Journal of Electrical Technology UMY (JET-UMY), Vol. XX, No. X, XXX XXX
ISSN 2550-1186 e-ISSN 2580-6823

DESIGN AND CONSTRUCTION OF GOODS SORTER CONVEYOR WITH COLOR DETECTION USING ESP-32 CAMERA BASED ON OPEN-CV PYTHON

Hafidz Maulana Ichsan, Indah Sulistiyowati*, Izza Anshory

Department of Electrical Engineering, Universitas Muhammadiyah Sidoarjo
Hafidzymaulanaichsan@gmail.com
*Corresponding author, e-mail: Indah_sulistiyowati@umsida.ac.id

Abstract – *The use of the OpenCV Python library can be developed in all fields of technology, including in the industrial sector. In the industrial world, there are goods sorting tools in the form of conveyors, where these tools are now increasingly sophisticated with the use of cameras as readers of the objects being sorted. The purpose of applying the OpenCV (Open Source Computer Vision Library) system to this goods sorter conveyor is to make it easier to sort objects based on the color we want and improve education on industrial technology in Indonesia. The method used to select this object is the OpenCV method of detecting objects based on color. The process of detecting this object starts by capturing RGB (red, green, blue) colored objects in real-time, converting RGB colors to HSV, followed by performing a threshold, and after that the morphological process to filter out noise that is not needed in the image by masking the object so that it is centered. The results of this study are to be able to distinguish objects based on RGB color when sorted later by taking into account the value of the HSV on the surface of colored objects, which later this research can be applied to conveyor equipment sorting goods based on color.*

Keywords*: OpenCV, Python, Color, Conveyor*

I. Introduction

Currently, the progress and use of AI (Artificial intelligence) or artificial intelligence technology in the industrial world is growing significantly[1]. So they slowly began to leave the traditional tools and machines and switch to modern tools and machines with automatic control. Classification or selection of goods, especially in the industrial sector can be grouped based on product type, color, weight, shape, etc[2]. Sorting can be done manually by humans, by barcode systems, or automatically by machines Sorting of goods in industry is generally done manually by humans, so the sorting of goods is slower, less accurate, and less reliable due to human fatigue[3],[4]. The classification or selection of goods in this sorting tool uses the esp-32 camera to detect the color of the goods. The ESP-32 camera is used to capture images of the goods being sorted on the conveyor. Color is one of the elements that can be detected well in the camera though. In particular, the colors captured are RGB (Red Green Blue)[5].

Systems for classifying objects based on color can be developed in various ways. A study by Euis W. et al used the TCS230 color sensor to detect color and classify goods using PLC (Programmable Logic Control) as the driving system[6]. The colors that can be detected are red, green, and blue. In a study by Ike Sari, et al. A classification tool was developed that can classify items with black, blue, green, red and white colors. The results of the study stated that the color sensor detects the color of objects in the specified color range and activates certain actuators[7]. And in the research of Wicaksono, F.R., et al, sorting goods using image processing where the sorted objects are captured by the camera and then processed into openCV[8]. Some of these studies use the sensor, Programmable Logic Control) methods, and image processing. However, this system is considered less effective because it does not guarantee the quality of the goods that are sorted after they are produced and there are no better sorting media[9]. From the weaknesses in the previous system, an innovation was made using the esp-32 camera as a verification input for counting or counting the number of items or objects that have been sorted by color and servo as the sorter can also

Manuscript received January 2022, revised January 2022

be observed using the python software display using the NumPy and open-cv methods which can be monitored in real time[5],[10]. From the background of the problem and part of the research results above. Therefore, the author designed a product sorting conveyor using two servo motors as a product sorter based on three different color codes. By using Arduino UNO R3 as a microcontroller to drive conveyors and sort goods[11]. The display of this tool uses a 16x2 LCD due to its simple use for everyone's understanding and. The working system of this tool swaths from the goods that have passed production will be sorted using this conveyor, the sorted goods will pass through the esp-32 camera and directly the goods will be separated according to their color, the appearance of color detection can be observed using the python display software real-time. There is a container for items that have been sorted by color[12],[13]. The main goal of the author to make a prototype of this color-based item sorting tool is to help and facilitate human work in sorting three color codes automatically and also counting the number of items that have been sorted[14],[15].

