autonomous production concept is widely used. In this sector, the production quality varies depending on the differences in the growing conditions of the products during the production stages. Considering the growing conditions, one of the products that require a difficult process of production is the mushroom [1].

In the literature, there are many technology-related studies on the aforementioned agricultural products. For example, in [2], described a new method in order to separate tomato pictures from each other by classifying them according to their maturity. In another study [3], necessary image processing techniques were applied after the images of hazelnut fruit were obtained from the camera. Then, the size and area data of the hazelnuts were calculated with the help of these techniques and these data were evaluated and divided into 3 classes: small, medium and large. In [4], introduced a mobile robot developed in order to ensure the quality control of agricultural products. In this project, Aforge.Net library is used with C # interface and the robot has 2 cameras. The first of these helps to display the path followed by the robot during the movement towards the location of the product. The other camera provides a better viewing and crop quality control by applying the necessary filtering processes for the detected product. In [5], described the application of the visual feedback control system in the robots used for harvesting. The movements of the fruits were analyzed in the image plane and along the optical axis of the camera. In [6], described a harvest robot with an image-based architecture designed for product detection and collection. In [7], discussed the design and simultaneous simulation problems for tomato harvest robots. In [8], have implemented a real-time product matching application with high accuracy. In [9], described fresh tomato harvesting robot design and testing works. In their study, tomatoes, which are determined grown up were collected by a mobile moving platform and a robot manipulator on it. In [10], mentioned the image processing methods to perform autonomous harvesting of the products. The mushroom was used as a sample product. The identification of the mushroom images taken from the image systems is done with the Support Vector Machine (SVM). After taking the necessary morphological processes on the mushroom picture, the feature extraction and evaluation problem of the mushroom growing is performed with SIFT. In the final stage, the mushroom evaluated is classified by the

SVM method. Apart from agricultural applications, there are other areas where image processing techniques are applied. In [11], described the simulation of the movement of robotic arms to movement of human arm by real-time image processing. In another study [12], suggestions for the solution of the problem of human detection, which may be needed in studies using intelligent systems, surveillance applications and advanced robots are presented and the results are evaluated with numerical data. In [13], Pfunder system (person finder) is a real-time system that enables people to detect and monitor their behaviour.

The motivations we get from the studies and the results, in this study, the design and implementation of an image based linear robotics platform was made. In this study, the robotic mushroom picking system was used to determine the location of the product in the case and to direct the linear actuator to the position where the product was determined. Image processing methods will be used to detect mushroom. The determination of the product as the mushroom is carried out by Haar-Cascade classifier which is one of the image processing object detection methods. Visual Studio C # .NET platform was used to test the Cascade classifier and write other evaluation codes. With the classifier used, the product will be detected and the mushroom will be monitored from the camera. While tracking the mushroom, the position information of the mushroom followed up at the same time will be sent to the PLC simultaneously with the help of camera calibration method. These position information sent to the PLC will be evaluated and the coordinates obtained as a result of this evaluation will be sent to the stepper motor drivers by PLC. At the last part of this, with the help of motors, drivers and encoders ensured that the linear actuator is transferred to the position in this coordinate.

II. MATERIALS AND METHODS

Robotic platform is an autonomous system that enables the detection of mushroom and location information. In this system, the identification of the mushroom and the location of it is realized by means of image processing techniques. Haar-Cascade classifier was used to determine the location of the mushroom. The XML file will then be created for use in the classifier training and software codes. Then the camera will be calibrated and the position information of the mushroom will be identified. The information received will then be sent to the PLC device as position information and the PLC will provide to drive the motors to the position where the mushroom is located.

A. Haar-Cascade Classifier

The Haar-Cascade Classifier is a classifier for object detection. This algorithm; The Haar features are composed of sections such as Integral Image, AdaBoost learning algorithm and Cascade classifier [14]. In this study, all steps of the classifier will be applied to determine the mushroom object. For object detection, the algorithm must be trained first. For this training, there is a need for a set of positive images that contain the object to be detected and a set of negative images without an object. First, the pictures of the mushroom object

are recorded as positive pictures. Afterwards, in these positive pictures, the areas where the mushroom object is located are introduced into the quadrature region and introduced to the classifier. The target object boundaries must be chosen in a proper way to ensure that the accuracy of object detection is high. After the target object selection is made, the classifier then initiates scanning operations by placing Haar-like features shown in Figure 1 into this quadrilateral field [14].

