OBJECT SORTING CONVEYOR WITH DETECTION COLOR USING ESP-32 CAMERA PYTHON BASED ON OPEN-CV

*INDAH SULISTIYOWATI, HAFIDZ MAULANA ICHSAN, IZZA ANSHORY

Department of Electrical Engineering, Universitas Muhammadiyah Sidoarjo

Jl. Mojopahit No.666 B, Sidoarjo

e-mail: indah_sulistiyowati@umsida.ac.id

*Corresponding author

ABSTRACT

The OpenCV Python library has been developed in all technology fields, including the industrial sector. In the industrial world, there are objects sorting tools in the form of conveyors. These instruments are presently more advanced since they employ cameras to read the things that should be sorted. The purpose of applying the Open Source Computer Vision Library (OpenCV) system to this object's sorter conveyor is to make it easier to sort objects based on color detection technology. The OpenCV method for object detection based on color was employed to select those objects. The first step in identifying the object in question is to capture RGB (red, green, and blue) objects in real time and transform their colors into HSV. Additionally, by masking the object to be centered and applying a threshold, the morphological process can remove unnecessary noise from the image. The investigation results include the ability to differentiate objects based on RGB color when sorted by considering the HSV value on the surface of colored objects.

Keywords: *OpenCV, Python, Colors, Conveyor, ESP-32*

1. INTRODUCTION

The progress and use of Artificial Intelligence (AI) technology in the industrial world is growing significantly [1]. As a result, they are gradually replacing traditional equipment and machines with modern versions that include automatic controls. Classification or selection of objects, especially in the industrial sector, can be grouped based on product type, color, weight, shape, etc. [2]. Sorting can be done manually by humans, with a barcode system, or automatically by machines. Sorting objects in the industry is generally done manually by humans, so object sorting is slower, less accurate, and less reliable. This is caused by human nature, which quickly gets tired. Therefore, a control tool is needed to sort objects that work automatically [3], [4].

Classification or selection of objects in the sorter system uses an ESP-32 camera to detect the color of the object. The Esp-32 camera is used to capture images of objects being sorted on the conveyor. Color is one of the elements that can be detected by the camera. Specifically, the colors captured are RGB (Red, Green, Blue) [5]. Systems for classifying objects by color can be developed in various ways. Research by Euis W., et al. uses the TCS230 color sensor to detect color and classify objects using a PLC (programmable logic control) as the driving system [6]. The colors that can be detected are red, green, and blue. A classification tool was developed that can classify items as black, blue, green, red, and white by Ika Sari [7]. In addition, the results of this research state that the color sensor detects the color of objects in a specified color range and activates certain actuators. Wicaksono, FR, et al., researched object sorting using image processing, where the sorted objects are captured by the camera and then processed into OpenCV [8].

The use of color sensors, PLC (programmable logic control) methods, and image processing have been carried out. However, the system is considered less effective because it does not guarantee the quality of the objects being sorted after production, and there is no better sorting media [9]. Based on the weaknesses in the previous system, an innovation was carried out by using an ESP-32 camera as a verification input to count or count the number of items or objects that had been sorted by color and a servo as a sorter which could also be observed using the Python software display using the number method and Open-CV. The advantages of this system can be monitored in real time [5], [10].

DOI: https://doi.org/10.54732/jeecs.v9i1.7

Therefore, the author has designed a product sorting conveyor using two servo motors. This design provides an efficient sorter based on three different color codes, by using Arduino UNO R3 as a microcontroller to drive the conveyor and object sorter [11]. The display of this tool uses an I2C 16x2 LCD because its use is simple and easy to understand.

The working system of this tool starts with the objects that have passed production and will be sorted using this conveyor, the objects that have been sorted will pass through the ESP -32 camera, and directly the objects will be separated according to their color, the appearance of color detection can be observed using Python display software in real-time. Next, a container was designed to accommodate objects that had been sorted by color [12], [13]. The main aim of creating a color-based item sorting tool prototype is to simplify human work in automatically sorting three color codes and also counting the number of items that have been sorted [14], [15].

