International Journal of Intelligent Robotics and Applications (2024) 8:179–192
https://doi.org/10.1007/s41315-023-00305-y

REGULAR PAPER

Experiments with cooperative robots that can detect object's shape, color and size to perform tasks in industrial workplaces

Md Fahim Shahoriar Titu[1] · S. M. Rezwanul Haque[1] · Rifad Islam[1] · Akram Hossain[1] · Mohammad Abdul Qayum[1] · Riasat Khan[1]

Received: 9 February 2023 / Accepted: 29 October 2023 / Published online: 25 November 2023

Abstract
Automation and human-robot collaboration are increasing in modern workplaces such as industrial manufacturing. Nowadays, humans rely heavily on advanced robotic devices to perform tasks quickly and accurately. Modern robots with computer vision and artificial intelligence are gaining attention and popularity rapidly. This paper demonstrates how a robot can automatically detect an object's shape, color, and size using computer vision techniques and act based on information feedback. In this work, a powerful computational model for a robot has been developed that distinguishes an object's shape, size, and color in real time with high accuracy. Then it can integrate a robotic arm to pick a specific object. A dataset of 6558 images of various monochromatic objects has been developed, containing three colors against a white background and five shapes for the research. The designed system for detection has achieved 99.8% success in an object's shape detection. Also, the system demonstrated 100% success in the object's color and size detection with the OpenCV image processing framework. On the other hand, the prototype robotic system based on Raspberry Pi-4B has achieved 80.7% accuracy for geometrical shape detection and 81.07%, and 59.77% accuracy for color recognition and distance measurement, respectively. Moreover, the system guided a robotic arm to pick up the object based on its color and shape with a mean response time of 19 seconds. The idea is to simulate a workplace environment where a worker will ask the robotic systems to perform a task on a specific object. Our robotic system can accurately identify the object's attributes (e.g., 100%) and is able to perform the task reliably (81%). However, reliability can be improved by using a more powerful computing system, such as the robotic prototype. The article's contribution is to use a cutting-edge computer vision technique to detect and categorize objects with the help of a small private dataset to shorten the training duration and enable the suggested system to adapt to components that may be needed for creating a new industrial product in a shorter period. The source code and images of the collected dataset can be found at: https://github.com/TituShahoriar/cse499B_Hardware_Proposed_System.

Keywords Automation · Bounding box regression · Computer vision · Detection · Image processing · Library · OpenCV · Robot

1 Introduction

Automation in industrial workplaces is increasing at a breath-taking speed and also, human-robot collaborations are becoming an important issue as more autonomous robots are roaming around human workplaces. Automation implies the machine performs its function automatically with minimum human communication (Grudin and Carroll 2017). The device can correctly perform its function without human interaction by executing command instructions from an application program (Maasoumy and Sangiovanni-Vincentelli 2016). For instance, a computer with such an application program will be able to determine whether or not people are wearing masks and use automation to decide whether to run medical testing or not. Programming for the automation process is executed on the machine, and the machine actively offers the result by performing tasks without human interaction (Rafique and Velasco 2018). Though automation began in the 1790s and 1840s, it is plausible to say that nowadays, automation has become a vital factor of the industrial revolution and industrial machinery (Imran et al. 2020).

✉ Riasat Khan
riasat.khan@northsouth.edu

[1] Electrical and Computer Engineering, North South University, Dhaka, Bangladesh

Researchers use automation to save time and resources, reduce human errors, increase accuracy, and achieve higher productivity and consistency. Again, many tasks that people find difficult and dangerous can be completed using industrial automation control systems like powerful computers or robots (Wettinger et al. 2018). Automation technology is predicted to cause the global GDP to rise to $1 trillion USD by 2030. Human-robot collaboration is a field of technology where users create application programs to make the machine more human-friendly, and the device performs its functions through human-guided commands (Kondratenko et al. 2021). Even robots in workplaces can not only perform tasks automatically but also can cooperate in work due to their powerful computational power, which also can act like a creative mind (Qayum et al. 2017).

This paper demonstrates how an intelligent system can automatically detect an object's shape, color, and size using image processing and computer vision-based approaches. And then, a robotic system has been added to the proposed system, which can independently perform these similar operations using the Raspberry Pi 4B microcontroller and arm that can handle objects. The followings are the main contributions to this work:

- A custom dataset of 6558 images of various monochromatic objects with three colors (red, blue and green) against a white background and five shapes objects: square, hexagon, cylinder, triangle, and circle, has been created. LabelImg, an open-source framework, has been used to annotate the dataset.
- YOLOv4 tiny deep learning model and PyTorch library are used for object detection. The proposed robotic system detects object shapes using the bound box regression approach.
- The proposed automatic system can detect the color and size of the object through the execution of the OpenCV function.
- The proposed robotic system employs Raspberry Pi 4B, Arduino Nano, Arduino Uno microcontrollers and a robotic arm with MG996R servo motors to pick up and place by sorting the color and shape of the items precisely.
- The performance has been defined by comparing the detection accuracy of the designed system with a real-time detection approach. Finally, the proposed automatic robotic device has been thoroughly investigated. Its performance in terms of object color, shape and size detection accuracy, response time, distance measuring accuracy, design expenditure, etc., has been shown.

The novelty of this work is to accelerate the development of an automatic object color, shape and size detection along with a pick and place device. The system employs a small private dataset of 6558 images instead of using existing large libraries. The system is also a low-cost implementation of an advanced embedded system built with a Raspberry Pi 4B, demonstrating its utility against existing literature with a comprehensive analysis.

In the next section, related studies in autonomous robots have been discussed, especially articles that can detect and handle objects' attributes. Section 3 discusses the proposed system in detail: its individual software and hardware components and the created dataset while outlining the processes for determining an object's color, shape, and size. Section 4 reviews the simulation and hardware results. Next, the limitations of the proposed system have been presented in Sect. 5. Finally, the contributions and potential features to upgrade this system for future work have been discussed in Sect. 6.

2 Related work

Robotics is currently being used extensively for industrial purposes. That is why the number of robotic vision-based projects has increased. With the advancement of artificial intelligence techniques, computer vision and deep learning techniques have been applied in many works for object detection and image segmentation (Ranjbarzadeh et al. 2022; Ranjbarzadeh et al 2023). In the following paragraphs, some recent articles on robotic systems have been discussed, which are operated by computer vision and artificial intelligence. Most of these works have used a wide range of hardware robotics devices, such as advanced microcontrollers and sophisticated sensing systems.