II. Literature Study

There is a review of several previous journals that have analyzed the system I designed. In a study conducted by Euis W., et al, the sorting of goods used a TCS230 color sensor for color detection using PLC (Programmable Logic Control) for the actuator system[6]. The detected colors are blue, green, and red. In a study conducted by Ike Sari, et al, a classification tool has been created that can sort items into black, white, red, blue and green. The results show that the color of the object can be detected by a color sensor with a predetermined color range and activates a certain servo motor[7]. And in the research of Wicaksono, F.R., et al, sorting goods uses image processing where the sorted objects are captured by the camera and then processed into openCV[8]. From the explanation above, the author will design a conveyor sorting goods with color detection using Arduino UNO R3 as a microcontroler, ESP-32 camera as a color detector for goods, a 16x2 I2C LCD as a display on the device, a dc motor as a conveyor controller and a servo motor as a sorter. goods, as well as a real-time video display that can be observed with python software in real-time.

II.1. *Image processing*

Image processing is a process for improving quality, taking, and changing information in an image. Image processing has gone through many developments made by researchers in object detection. This research shows that color can be used as a reference value for detecting objects directly through the camera[16]. In image processing, sampling states the size of the pixels (points) in the image, while quantization states the brightness level at the grayscale level according to the number of binary bits contained in an optical device[17]. Typically, the size of an image array is a few hundred pixels times a few hundred pixels and there are several tens of possible gray level differences.

II.2. *OpenCV Python*

In this study, OpenCV functions as the processing of video results for red, green, and blue objects. Open Source Computer Vision Library (OpenCV) functions to process images and videos to allow users to extract information from images. OpenCV can be run using programming languages such as Python, C, C++, and Java, and can run on various platforms such as Linux, Windows, Android, Mac OS, and IOS[16]

II.3. *RGB (Red, Green, Blue)*

RGB in this study is the color of the object to be moved, namely red, green, and blue (red, green, blue). Each color in the photo/image is a color combination of RGB (Red, Green, Blue). A color reading from 0% to 100% is 0 to 255. RGB (255,0,255), (255,255,0), and (0,255,255) produce red, green, and blue colors[16].

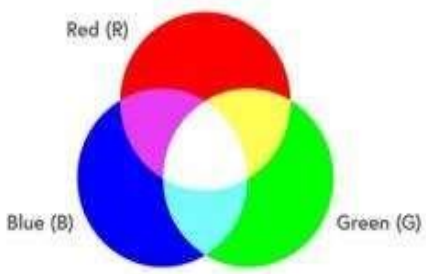

Fig 1. RGB color

II.4. *HSV (Hue, Saturation, Value)*

In this study, HSV serves as a color detector that will be entered into Python later. This HSV model requires RGB primary colors as the basis for color detection. H (hue) is the color angle on the axis of the conical circle, with red as the 0° axis. V (value) is the color component of 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. S

(saturation) is the degree of saturation that contains a lot of white light or color purity, and its value is the radians of the cone[16].

II.5. *ESP-32 Camera*

In this study, the Esp-32 camera serves as an input to capture the color of goods passing through the conveyor. ESP-32 camera is a board that contains WIFI/Bluetooth with an ESP-32 microcontroller and camera[18]. There is no USB to a serial interface on this board. Programming is carried out via an external interface. Generally, use USB to Serial FTDI for the programming process. Esp-32 camera works with 3V voltage with pin RX/TX[19].

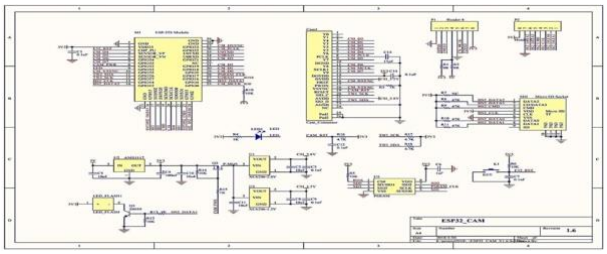

Fig 2. ESP32 CAMERA Module

ESP-32 tripod address location for USB to Serial (FTDI): for USB to Serial (FTDI). The Esp-32 VDC camera pins are connected to the USB to Serial (FTDI) VCC pins, the Esp-32 camera ground is connected to the USB to Serial (FTDI) ground pins, and the Esp-32 camera's ESP OUR pins are connected to the USB to FTDI serial TX pins, the OUT pin of the Esp-32 camera to the RX pin of the USB pin, and GPIO_0 to ground[5].