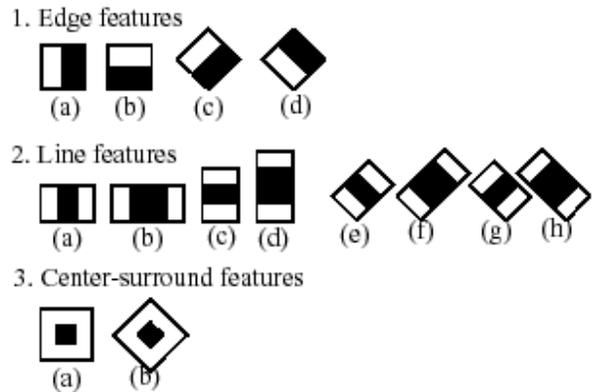


Fig.1. Haar-like features

After each feature is inserted, the sum of the values of the remaining pixels in the black and white areas is found. Then, as a result of the difference between the values obtained from this black region and the white region, the density of the feature is calculated in Equation 1.

$$f(x) = \sum_{BlackArea}(Density) - \sum_{WhiteArea}(Density) \quad (1)$$

The density of the feature in the equation indicated above is also considered to be a weak classifier. In the next step, these weak classifiers are collected according to a particular formulation, such as in Equation 2, and strong classifiers are obtained. In the Equation 2, the use of features of weak classifiers will form a strong classifier.

$$F(x) = a_1 f_1(x) + a_2 f_2(x) + a_3 f_3(x) + \dots \quad (2)$$

In this equation $f_n(x)$ is considered as weak classifier, and $F(x)$ as strong classifier. In the last step of the algorithm, the cascade classifier shown in Figure 2 is obtained with the successive application of these strong classifiers. Each $C_1, C_2 \dots C_N$ element is a strong classifier. In the rectangular area where the object is located, using each of the weak classifiers and applying the required formulations as in Equation 2, strong classifiers ($F(x)$ or C_N) will be obtained. These strong classifiers obtained in the final stage of all these processes are applied to the image taken into the system and the detection process is performed to determine whether the object is present in the image.

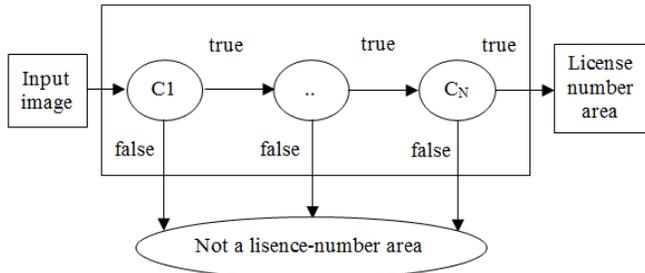


Fig. 2. Cascade Classifier

B. Camera Calibration

In this study, the camera calibration process is performed by transforming the coordinate systems shown in Figure 3 in order for the system to work properly.

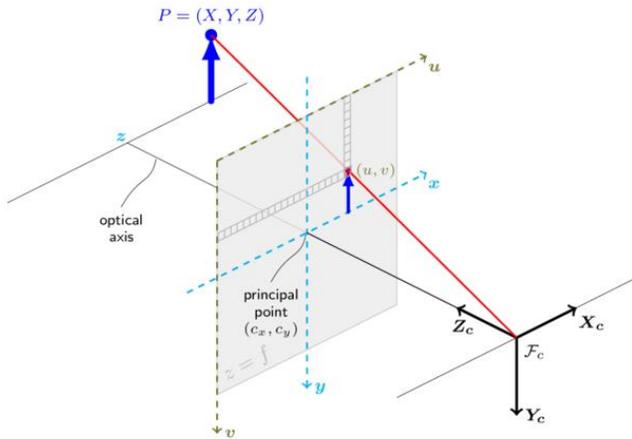


Fig. 3. Camera Calibration Coordinate System

For calibration, the transformation of the actual coordinate system to the camera coordinate system is provided by Equation 3 below.

$$\begin{pmatrix} X_C \\ Y_C \\ Z_C \end{pmatrix} = R \begin{pmatrix} X_P \\ Y_P \\ Z_P \end{pmatrix} + t \tag{3}$$

Here, R is a 3x3 dimensional rotational matrix and t is a 3x1 dimensional transformation matrix. Then the camera coordinate system is then obtained by the Equation 4 which is given below the physical coordinates in the image plane coordinate system using the pinhole camera model method [15].

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} X_C/Z_C \\ Y_C/Z_C \end{pmatrix} \tag{4}$$

Including radial and tangential lens distortions, the extended physical coordinates in the picture plane coordinate system are obtained from Equation 5 below.

$$\begin{pmatrix} x_q \\ y_q \end{pmatrix} = (1 + k_1 r^2 + k_2 r^4) \begin{pmatrix} X_C/Z_C \\ Y_C/Z_C \end{pmatrix} + \begin{pmatrix} 2p_1 xy + p_2(r^2 + 2x^2) \\ p_1(r^2 + 2y^2) + 2p_2 xy \end{pmatrix} \tag{5}$$

Here, the radial distortion coefficients k_1 and k_2 , p_1 and p_2 are the tangential distortion coefficients and also $r_2 = x_2 + y_2$. Finally, the resulting extended physical coordinates are converted to pixel coordinates as follows.

