Engineering and Technology Journal
Journal homepage: https://etj.uotechnology.edu.iq

Smart Robot Vision for a Pick and Place Robotic System

Momena M. Mohammed*, Mohanned M. H. Al-Khafaji, Tahseen F. Abbas

Production Engineering and Metallurgy Dept., University of Technology-Iraq, Alsina'a street, 10066 Baghdad, Iraq.
*Corresponding author Email: pme.19.12@grad.uotechnology.edu.iq.

HIGHLIGHTS

- The movement of the 5 DOF robotic arm was controlled through the geometric approach in inverse kinematics analysis
- HSV color space, a series of filters (Median, Bilateral, and Gaussian), and the draw-contour method to discover objects' colors, shapes, and centroid were implemented.
- The shapes and colors of the objects in different lighting conditions ranging from 5 to 7000 lux with an accuracy of 93.83% were discovered.

ARTICLE INFO

Handling editor: Omar Hassoon

Keywords:
Robot vision; Raspberry pi; Arduino; different illumination conditions; HSV.

ABSTRACT

The main contribution of this paper is to develop an innovative algorithm to accurately detect and identify the shape and color of objects under various light intensities and find their location to be manipulated by a pick-and-place robotic arm. Workpieces of various shapes and colors are dispersed on the robot's work plane and manipulated according to its specifications. The proposed algorithm utilizes the HSV color model to distinguish between different object colors and shapes. The S channel is used to detect the shapes of objects. After that, a series of filters (Median, Bilateral, and Gaussian) are applied to reduce the noise of the segmented image to make the process of discovering the shape and coordinates of the objects successful. The draw-contour method is used to discover the object's shapes. After the shape of the object is discovered, the centroid coordinates are calculated. After extensive testing on 354 images that are captured in various lighting conditions in the range of (5-7000 lux), the overall system performance of 93.83% is achieved, and the average execution time is 2.21s. Finally, we had a dependable flexible automatic pick and place system that could correctly detect and identify the objects based on their features.

1. Introduction

In the field of industrial applications, with the invention and evolution of the industrial revolution, industries now have a wide range of robotic systems, the majority of which perform tedious tasks. Thus, Cognitive abilities must be added to create more intelligent systems and eliminate the tedious behavior of manipulators [1] s. In these applications, there is a need for object recognition systems where different objects of variable shapes and sizes should be handled [2]. For this reason, computer vision has grown in popularity in industrial applications of robotic arms such as packaging, sorting, and others [3]. Where Image-based systems can automate this application by controlling the industrial robotic arms [4].

To implement the object recognition process, Perfect lighting is required for system operation. The lighting system is one of the most important factors that affect features from extraction from captured images [5]. Thus, the change in the intensity of the lighting will hinder the process of image processing and thus impede the process of picking up and location. The real challenge is how to improve automated system applications which can work in different light conditions [6].

Furthermore, low light is a challenging environment for image processing and computer vision tasks, either in contrast enhancement for better visibility and quality, or application-oriented tasks such as detection [7]. Moreover, images and videos captured under weak illumination conditions often suffer from noticeable degradation of visibility, brightness, and contrast. Existing methods show limitations when they are used to enhance weakly illuminated images, especially for the images captured under diverse illumination circumstances [8]. Therefore, computer vision algorithms that can assist in such conditions are highly valuable [7].

The rest of this paper is organized as follows. Section 2 introduces the related works and the motivation for this research. Materials and Methods are presented in Section 3. Section 4 detailed the proposed algorithm. Section 5 discusses the results and finally, Conclusions are given.

http://doi.org/10.30684/etj.2023.135966.1292
Received 16 September 2022; Accepted 13 January 2023; Available online 28 January 2023
2412-0758/University of Technology-Iraq, Baghdad, Iraq

2. Related Works

Recent years have shown extensive research interest in the development of image processing in industrial robotic applications to detect the color and geometric shape of products for sorting operations. Kumar et al. [9] developed the image processing algorithm for a pick-and-place robotic arm to sort objects based on their types and coordinates. Qul'am et al. [10] discussed the use of the edge detection method as visual input for a 4 DOF robot manipulator tasked with picking and placing a tomato among three other objects. Chakole and Ukani [11] Described and explained a vision-based pick-and-place robot using an industrial ABB IRB 140 robot and a web camera. Shankar et al. [12] combined image processing with robotic arms to identify the colors of objects and classify them appropriately. Ali et al. [13] presented a robotic arm-based vision-based autonomous object sorting system using an industrial robot named ScorbotER 9Pro. Abbas et al. [3] presented an image-processing-based mechatronic sorting system. Hameed et al. [14] presented a method for identifying and determining the orientation of an object in a scene.

2.1 Motivation

Although these research and excellent works have reviewed the methods of object detection, and have shown good performance, especially in controlled conditions. Nevertheless, accurate detection is still hard to do in uncontrolled conditions. No research presents a complete method of the existing different methods in the case of different light intensities. While the lighting system is one of the most important factors affecting feature extraction from the captured image [5]. Motivated by these studies, the present study aimed to design an efficient robotic system that can recognize an object in various light conditions. According to the proposed system, the developed algorithm is used for color and shape detection in different light conditions from captured images and the details are presented in section 4.

3. Materials and Methods

In this section, the developed algorithm is described. This algorithm is used to detect the color and shape of captured images in different light conditions to provide an intelligent system for the pick-and-place robotic process. The details of this proposed system are explained detailed in the following subsections.

3.1 Pick and Place Robotic System

The main objective of this research is to control the robotic arm to perform exact movements for picking up color (red, blue, green, and orange) objects with different geometric shapes (square, cylinder, and rectangle) from the conveyor belt and put it into an assigned discharge station.

In this work, a webcam mounted above the conveyor belt and connected via USB to an RPi is used as a sensor input. The images of the object captured by the camera were sent to a Python program to extract the shape, color, and centroid of the objects, as outlined in Section 4. A block diagram of the essential components of a robotic system is depicted in Figure 1.

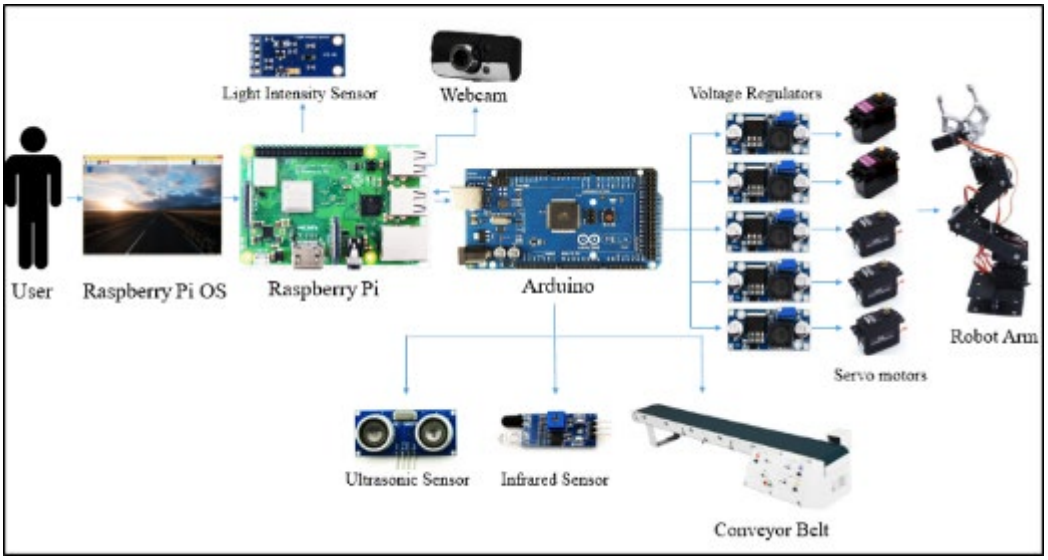

Figure 1: The block diagram of the robotic system's essential components

Two controllers RPi /Arduino are used in the master/slave configuration to perform the automated process. The RPi takes a command from the user to start the automated process and passes this command to Arduino through serial communication between them. After that, the Arduino starts performing the predefined pick and place process. The 5 DOF robotic arm is used to perform the pick and place process. The coordinate frame assignment is shown in Figure 2, while table 1 indicates the DH parameter of the arm. When the moving object on the conveyor belt is being sensed by the IR sensor, the Arduino stops the movement of the conveyor belt. The ultrasonic sensor starts measuring the high of the object. After that Arduino sends a command to RPi to start image processing through python code to determine the object's shape and color, the details are

presented in subsection 4.1. Image processing output data are sent through the serial port to the robot controller (Arduino) to perform pick operation through the inverse kinematic and place the object in the desired position.