2.1 Object detection

Earlier research endeavors in this domain primarily focused on exclusively implementing object tracking and detection. Banerjee and his group (Banerjee et al. 2018), for instance, initiated an automated device that can accurately test object-tracking algorithms. The authors used an automation system to develop their object tracking feature and track objects accurately. They used a robotic arm to execute a test case of one or more objects being tracked by motion. This proposed device comprised a track so that the user would select the area of the object through the camera preview via bound-box regression. The area inside the bound box was considered a template. This object tracking feature calculated the accuracy by contrasting its effectiveness with the actual data, ground truth. Finally, the authors applied this tracking feature to three distinct objects of various sizes. The obtained tracking accuracy was 78.7% without using any colored backgrounds. Ahmed et al. (2021) proposed detecting and tracking objects using computer vision systems

and intelligent cameras or optical sensors. The authors used SSD and YOLO-based deep learning models for object detection and localization. They also used intelligent robotic cameras with visual processing. Their system achieved a maximum 93–90% detection rate from the applied models, and the success rate in object tracking accuracy was 90–94%. Othman et al. (2018) used an advanced technique to identify objects in real-time video streams employing webcams. The authors used the Canny edge detection algorithm to identify objects instantaneously. They used dilation and irrational algorithms to fill in the blanks on the edge. Then the dimensions of the object are measured by selecting the contours. They used the OpenCV software library, the Raspberry Pi 3 as a microcontroller, and the Raspberry Pi camera as a webcam to implement their proposed technique. The authors accomplished 96% success in determining the size of any object. Roy et al. (2019) designed an automatic object-sensing system combining ultrasonic sensor-based hardware components and computer vision techniques. The developed device used Arduino Uno and an IP camera to detect surrounding objects. Some of these articles did not incorporate these automatic devices into hardware designs. Gupta et al. (2016) applied a partitioning technique to identify objects from input images based on their size and geometric foundation. This system can detect the object's shape, which will help determine the entity. This group used linear SVM and KNN machine learning techniques as object recognition classifiers. The authors collected 1020 images and created 20 photos of 51 objects for the test. Their success rates were 81% and 74% for the linear SVM and KNN models, respectively.

2.2 Object's color, size and shape classification

In recent studies, the emphasis has shifted toward classifying the detected objects' color, size, and shape. Magadum and Bombale (2019) developed a robotic arm for various color recognition of buckets, which was implemented with an image processing approach and a color identification sensor. They used an Arduino microcontroller and a robotic arm with multiple servo motors comprising four degrees of freedom. Divya and Kumar (2020) described color identification based on automatic eyesight. The authors employed a robotic arm and an Arduino Uno microcontroller in their research. They used LabVIEW, which uses image processing and a camera installed on the robotic arm to identify a given color. Robotic arms recognize a specific color that has been selected in LabVIEW's front panel using color recognition technology. The advantages of this method over manual labor include reduced errors, enhanced precision and savings of time. Sawant et al. (2022) constructed a color-sorting robotic hand employing the Faster RCNN deep learning approach. The Raspberry Pi integrated device accomplished approximately 80% accuracy in detecting and sorting red and blue objects. Cong et al. (2023) developed an intelligent module to classify different objects automatically. The authors employed a Raspberry Pi 4 embedded device and image processing techniques. The system correctly classified 238 objects from a total of 240 according to their distinct colors. Fernandes and Shivakumar (2020) applied image processing and OpenCV libraries to identify the color and shape of different solid objects. ATmega328 microprocessor and a webcam have been used to detect three categories of objects instantaneously. Zhang et al. (2020) introduced an automatic color recognition and sorting robotic hand employing an STC15F2K60S2 microcontroller and CCD camera. The authors used various image processing approaches and morphological operations to detect the objects. Shaikat et al. (2020) designed an object color and height recognition automated robotic device using Haar cascade-based image processing technique and OpenCV framework. The authors deployed the system into Arduino Nano and a camera to sort the detected products. Using inverse kinematics and artificial neural networks, Sanjaya (2018) developed a robotic arm with automatic 5-DOF color identification and sorting capabilities. A digital webcam and the inbuilt Arduino framework were used to locate three different types of colored items in predetermined locations. Noman et al. (2022) computer vision system-based robotic arm can recognize an object's color, shape, and size. Pixy-CMU uses a camera sensor in this instance to detect colors with an accuracy of 80%. The OpenCV image processing library is used for shape and size identification. As it is currently configured, the system is almost 100% capable of identifying an object's size and shape. Intisar et al. 2021 used a robotic arm controlled by an interactive graphical user interface. They used the OpenCV library to detect the object's color, size, and shape with 93.33%, 98.60% and 94.60% accuracies, respectively. They used the inverse kinetics algorithm to capture the object and outline the robotic arm's positioning. In Gode and Khobragade (2016), the authors developed an algorithm established on image processing techniques to detect and identify objects using their size and color. The authors designed this work by separating the objects using color information and comparing them with their corresponding contours. They used the Fourier descriptor method for the detection process. Their proposed technique can accurately identify objects during experiments with approximately a 9.10% error rate.

2.3 Automatic object sorting and grasping

Lately, the trend in this topic has moved to sort and grasp the detected objects automatically based on their predefined characteristics. Sekkat et al. (2021) developed a robot that can capture an object in multiple dimensions

using a complex determining principle. The robot gets a multi-degree rotation to grasp an object. The device identified the angle of the robotic joints by using inverse kinetics so that the grip joint of the robot could reach the target. Liu and Jin (2020) recommended multi-objective visual placement and identification methods for object sorting employing cost-effective robotic devices. They developed a sorting robot system with visual perception for their suggested method. According to the study's findings, these autonomous devices might be used to pick up objects and position them in predetermined locations. Najmurrokhman et al. (2019) discussed a manipulator arm system for an industrial robotic system that can pick and sort lightweight objects. The ATmega2560 microcontroller's output sent a signal to four servomotor drivers, moving the robot arm in the desired direction. The robot selected the objects and consequently placed them in their proper places. The trial's results showed that the manipulator's arm functions effectively in line with design objectives, including color accuracy and quick robot movement responsiveness from beginning to end. Chen and Hu (2023) developed a deep learning-based robotic arm for gripping objects. An AI depth camera detects various objects employing the YOLOv3 deep learning technique. The designed system achieved an accuracy of approximately 90% on grasping different objects. A robotic hand for object location control was developed by Hamzah et al. (2022) using the Mamdani fuzzy inference technique. Fuzzy sets, defined by membership functions that assign degrees of membership to components in a universe of discourse, are used in this device to represent inputs and outputs. The Arduino Uno-based device accomplished 99.33% accuracy for object location placement. Sivapriyan et al. (2020) implemented an efficient strategy to manipulate the existing oscillation sensor control system. The researchers employed various advanced image processing techniques, kinetic models, and general filtering algorithms for speed manipulation that were used alongside the AI-present oscillation sensor to manipulate the control system. This research paper achieved 88% success using a manipulator control system. The PyTorch library is used in this work for shape detection in the software's proposed system. Consequently, the color and size of the object have been detected using Bound Box regression using the object's shape and OpenCV framework. Abdi et al. (2022) devised object detection and direction planning for a robotic arm using deep learning and reinforcement learning approaches. YOLO-based deep learning model and a Q-learning technique were used for object detection and route planning, respectively.