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f_x x_q + u_0 \\ f_y y_q + v_0 \end{pmatrix}, \quad f_x = f \cdot \frac{s}{\Delta x} \quad \text{ve} \quad f_y = f / \Delta y \tag{6}$$

In the equation (u_0, v_0) is the basic point and is usually selected as the center of the picture. f is the focal length of the camera. s is the scale factor. Δx is the value in mm of the distance between two adjacent pixels in the horizontal axis. Δy is the value in mm of the distance between two adjacent pixels in the vertical axis. Using these equations the following equation is obtained:

$$s \cdot p = A(R|t) \cdot P \tag{7}$$

In this equation, $p = (u, v, 1)^T$ is the homogeneous coordinate of the pixel point of the picture. $P = (X, Y, Z, 1)^T$ is the homogeneous coordinate of the point in the spatial plane. A is the matrix in which the camera's intrinsic parameters are located. $(R|t)$ is the matrix with extrinsic parameters. This equation expresses the relationship between a point in the image and a point in the space plane [16].

In accordance with the above, in the stage of performing camera calibration in the system we propose, calibration will be performed visually using the chessboard pictures with the Emgu.CV library. Then the parameters to be obtained by calibration, and using the location of the object detected in the image of the center pixel, the actual location of the object will be determined. For calibration, as shown in Figure 4, the picture of the chessboard will be displayed from different angles to the camera and the pictures will be saved. After that, the corner points of these images will be provided. After the corner points are found, intrinsic and extrinsic parameters will be found and error analysis will be performed. The position of the mushroom in the space plane will be determined by using the location information of the pixel in the centre of the image of the mushroom detected with these intrinsic and extrinsic parameters. By means of the determined position, the robot will be directed to this position.

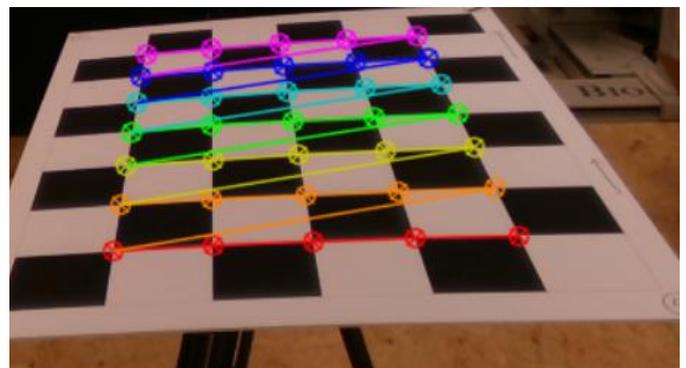


Fig. 4. Detecting the corners of the chessboard for Camera Calibration

Normally, the 3-dimensional axis is used to obtain the projection feature. To use the projection feature, the stereo camera system must be used. However, in this study, since a single camera is used for product identification and location information in the system, location calculation operations will be performed in a different way.

With this calibration process carried out in the system we recommend, focal lengths in the X and Y axis of the camera used are taken as a priority. These focal lengths are then used in the calculation of the position and the positions in the X, Y, and Z axis are obtained by adapting the formulations (Equations 8-10) used in the study for the determination of the mushroom position to our system [17].

$$d_z = \frac{((f_x + f_y)/2) * ((dg_x + dg_y)/2)}{((mrm_x + mrm_y)/2)} \quad (8)$$

$$d_x = \frac{((-d_z) * ((mrb_x + \frac{mrm_x}{2})) - (kg_x))}{(f_x)} \quad (9)$$

$$d_y = \frac{((-d_z) * ((mrb_y + \frac{mrm_y}{2})) - (kg_y))}{(f_y)} \quad (10)$$

where d_z is the position on the Z axis, d_y is the position on the Y axis, d_x is the position on the X axis, also f_x, f_y are the focal lengths of camera on the X and Y axis, respectively. mgb_x, mgb_y are actual horizontal and vertical length of mushroom, respectively. mrb_x, mrb_y are horizontal and vertical length of the image plane of mushroom, respectively. $mrbn_x, mrbn_y$ are the value of the starting point in the image plane of the mushroom on the X and Y axis, respectively. kg_x, kg_y are the length of the image which is taken from the camera on the X and the Y axis, respectively.

III. HARDWARE AND SOFTWARE OF THE PROPOSED SYSTEM

A. PLC and TIA PORTAL Software

In the control system with the PLC device which is the second stage of the robotic mushroom collection system, coordinate data of mushroom which is determined by image processing techniques is transferred to the PLC device. Then, this coordinate data coming to the PLC device pass through the necessary evaluation stages and the robot is sent to these coordinates. These coordinates are the coordinate data in the actual coordinate system of the central pixel of the detected object. This data is sent to the stepper motor drives after passing the necessary evaluation stages in the PLC device and the motors are sent to these coordinates with the help of the drivers. The PLC type to be used in this study is S7-1200 type. The picture of the model is presented in Figure 5.

Technical details about S7-1200 type PLC can be found [18].

In order to program the PLC, the TIA PORTAL interface has been developed. The TIA (Totally Integrated Automation) PORTAL software is a very useful interface which is developed by SIEMENS and providing solutions to problems such as communication with PLC, HMI, Scada systems and similar problems.