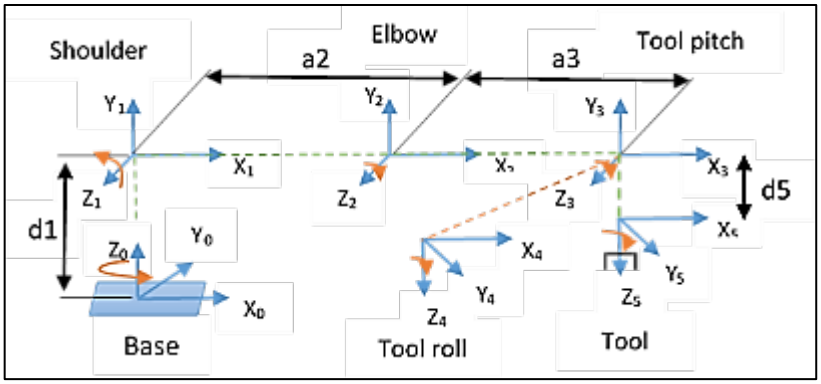

Figure 2: Coordinate Frame Assignment

Table 1: D-H parameters for the robotic arm

link	αi(mm)	*αi*(degree)	di(mm)	*θi*(degree)
1	0	90	105	θ 1
2	105	0	0	θ 2
3	100	0	0	θ 3
4	0	90	0	θ 4
5	0	0	150	θ 5

To control the movement of this arm the geometric approach in inverse kinematics analysis, which is illustrated in the following section, is used to find the joint angles of the robotic arm.

3.1.1 Inverse kinematic analysis

Inverse kinematics is concerned with determining the required joint angles to accomplish a specific end-effector location and orientation in Cartesian space. When the final position and orientation of the end-effector are specified, inverse kinematics can be calculated using a geometric approach with the input data (a2, a3, d1, d5):

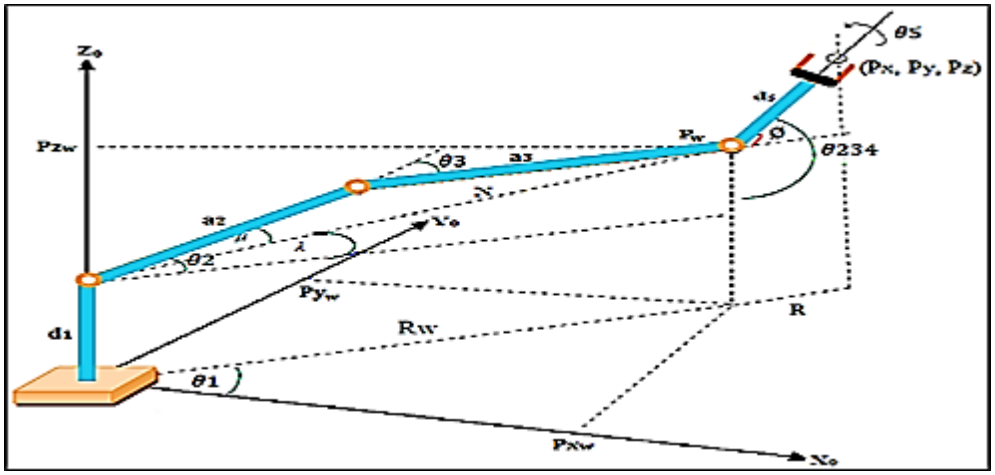

Figure 3: The 4-link articulated robot [15]

From Figure 3 the relation between (θ_2, θ_3 and θ_4) is:

$$\theta_{234} = \theta_2 + \theta_3 + \theta_4 \quad (1)$$

Where θ_{234} can be calculated based on pitch wrist orientation angle $\emptyset$

$$90 - \theta_{234} = \pm\emptyset \quad (2)$$

The following steps are used to determine the articulated robot's joint angles:

$$Pz_w = Pz + d_5 \sin\emptyset$$

The $$R_w = \sqrt{Px^2 + Py^2} - d_5 * \cos \emptyset \quad (4)$$

$$N = \sqrt{(Pz_w - d_1)^2 + R_w^2} \quad (5)$$

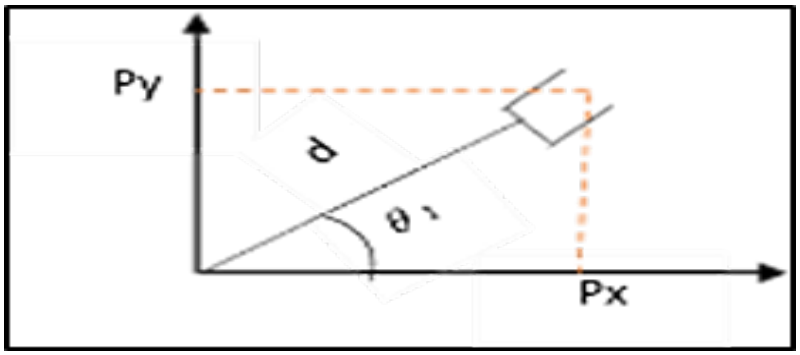

Figure 4: Top View of Robot

Step1: Solution for θ_1
From Figure 4, θ1 can be calculated from the following equation

$$\theta_1 = tan^{-1}\left(\frac{py}{px}\right) \quad (6)$$

Step 2: Solution for θ_2
By using the law of cosines:

$$\mu = cos^{-1}\left(\frac{N^2 + a_2^2 - a_3^2}{2a_2N}\right) \quad (7)$$

$$\lambda = tan^{-1}\left(\frac{Pz_w - d_1}{R_w}\right) \quad (8)$$

$$\theta_2 = \lambda \mp \mu \quad (9)$$

Step 3: Solution for θ_3

$$\theta_3 = \pm cos^{-1}\left(\frac{N^2 - a_2^2 - a_3^2}{2a_2a_3}\right) \quad (10)$$

Step 4: Solution forθ_4

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 \quad (11)$$

Thus, the robotic arm joins angles can be determined by utilizing the above equations with the known variables (d1, d5, a2, and a3), In addition to the known desired position Px, Py, Pz.

4. Image Processing and Feature Extraction Algorithm

Image processing is the process of converting an image into a digital format and applying specific operations to it to obtain an improved image or information from it [16]. In this work, the implementation of the pick-and-place process depends on the results of the image processing. The results of image processing include the object's shape, color, and coordinates so that the Robotic Arm can perform the pick-and-place task. Shape recognition is based on the 2D image of the workpieces. While color recognition relies on the hue value of the HSV color model. Additionally, the computation of the workpiece's centroid, which is used to identify the workpiece's location on the work plane, is presented. The software was developed in the Thonny python program in RPi using the OpenCV library. The block diagram of the image processing process is shown in Figure 5, while the flowchart of the image processing is depicted in Figure 6 and the details of this flowchart are explained detailed in the following subsections.