Fig 3. ESP-32 Camera

II.6. *Arduino-UNO*

In this study, the Arduino Uno R3 board type was used as the microcontroller of this tool. Arduino Uno R3 is a microcontroller board based on ATmega328[4]. Arduino Uno R3 specifications in table 1, are as follows:

TABLE I
ARDUINO-UNO SPECIFICATIONS TABLE

Arduino-UNO	Usage
Mikrokontroler	ATmega328
Number of digital I/O pins	14
Number of analog input pins	6
DC current per I/O pin	40 mA
DC current for the 3.3V pin	50 mA
Input voltage limit	6-20V
operating voltage	5V Recommended input voltage: 7-12V

Fig 4. Arduino-UNO

III. Research Method

III.1. *Block Diagram System*

The block diagram of a system is useful for determining the design basis. The block diagram of the Conveyor Sorting Conveyor Design System with Color Detection Using the OpenCV Python-Based Esp-32 Camera is as follows:

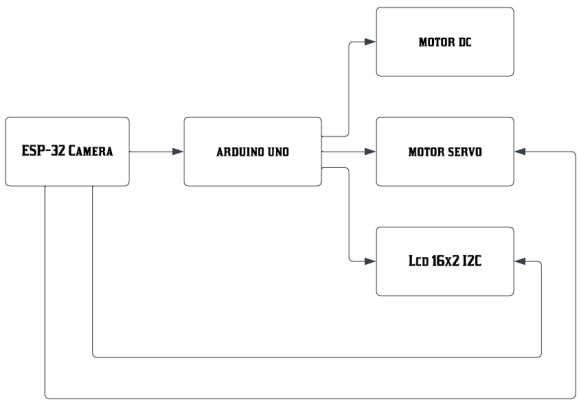

Fig 5. The Block Diagram Tool

There are 3 parts to the Design of a Goods Sorting Conveyor with Color Detection Using the Esp-32

Camera Based on OpenCV Python, namely: input, process, and output. In the input section, there is an Esp-32 camera module as input for capturing moving objects on the conveyor, as well as for identifying colors and counting moving objects[20],[21]. In the process section, there is an Arduino UNO which is used to carry out the process of moving dc motors, and servo moors, and turning on the 16x2 I2C LCD[22]. At the output, there is a dc motor to drive the conveyor, a servo motor to sort objects or goods, and a 16x2 I2C LCD to display the number of items sorted by color[23]. In an electronic device design, there is a picture of a tool-making circuit[17]. The following is a picture of a series of Goods Sorting Conveyor tools with Color Detection Using the OpenCV Python-Based Esp-32 Camera :

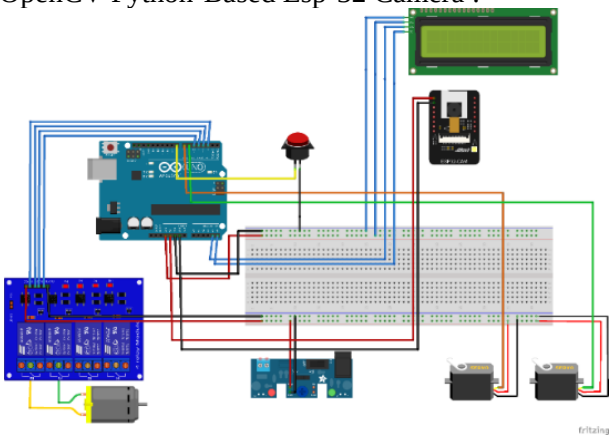

Fig 6. Wiring design

There is an Arduino UNO module component as a microcontroller. There is a project board that is used for connecting pins between components, including Arduino Uno, push buttons, power supply, 16x2 i2c LCD, relay, and servo motors[24].

TABLE II
PIN CONFIGURATION TABLE

No.	Component	Component Pin Addresses	Arduino UNO Pin Addresses
1.	Lcd 16x2 I2C	GND	GND
		VCC	3V
		SDA	A4
		SCL	A5
2.	Esp-32 Cam	5V	5V
		GND	GND
3.	Push Button	OUT	D8
		GND	GND
4.	Relays	VCC	12V PSU
		IN4	PWM 2
		IN3	PWM 3
		IN2	PWM 4
		IN1	D5
		GND	GND PSU
5.	Servo Motors 1	J1 pulse	D7
		VCC	12V PSU
		GND	GND
6.	Servo Motors 2	J2 pulse	D6
		VCC	12V PSU
		GND	GND

III.2. *Program Flowchart*

Software design is a program flow planning that will be made. The software design is shown in the program flowchart. The program begins with initializing (giving an initial value) to the input/output, then by pressing the ON button, the conveyor will activate then the colored objects enter, then connect to the internet network, esp-32 cam is connected to the internet network, if it is connected ESP-32 cams will detect objects in red, green, blue. If a red object is detected by the esp-32 cam, the LCD will issue the words "red: 1" and the servomotor 1 will move 45^o to make the red object fall into the first container. Likewise, the green object will be detected by the esp-32 cam then the LCD will issue the words "green: 1" and the servo motor 2 will move 45^o making the green object fall into the second container. Finally, if a blue object is detected by the esp-32 cam, the LCD will issue the words "blue: 1" and will immediately fell into the third container. The following is a flowchart of the Goods Sorting Conveyor Design Program with Color Detection Using the ESP-32 Camera based on OpenCV Python.