2.4 Similarities and contrasts between the presented work and existing literature

The literature reviews illustrate that extensive works have been done on object size, shape, and color detection employing image processing and artificial intelligence techniques. However, the integration of object size, shape, distance, color detection and pick and place the detected objects in a specific location using advanced library functions of image processing approaches are yet to be done. An automated detection process using the Raspberry Pi 4B or similar contemporary cutting-edge microcontrollers has not been initiated in most of these articles. The majority of these works did not perform in-depth analyses of their developed devices. This work contrasts the outcomes of automatic detection using the Raspberry Pi 4B microcontroller with those obtained using advanced computer vision-based library functions. The Raspberry Pi 4B microcontroller employs the YOLOv4 tiny deep learning approach for object identification, achieving efficient detection results and faster response time than other works. Extensive investigation of the proposed automatic robotic device has been performed in this study, demonstrating its object color, shape and size detection accuracy, response time, distance measuring accuracy, implementation cost, etc.

3 Proposed system

3.1 Dataset creating and preprocessing

A significant contribution of this research is to create a unique dataset of various solid objects comprising five classes of shapes, square, hexagon, cylinder, cone, and circle, and three monochromatic colors, green, red, and blue. 6558 images for the custom dataset have been collected, of which 2186 photos are of each color and 1500 are of each shape except the circles.

The number of various colors and shaped objects of the created dataset has been illustrated in Table 1. The

Table 1 Number of various colors and shapes objects of the created dataset

Circle			Cylinder			Hexagon			Square			Cone		
Red	Blue	Green	Red	Blue	Green	Red	Blue	Green	Red	Blue	Green	Red	Blue	Green
186	186	186	500	500	500	500	500	500	500	500	500	500	500	500

obtained dataset has been annotated in YOLO.jpg format and PASCAL VOC.jpg specification for the software (laptop PC) and hardware (Raspberry Pi) systems, respectively, using the LabelImg framework. The proposed hardware device used a dataset of approximately 925 images because of the RAM limitations of the Raspberry Pi 4B microcontroller. First, images of similar colors have been selected in the data cleaning and preprocessing step. White backgrounds to all these images are applied. After that, the video of the object has been captured at 306° rotation and various angles with a high-resolution camera on an Apple iPhone 12 Pro Max smartphone. We used an open-source visual effect tool to edit every video's shape color in red, green, and blue. The videos into corresponding images are converted using this tool and some images with noises are removed. Various objects with different colors and shapes of the created dataset have been depicted in Fig. 1.

3.2 Software tools

Visual Studio: Visual studio is a source code editor compatible with every operating system (Bai 2020). It is fast in its operation, and it has the best readability. It has a coloring statement that helps to understand the code's work sequence.

Anaconda: All the necessary packages can be obtained in one place for this research in Anaconda software (Phelps et al. 2000). It is faster and requires less space in storage.

Python: Python is a popular, simple, and high-level programming language (Parveen and Shah 2021). Python programs are relatively easy to write and understand. Python has a large number of standard libraries. Therefore, writing programs in Python provides access to many of the required free libraries of the proposed system.

PyTorch: This work utilizes the PyTorch framework and NumPy for calling the dataset for object detection. In the PyTorch library, a lightweight package can be used, and a few lines of code are needed to initialize the mode (Arunachalam et al. 2022). So, in terms of detecting objects in a computer vision operating robotic system model (Krishnadas and Sampathila 2021), this library will work faster than any other library.

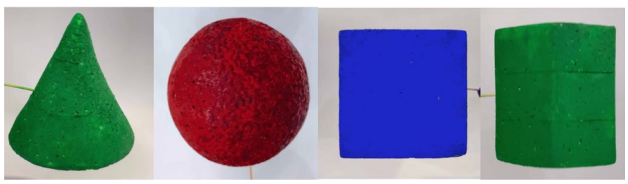


Fig. 1 Sample images of various objects with different colors and shapes of the created dataset

3.3 Hardware components

Raspberry Pi 4B 2GB: Raspberry Pi 4B microcontroller has been used in the proposed system. It is a 1.5 GHz 64-bit quad-core ARM Cortex-A72 processor. Raspberry Pi is a single-board computer (Wang and Long 2022). It has a processor, RAM, and I/O port like a computer. Raspberry Pi works as the central computational unit of the robotic arm, facilitating the identification of objects for subsequent picking and placing operations.

Arduino Uno: Similar to Arduino Nano, Arduino Uno is an 8-bit, microcontroller-based development board (Intisar et al. 2021). The Arduino Uno undertakes computational tasks related to the robot's stepper motors, controlling its degrees of freedom. The Uno receives command signals from both a computer (e.g., a worker displaying an object to the computer's camera) via serial communication and from the Raspberry Pi (RPi).

Arduino Nano: Arduino Nano is an 8-bit, microcontroller-based development board (Najmurrokhman et al. 2019). The connection between the Uno and RPi is facilitated through an Arduino Nano, which operates as an intermediary, regulating the flow of information between the RPi and Uno. This arrangement involves the Nano buffering information while the RPi executes computations to determine object detection related actions.

Servo Motor MG996R: Servo motor MG996 is a metal gear servo motor (Chen et al. 2021). It can take a heavy load, a maximum of 10 kg. This metal servo motor is used to move the robotic arm.

LM2596 Buck Converter: LM2596 is a variable linear buck converter (Sudhoff 2014). It can convert any voltage from 1.50 Volts to 38 Volts. In this work, this converter has been used to power the servo motors employing a 12V power supply.

Web Camera: A full HD 1080P USB web camera, Xiaomi-CMSXJ22A, has been used in this research (Azad and Misbahuddin 2018). It has been used for capturing video to detect various objects in real-time.

3.5-inch LCD Display: Realtime Raspberry Pi monitoring and control has been executed using a 3.5-inch LCD (Kafadar 2021). It has a built-in touch system. Raspberry Pi can be easily controlled using this touchscreen display.

Figure 2 demonstrates the complete setup of the proposed hardware system. All of the items to design the proposed hardware device have been purchased from the local electronics and robotics shop in Dhaka, Bangladesh. The system's total cost was approximately $298 USD.