2. RESEARCH METHODOLOGY

The main objective of this research is to create a prototype of a color-based item sorting tool that can help and simplify human work in automatically sorting three color codes and also counting the number of items that have been sorted. The method used in this research is image processing which utilizes the OpenCV library in Python.

2.1 The Workflow of Classification

The steps in classifying colored objects using the proposed method can be explained as follows:

1. The image data collected is objects that are red, green and blue.
2. Converting an object or video image into the Lab color space to get a red, green, or blue color classification.
   The result of changing this value is an image or video feature to group images based on color which can be determined by the number of items. This work uses Python and Arduino IDE software.
3. Classify the color of objects as red, green, or blue.
   This classification uses the OpenCV and HSV (Hue, Saturation, Value) methods. The Open Source Computer Vision Library (OpenCV) functions to process images and videos, allowing users to extract information from images. This HSV model requires RGB primary colors as the basis for color detection. H (hue) is the color angle on the circular axis of the cone, with red as the 0° axis. V (value) is the color component on the vertical axis of the cone. And the value V = 0 is at the end of the black axis and the value V = 1 is at the end of the white axis. This V axis represents all types of gray colors. S (saturation) is the level of saturation that contains a lot of white light or color purity, and the value is the radian of the cone. The feature values used by the RGB box (Red, Green, Blue) are the upper and lower values to get the desired RGB color value.
4. Perform image processing for conveyor sorting on colored objects and count the number of objects based on their color.
   The next step is to calculate the overall identification results to obtain testing accuracy by matching the identification results using the OpenCV method with the actual results.

2.2 Design of Device

The conveyor design is adapted from the shape of a long beam, but the size is larger because it is intended for packages 1 meter long, 40 cm wide, and 30 cm high. An Esp-32 camera to capture images of objects to detect color and count objects that are running on a conveyor placed on a conveyor placed on a 40 cm high support. The Arduino Uno and I2C 16x2 LCD are placed on the right side of the conveyor in a protective box so that they are not easily damaged. Behind the conveyor, there are two dc motors on both sides to drive the conveyor. At the front, there are 2 servo motors with a distance of 30 *cm* apart left and right. The design of the packaging box is shown in Figure 1 and 2.