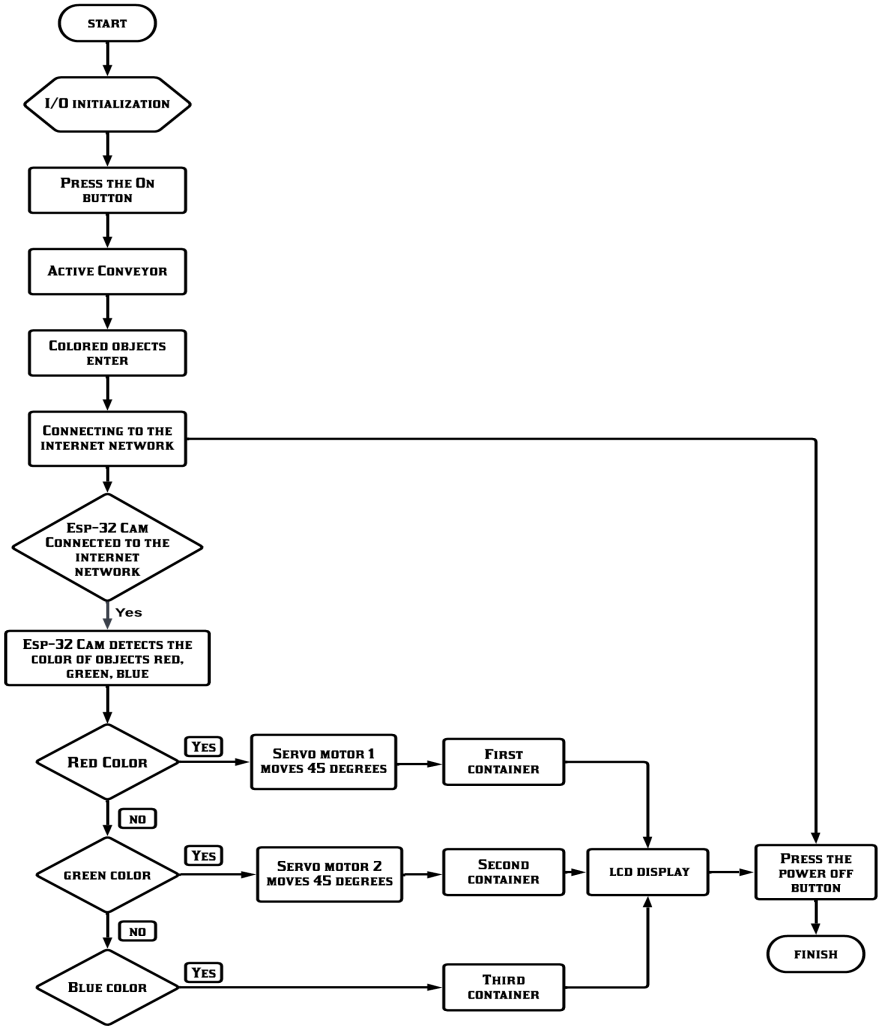

Fig 7. System Flowchart

III.3. *Equations*

The calculation of RGB conversion to HSV can be formulated as follows :

$$H = \tan\left(\frac{3(G-B)}{(R-G)+(R-B)}\right) \quad (1)$$

$$S = 1 - \frac{\min(R.G.B)}{v} \quad (2)$$

$$V = \frac{R+G+B}{3} \quad (3)$$

If the value of S = 0, then H cannot be determined. Therefore it is necessary to normalize RGB first using the following formula.

$$r = \frac{R}{(R+G+B)} \quad (4)$$

$$g = \frac{G}{(R+G+B)} \quad (5)$$

$$b = \frac{B}{(R+G+B)} \quad (6)$$

After the r, g, and b values are normalized, the RGB to HSV transformation formula becomes as follows.

$$v = \max(r, g, b) \quad (7)$$

$$s = \begin{cases} 1 - \frac{\min(r,g,b)}{v}, V > 0 \\ 0\ , jika\ V = 0 \end{cases} \quad (8)$$

IV. Results and Discussion

IV. 1. *Result of tool realization*

In the conveyor design, the long beam shape is adapted, but the size is larger because it is intended for packages with a length of 1 meter, a width of 40 cm and a height of 30 cm. With the Esp-32 camera to capture images of objects to detect color and count objects that are running on the conveyor placed on the conveyor placed on a 40 cm support. Arduino uno and 16x2 I2C LCD display are placed on the right side of the conveyor in a protective box so they are

Manuscript received January 2022, revised January 2022

not easily damaged. Behind the conveyor there are two dc motors on either side to drive the conveyor. At the front are placed 2 servo motors with a distance of 30 cm which are separated left-right. Here is the design of the package box :