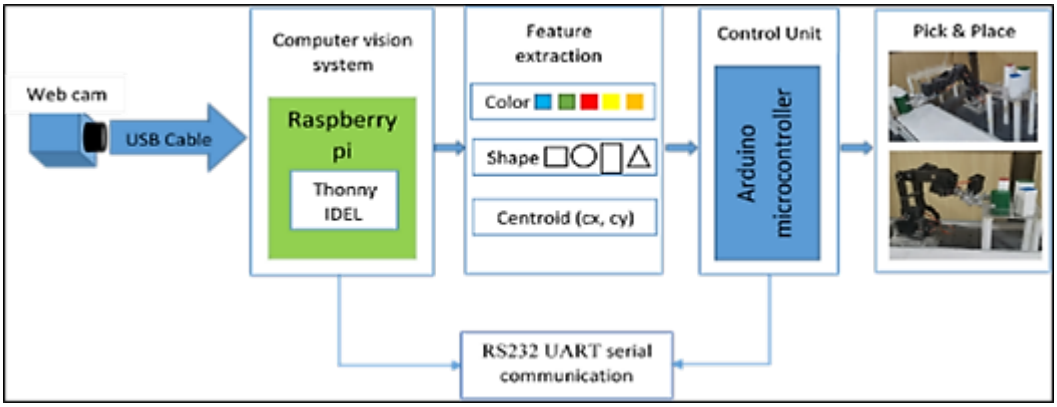

Figure 5: The block diagram of the image processing

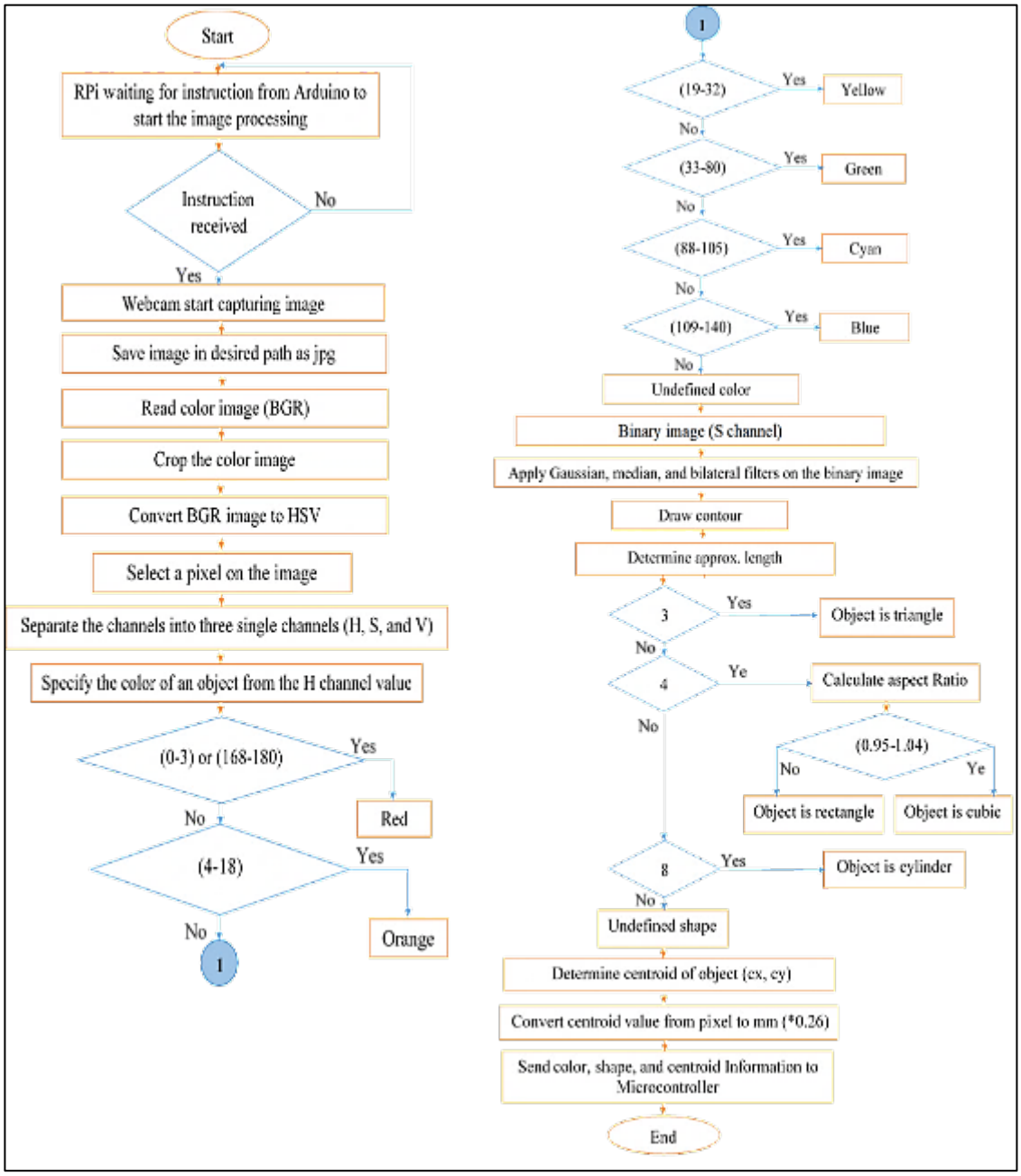

Figure 6: Flowchart of the image processing

4.1 Algorithm Description

The proposed algorithm included three stages: a color detection stage, a shape detection stage, and a centroid detection stage. The color detection stage was performed using HSV color space analysis. The shape detection stage was performed using contour drawing techniques. The image processing steps are illustrated in the following subsections.

4.1.1 Image acquisition

Image acquisition is the first step in the work. A web camera with a maximum video resolution of 480 x 640 pixels and an effective photo resolution of 8 MP was used to acquire the images under a variety of illuminations and saved them in JPG format. A top-view camera position was adopted and the camera was mounted above the conveyor belt. At a certain instant, it captured the image of the objects placed on the conveyor belt in front of the robot. The objects were wood objects with several shapes, sizes, and colors as shown in Figure 7.

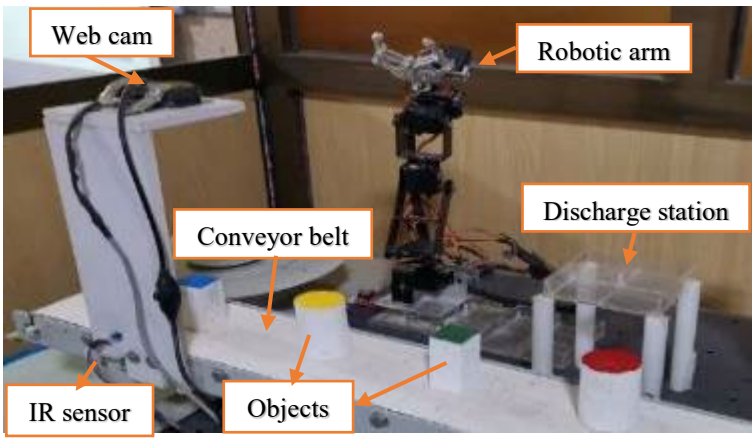

Figure 7: Pick and place robotic system with the test objects

4.1.2 Image pre-processing

Following image acquisition is image preprocessing. Original camera images are frequently required to be pre-processed to facilitate image segmentation [17]. Pre-processing steps are applied to enhance the visual appearance for image segmentation. The image pre-processing steps used for this project are image cropping, color conversion, and image filtering.

4.1.2.1 Cropping

Cropping is a technique that splits a given image into small regions to obtain a more easily processed image representation. This splitting process allows unwanted parts of the images to be removed, leaving only the object of interest [18]. In this work, the image is cropped to 240x240 pixels to remove the unwanted parts that appear in the image and influence the extraction process like threshold, contour drawing, and color detection.

4.1.2.2 Color Conversion

In this work, mapping from BGR to HSV is chosen due to its uncorrelated Color Space, which means that the modification to one channel does not affect the image quality [19]. Furthermore, HSV yields better results for image segmentation than the RGB color space. Moreover, HSV images could be more suitable for feature extraction and classification purposes [20]. It is robust to illumination variation [17]. In addition, HSV-based detection is best suited for simple images with uniform backgrounds. Moreover, if there is a lot of fluctuation in the value of the color information (hue and saturation), pixels with small and large intensities are not considered [21]. Figure 8 shows the split channel of the HSV color model.

The HSV color space channels are as follows: Hue: Describes a pure color. The lowest and highest values for the H space are and (0-180). While Saturation: Describes how much a pure color is diluted with white light. Value: Refers to the brightness of the color. S and V have minimum and maximum values ranging from 0 to 255 [22]. For details, readers may refer to Reference [23].