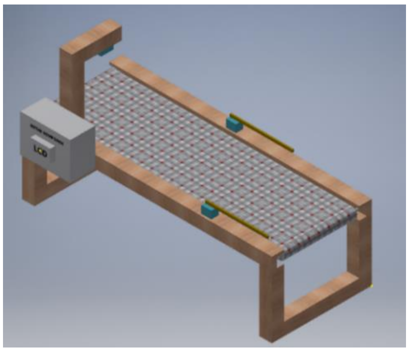


Figure 1. Tool Design

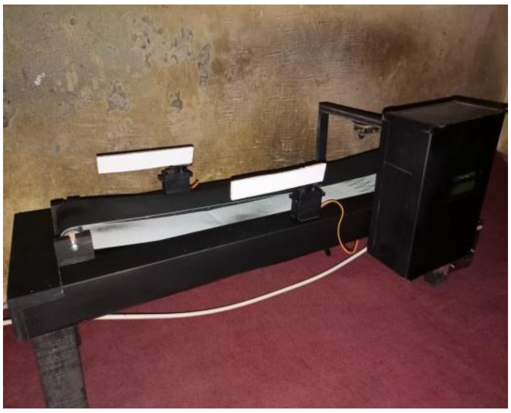


Figure 2. Tools produced

2.3 Suite

A block diagram of a system is useful for determining the basis for design. Block diagram of a conveyor sorter with Color Detection Using Esp-32 Camera Based on OpenCV Python is as presented in Figure 3. There are 3 parts to the conveyor Object Sorter with Detector Color Using Esp-32 Camera Based on OpenCV Python, namely: input, process, and output. In the input section there are module Esp-32 cameras as input for capturing object moving on the conveyor, as well as to identify color and count moving objects [16]. In the process section, there is an Arduino UNO which is used to carry out the process of moving the dc motor, servo motor, and turning on the 16x2 I2C LCD [23]. In the output section, there is a DC motor to drive it conveyor, servo motor for sorting objects or objects, LCD I2C 16x2 to display the amount of sorted items based on color [24]. In planning tool electronics, there are pictures of the tool-making circuit [17]. The Suite tool Conveyor Object Sorter with Detector Color Using OpenCV Python-Based Esp-32 Camera is presented in Figure 4.

There is a component Arduino UNO module as a microcontroller. There is a project board that is used to connect the pins between components including Arduino Uno, push button, power supply, 16x2 i2c LCD, relay, and servo motor [18].