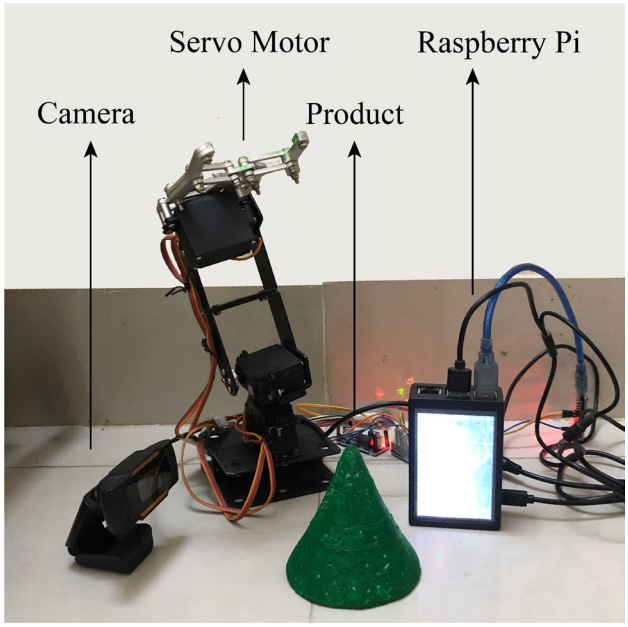


Fig. 2 Design of the proposed hardware system

3.4 Software and hardware system design

3.4.1 Color detection workflow

In this work, the OpenCV framework has been applied for the color sorting of various objects. At first, the video is loaded from the webcam in an image frame. The range for the RGB color model is set, and the corresponding mask is defined. Finally, a program terminator is initiated. Figure 3 demonstrates the working sequences for color detection of the proposed system.

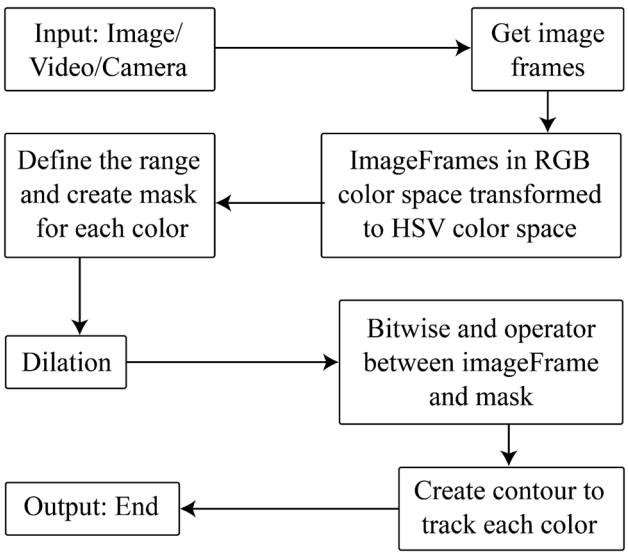


Fig. 3 Color detection of the proposed system

3.4.2 Shape detection

The bounding box regression technique has been implemented for shape detection in this work. The following attributes are present in every bounding box in the image, i.e., width, height, class, and bounding box center (W, H). This model can predict objects' height, width, center, and class. An open-source software, LabelImg, has been utilized to label the images. Those images represent the probability of an object appearing in the bounding box. Various predefined classes have been developed, e.g., square, hexagon, ball, cylinder, and cone. The classified and annotated images are finally saved in the YOLO format. YOLO framework regulates the image's size during the training phase.

3.4.3 Size measurement

Figure 4 depicts the object's size detection working sequences of the proposed system. *contrib* and *imutils* (Koodtalang and Sangsuwan 2019) modules of OpenCV are used for size measurement. In this research, size in three categories has been defined, i.e., small, medium, and large. In this process, an image is taken as input. The program will detect the edge of the image and set the points accordingly (Wang et al. 2019). This step helps verify the object's height and weight, considering the edge point's distance. It compares these attributes with the area of the image. Finally,

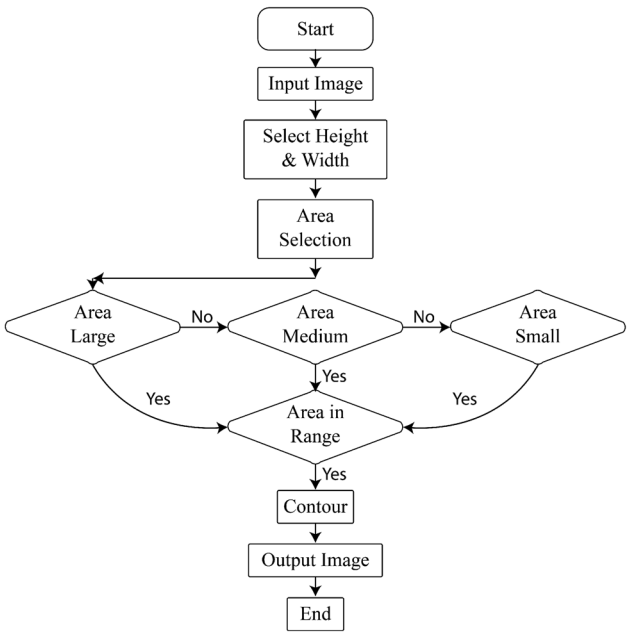


Fig. 4 Object size detection of the implemented device

this process will give an output image with a new size label and the object shape detection process ends.

$$S_{girth} = (2 \times width) + (2 \times Height) \tag{1}$$

$$Length\ of\ Cone = \sqrt{(H^2 + (\frac{w}{2})^2)} \tag{2}$$

In (1), *girth* indicates measuring around the center of any object. According to this equation, to measure the object's perimeter, the height and width are doubled and added together (Shahid et al. 2018). For measuring the length of the cone, as illustrated in (2), the width has to be taken as the diameter, i.e., the radius has to be found by dividing the width by two.

3.4.4 Software system workflow

The overall workflow of the proposed software system in a laptop PC environment has been depicted in Fig. 5. Three different aspects of an object have been considered in this work. Firstly, a webcam or preprocessed videos or images have been used. The program will read the frame from the input data. After that, it will define the object in terms of shape and color. There can be five different shapes, e.g., square, hexagon, cone, cylinder, and ball. The color of the object will be red, blue, or green. After completing the detection and recognition part, the program will search for the item and be able to define the object's size into three categories: medium, small and large. Finally, the program will merge the three features into one output.

3.4.5 Hardware system workflow

Figure 6 shows the overall workflow of the proposed hardware system. This system will detect the shape and recognize the color using the program's appropriate commands. The camera has to be activated to start the program and send data to the Raspberry Pi. This video analyzes the data using the TensorFlow Lite framework. The display will show the shape's name, color, and distance when the shape is detected successfully from the video input. If it finds no objects, it will continue to check the shape. The detected shape specification will then be sent to Arduino Nano, and this data will be sent to Arduino Uno, which will work as a communicator. Arduino Uno will wait for the user's command to determine which product needs to be grasped by the robotic arm. The user will give power and compare the result with shape detection. Then, the shape will be carried using the robotic arm and placed in the correct position, as described in the Arduino Uno program.

Figure 7 shows the detailed working process of the hardware system. The first part of the image shows the robotic arm's ability to grip a blue ball. The second step of the image shows that the arm is carrying the ball into the air. In the final stage of the picture, the arm keeps the object in different locations.