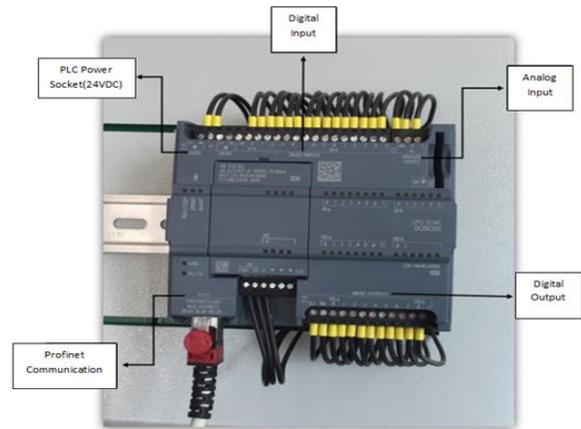


Fig. 5. S7-1200 PLC Device

B. Stepper Motor and Driver

In the robotic platform, stepper motor EMMS-ST-57-S-SE-G2 which is a product of FESTO company will be used. The image of the motor is shown in Figure 6 and the technical information is shown in Table-1.

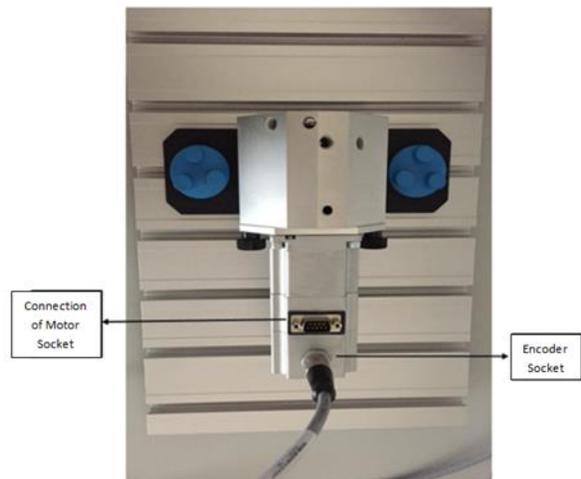


Fig. 6. Stepper Motor and Connection Parts

TABLE-1
TECHNICAL DETAILS OF STEPPER MOTOR

Size of Motor		57-S
Nominal Voltage	[V DC]	48
Nominal Current	[A]	5
Max. rotational speed	[rpm]	2720
Holding torque	[Nm]	0.8
Stepper angle	[°]	1.8±5%
Winding resistance	[Ω]	0.15±10%
Winding inductance	[mH]	0.5
Output mass moment of inertia	[kgcm ²]	0.29/0.30
Radial load on shaft	[N]	52
Axis load on shaft	[N]	10
Mass moment of inertia of rotor	[kgcm ²]	0.29

The motor drives are used for two stepper motors in robotic platform. These drives are also CMMS-ST-C8-7-G2 stepper motor drivers manufactured by FESTO, which is the stepper motor manufacturer. Figure 8 shows the image of the motor driver used in the system.

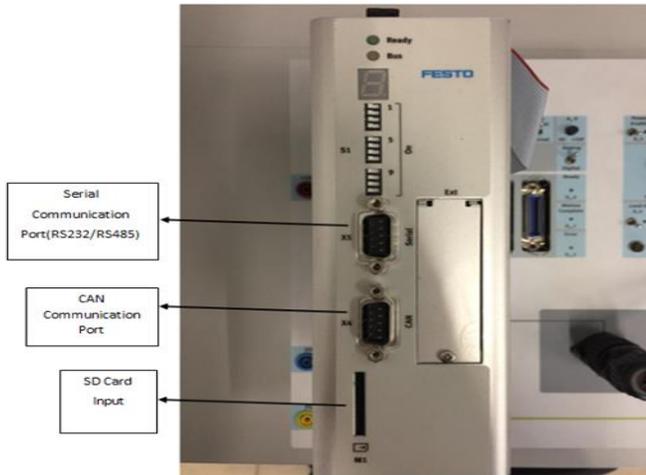


Fig. 8. Stepper Motor Driver used in the system

The motor drive and motor connection in the system are shown in Figure 9.