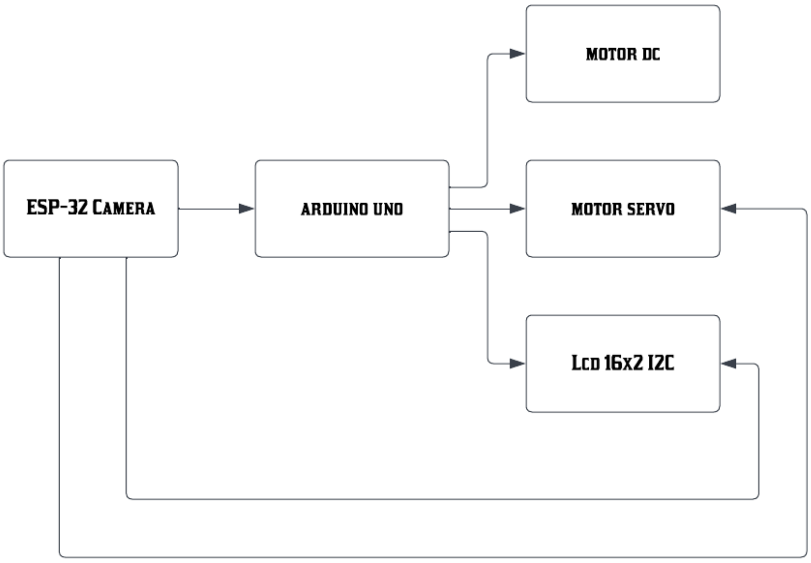


Figure 3. Block diagram

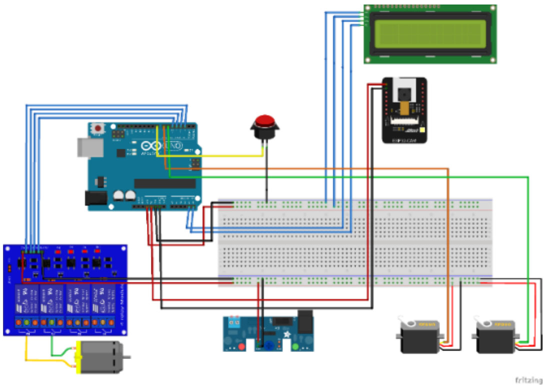


Figure 4. Circuit

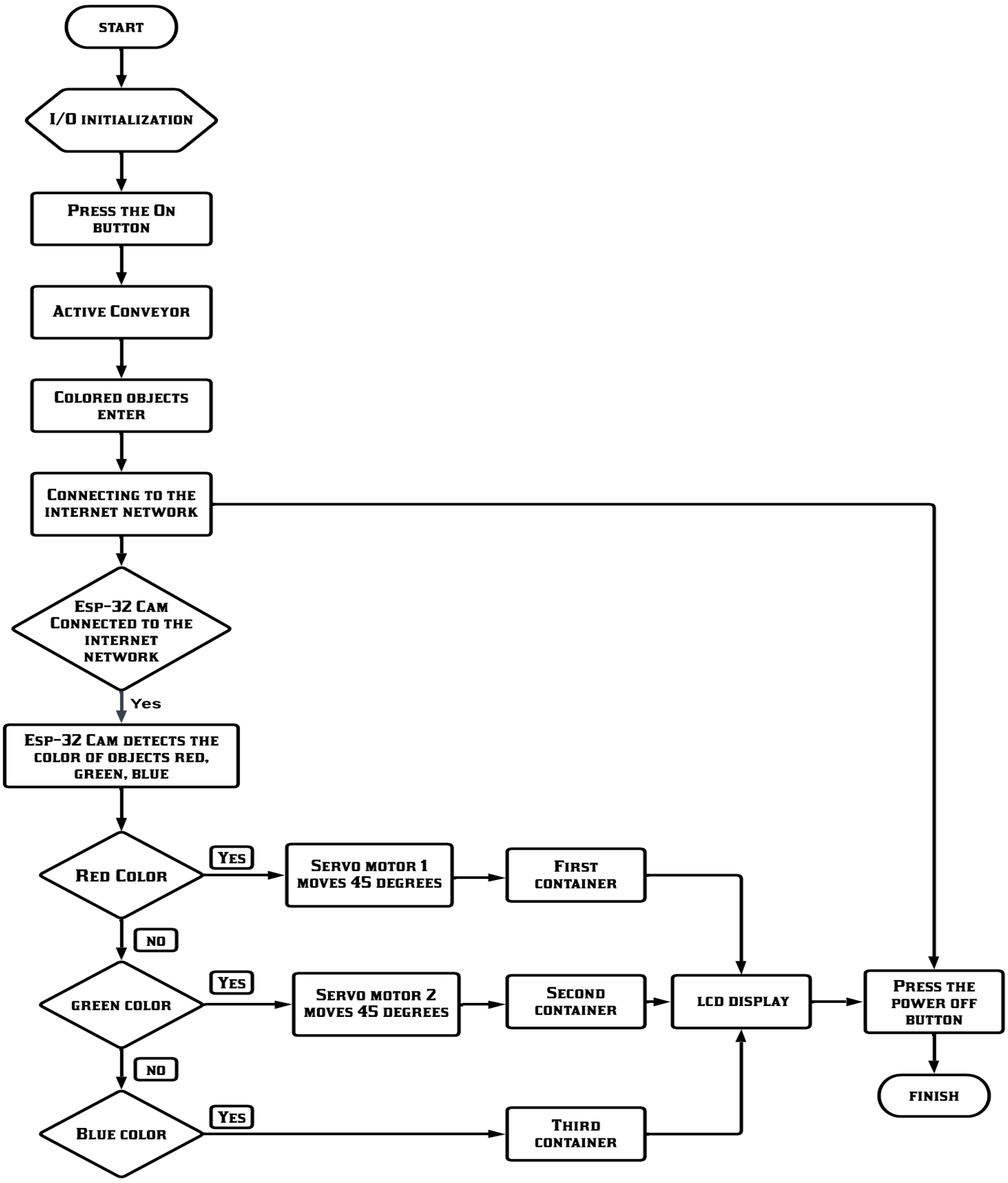

START
I/O INITIALIZATION
PRESS THE ON BUTTON
ACTIVE CONVEYOR
COLORED OBJECTS ENTER
CONNECTING TO THE INTERNET NETWORK
ESP-32 CAM CONNECTED TO THE INTERNET NETWORK
Yes
ESP-32 CAM DETECTS THE COLOR OF OBJECTS RED, GREEN, BLUE
RED COLOR
YES
SERVO MOTOR 1 MOVES 45 DEGREES
FIRST CONTAINER
NO
GREEN COLOR
YES
SERVO MOTOR 2 MOVES 45 DEGREES
SECOND CONTAINER
LCD DISPLAY
PRESS THE POWER OFF BUTTON
FINISH
NO
BLUE COLOR
YES
THIRD CONTAINER