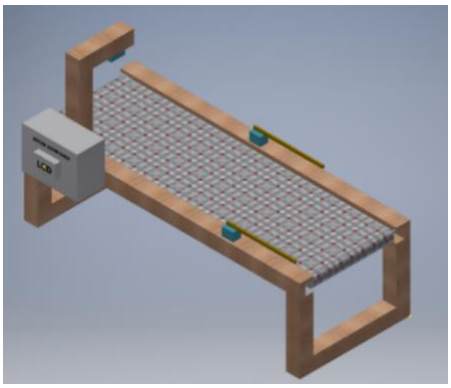

Fig 8. Design

Fig 9. Results

IV. 2. Program Results

First make a program on the Arduino IDE for programming the ESP-32 Cam connection with python, the following programs on the Arduino IDE and the python program are shown in figures (10) and (11) :

IP_ESP-32_CAMERA§ | CONVEYOR_UTAMA | LCD_DISPLAY | MOTOR_DC_DRIVER_MOTOR | SERVO_PENYORTIR

```
# GOODS SORTING CONVEYOR DESIGN PROGRAM WITH COLOR DETECTION USING OPEN-CV PYTHON BASED ESP-32 CAMERA #
                    # ELECTRICAL ENGINEERING THESIS EXAMINATION #
                       # HAFIDZ MAULANA ICHSAN_191020100083 #
                   # COLOR DETECTION AND OBJECT COUNTING PROGRAMS #

#include <WebServer.h>
#include <WiFi.h>
#include <esp32cam.h>

// Enter your wifi ID and Pass
const char* WIFI_SSID = "QQ"; //ID wifi
const char* WIFI_PASS = "open12345"; //Password wifi

WebServer server(80);

// Resolution for ESP-32 Camera
static auto loRes = esp32cam::Resolution::find(320, 240); //Low camera resolution
static auto midRes = esp32cam::Resolution::find(350, 530); //Medium camera resolution
static auto hiRes = esp32cam::Resolution::find(800, 600); //High camera resolution
void serveJpg()
{
  auto frame = esp32cam::capture();
  if (frame == nullptr) {
    Serial.println("CAPTURE FAIL");
    server.send(503, "", "");
    return;
  }
  Serial.printf("CAPTURE OK %dx%d %db\n", frame->getWidth(), frame->getHeight(),
                static_cast<int>(frame->size()));
```

Fig 10. programming on ESP32 Cam using Arduino IDE

```
# GOODS SORTING CONVEYOR DESIGN PROGRAM WITH COLOR DETECTION USING OPEN-CV PYTHON BASED ESP-32 CAMERA #
                    # ELECTRICAL ENGINEERING THESIS EXAMINATION #
                       # HAFIDZ MAULANA ICHSAN_191020100083 #
                   # COLOR DETECTION AND OBJECT COUNTING PROGRAMS #

import cv2
import urllib.request
import numpy as np
from typing import AsyncIterable
from pyfirmata import Arduino, SERVO, util
import time

def nothing(x):
    pass

# change the IP address below according to the
# IP shown in the Serial monitor of Arduino code

url = 'http://192.168.170.120/cam-hi.jpg'
##'''cam.bmp / cam-lo.jpg /cam-hi.jpg / cam.mjpeg '''
cv2.namedWindow("live transmission", cv2.WINDOW_AUTOSIZE)

# Counting Color
counter_green = 0  # Hijau
counter_blue = 0  # Biru
counter_red = 0  # Merah

# Line object detection
Deteksi_Objek = cv2.createBackgroundSubtractorMOG2()
##cek1 = 0
##cek2 = 0

# Arduino port declaration to computer
port = 'COM6'
pin1 = 10
pin2 = 11
board = Arduino(port)

board.digital[pin1].mode = SERVO
board.digital[pin2].mode = SERVO

while(1):
```

Fig 11. programming on ESP32 Cam using Python

Next is programming to determine the input value to get the desired color, namely RGB, by entering the Python OpenCV library, which is shown in figure (12) :

```
#_____Green Detection_____#
    # Set range for green color and
    # define mask
    green_lower = np.array([25, 52, 72], np.uint8)
    green_upper = np.array([102, 255, 255], np.uint8)
    green_mask = cv2.inRange(hsvFrame, green_lower, green_upper)

    # For green color
    green_mask = cv2.dilate(green_mask, kernal)
    res_green = cv2.bitwise_and(frame40, frame40, mask = green_mask)

    # Creating contour to track red color
    contours_green, hierarchy_green = cv2.findContours(green_mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    for pic, contour in enumerate(contours_green):
        area_green = cv2.contourArea(contour)
        if(area_green > 300):
                x, y, w, h = cv2.boundingRect(contour)
                frame40 = cv2.rectangle(frame40, (x, y), (x + w, y + h), (0, 255, 0), 2)

                cv2.putText(frame40, "Green Colour", (x, y), cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0, 255, 0))

                if((x+int(w/2))>int(Width/2)-15)and((x+int(w/2))<(int(Width/2)+30)):

                        counter_green = counter_green + 1

                print("Number of Green Objects = ", counter_green)

    # REPEAT COUNTING
    for cnt_green in contours_green:
        area_green = cv2.contourArea(cnt_green)
        if area_green < 3000 or area_green > 30000:
            continue
#_____Blue Detection_____#
    # Set range for blue color and
    # define mask
    blue_lower = np.array([94, 80, 2], np.uint8)
    blue_upper = np.array([120, 255, 255], np.uint8)
    blue_mask = cv2.inRange(hsvFrame, blue_lower, blue_upper)
```