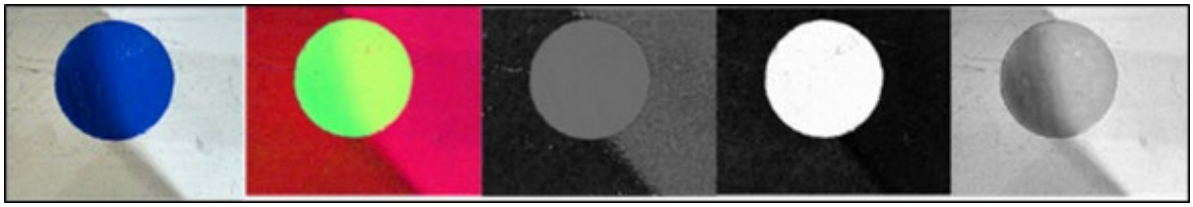

Figure 8: The conversion from BGR to HSV color model

4.1.3 Color feature extraction

In this work, the parts move on the conveyor belt and when they are detected by the IR sensor, the conveyor stops moving and the camera starts taking pictures to perform the image processing process. This means that the objects will stop at approximately the same place. Thus, a pixel on the image has been selected. The location of this pixel has been determined by trial and error method to ensure that it will be within all objects.

To extract color information, the values of the channels H, S, and V of the previously selected pixels are read. The color information is extracted from the H channel. According to this value, the color is specified. If the value of the channel ranges from (4-18), it is marked as orange. If the channel reading is (19-32) it is marked as yellow. If the reading of the channel ranges (33-80) then it is distinguished as green. If the reading of the channel ranges (109-140) then it is distinguished as blue. If the reading of the channel ranges (0-3) or (168-180) then it is distinguished as red. This value is selected based on 324 tested images in the different light conditions as shown in Figure 9, in the range (5-7000 lux).

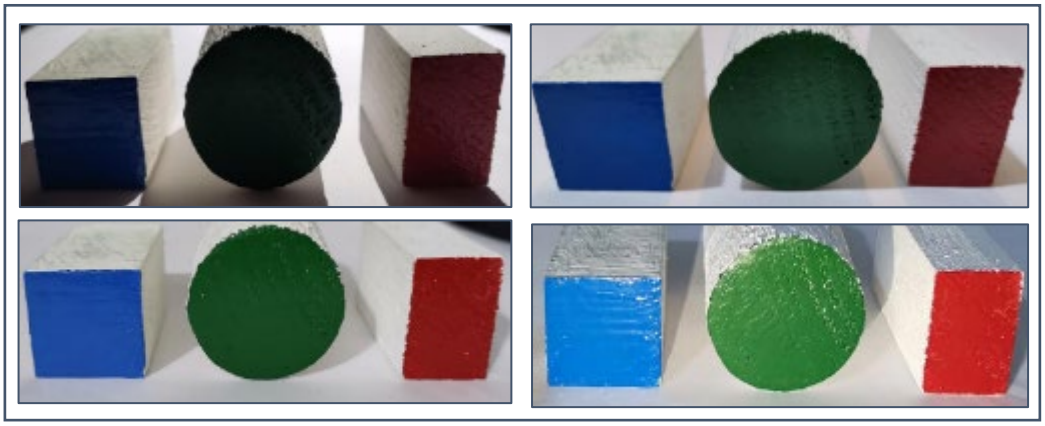

Figure 9: Training images under different light illumination

4.1.4 Shape feature extraction

To extract the shape of the object, there are several steps should be applied to the image. These steps are image segmentation, noise filtering, and then drawing contouring to have the shape. All these steps are illustrated in the following subsections.

4.1.4.1 Image segmentation

One of the most crucial stages in artificial vision systems' object recognition is image segmentation [22]. It's a very important part of image analysis and computer vision [24]. It is the separation of an object of interest from its background. There are various methods of segmentation [25]. Thresholding is one of the most significant and efficient image segmentation tools [24]. It is commonly used as the first step in a variety of algorithms for image analysis, object representation, and visualization [26]. It should be noted that the majority of this technique's success stems from the fact that all environmental factors, including background colors and lighting, among others, are carefully monitored and controlled to produce the best outcomes [16]. Finding suitable threshold values is crucial for the threshold-based approach. An easy method for determining thresholds is based on empirical knowledge and produces a fixed threshold, but this fixed value is sensitive to changes in lighting and is only appropriate for specific applications [17].

To make the proposed algorithm work well in a wide range of lighting conditions, there is a need for a dynamic and automatic threshold calculating approach to make the proposed algorithm robust to the various illumination conditions. The S channel is a good option for automatically determining a light-dependent threshold because it correlates with the light level. Therefore, it was chosen to perform the image segmentation process on it.

4.1.4.2 Noise filtering

Eliminating noise from images is one of the most crucial topics in the field of image processing. Image denoising is a pre-process in image processing that attempts to get an image as close to the real one as possible by removing the noise. The success rate of image segmentation, classification, and other similar procedures is impacted by the success of the image denoising procedure. Impulse noise can be caused by a variety of factors, including faulty pixels in camera sensors or memory locations in faulty hardware, and it can cause images to be distorted. Salt and pepper noise (SAP) and random valued noise are two types of impulse noise [27]. Different filters should be used depending on the noise type [18].

There are different filtering techniques like averaging filter, median filter, Wiener filter, adaptive filter, etc. [28] performed a comprehensive comparison to cover all the denoising methods in detail and the results they yield. Furthermore, it summarized the progress that has been made over the years in all applications involving image processing.

According to [18], the filters are selected according to the intensity of the lighting of the captured image. In this work, a series of filters applied on the S channel as shown in Figure 10 to eliminate noise and undesirable elements in the image.

Figure 10: Filters on the S channel

Median Filter (MF) is the most common filter for removing salt and pepper (SAP) noise and its derivatives. [27]. A bilateral filter is an edge-preserving denoising filter. Its performance greatly depends upon the selection of spatial and radiometric parameters [29]. Gaussian filter is the most common approach to apply In the case of noise filtering [18].

All these filters are applied to the segmentation image. This method is reached through an experiment on all captured images under different light conditions. Until a successful way has been reached to obtain a clear image without noise, through which the shape information can be extracted correctly for a group of shapes in different images.

Before applying the proposed method with test images, the algorithm parameters were optimized using training images captured under varying lighting conditions. During the testing phase, the system is evaluated with a completely different set of images using the parameters chosen with the training data.

The selection of filter parameters is done through the trial and error method. The dependency of the bilateral filter lies on two controlling parameters such as σs and σr. these parameters influence the entire process of filtering.

4.1.5 Shape feature extraction

After segmentation of the image, it will be easy to extract the shape of the object using the Open CV findContours function. which is about enclosing the change in the value of the pixels in the image, where curves or lines surrounding the shape are drawn, and based on contour approximation values, the shape of each element is determined, if the approximation value is equal to 3 then the shape is a triangle if it is equal to 4 the aspect ratio value is calculated to determine if the shape is square or rectangular, if the value is equal to 8 then the shape is a circle, and finally if the approximation number is not equal to the above values then the shape is defined as the undefined shape. These values depend on the resolution of the camera used.

It is worth mentioning that the previous steps play an important role in the success of extracting the shape feature from the image. Where a lack of image segmentation and noise in the image will cause unclosed or distorted drawn contour which affects the shape detection process.

4.1.6 Centroid features

The final step is to determine the centroid of the objects which enables the manipulator to pick the objects and place them in the predetermined locations using the given coordinates. The centroid of the object is given by the cv2.moments () function. The value obtained of Xc and Yc are in pixels, thus to convert it to millimeters it will multiply by 0.26. Figure 11, illustrates the importance of each step to the success of the process of extracting information from the captured image.