Figure 5. Flow diagram

2.4 Device Design Soft

Planning device soft is planning program flow to be created. The Planning device soft shown in the program flow diagram. The program begins with initialization (giving initial value) at input/output, then by pressing the ON button the conveyor will active Then object colored in, and then connect to internet network, esp-32 cam is connected to the internet network, if connected esp-32 cam will detect object colored red, green, or blue. If detected object is detected red by the esp-32 cam then the LCD will display the words “red:1” and servo motor 1 will move 45^o so object is colored red the fall to receptacle First. Likewise objects colored green will detected by cam esp-32 then LCD will displays the words “green:1” and servo motor 2 will moving 45^o making objects colored green the fall to receptacle second. Lastly, if There is an object colored blue detected by the esp-32 cam, then the LCD will it will say “blue:1” and it will immediately fall to receptacle third. The Conveyor Program flowchart Sorting Objects with Detection Color Using ESP-32 Camera Based on OpenCV Python is presented in Figure 5.

3. RESULTS AND DISCUSSIONS

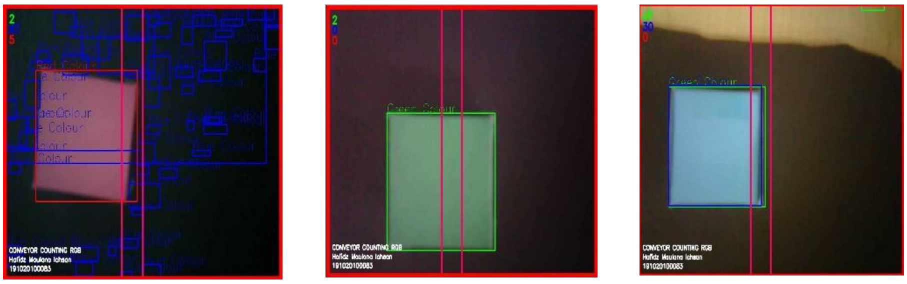
Test procedure carried out on each component and system to ensure the machine runs as designed. Testing is carried out on each part as follow.

3.1 Testing the ESP-32 Camera

Testing the ESP-32 Camera as RGB color reading in presented in Figure 6. From Figure 6 (a), (b), and (c), the accuracy results for reading each RGB color from a distance of 12 cm are obtained, as explained in the Table 1. The display in the Python software shows that the ESP32 Cam is very vulnerable to reading light so noise often occurs when reading color.

3.2 Experiment calculation sorted objects

This test is carried out to calculate the items to be sorted. The results of calculating the sorted items show accuracy in capturing the suitability of each 10 objects according to RGB colors, which is explained in Figure 7. From experiments to determine the accuracy of the ESP32 Cam in calculating each RGB color, it was found that according to Figure 8 above, red has 80% accuracy, green has 100% accuracy, and blue has 80% accuracy.



a. Color Red (R)

b. Color Green (G)

c. Color Blue (B)

Figure 6. Color detection

.Table 1: RGB Color Chart

RGB Color Reading Results					
Colour	**HSV value**			**Distance (cm)**	**Accuracy (%)**
	H (Hue)	**S (Saturation)**	**V (Value)**		
Ed	Lower: 136	Lower: 87	Lower: 111	12 cm	80%
	Upper: 180	Upper: 255	Upper: 255		
Green	Lower: 25	Lower: 52	Lower: 72	12 cm	100%
	Upper: 102	Upper: 255	Upper: 255		
Blue	Lower: 94	Lower: 80	Lower: 2	12 cm	80%
	Upper: 120	Upper: 255	Upper: 255		