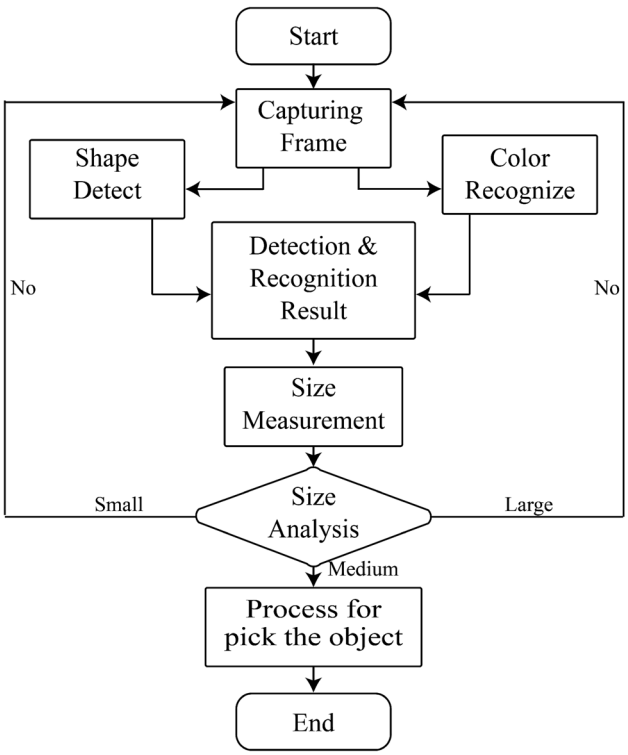


Fig. 5 Overall workflow of the proposed software system

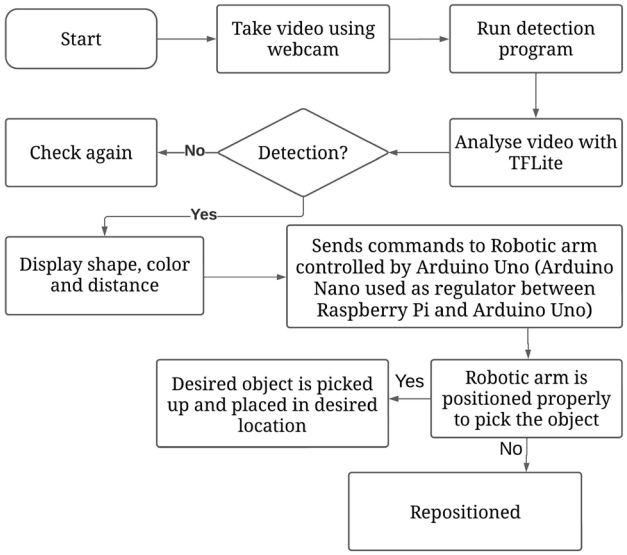


Fig. 6 Overall workflow of the proposed hardware system

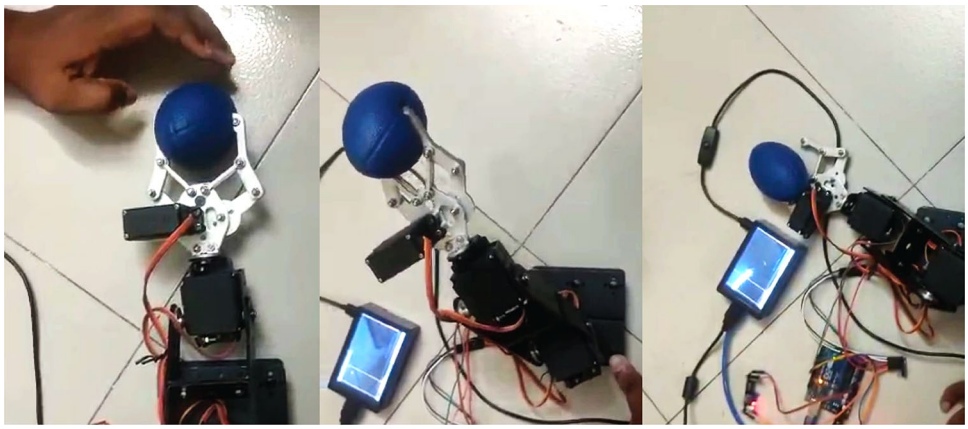


Fig. 7 The working process of the proposed system

Start
Step 1: Training of the proposed system (only needed once).
Step 1.1: Trained YOLOv4 tiny model is loaded in Raspberry Pi 4B.
Step 2: Initialize the following system components:
Step 2.1: Raspberry Pi 4B as the main controller.
Step 2.2: Webcam and YOLOv4 tiny model for object detection.
Step 2.3: Arduino Nano and Arduino Uno for controlling the 6-DOF robotic arm.
Step 3: Receive user input for the desired object's shape, size, and color.
Step 4: Set up the object detection system.
Step 4.1: Capture of live video feed by webcam.
Step 4.2: Transmitting video frames through the YOLOv4 framework for object detection.
Step 4.3: Accurate detection of shape, color, size, and distance using PyTorch bounding box regression and OpenCV.
Step 5: System sends the calculated commands to the robot.
Step 6: Calibrate the robot for object picking and placement.
Step 6.1: Place the object according to the user's preference using the arm's motion control.
End

Algorithm 1 Algorithm of the proposed hardware robotic system

Table 2 Color sorting accuracy

Color	Attempts	Failure	Success rate (%)
Red	30	0	100
Blue	30	0	100
Green	30	0	100

Table 3 Shape detection rate of the proposed system

Shape	Attempts	Success	Failure	Success rate (%)
Square	145	145	0	100
Hexagon	149	147	2	98.68
Cylinder	154	154	0	100
Cone	151	151	0	100
Ball	56	56	0	100
Overall	657	655	02	99.7

4 Results and discussion

4.1 Performance of the proposed software system

This section discusses the software results of the proposed robotic system's object color, shape, and size detection. The software results refer to the performance of the proposed system in a laptop PC with 16GB RAM and 256GB SSD.

Color sorting performance of the proposed robotic device has been illustrated in Table 2. In this work, the OpenCV library function is used to detect the object's color. The program is designed to recognize three colors—red, blue, and green. We set the color range for RGB and define the mask. By attempting 90 times, i.e., 30 times for each color, the accuracy was overall 100% in color sorting when the camera and the light work ideally, and the background is white or black. However, the result can fluctuate due to light sensitivity and the camera's position.

Table 3 displays the performance of shape detection of the designed system. In 657 attempts, the proposed device

Table 4 Object size detection accuracy

Size	Attempts	Success	Failure	Success rate (%)
Large	7	7	0	100
Medium	35	35	0	100
Small	5	5	0	100
Overall	40	40	0	100

Fig. 8 The output of the proposed software system

successfully identified the shape of numerous items in 655 out of 657 tests, failing only twice. The proposed system recognized most of the shapes except the hexagons. According to this table, the designed system achieves 99.7% accuracy in shape detection rate using bounding box regression.