Fig 12. OpenCV python programming

Next, the sorting program uses 2 servo motors in the python program, which is explained in figure (13) as follows :

```
    def rotateservo(pin1,angle):
        board.digital[pin1].write(angle)
    def rotateservo(pin2,angle):
        board.digital[pin2].write(angle)

    while True:
        x = input("input : ")
        if x=="1":
            for i in range(0,45):
                rotateservo(pin1,i)
        elif x=="2":
            for i in range(0,10):
                rotateservo(pin1,i)
        if x=="3":
            for i in range(0,45):
                rotateservo(pin2,i)
        elif x=="4":
            for i in range(0,10):
                rotateservo(pin2,i)
```

Fig 13. Servo motor sorting program

IV. 2. *Testing The ESP-32 Camera as RGB Color Reading*

In testing the ESP-32 Camera as an RGB color reading, it was carried out using Python and Arduino IDE software, the test was carried out to obtain the desired RGB color image data, the following are the results of the RGB color reading shown in figures 12, 13, 14 :

Fig 14. Blue Color

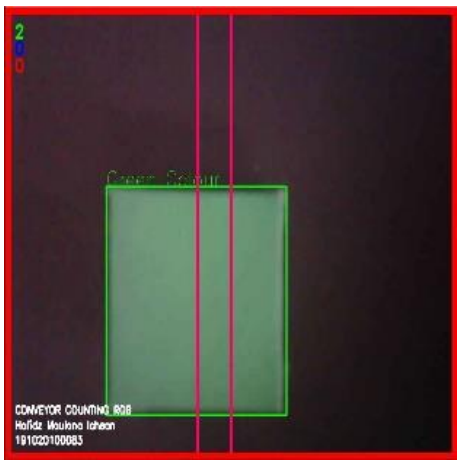

Fig 15. Green Color

Fig 16. Blue Color

From figure 12, 13, 14, the results of the accuracy of reading each RGB color from a distance of 12cm are obtained, described in table III as follows.

TABLE III
TESTING THE ESP-32 CAMERA AS RGB COLOR READING

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

The display in the python software shows that the ESP32 Cam is very susceptible to light readings, so noise often occurs when reading colors.

IV.3. *Sorted Object Count Test*

In this test for the calculation of items to be sorted. The results of the calculation of sorted items show the accuracy of capturing the suitability of every 10 objects according to the RGB color, which is explained in table IV as follows.

TABLE IV
COUNTED NUMBER OF COLORED OBJECTS

Counted number of colored objects		
Color	Number of colored objects (10 obejct)	Accuracy (%)
Red	8	80%
Green	10	100%
Blue	8	80%

From an experiment to find out the accuracy of the ESP32 Cam in calculating each RGB color, it was found that according to figure 13 above, the red color has an accuracy of 80%, the green color has 100%, and the blue color has 80%.

IV.4. *Sorter Accuracy Testing and Detection for sorted items*

In this test to measure the accuracy of sorting goods to be sorted. The results of sorting the goods show the accuracy of the items that are sorted every 10 times, described in table V as follows.

TABLE V
SORTER ACCURACY TESTING AND DETECTION FOR SORTED ITEMS

Color	testing	Sorter 1	Sorter 2	Sorter 3	Accuracy (%)
Red	10 times	8	-	-	80%
Green	10 times	-	10	-	100%
Blue	10 times	-	-	8	80%

From the experiment to find out the accuracy of the servo motor in sorting RGB color objects which was carried out 10 times, it was found that according to table 3 above, the red color has an accuracy of 80%, the green color is 100%, the blue color is 80%.