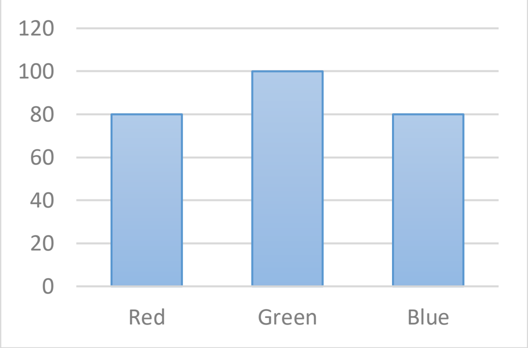


Figure 7. Counting the number of colored objects

Table 2. Accuracy testing and detection for sorting objects against the objects to be sorted

Colour	testing	distribution channel 1	distribution channel 2	distribution channel 3	Accuracy (%)
Ed	10 times	8	-	-	80%
Green	10 times	-	10	-	100%
Blue	10 times	-	-	8	80%

3.3 Testing Accuracy Sorting and Detection Sorting Sorted Objects

This test is used to measure the accuracy of sorting the objects to be sorted. The object sorting results show the accuracy of the items sorted every 10 times, explained in Table 2. From experiments to determine the accuracy of servo motors in sorting RGB color objects which were carried out 10 times, it was found that based on table 2 above, red has an accuracy of 80%, green has 100% accuracy, and blue color has 80%. The calculation results of the sorted objects show precise accuracy, according to the desired color.

4. CONCLUSION

Based on the results of the tests carried out in this research, several conclusions can be obtained that image processing using the OpenCV method on the ESP-32 Camera can run well, with 3 color categories, namely: red, green, and blue. The Python program that has been created can function properly and can work as designed. With the OpenCV method for reading these 3 colors, you can easily separate red, green, and blue objects.

There is a drawback to the ESP32 Cam in that it is very susceptible to light, causing interference during the reading process. The ESP32 Cam also has a weakness for hot temperatures on the device which will cause delays in the reading process. In subsequent research, a camera was added to make recognition more precise and the enumeration pattern was improved to minimize the influence of shadows that interfere with the image recognition process. It is necessary to control the speed of the conveyor motor, so that object selection can be better.

REFERENCES

[1] R. Bangun *et al.*, (2020), “Rancang Bangun Alat Kendali Sortir Barang Berdasarkan Empat Kode Warna,” *JTEV (Jurnal Teknik Elektro dan Vokasional)*, vol. 6, no. 2, pp. 391–402, doi:10.24036/JTEV.V6I2.109360.

[2] D. R. Rajagukguk and Iqbal Panjaitan, (2020), “Compression of Color Image Using Quantization Method,” *Jurnal Info Sains : Informatika dan Sains*, vol. 10, no. 2, pp. 13–18, doi:10.54209/infosains.v10i2.32.

[3] F. Pfitzner, A. Braun, and A. Borrmann, (2024), “From data to knowledge: Construction process analysis through continuous image capturing, object detection, and knowledge graph creation,” *Automation in Construction*, vol. 164, p. 105451, doi:10.1016/J.AUTCON.2024.105451.

[4] S. Suwarno and K. Kevin, (2020), “Analysis of Face Recognition Algorithm: Dlib and OpenCV,” *Journal of Informatics and Telecommunication EngineerinG*, vol. 4, no. 1, pp. 173–184, doi:10.31289/JITE.V4I1.3865.

[5] S. Mala and Rasiban, (2021), “Klasifikasi Kematangan Buah Pisang Tanduk Berdasarkan HSV,” *Jurnal Jupiter*, vol. 1, no. 1, pp. 197–207.

[6] T. J. Nuva, M. I. Ahmed, and S. S. Mahmud, (2022), “Design & Fabrication of Automatic Color & Weight-Based Sorting System on Conveyor Belt,” *Journal of Integrated and Advanced Engineering (JIAE)*, vol. 2, no. 2, pp. 147–157, doi:10.51662/jiae.v2i2.87.

[7] M. I. Sari, R. Handayani, S. Siregar, and B. Isnu, (2018), “Pemilah Benda Berdasarkan Warna Menggunakan Sensor Warna TCS3200,” *TELKA - Jurnal Telekomunikasi, Elektronika, Komputasi dan Kontrol*, vol. 4, no. 2,

pp. 85–90, doi:10.15575/TELKA.V4N2.85-90.