Table 4 shows the accuracy of the size detection, where the size was defined into three categories, e.g., small, large, and medium. The average size detection accuracy rate is 100% using the OpenCV library function. The library perfectly measures the shape's boundary area by the OpenCV image processing function.

$$Dimension = \begin{Bmatrix} 5\ cm: & Small \\ (5-8)\ cm & Medium \\ 8\ cm\ and\ above & Large \end{Bmatrix}$$

Figure 8 shows the output of the entire software system for a red-colored square medium object. The output shows that the system detected the object as a red square and detects the object's size as medium with the dimension of the object defined in the system, as expressed in (IV-A).

The comparison of the software tool of the proposed system with some of the related works is demonstrated in Table 5. Many of these studies used OpenCV and image processing techniques to recognize object color, shape, and size. In contrast, this work uses YOLOv4 deep learning technique for object detection. The proposed software system demonstrates superior performance to most other studies regarding its color, shape and size detection accuracies.

4.2 Performance of proposed hardware system

This section presents the real-time experimental results of the object's color, shape, and size detection of the proposed robotic system with a Raspberry Pi 4B microcontroller.

Table 6 shows the accuracy of the proposed hardware system for shape detection. The success rate from 502 attempts for detecting the object's shape in the hardware system is 81.07%. The accuracy of the cone-shaped object is the highest, with 91.81% success. Conversely, the proposed hardware

Table 5 Comparison table for software tools

References	Color detection	Shape detection/object detection	Size detect	Results
Sekkat et al. (2021)	N/A	YOLOv5	N/A	Detection accuracy: 98%
Intisar et al. (2021)	HSV	OpenCV contours	OpenCV contours	Color detect acc.: 93.33% Shape detect acc.: 94.6% Shape detect acc.: 98.6%
Noman et al. (2022)	N/A	OpenCV contours	N/A	Color detect acc.: 80% Shape detect acc.: 100% Shape detect acc.: 100%
Abdi et al. (2022)	N/A	YOLO	N/A	N/A
Proposed	HSV and gray mode	PyTorch bounding box regression (YOLOv4)	OpenCV contours	Color detect acc.: 100%, Shape detect acc.: 99.7% Shape detect acc.: 100%

Table 6 Shape detection rate of the proposed hardware system

Shape	Attempts	Success	Failure	Success rate (%)
Square	111	92	19	82.9
Hexagon	110	82	28	74.54
Cylinder	111	84	27	75.7
Cone	110	101	09	91.81
Ball	60	48	12	80
Overall	502	407	95	81.07

Table 7 Color sorting accuracy of the proposed hardware device

Color	Attempts	Failure	Success rate (%)
Red	20	01	95
Blue	20	06	70
Green	17	04	76.5
Overall	57	11	80.7

Table 8 Size detection performance of the proposed robotic device

Size	Attempts	Success	Failure	Success rate (%)
Large	6	5	1	83.33
Medium	18	15	3	83.33
Small	5	4	1	80
Overall	29	24	5	82.75

Table 9 Object distance assessment accuracy of the proposed hardware device

Distance	Attempts	Success	Failure	Success rate (%)
(0.0–1.0) cm	32	19	13	59.37

device accomplished the lowest detection accuracy for hexagon-shaped objects.

Tables 7 and 8 show the accuracy of the object color and size detection of the hardware proposed system, respectively, in a real-time video camera with a Raspberry Pi microcontroller. The proposed hardware system's accuracy is 80.7% after attempting it 57 times in color recognition for various instances. The constructed hardware robotic device accomplished the highest accuracy in recognizing red-colored products with a 95% success rate. The proposed robotic device attained 82.75% in the detection of object sizes.

The performance of the measured distance of the desired object from the real-time video camera is depicted in Table 9. Various objects have been placed from the webcam at different distances of 0.0–1.0 cm. The proposed system correctly determined the distance for 19 out of 32 instances with approximately 60% success rate. The capabilities of the camera and the surrounding lighting are the key determinants of accuracy.

Table 10 Response time of pick and place of the detected objects of the proposed robotic arm

Object	Response time (s)
Green cone	19
Green ball	18
Red square	20
Red ball	19
Blue ball	19
Blue cylinder	19
Average	19

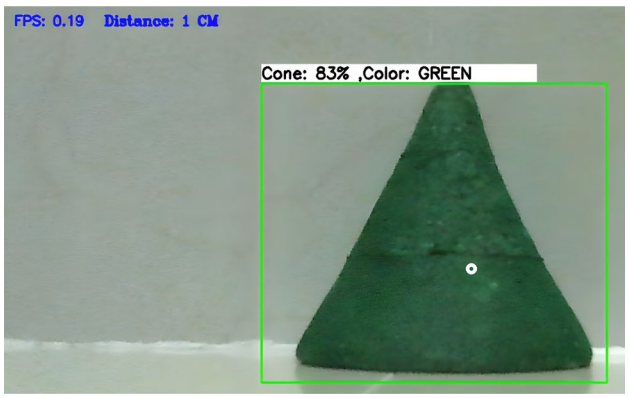


Fig. 9 Output of the proposed hardware system

The response time of pick and place of various objects of varying colors, shapes and sizes of the proposed robotic device has been illustrated in Table 10. According to this table, the proposed robotic arm can pick and locate different objects with a mean response time of 19 s. In this work, the YOLOv4 tiny deep learning model has been used for the object detection of the Raspberry Pi hardware system. This model comprises reduced complexities and parameters and is considered one of the fastest real-time object detection models.

Figure 9 represents a sample output of the proposed hardware system identifying a green-colored cone with a distance of 1 cm.

An illustration of multiple objects' simultaneous detection, i.e., a green-shaped circle and a red-shaped square, of the proposed robotic device has been illustrated in Fig. 10.

Table 11 illustrates a comparison of the proposed hardware device with other existing works. According to this table, most of the works did not develop an integrated system that can detect color, size, and shape and pick and place the detected object simultaneously. The majority of these works did not perform in-depth analyses of their developed devices. The main hardware processing unit for this research

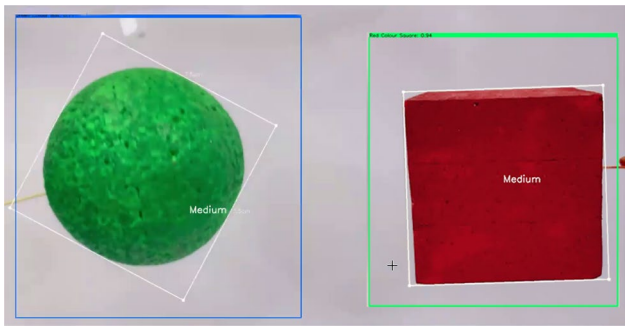


Fig. 10 Output of the proposed hardware system for multiple objects

is a Raspberry Pi 4B microcontroller, along with an HD 1080P USB web camera. The proposed system is relatively simple to use and provides acceptable accuracy for detecting the object's color and shape and measuring distance.