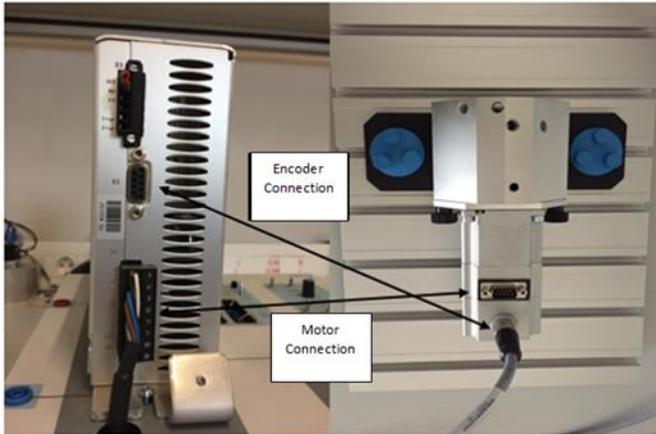


Fig. 9. Connection between Stepper Motor and Stepper Motor Driver

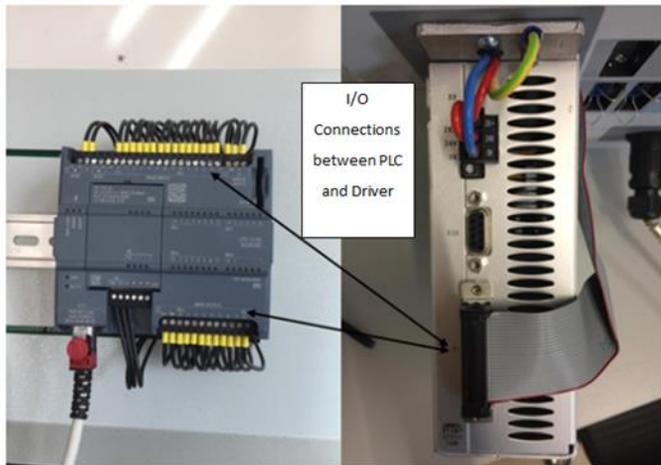


Fig. 10. Connection between PLC and Stepper Motor Driver

The connections between the PLC device and the motor drive are shown in Figure 10. Here is shown the sockets which are used for the output signals from the PLC to the drive and

input signal pins from the drive to the PLC device. In the driver part, the speed and position information from the encoder is provided by the connection from the encoder socket to the drive. This speed and position information from the encoder are sent as an output signal from the driver to the PLC device. The driver information to be supplied to the motor after this information is evaluated in the PLC is sent as input to the driver from the output terminals and is output to the motor connected to the motor socket after evaluating on the drive.

C. Communication of Visual Studio C# and TIA PORTAL

Visual Studio C# is the software in which image processing steps are performed. The PLC device is the hardware part where the control part takes place. Communication with the PLC device is carried out with TIA PORTAL. It is impossible for C# to communicate directly with TIA software. To eliminate this problem, there are several libraries of C# language. In this study, S7.Net library is used and this section will compile the code and communicate with TIA PORTAL [19].

S7.Net is a library that works only with the SIEMENS PLC and communicates only using Ethernet protocol. This library is written for C#. Therefore, without the need for any extra files, only the library can be compiled by adding the library. The following codes are written in the C# program language;

```
Plc Plc = new Plc(CpuType.S71200, "192.168.0.1", 1, 0); (11)
```

In this command, the PLC definition is performed. In this way, PLC is defined by entering the PLC type, communication IP address, Rack and Slot parameters.

```
Plc.Open (); (12)
```

In this command (12), at the start of the process, the PLC is ready for communication.

```
Plc.Write ("DB9.DBW0.0", axis_x); (13)
```

```
Plc.Write ("DB9.DBW4.0", axis_y) (14)
```

In DataBlock-9, an integer is assigned to the address 0.0 of the PLC to the mushroom position of the x axis and other integer is also assigned to the address 4.0 of the PLC to the mushroom position of y axis. In this system, there are only two motors that provide movement in the x and y axis, so only the position information of these axis is sent. In case of need, third axis or other information can be sent to TIA interface.

IV. EXPERIMENTAL RESULTS

In the mushroom picture used as sample data in the real-time testing phase of the system, the mushroom was first detected. Then, the location of the mushroom was determined and these position information obtained from the TIA Portal software communicating in real time with the Visual Studio C#.NET software were sent to the PLC instantly. The product detection and location information carried out in the testing phase with the PLC is shown as in Figure 11.