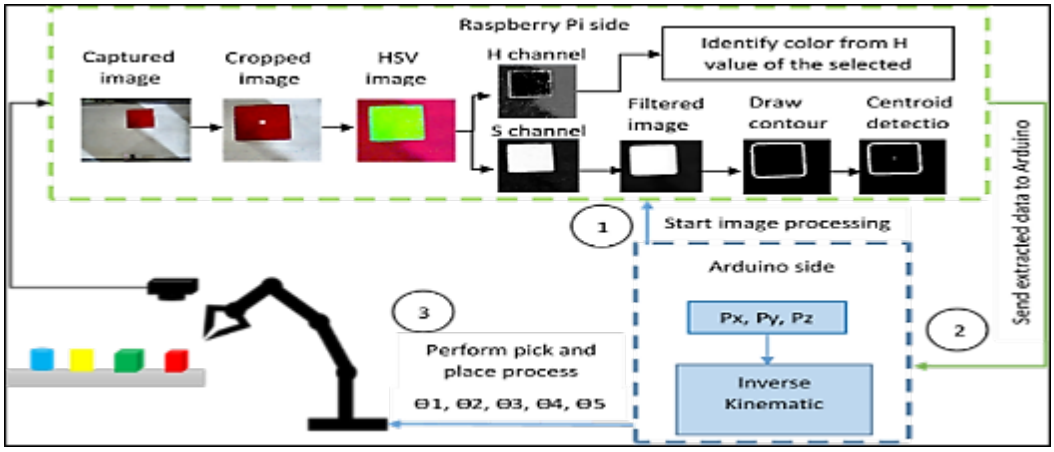

Figure 11: The block diagram of the entire pick and place process

5. Results and Discussion

The algorithm proposed in the paper is trained under 324 images under different illumination conditions in indoor and outdoor environments. The results are presented in the following sections.

5.1 Features Detection Results Under Different Illumination Conditions

In this section, we demonstrate results that represent our findings from these experiments. The system accuracy is measured using the following Equation [30]:

$$Accuracy = \frac{total\ number\ of\ images - total\ number\ of\ erroe\ images}{total\ numner\ of\ images} \times \frac{100}{1} \quad (12)$$

5.1.1 Accuracy of color feature detection

The graph below shows the results obtained from examining the training images of five different colors (red, blue, green, yellow, and orange) with different surface roughness and under different lighting conditions that ranged from (5-7000 lux) to check the accuracy of the developed algorithm. Through the results obtained from the experiments that are conducted, it is found that the selected channel can detect colors in varying illumination conditions. As shown in Figure 12, all five colors Red, Blue, Green, Yellow, and Orange have the highest detection accuracy of 100% when the brightness is around 500-1000 lux. While the lowest accuracy of the system was from 2001 to 3000 lux, which is 81.8%. In addition, it can be noted that the yellow color had the highest detection accuracy under all conditions, as the total accuracy of this color reached 98.88%. While the lowest detection accuracy of the red color was 94.84%. As for the rest of the colors, it was 98.44% for the blue color, 97.43 for the orange color, and 96.15% for the green color. Finally, the overall accuracy of the algorithm for detecting colors is 97.11%.

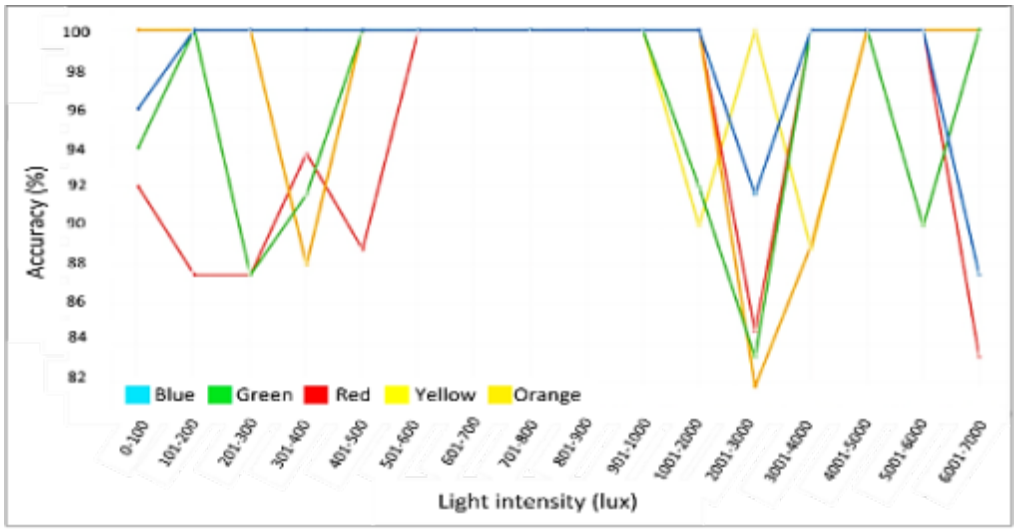

Figure 12: Color detection accuracy vs. light intensity

5.1.2 Segmentation and shape detection results

As for shape detection results, Figure 13 shows the accuracy of the algorithm for three shapes (square, rectangle, and circle) detection under different lighting conditions. It can be seen that the largest accuracy obtained for the three shapes is at 4001-7000 lux .Which is 100%. While the lowest accuracy is from 601 to 700 lux, which is 41% for the square object. As for the overall accuracy of each shape in all specified illumination conditions, it was found that the highest detection accuracy is for the rectangular shape which is 92.27%. While less accuracy was for the square shape at 86.10%. As for the circular shape, it has reached 87.73% accuracy. Thus, the overall accuracy of the algorithm for detecting shapes is 88.70%.

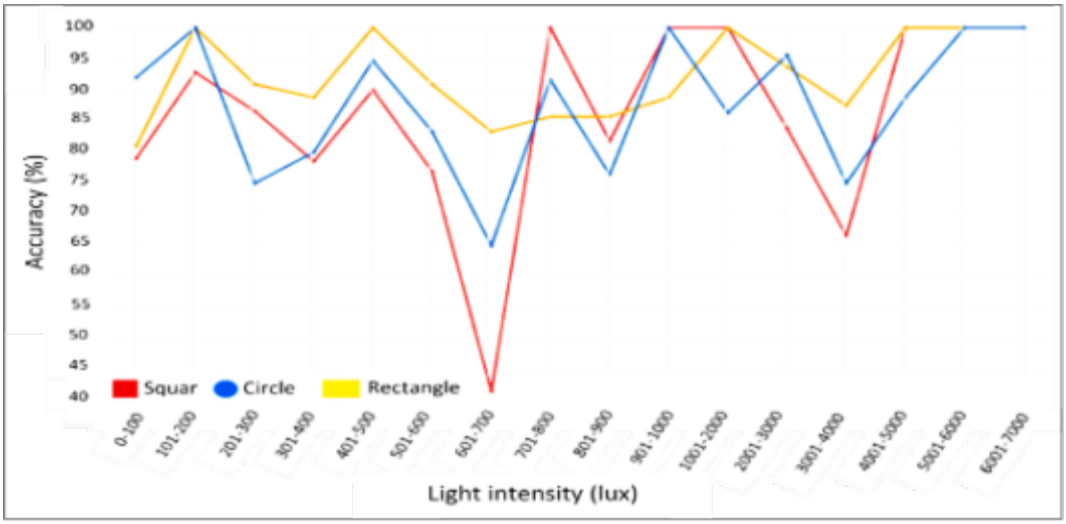

Figure 13: Shape detection accuracy vs. light intensity

5.1.3 Centroid detection results

The accuracy of discovering the center of the objects is the same as the accuracy of discovering the shapes, as its accuracy depends on the drawing of an integrated contour of the part. Figure 14 shows the results of detecting the x and y coordinate positions of the different object shapes, and as it is detected, Raspberry Pi sends these coordinates to the Arduino microcontroller to move the manipulator to grasp the object and perform the sorting process.

Figure 14: Centroid detection for different shapes

5.2 Result Of Pick and Place Process

The final step is to test the system in a real environment to ensure the proposed algorithm's accuracy. The presented image processing methodology has been validated by the two experiments as shown in Figure 15(a) and (b), conducted to evaluate the developed algorithm. The captured images are shown in Figures 16 and 17, and the image processing results are shown in Tables 2 and 3.