5 Limitations of the proposed system

An automatic robotic arm for classifying solid objects based on colors, shapes and sizes has been proposed in this work. The state-of-the-art deep learning model, YOLOv4 tiny, has been used to detect solid objects. However, the proposed system experiences some limitations, too. For instance, the implemented device can not classify the categories of the detected objects. For example, the proposed system can

Table 11 Comparison of the hardware system with existing works

References	Primary features	Processing unit	Distance measure	Position computation	Reported results
Sekkat et al. (2021)	Object detection and measure size	Raspberry Pi 3, Pi camera	N/A	Inverse Kinetic	Object grasping accuracy: 93%
Noman et al. (2022)	Object color, size and shape detection	Arduino Mega 2560 R3, Pixy Cam CMU	N/A	N/A	Color detect acc.: 80% Shape detect acc.: 100%, Shape detect acc.: 100% Design cost: $150
Intisar et al. (2021)	Object color size and shape detection	Arduino Uno, Webcam	N/A	Kinetic	Color detect acc.: 93.33% Shape detect acc.: 94.6% Shape detect acc.: 98.6% Design cost: $80
Chen and Hu (2023)	Object detection and grasping	RealSense D435 AI camera	N/A	N/A	Object gripping accuracy: 90%
Cong et al. (2023)	Object color detection	Raspberry Pi 4, Sony IMX219 camera	Pythagorean theorem	Preprogrammed position	Color detect acc.: 99.17% Sorting time: 1.8s
Sawant et al. (2022)	Color detection and positioning	Raspberry Pi 3B, OV5674 camera	N/A	Inverse Kinetic	Color detect acc.: 78.8%
Fernandes and Shivakumar (2020)	Color, shape detection and sorting	ATmega328 microcontroller, Webcam	Douglas-Peucker approach	N/A	N/A
Zhang et al. (2020)	Color detection and sorting	STC15F2K60S2 microcontroller, CCD camera	Predefined	N/A	N/A
Shaikat et al. (2020)	Color, height detection and sorting	Arduino Nano, camera	Predefined	N/A	Color detect acc.: 99% Height detect acc.: 99%
Sanjaya (2018)	Color detection and sorting	Arduino, Webcam	N/A	Inverse Kinematic	N/A
Hamzah et al. (2022)	Object position control	Arduino Uno, Webcam	N/A	Mamdani fuzzy inference system	Position control acc.: 99.33%
This work	Object color, size and shape detection	Raspberry Pi 4B, Webcam	Predefined	Preprogrammed position	Color detect acc.: 100% Shape detect acc.: 99.7% Shape detect acc.: 100% Response time: 19 s Design cost: $298

detect distinct object shapes, e.g., circle, but can not categorize ball, orange, etc. Different open-source datasets, e.g., MS COCO, ImageNet, CIFAR, etc., can be used to train the system to detect various classes of objects. Conversely, utilization of these large-scale datasets will reduce the instantaneous response time of the hardware device. There are scopes to improve the device's efficiency by introducing reinforcement learning techniques, Q-learning, SARSA, etc. In this technique, the robotic system will interact with its surrounding environment by taking action. The environment responds with feedback in the form of rewards or penalties based on its actions. Currently, the 6-DOF robotic arm can pick up and position objects within a maximum distance of about 40 cm. However, with the integration of a mobile platform, the robotic arm can achieve further mobility, but it will require an obstacle detection and avoidance system. The motivation of this work is to perform tasks in industrial workplaces alongside humans in a less mobile setup, such as sorting, picking and placing objects on products that are moving through a conveyor belt according to their color, shape and size.

We acknowledge that the concept presented in this article is a generic idea used in advanced cooperative robots such as Baxter, ABB Yumi, Motoman SDA-5F, etc. However, stability conditions in the robot's control loop and generating automatic setpoints for human-robot interaction are beyond the scope of this article. To understand the environment effectively and efficiently, depending on the importance of the ambient workplace, sensors ranging from simple, such as touch and avoid (e.g., iRobot), or complexes, such as LIDAR (e.g., autonomous cars), can be implemented. Nevertheless, our proposed system primarily focuses on expediting the training and action process for automatic detection. It serves as a cost-effective implementation to demonstrate the proof of concept.

6 Conclusions and future work

This paper designs and develops a computer vision-based automated system for classifying solid objects based on colors, shapes and sizes. A custom dataset of 6558 images of various monochromatic objects comprising five shapes, three colors and three sizes has been created. The proposed system identifies the color using the OpenCV framework, detects the shape using the PyTorch-based bounding box regression approach, and measures the size using the OpenCV image processing technique. This work achieved 100%, 99.7%, and 100% accuracy in color, shape, and size detection, respectively, in a powerful computing platform. A robotic system has also been developed using a Raspberry Pi 4B microcontroller to command a robotic arm to pick up the object by recognizing the object's shape and color. The robotic system with the Raspberry Pi 4B microcontroller achieved 80.7% accuracy in color detection, 81.07% and 59.77% accuracies in shape and distance measurement, respectively. The implemented device involves leveraging a small-scale private dataset and lightweight YOLOv4 tiny deep learning technique to reduce training duration and facilitate the adaptability of the proposed system to various components required for the development of new industrial products. In the future, more efficient and lightweight deep learning models can be used for object detection. Jetson Nano or more advanced edge devices and ultra-motion cameras can be employed to improve the performance of the robotic system. Reinforcement learning techniques can be used to enhance the performance of the proposed hardware device.

Declarations

Conflict of interest We certify that there is no actual or potential conflict of interest in relation to this article.

Supplementary material The source code and images of the collected dataset can be found at: https://github.com/TituShahoriar/cse499B_Hardware_Proposed_System.

References

Abdi, A., Ranjbar, M.H., Park, J.H.: Computer vision-based path planning for robot arms in three-dimensional workspaces using Q-Learning and neural networks. Sensors **22**(5), 1697 (2022)

Ahmed, I., Din, S., Jeon, G., Piccialli, F., Fortino, G.: Towards collaborative robotics in top view surveillance: a framework for multiple object tracking by detection using deep learning. IEEE/CAA J. Autom. Sin. **8**, 1253–1270 (2021)

Arunachalam, M., Sanghavi, V., Kaira, S., Ahuja, N.A.: End-to-end industrial IoT: software optimization and acceleration. IEEE Internet Things Mag. **5**, 48–53 (2022)

Azad, A., Misbahuddin, M.: Web-based object tracking using collaborated camera network. Adv. Internet Things **8**, 13–25 (2018)

Bai, Y.: Data Selection Query with Visual Basic.NET, pp. 215–325. Wiley-IEEE Press (2020)

Banerjee, D., Yu, K., Aggarwal, G.: Object tracking test automation using a robotic arm. IEEE Access **6**, 56378–56394 (2018)