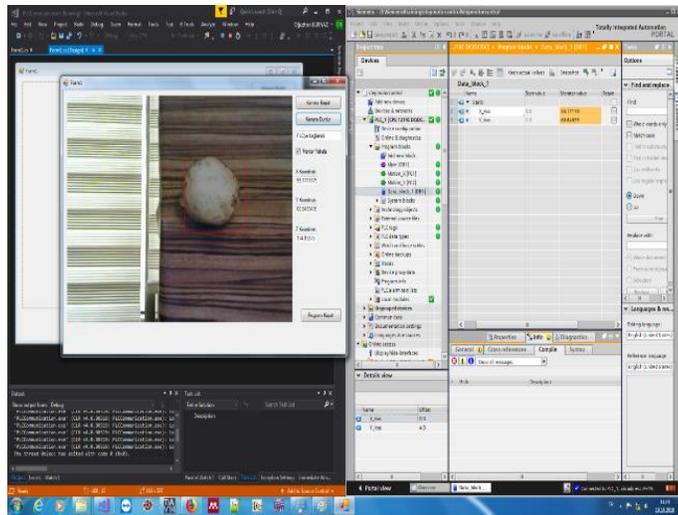


Fig. 11. Mushroom Detection and Sending Position Data to PLC

In Figure 11, the state of the system is shown. Firstly, the compilation process of the form interface developed by C# software language was performed on the Visual Studio platform. In the event that no error related to the code part is encountered after the compilation process, the form application has been executed and the image acquisition of the object from the camera has been provided by pressing the form 'Start Camera' button on the form interface. In the case of communication with the PLC connected to the system, the text box shows 'Connected to PLC System'. After this text appears, PLC communication with Visual Studio has been realized and the system has been made available for data exchange. The next step is the detection of the mushroom and if the 'Catch Mushroom' option is selected under the text box, the system is ready for the detection of the mushroom object. In the upper picture, the option is active and the detected mushroom object is enclosed in a red rectangle. The position of the center point of the quadrilateral area in the XYZ plane is shown in the 'X Coordinate', 'Y Coordinate' and 'Z Coordinate' in boxes. Because of the task performed in the system sends position information to the motors moving in X and Y plane, X and Y coordinate information is sent to the PLC. The right side of the picture shows the connected PLC device. The variables defined in the 'Data Block' section appear on the left side. These defined variables are used in the 'Motion_X' and 'Motion_Y' function blocks on the left side and the stepper motors are controlled. 'X Coordinate' value obtained in the Visual Studio form interface is sent to the PLC device and assigned to the 'X_Axis' variable. Similarly, 'Y Coordinate' value obtained in the Visual Studio form interface is sent to the PLC device and assigned to the 'Y_Axis' variable. These values can be seen simultaneously on both the form interface and the PLC device. The orange-coloured parts on the right show that the PLC device is connected and the incoming data is displayed instantly. The graph with position and speed information is shown in Figure 12 during the desired motion.

The graph shown in Figure 12 shows the change of the position data of the X axis depending on time from the form

interface. Location data are given in cm. In the blue part, the parts belonging to the Y axis are shown. Local changes in a period of approximately 70 seconds are shown graphically. In the pink section, the speed data of the X axis are shown. The motor which moves in X axis is moving at constant speed and 100 mm/Sec. This motor has a constant speed and its representation is linear. Similarly, in the turquoise colour, the speed data of the motor which provides the movement of the Y axis is shown. Again in this axis, the motor moves at 100 mm / sec and the speed of the motor is constant and the its representation is linear.

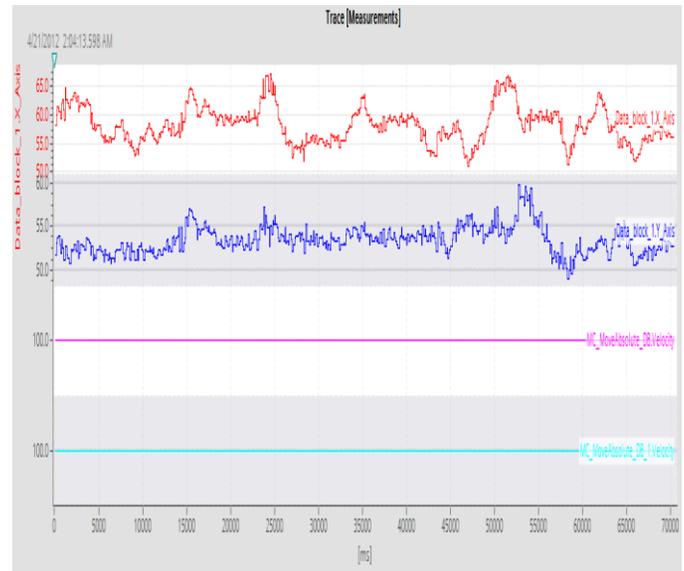


Fig. 12. Graphics of position and velocity data on X and Y axis

The following table shows the location information in which the mushroom detected in 10 different positions is detected and the details of the points in the global coordinate system are given in Table 2. The given position information is calculated in mm. Changes in X axis and Y axis are shown in Figure 13 and Figure 14.

TABLE-2
ACTUAL AND CALCULATED POSITION INFORMATION AND ERROR RATES IN X AND Y AXES

Data	Real location		Calculated Location		Error Rate in X Axis	Error Rate in Y Axis
	X	Y	X	Y		
1	146	44	135	38	7,53%	13,64%
2	140	132	144	132	2,86%	0,00%
3	96	78	92	76	4,17%	2,56%
4	148	132	144	107	2,70%	18,94%
5	140	126	143	106	2,14%	15,87%
6	122	106	123	85	0,82%	19,81%
7	110	104	108	81	1,82%	22,12%
8	102	110	102	109	0,00%	0,91%
9	120	104	119	103	0,83%	0,96%
10	118	104	134	95	13,56%	8,65%