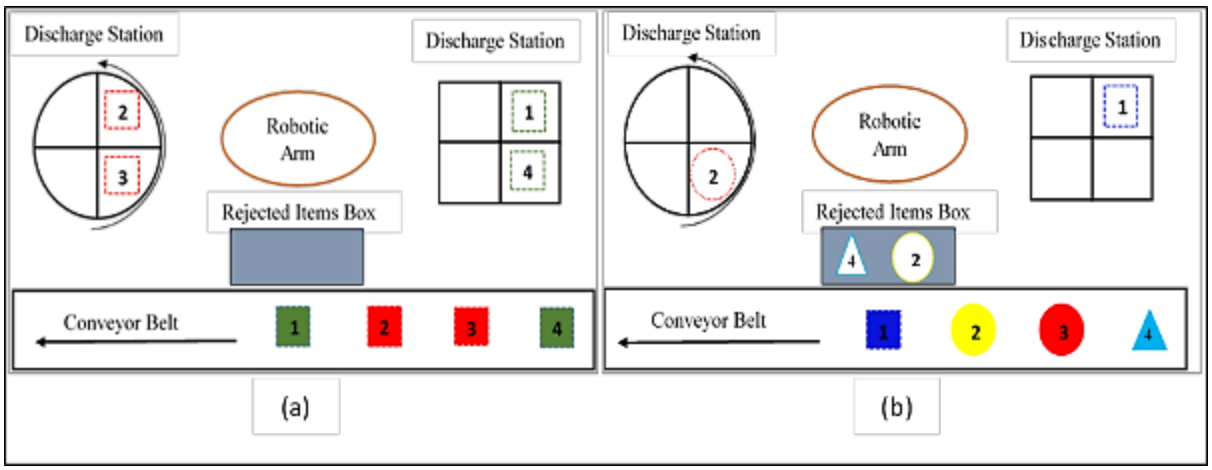

Figure 15: The strategy of pick and place: (a) for case 1, (b) for case 2

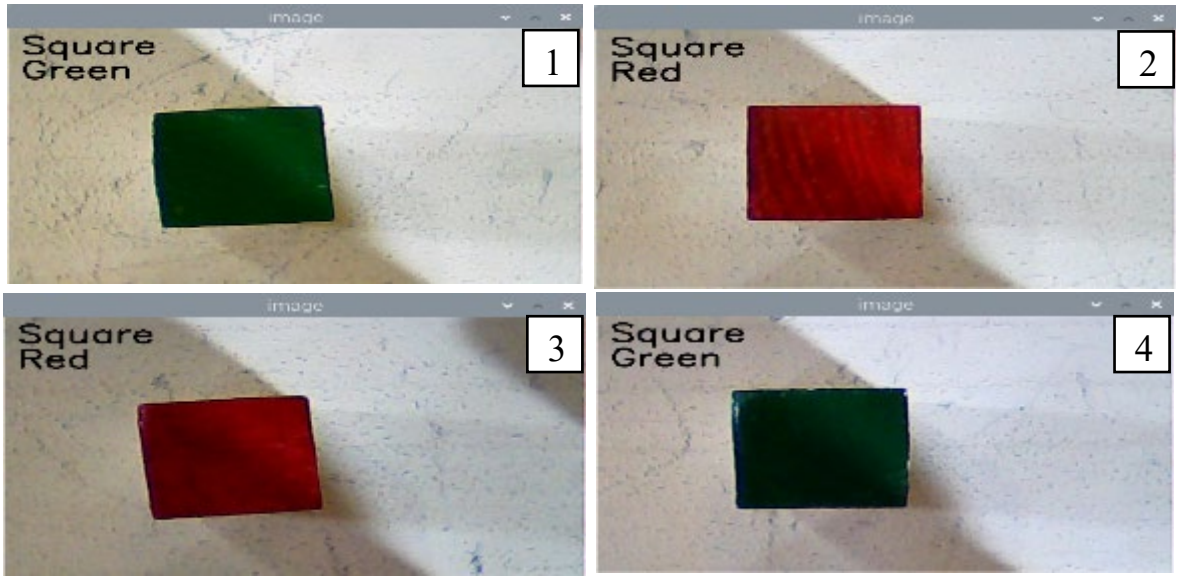

Figure 16: Image-processing output of case (1)

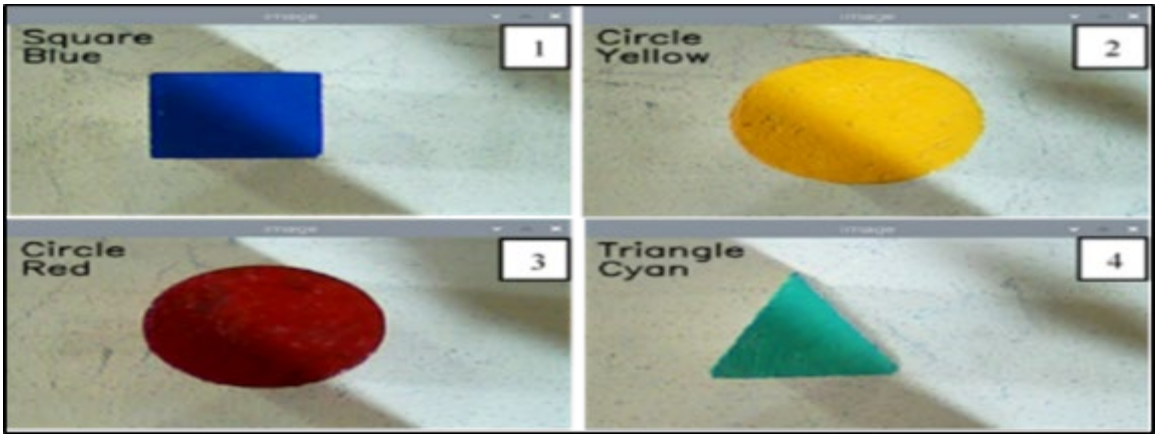

Figure 17: Image-processing output of case (2)

Table 2: Image-processing output of case (1)

Item	Intensity (lux)	color	shape	Centroid (cx, cy)
1	70	Green	Square	(48.36, 43.42)
2	70	Red	Square	(48.88, 41.08)
3	70	Red	Square	(46.02,43.42)
4	70	Green	Square	(46.28, 41.34)

Table 3: Image-processing output of case (2)

Item	Intensity (lux)	color	shape	Centroid (cx, cy)
1	87	Blue	Square	(47.84, 37.96)
2	87	Yellow	Cylinder	(57.46, 39.52)
3	87	Red	Cylinder	(54.08, 37.44)
4	87	Cyan	Triangle	(48.1, 36.92)

By relying on the developed algorithm, the system succeeded in discovering the shapes, colors, and centroid of the objects. After features detection, the objects are sorted into predefined positions with the help of a pick-and-place robot arm as shown in Figure 18(a) for the first case and Figure 18 (b) for the seconde case. If the system fails to detect the object it is rejected by the system and placed in the rejected item box. But in case 2 we defined the yellow and cyan colors as rejected objects thus the system put them in the rejected item box. Tables 4 and 5 show the result of the pick and place process for each case.

The results obtained show that the geometric technique in inverse kinematics analysis is effective in determining the joint angles of the robotic arm. After that, the robotic arm will pick up the object and place it in the desired location. As shown in Figure 8, the robotic arm could perfectly carry out the instruction to move within the necessary angular displacement degree. In addition, the arm can perform repeated commands for the same position. The coordinate system is used to measure the errors in the end effectors' positions in the x, y, and z axes. It can be noted that the largest error rate in the tool center point (TCP) position occurred when arm2 transferred the samples to the rotary carousel, as shown in Tables 4 for items 2 and 3, and also table 5 for items 3. This is because the distance from the rotary carousels and arm base center is more than containers and rejected boxes. However, all these errors were within the specified tolerances.