[8] F. R. Wicaksono, A. Rusdinar, I. Prasetya, and D. Wibawa, (2018), "Perancangan Dan Implementasi Alat Penyortir Barang Pada Konveyor Dengan Pengolahan Citra Design and Implementation of Items Device Sorting on Conveyor With Image Processing," *e-Proceeding of Engineering*, vol. 5, no. 1, pp. 40–47.

[9] E. D. Febriyanti, A. Alimuddin, and H. M. Putra, (2024), "Rancang Bangun Penerimaan Box Paket Berbasis Internet of Things (IOT)," *Printer : Jurnal Pengembangan Rekayasa Informatika dan Komputer*, vol. 2, no. 1, pp. 37–45.

[10] H. Attar, A. T. Abu-Jassar, V. Yevsieiev, V. Lyashenko, I. Nevliudov, and A. K. Luhach, (2022), "Zoomorphic Mobile Robot Development for Vertical Movement Based on the Geometrical Family Caterpillar," *Computational intelligence and neuroscience*, vol. 2022, pp. 1–19, doi:10.1155/2022/3046116.

[11] O. A. Putra and R. Handika, (2022), "Rancang Bangun Sistem Keamanan Lalu Lintas Menggunakan Smartphone Dan Esp32cam Berbasis Arduino Mega 2560," *Jurnal Sains dan Teknologi (JSIT)*, vol. 2, no. 2, pp. 120–130, doi:10.47233/jsit.v2i3.202.

[12] N. I. Humaira B, M. Herman, N. Nurhikma, and A. B. Kaswar, (2021), "Klasifikasi Tingkat Kualitas Dan Kematangan Buah Tomat Berdasarkan Fitur Warna Menggunakan Jaringan Syaraf Tiruan," *Journal of Embedded Systems, Security and Intelligent Systems*, vol. 2, no. 1, p. 18, doi:10.26858/jessi.v2i1.20329.

[13] J. Jusrawati, A. Futri, and A. B. Kaswar, (2021), "Klasifikasi Tingkat Kematangan Buah Pisang Dalam Ruang Warna RGB Menggunakan Jaringan Syaraf Tiruan (JST)," *Journal of Embedded Systems, Security and Intelligent Systems*, vol. 2, no. 1, pp. 49–54, doi:10.26858/JESSI.V2I1.20327.

[14] A. M. Elhanafi, R. Siregar, M. Yeni, and S. An-nissa, (2022), "Cryptography Application on RGB Overlapping Block Based PVD Using AES," *Jurnal dan Penelitian Teknik Informatika*, vol. 7, no. 3, pp. 2116–2124.

[15] D. S. Febriyan and R. D. Puriyanto, (2021), "Implementation of DC Motor PID Control on Conveyor for Separating Potato Seeds by Weight," *International Journal of Robotics and Control Systems*, vol. 1, no. 1, pp. 15–26, doi:10.31763/ijrcs.v1i1.221.

[16] H. S. Ghifari and F. Utaminingrum, (2022), "Klasifikasi Kualitas Minyak Goreng berdasarkan Fitur Warna dan Kejernihan dengan Metode K-Nearest Neighbour berbasis Arduino Uno," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 7, pp. 3269–3274.

[17] N. M. Syahrian, P. Risma, and T. Dewi, (2017), "Vision-Based Pipe Monitoring Robot for Crack Detection Using Canny Edge Detection Method as an Image Processing Technique," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, vol. 2, no. 4, pp. 243–250, doi:10.22219/kinetik.v2i4.243.

[18] F. E. Saputra, R. Cahya Wihandika, and A. W. Widodo, (2021), "Penentuan Kualitas Biji Kopi Menggunakan Local Ternary Patterns Dan RGB-HSV Color Moment Dengan Learning Vector Quantization," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 6, pp. 2299–2307.