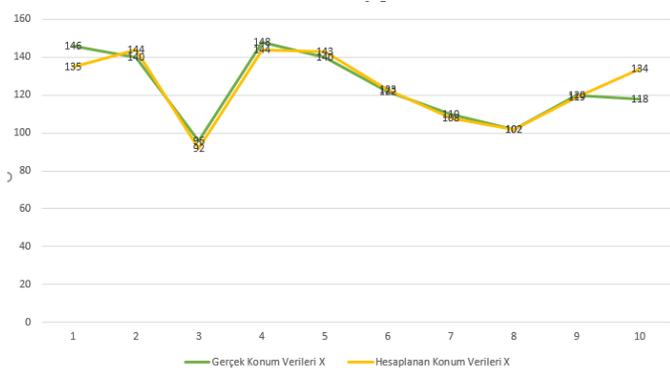


Fig. 13. Graphical representation of the actual position and calculated positions in the X axis

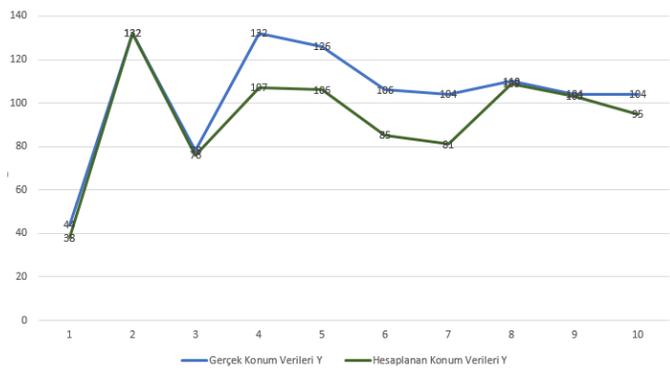


Fig. 14. Graphical representation of the actual position and calculated positions in the Y axis

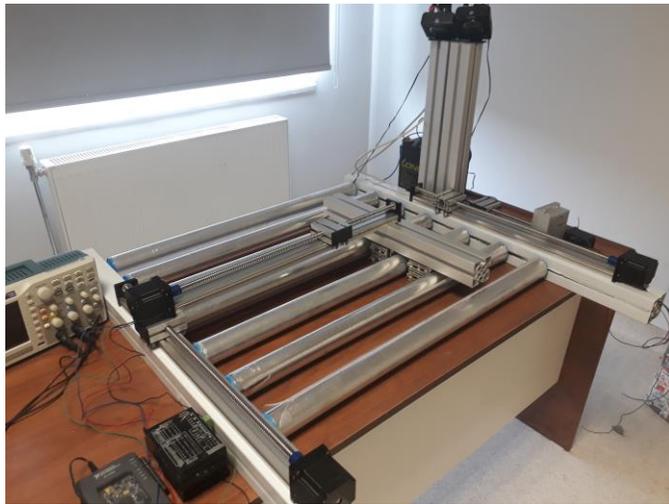


Fig. 15. Graphical representation of the test platform

In the test platform shown in Figure 15, the desired orbital movements were also applied.

V. CONCLUSION

In this study, the position of a mushroom in three-dimensional space was determined by means of an image-based controlled robotic platform. One of the biggest achievement of this work is to determine the actual position in the global coordinate

from the pixel-based location of the object in the image using an ordinary USB camera or built-in camera by means of calibration technique.

In the robotic platform, the first step was to determine the product by using image processing techniques. Firstly Haar-Cascade classifier was used for mushroom detection. For the use of this classifier, firstly, the stage of classifier training was carried out. Classifier training was carried out by introducing positive and negative images to software. The places where the mushrooms are located are called positive pictures. The places where the mushrooms are not located are called negative pictures. In the positive pictures, the parts of the mushroom are introduced to the system and the training is carried out. At the end of this process, .xml file is obtained. Thanks for this file introduced in the software, it is ensured that the mushroom object is detected when the system is running. After the solution of the problem of mushroom detection by using image processing techniques and camera, In the PLC, the position information evaluation steps are started. After making the necessary optimizations, these locations are sent to the stepper motor drivers and the motors were sent to these points. Using these position information, the motors are sent to this position. In the studies we are planning to perform in the future, the evaluation of the growth status of the detected mushroom and the studies of the harvesting of the product according to the result of this evaluation will be performed.