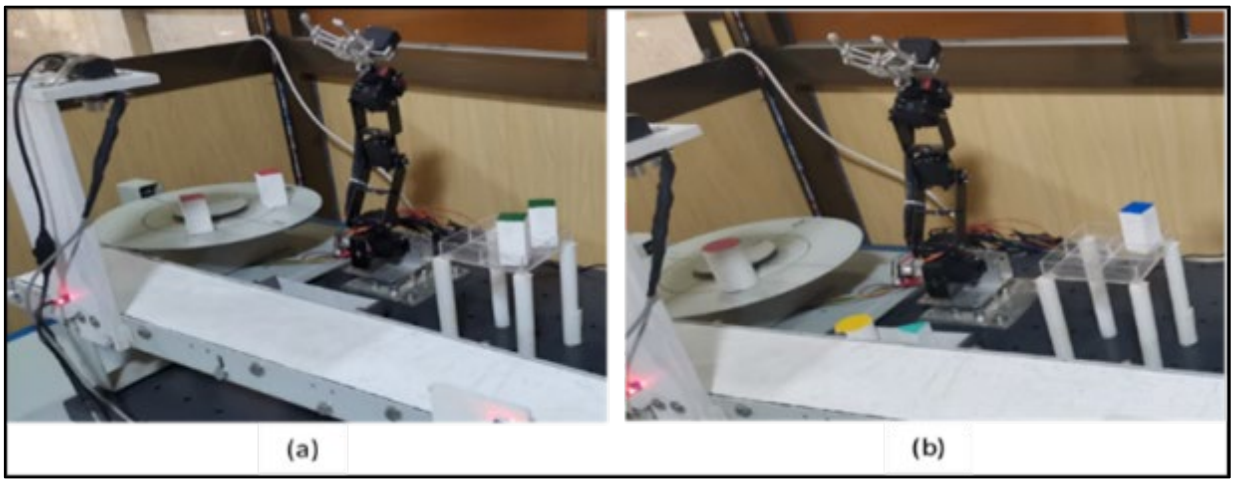

Figure 18: Implementation of sorting process: (a) for case 1, (b) for case 2

Table 4: The results of the position and error of case 1

Item No.	End Effector Position (True) (mm)	End Effector Position (Measured) (mm)	Joint Angles (degree)	Absolute $\text{Error} = \lvert\frac{\text{True}-\text{Measured}}{\text{True}}\rvert \times 100\%$
1	Px = -248 Py = 40 Pz = 245	Px =-249 Py =43 Pz =248	θ1 = 171 θ2 = 82 θ3 = -53 θ4 = 56 θ5 = 90	1.2 % 0 % 1.2 %
2	Px =179 Py = 103 Pz = 181	Px =178 Py =100 Pz =187	θ1 = 30 θ2 =105 θ3 = -87 θ4 = 50 θ5 =90	0.5 % 2.9 % 3.3 %
3	Px =179 Py = 103 Pz = 181	Px =174 Py =99 Pz =176	θ1 = 30 θ2 =105 θ3 = -87 θ4 = 50 θ5 =90	2.7 % 3.8 % 2.7 %
4	Px = -248 Py = 91 Pz = 245	Px =-246 Py =94 Pz =247	θ1 = 160 θ2 =74 θ3 = -42 θ4 = 53 θ5 =90	1.2 % 1 % 0.8 %

Table 5: The results of the position and error of case 1

Item No.	End Effector Position (True) (mm)	End Effector Position (Measured) (mm)	Joint Angles (degree)	Absolute $\text{Error} = \lvert\frac{\text{True}-\text{Measured}}{\text{True}}\rvert \times 100\%$
1	Px = -248 Py = 40 Pz = 245	Px =-250 Py =41 Pz =243	θ1 = 171 θ2 = 82 θ3 = -53 θ4 = 56 θ5 = 90	0.8 % 2.5 % 0.8 %
2	Px = 13 Py = 148 Pz = 160	Px =13 Py =148.5 Pz =159	θ1 = 85 θ2 =130 θ3 = -100 θ4 = 30 θ5 =90	0 % 0.3 % 0.6 %
3	Px =179 Py = 103 Pz = 181	Px =173.5 Py =100 Pz =178	θ1 = 30 θ2 =105 θ3 = -87 θ4 = 50 θ5 =90	3 % 2.9 % 1.6 %
4	Px = 13 Py = 148 Pz = 160	Px =13.5 Py =146 Pz =157	θ1 = 85 θ2 =130 θ3 = -100 θ4 = 30 θ5 =90	3.8 % 1.3 % 1.8 %

The results indicate that the kinematics analysis derived from the 5 DOF robotic arms was correct in performing the movement of the robotic arm with errors within acceptable limits. These results are based on the real robotic arm structure. Thus, these positional errors occurred depending on the robot's precision, which was affected by the type of drive, stiffness, thermal stability, object weights or effects dependent on time, such as gear wear and component damage, as mentioned previously. Thus, the endpoint pose is not correctly predicted by the robot's controller based on the joint angles.

To address this problem, continuous calibration of the robot is needed to figure out the static errors, such as those caused by changes in link dimensions, gear wear, elastic bending of links, etc., as well as dynamic errors, for instance, those caused by vibration.

5.3 Processing Time

Typically, image processing (IP) algorithms require extensive processing time for object recognition to be successfully implemented. Another important factor to consider when comparing the performance of the system is execution time during the testing phase. The following Equation is used to determine the execution time of the proposed image processing algorithm [30].

$$Avarage\ Processing\ Time\ (s)\ = \frac{Total\ Processing\ Time\ for\ each\ object}{Total\ Number\ of\ Images} \quad (13)$$

The computation time is based on 20 experiments conducted in RPi 3 model B, which has 1GB memory and 1.2GHz Quad-Core ARM Cortex-A53 for an image size of 240x240. The results show that the average processing time of the proposed system is approximately 2.21s. As the maximum extraction time was 2.24 s and the minimum time was 1.98 s. This value allows the detection of the produced pieces without slowing down the actual production rate.

5.4 Final Results

The total algorithm accuracy for both shape and color detection of training images is 93.95%. After determining the accuracy of the developed algorithm on the training images. It must be tested on a new set of images to ensure its accuracy. Thus, it was tested on 30 new images that were not present in the training images. The algorithm is shown good results in discovering the shape, as the accuracy of color and shape testing reached 93.71%. Thus, the overall system performance is 93.83%. It is worth noting that the algorithm failed to detect features in lighting conditions higher than the specified range.

All in all, the total time for performing the pick and place process takes 27 seconds for one object, starting from sending the instruction to the RPi until the object is placed in the desired position and the manipulator returns to the home position.

By comparing the results obtained from the proposed algorithm with that developed by [9] for shape detection which was based on the HSV color space and canny edge detection method, Gaussian Filter, and Otsu threshold technique to convert the grayscale image into a binary image. The efficiency rate achieved was 83.64 %. While the accuracy of the proposed algorithm for shape detecting is 88.70%. We conclude from this that the technique used in this research to discover the shape by relying on the S-channel as a segmented image, and use filters (median, bilateral, and gaussian) to filter the image, then applying the contour drawing technique to discover the shape was better by 5.06 % than [9].

As for color detection, the proposed design in [11] presents the discovery of only two colors, blue and green, and in constant lighting conditions. As for the proposed algorithm, it works on discovering five colors (blue, green, yellow, red, and orange) in different lighting conditions (5-7000 lux) in indoor and outdoor environments with smooth and rough surface objects. As well as it has reached an accuracy of 100% for various colors in various illumination as shown in Figure12. Thus, the proposed method is more effective and has more flexibility in use.

6. Conclusion

This research presents the development of an algorithm that works as a vision of a manipulator used for the pick-and-place process based on shape, color, and centroid detection in different light environments. The HSV color model is used to detect the color from the H channel. The draw-contour method is used to enclose the objects and discovers their shapes. the evaluation experiments using wood objects with different colors, shapes, and surface roughness shows that the developed image processing algorithm proved its efficiency in discovering the shapes and colors of the objects in different lighting conditions ranging from 5 to 7000 lux with an accuracy of 93.83%, and an average process time is 2.4 s, which enhanced the accuracy and efficiency of the system. The total time for performing the pick and place process takes 102 seconds, starting from sending the instruction to the RPi until the object is placed in the desired position and the manipulator returns to the home position. The future work is to develop the algorithm and make it workable in lighting conditions higher than 7000 lux.