IV.5. *Overall Test Results*

The results of the calculation of sorted objects show the right accuracy, according to the desired color. From the three test results above, a graph of the overall research results is obtained, which is explained in figure 15 as follows:

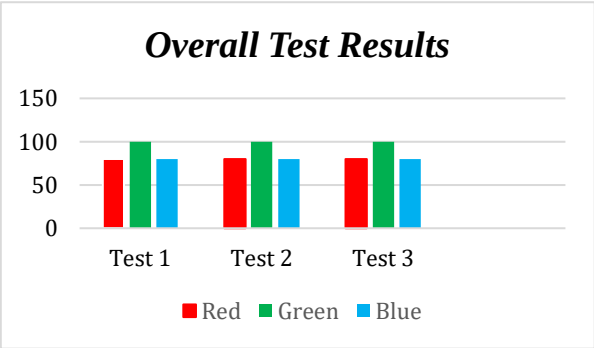

Fig 17. Overall Test Results

Conclusion

Based on the results of that test carried out in this study, can be obtained some conclusions as follows.

1. Image processing using the OpenCV method on the ESP-32 Camera can work well, with 3 color categories namely: red, green, blue.
2. The python program that has been created can function properly and can work according to what is designed.
3. With the OpenCV method for reading these 3 colors, it can easily separate red, green, blue objects.
4. there is a deficiency in the ESP32 Cam where it is very susceptible to light, which creates disturbances during the reading process.
5. ESP32 Cam also has a weakness against hot temperatures on the device, which will cause a delay during the reading process.

In subsequent studies, the camera used was added to make the recognition more precise and the enumeration patterns increased to minimize the influence of shadows that interfere with the image recognition process. Need to control the speed of the conveyor motor, so that the selection of objects can be better.

Acknowledgements

The author would like to thank profusely to all parties who have helped in writing this research, especially to the electrical engineering study program at the Universitas Muhammadiyah Sidoarjo.

References

[1] A. Safaris and H. Effendi, “Rancang Bangun Alat Kendali Sortir Barang Berdasarkan Empat Kode Warna,” *JTEV (Jurnal Tek. Elektro dan Vokasional)*, vol. 6, no. 2, pp. 399–410, 2020, [Online]. Available: http://ejournal.unp.ac.id/index.php/jtev/index

[2] D. M. Rajagukguk and I. Panjaitan, “Compression of Color Image Using Quantization Method,” vol. 10, no. 02, pp. 13–18, 2020.

[3] F. Jalled and I. Voronkov, “Object Detection using Image Processing,” pp. 1–6, 2016, [Online]. Available: http://arxiv.org/abs/1611.07791

[4] S. Suwarno and K. Kevin, “Analysis of Face Recognition Algorithm: Dlib and OpenCV,” *J. Informatics Telecommun. Eng.*, vol. 4, no. 1, pp. 173–184, 2020, doi: 10.31289/jite.v4i1.3865.

[5] S. Mala *et al.*, “Klasifikasi Kematangan Buah Pisang Tanduk Berdasarkan Warna Menggunakan Metode Hue Saturation Value (HSV) 1,” pp. 197–207.

[6] H. Soleh, E. W, H. Witarsa, M. Ferdian, D. Yuniarti, and C. Caroline, “Prototipe Penyortir Barang Berdasarkan Warna, Bentuk Dan Tinggi Berbasis Programmable Logic Controller (Plc) Dengan Penggerak Sistem Pneumatic,” *J. Mikrotiga*, vol. 1, no. 2, 2014.

[7] M. I. Sari, R. Handayani, S. Siregar, and B. Isnu, “Pemilah Benda Berdasarkan Warna Menggunakan Sensor Warna TCS3200,” *TELKA - Telekomun. Elektron. Komputasi dan Kontrol*, vol. 4, no. 2, pp. 85–90, 2018, doi: 10.15575/telka.v4n2.85-90.

[8] F. R. Wicaksono *et al.*, “Perancangan Dan Implementasi Alat Penyortir Barang Pada Design and Implementation of Items Device Sorting on Conveyor,” *JTEV (Jurnal Tek. Elektro dan Vokasional)*, vol. 5, no. 1, pp. 40–47, 2018.

[9] Fauzan, “Menggunakan Modul Esp32-Cam,” pp. 393–400, 2020.