REFERENCES

- [1] Anonymous, "Mantar yetiştiriciliği", ss. 2–5, 2017. [Online]. Available at: <https://istanbul.tarim.gov.tr/Belgeler/KutuMenu/Brosurler/Sebzeçili k/mantar.pdf>. [Access: 27-Oct-2018].
- [2] Y. A. Sari ve S. Adinugroho, "Tomato ripeness clustering using 6-means algorithm based on v-channel otsu segmentation", *5th International Symposium on Computational and Business Intelligence, ISCBI 2017*, ss. 32–36, 2017.
- [3] S. SOLAK ve U. ALTINIŞIK, "Görüntü İşleme Teknikleri ve Kümeleme Yöntemleri Kullanılarak Fındık Meyvesinin Tespit ve Sınıflandırılması", *SAÜ Fen Bilimleri Enstitüsü Dergisi*, c. 22, num. 1, ss. 1–1, 2018.
- [4] M. S. A. Mahmud, M. S. Z. Abidin, ve Z. Mohamed, "Development of an autonomous crop inspection mobile robot system", *2015 IEEE Student Conference on Research and Development, SCORed 2015*, ss. 105–110, 2015.
- [5] S. S. Mehta ve T. F. Burks, "Adaptive Visual Servo Control of Robotic Harvesting Systems", *IFAC-PapersOnLine*, c. 49, num. 16, ss. 287–292, 2016.
- [6] Y. Zhao, L. Gong, Y. Huang, ve C. Liu, "A review of key techniques of vision-based control for harvesting robot", *Computers and Electronics in Agriculture*, c. 127, ss. 311–323, 2016.
- [7] W. Jun, Z. Zhou, ve D. U. Xiaodong, "Design and Co-simulation for Tomato Harvesting Robots", *Proceedings of the 31st Chinese Control Conference*, ss. 5105–5108, 2012.
- [8] B. Li, M. Wang, ve L. Li, "A real-time pineapple matching system based on speeded-up robust features", *Proceedings - 2010 International Conference on Computational Intelligence and Security, CIS 2010*, ss. 243–247, 2010.
- [9] Q. Feng, X. Wang, G. Wang, ve Z. Li, "Design and test of tomatoes harvesting robot", *2015 IEEE International Conference on Information and Automation, ICIA 2015 - In conjunction with 2015*

- IEEE International Conference on Automation and Logistics*, num. August, ss. 949–952, 2015.
- [10] A. Masoudian, “COMPUTER VISION ALGORITHMS FOR AN AUTOMATED HARVESTER (Thesis format: Monograph)”, num. December, 2013.
- [11] A. Shaikh, G. Khaladkar, R. Jage, ve T. P. J. Taili, “Robotic Arm Movements Wirelessly Synchronized with Human Arm Movements Using Real Time Image Processing”, *2013 Texas Instruments India Educators' Conference*, ss. 277–284, 2013.
- [12] M. Enzweiler ve D. M. Gavrilu, “Monocular pedestrian detection: Survey and experiments”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, c. 31, num. 12, ss. 2179–2195, 2009.
- [13] C. R. Wren, A. Azarbayejani, T. Darrell, ve A. P. Pentland, “Pfinder: real-time tracking of the human body”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, c. 19, num. 7, ss. 780–785, 1997.
- [14] P. Viola ve M. Jones, “Rapid object detection using a boosted cascade of simple features”, *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, c. 1, s. I-511-I-518, 2001.
- [15] C. Olsson, “Lecture 1 : The Pinhole Camera Model”, c. 3, num. 2, ss. 2–5, 2013.
- [16] Y. M. Wang, Y. Li, ve J. B. Zheng, “A camera calibration technique based on OpenCV”, *3rd International Conference on Information Sciences and Interaction Sciences (ICIS 2010), Chengdu, China*, ss. 403–406, 2010.
- [17] E. Wood ve A. Bulling, “Eyetable: Model-based gaze estimation on unmodified tablet computers”, *Proceedings of the Symposium on Eye Tracking ...*, ss. 3–6, 2014.
- [18] Anonymous, “S7-1200 DataSheet”, 2012. [Online]. Available at: https://cache.industry.siemens.com/dl/files/465/36932465/att_106119/v1/s71200_system_manual_en-US_en-US.pdf. [Access: 20-Nov-2018].
- [19] Anonymous, “S7.Net documentation Create a PLC instance , connect and disconnect”, 2016. [Online]. Available at: <https://github.com/S7NetPlus/s7netplus/blob/master/Documentation/Documentation.pdf>. [Access: 23-Sep-2018].

Research Assistant with the Department of Mechatronic Engineering, College of Engineering and Nature Sciences, Bursa Technical University, Bursa, Turkey from June 2018. His works have focused on the developments and applications of object detection, feature extraction, object tracking, object recognition, and image processing. His areas of interest also include Machine Learning Methods, Artificial Intelligence Algorithms, Deep Learning Methods, Neural Networks and their applications to mechatronic systems.

BIOGRAPHIES



KAĞAN KORAY AYTEN was born in Ankara, Turkey, in 1981. He received the B.S. degree in electronic engineering from Pamukkale University, Denizli, Turkey, in 2004, the M.S. degree in electrical & electronic engineering from Cardiff University, UK, in 2007, and the Ph.D. degree in electrical engineering from Bath University, England, UK, in 2013. He is currently an Associate Professor with the Department of Electrical and Electronics Engineering, College of Engineering, Erzurum Technical University, Erzurum. His work has focused on the development and application of control theory to a variety of mechatronic systems, with a focus on observation and estimation-based control. Much of his recent work has focused on the analysis and control of dynamical systems that arise in engineering applications.



OĞUZHAN KURNAZ was born in Erzurum, Turkey, in 1993. He received the B.S. degree in Mechatronic Engineering from Erciyes University, Kayseri, Turkey, in 2016. He continues his M.S. degree from 2016 in Erzurum Technical University, Erzurum, Turkey. He is currently a