Acknowledgment

The authors would like to thank the Department of Production Engineering and Metallurgy, University of Technology, Iraq, for providing facilities and support.

Author contribution

All authors contributed equally to this work.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability Statement

Not applicable.

Conflicts of Interest

The authors of the current work do not have a conflict of interest.

References

[1] V. Batra and V. Kumar, Real-Time Object Detection and Localization for Vision-Based Robot Manipulator, SN Comput. Sci., 2 (2021) 1-10. https://doi.org/10.1007/s42979-021-00561-4

[2] P. Tsarouchi, S. A. Matthaiakis, G. Michalos, S. Makris and G. Chryssolouris, A method for detection of randomly placed objects for robotic handling, CIRP J Manuf Sci Technol., 4 (2016) 20-27. https://doi.org/10.1016/j.cirpj.2016.04.005

[3] T. F. Abaas, A. A. Khleif and M. Q. Abbood, Computer Vision-Based System for Classification and Sorting Color Objects, IOP Conf. Series: Mater. Sci. Eng., 745, 2020, 012030. https://doi.10.1088/1757-899X/745/1/012030

[4] G. D. Leo, C. Liguori, A. Pietrosanto and P. Sommella, A vision system for the online quality monitoring of industrial manufacturing, Opt. Lasers Eng.,89 (2016) 162-168. https://doi.org/10.1016/j.optlaseng.2016.05.007

[5] R. V. Sharan and G. C. Onwubolu, Automating the Process of Work-Piece Recognition and Location for a Pick-and-Place Robot in a SFMS, in: Int. J. Image. Graphics. Signal Process., 6 (2014) 9-17. https://doi.org/10.5815/IJIGSP.2014.04.02

[6] A. A. Ata, S. F. Rezeka, A. El-Shenawy and M. Diab, Design and Development of 5-DOF Color Sorting Manipulator for Industrial Applications, in: World Academy Sci. Eng. Technol. Int. J. Mech. Aerosp. Ind. Mechatron. Manuf. Eng. 7 (2013) 2457-2464.

[7] Y. P. Loh, X. Liang and C. S. Chan, Low-light image enhancement using Gaussian Process for features retrieval, Signal Process Image Commun., 74 (2019) 175-190. https://doi.org/10.1016/j.image.2019.02.001

[8] C. Li, J. Guo, F. Porikli and Y. Pang, LightenNet: A Convolutional Neural Network for weakly illuminated image enhancement, Pattern Recognit. Lett., 104 (2018) 15-22. https://doi.org/10.1016/j.patrec.2018.01.010

[9] R. Kumar, S. Kumar, S. Lal and P. Chand, Object detection and recognition system for pick and place robot, in: Asia-Pacific World Congress on Computer Science and Engineering, IEEE. 2014,1-7. https://doi.org/10.1109/APWCCSE.2014.7053853

[10] H. M. Qul'am, T. Dewi, P. Risma, Y. Oktarina and D. Permatasari, Edge detection for online image processing of a vision guide pick and place robot, in: 2019 International Conference on Electrical Engineering and Computer Science (ICECOS) IEEE.2019, 102-106. https://doi.org/10.1109/ICECOS47637.2019.8984522

[11] S. Chakole and N. Ukani, Low-Cost Vision System for Pick and Place application using camera and ABB Industrial Robot, in: 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) IEEE. 2020, 1-6. https://doi.org/10.1109/ICCCNT49239.2020.9225522

[12] P. Shankar, V. Babureddy, S. C. S. Vigneshwaran and C. A. Prakash, Design and analysis of pick and place robot for colour based sorting applications, in: AIP Conference Proceedings.2247, 2020, 020025. https://doi.org/10.1063/5.0004429

[13] M. H. Ali, A. K., Y. K. and Z. T. a. A. O, Vision-based Robot Manipulator for Industrial Applications, Procedia Comput. Sci.133 (2018) 205–212. https://doi.org/10.1016/j.procs.2018.07.025

[14] F. S. Hameed, H. M. Alwan and Q. A. Ateia, Pose Estimation of Objects Using Digital Image Processing for Pick-and-Place Applications of Robotic Arms, ETJ. 38 (2020) 707-718.

[15] T. F. Abaas, A. A. Khleif and M. Q. Abbood, Inverse Kinematics Analysis and Simulation of a 5 DOF Robotic Arm using MATLAB, KECBUJ.16 (2020) 1- 10.

[16] P. V. Garad, Object Sorting Robot Based On the Shape, IJARIIT. 3 (2017) 129-134.

[17] A. Wang, W. Zhang and X. Wei, A review on weed detection using ground-based machine vision and image processing techniques, Comput. Electron. Agric.,158 (2019) 226-240. https://doi.org/10.1016/j.compag.2019.02.005

[18] H. O. Velesaca, P. L. Suarez, R. Mira and A. D. Sappa, Computer vision based food grain classification: A comprehensive survey, Comput Electron Agric. 187 (2021) 106287. https://doi.org/10.1016/j.compag.2021.106287

[19] K. Muhammad, M. Sajjad, S. Rho and S. W. Baik, Image steganography using uncorrelated color space and its application for security of visual contents in online social networks, Future Gener. Comput. Syst., 86 (2018) 951-960. https://doi.org/10.1016/j.future.2016.11.029

[20] G. Bargshady, X. Zhou, R. C. D. J. Soar, F. Whittaker and H. Wang, The modeling of human facial pain intensity based on Temporal Convolutional Networks trained with video frames in HSV color space, Appl. Soft Comput., 97 (2020) 106805. https://doi.org/10.1016/j.asoc.2020.106805

[21] K. B. Shaik, G. P, V.Kalist, B.S.Sathish and J. M. Jenitha, Comparative Study of Skin Color Detection and Segmentation in HSV and YCbCr Color Space, Procedia Comput. Sci., 57 (2015) 41-48. https://doi.org/10.1016/j.procs.2015.07.362

[22] F. G. Lamont, J. Cervantes, A. López and L. Rodriguez, Segmentation of images by color features: A survey, Neurocomputing.292 (2018) 1-27. https://doi.org/10.1016/j.neucom.2018.01.091

[23] D. Wu and D.-W. Sun, Colour measurements by computer vision for food quality control - A review, Trends Food Sci Technol., 29 (2013) 5-20. https://doi.org/10.1016/j.tifs.2012.08.004

[24] S. Kotte, P. R. Kumar and S. K. Injeti, An efficient approach for optimal multilevel thresholding selection for gray scale images based on improved differential search algorithm, Ain Shams Eng. J., 9 (2018) 1043-1067. https://doi.org/10.1016/j.asej.2016.06.007

[25] D. Bulanon, T. Kataoka, Y. Ota and T. Hiroma, A segmentation algorithm for the automatic recognition of Fuji apples at harvest, Biosyst Eng., 83 (2002) 405-412. https://doi.org/10.1006/bioe.2002.0132

[26] T. Y. Goh, S. N. Basah, H. Yazid, M. J. A. Safar and F. S. A. Saad, Performance analysis of image thresholding: Otsu technique, Mea., 114 (2018) 298-307. https://doi.org/10.1016/j.measurement.2017.09.052

[27] U. Erkan, L. Gökrem and S. Enginoglu, Different applied median filter in salt and pepper noise, Comput. Electr. Eng., 70 (2018) 789-798. https://doi.org/10.1016/j.compeleceng.2018.01.019

[28] B. Goyal, A. Dogra, S. Agrawal, B. Sohi and A. Sharma, Image denoising review: From classical to state-of-the-art approaches, Inf. Fusion., 55 (2020) 220-244. https://doi.org/10.1016/j.inffus.2019.09.003

[29] J. Joseph and R. Periyasamy, An image driven bilateral filter with adaptive range and spatial parameters for denoising Magnetic Resonance Images, Comput. Electr. Eng., 69 (2018) 782-795. https://doi.org/10.1016/j.compeleceng.2018.02.033

[30] A. K. Sat and T. Tint, Object detection and recognition system for pick and place robot, in: International Conference on Big Data Analysis and Deep Learning Applications, Springer.2018, 315-323. https://doi.org/10.1007/978-981-13-0869-7_35