[10] M. D. Cookson and P. M. R. Stirk, "ZOOMORPHIC MOBILE ROBOT DEVELOPMENT FOR VERTICAL MOVEMENT BASED ON ESP 32-CAM Vladyslav," 2019.
[11] V. No, J. Hal, A. Mega, O. A. Putra, and R. Handika,"RANCANG BANGUN SISTEM KEAMANAN LALU LINTAS MENGGUNAKAN SMARTPHONE DAN ESP32CAM BERBASIS," vol. 2, no. 1, pp. 120–130, 2022.
[12] A. Mordvintsev and K. Abid, "OpenCV-Python Tutorials Documentation," *OpenCV Python Doc.*, p. 269, 2017,[Online].Available:https://media.readthedocs.org/pdf/opencv-python-tutroals/latest/opencv-python tutroals.pdf
[13] S. Tinggi and I. Komputer, "Jaringan Syaraf Tiruan LVQ Berbasis Parameter HSV dalam Penentuan Uang Jaringan Syaraf Tiruan LVQ Berbasis Parameter HSV dalam Penentuan Uang Rupiah Palsu," vol. 13, no. January 2019, pp. 47–52, 2020.
[14] A. M. Elhanafi, R. Siregar, M. Yeni, and S. Annissa, "Cryptography Application on RGB Overlapping Block Based PVD Using AES," vol. 7, no. 3, pp. 2116–2124, 2022.
[15] D. Sonny Febriyan and R. Dwi Puriyanto, "Implementation of DC Motor PID Control On Conveyor for Separating Potato Seeds by Weight," *Int. J. Robot. Control Syst.*, vol. 1, no. 1, pp. 15–26, 2021, doi: 10.31763/ijrcs.v1i1.221.
[16] M. Z. Andrekha and Y. Huda, "Deteksi Warna Manggis Menggunakan Pengolahan Citra dengan Opencv Python," *Voteteknika (Vocational Tek. Elektron. dan Inform.*, vol. 9, no. 4, p. 27, 2021, doi: 10.24036/voteteknika.v9i4.114251.
[17] N. M. Syahrian, P. Risma, and T. Dewi, "Vision-Based Pipe Monitoring Robot for Crack Detection Using Canny Edge Detection Method as an Image Processing Technique," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 2, no. 4, pp. 243–250, 2017, doi: 10.22219/kinetik.v2i4.243.
[18] R. RAGIL FANNY SETIYA AJI and I. Sulistiyowati, "Mesin Penetas Telur Burung Murai Batu Dengan Monitoring Camera ESP32 Berbasis IoT," *JASEE J. Appl. Sci. Electr. Eng.*, vol. 2, no. 02, pp. 87–99, 2021, doi: 10.31328/jasee.v2i02.173.
[19] S. C. S. Yanti and I. Sulistiyowati, "An Inventory Tool for Receiving Practicum Report Based on IoT by Using ESP32-CAM and UV Sterilizer: A Case Study at Muhammadiyah University of Sidoarjo," *J. Electr. Technol. UMY*, vol. 6, no. 1, pp. 49–56, 2022, doi: 10.18196/jet.v6i1.14607.
[20] S. W. Utama and A. Kusumawardhani, "Aplikasi Pendeteksi Plat Nomor Negara Indonesia Menggunakan OpenCV dan Tesseract OCR pada Android Studio," no. December, pp. 1–6, 2018.
[21] H. S. Ghifari and F. Utaminingrum, "Klasifikasi Kualitas Minyak Goreng berdasarkan Fitur Warna dan Kejernihan dengan Metode K-Nearest Neighbour berbasis Arduino Uno," vol. 6, no. 7, pp. 3269–3274, 2022.
[22] R. I. Gimeno, "TREBALL DE FI D ' ESTUDIS," 2021.
[23] H. Rini, Puspa dan Sugiarto, "Perancangan Sistem Informasi Pembelian Bahan Baku Roller Pada PT.Vortex Conveyor International," *IJCIT (Indonesian J. Comput. Inf. Technol.*, vol. 3, no. 2, pp. 157–164, 2018.
[24] F. E. Saputra, R. Cahya Wihandika, and A. W. Widodo, "Penentuan Kualitas Biji Kopi Menggunakan Local Ternary Patterns Dan RGB-HSV Color Moment Dengan Learning Vector Quantization," vol. 5, no. 6, pp. 2299–2307, 2021, [Online]. Available: http://j-ptiik.ub.ac.id

Authors' information

Hafidz Maulana Ichsan was born in Jombang, August 29 2001, currently studying at the Muhammadiyah Sidoarjo University majoring in electrical engineering, he was chairman of the Sidoarjo Student Association in 2020, he has participated in the POSI competition and won silver and bronze medals.

Indah Sulistiyowati. is a researcher who specializes in Renewable Energy Power Systems and Control Systems. Completed her master's degree at Institut Teknologi Sepuluh Nopember majoring Control System Electrical Engineering Subsidiary, and now is a lecturer at Department of Electrical Engineering, Faculty of Science and Technology Universitas Muhammadiyah Sidoarjo.

Izza Anshory was born on December 9, 1975, currently serves as head of the electrical engineering study program and is a lecturer in electrical engineering at the Muhammadiyah University of Sidoarjo, he has made many journals, the latest in 2021 entitled Transfer function modeling and optimization speed response of bldc e-bike motors using intelligent controllers published in the Journal of Engineering Science and Technology.