Chen, L., Sun, H., Zhao, W., Yu, T.: Robotic arm control system based on AI wearable acceleration sensor. Math. Probl. Eng. **2021** (2021)

Chen, C.-S., Hu, N.-T.: Eye-in-hand robotic arm gripping system based on machine learning and state delay optimization. Sensors **23**, 1076 (2023)

Cong, V.D., Phuong, L., Chi, H., City, M., Kiji, S.: Design and development of a delta robot system to classify objects using image processing. Int. J. Electr. Comput. Eng. **13**, 2669–2676 (2023)

Divya, Y., Kumar, C.P.: Machine vision based color recognition by robotic arm using LabVIEW. CVR J. Sci. Technol. **18**, 100–104 (2020)

Fernandes, L., Shivakumar, B.: Identification and sorting of objects based on shape and colour using robotic arm. In: International

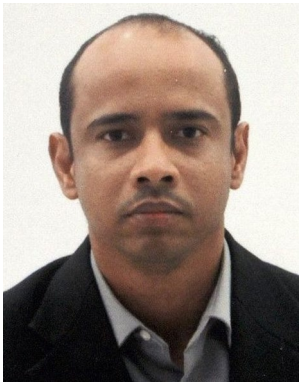
Conference on Inventive Systems and Control, pp. 866–871 (2020)
Gode, C.S., Khobragade, A.S.: Object detection using color clue and shape feature. In: International Conference on Wireless Communications, Signal Processing and Networking, pp. 464–468 (2016)
Grudin, J., Carroll, J.M.: From Tool to Partner: The Evolution of Human-Computer Interaction. Morgan & Claypool, San Rafael (2017)
Gupta, S., Singh, Y.J., Kumar, M.: Object detection using multiple shape-based features. In: International Conference on Parallel, Distributed and Grid Computing, pp. 433–437 (2016)
Hamzah, M., Thamrin, N., Tajjudin, M.: Robotic arm position control using Mamdani fuzzy logic on Arduino microcontroller. J. Mech. Eng. **19**, 235–255 (2022)
Imran, M.A., Hussain, S., Abbasi, Q.H.: Ultra Reliable Low Latency Communications as an Enabler for Industry Automation, pp. 89–107. Wiley (2020)
Intisar, M., Khan, M., Islam, M., Masud, M.: Computer vision based robotic arm controlled using interactive GUI. Intell. Autom. Soft Comput. **27**, 533–550 (2021)
Kafadar, O.: RaspMI: Raspberry Pi assisted embedded system for monitoring and recording of seismic ambient noise. IEEE Sens. J. **21**, 6306–6313 (2021)
Kondratenko, Y.P., Kuntsevich, V.M., Chikrii, A.A., Gubarev, V.F.: Part I: Advances in Theoretical Research on Automatic Control, pp. 1–1. River Publishers, Aalborg (2021)
Koodtalang, W., Sangsuwan,T.: The chicken's legs size classification using image processing and deep neural network. In: International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics, pp. 183–186 (2019)
Krishnadas, P., Sampathila, N.: Automated detection of malaria implemented by deep learning in PyTorch. In: International Conference on Electronics, Computing and Communication Technologies, pp. 01–05 (2021)
Liu, X., Jin, S.: Multi-target visual recognition and positioning methods for sorting robots. In: IOP Conference Series: Materials Science and Engineering, vol. 892 (2020)
Maasoumy, M., Sangiovanni-Vincentelli, A.: Smart connected buildings design automation: Foundations and trends. In: Foundations and Trends in Electronic Design Automation, pp. 1–143 (2016)
Magadum, R.A., Bombale, U.L.: Intelligent color based object sorting using robotics. Int. J. Robot. Res. Dev. **9**, 43–48 (2019)
Najmurrokhman, A., Kusnandar, K., Maulana, F., Wibowo, B., Nurlina, E.: Design of a prototype of manipulator arm for implementing pick-and-place task in industrial robot system using TCS3200 color sensor and ATmega2560 microcontroller. J. Phys. Conf. Ser. **1375**, 1–7 (2019)
Noman, A., Eva, A., Yeahyea, T., Khan, R.: Computer vision-based robotic arm for object color, shape, and size detection. J. Robot. Control **3**, 180–186 (2022)
Othman, N.A., Salur, M.U., Karaose, M., Aydin, I.: An embedded real-time object detection and measurement of its size. In: International Conference on Artificial Intelligence and Data Processing, pp. 1–4 (2018)
Parveen, S., Shah, J.: A motion detection system in Python and OpenCV. In: International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, pp. 1378–1382 (2021)
Phelps, R., Krasnicki, M., Rutenbar, R., Carley, L., Hellums, J.: Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **19**, 703–717 (2000)
Qayum, M.A., Nahar, N., Siddique, N.A., Saifullah, Z.: Interactive intelligent agents with creative minds: experiments with mobile robots in cooperating tasks by using machine learning. In: International Conference on Imaging, Vision & Pattern Recognition, pp. 1–6. IEEE (2017)
Rafique, D., Velasco, L.: Machine learning for network automation: overview, architecture, and applications. J. Opt. Commun. Netw. **10**, 126–143 (2018)
Ranjbarzadeh, R., Caputo, A., Tirkolaee, E.B., Jafarzadeh Ghoushchi, S., Bendechache, M.: Brain tumor segmentation of MRI images: a comprehensive review on the application of artificial intelligence tools. Comput. Biol. Med. **152** (2023)
Ranjbarzadeh, R., Ghoushchi, S., Anari, S., Safavi, S., Sarshar, N., Tirkolaee, E. Babaee, Bendechache, M.: A deep learning approach for robust, multi-oriented, and curved text detection. Cogn. Comput. 1–13 (2022)
Roy, S., Hazera, C., Das, D., Pir, R., Ahmed, A.S.: A computer vision and artificial intelligence based cost-effective object sensing robot. Int. J. Intell. Robot. Appl. **3**, 457–470 (2019)
Sanjaya, W.S.M., et al.: Colored object sorting using 5 DoF robot arm based artificial neural network (ANN) method. J. Phys. Conf. Ser. **1090**, 1–10 (2018)
Sawant, N., Tyagi, A., Sawant, S., Tade, S.L.: Implementation of faster RCNN algorithm for smart robotic arm based on computer vision. In: International Conference on Computing, Communication, Control and Automation, pp. 1–6 (2022)
Sekkat, H., Tigani, S., Rachid, S., Chehri, A.: Vision-based robotic arm control algorithm using deep reinforcement learning for autonomous objects grasping. Appl. Sci. **11**, 7917 (2021)
Shahid, M.A., Iftikhar, M.A., Gondal, Z.A., Adnan, M., Rathore, S.: Object size measurement through images: an application to measuring human foot size. In: International Conference on Frontiers of Information Technology, pp. 298–302 (2018)
Shaikat, A.S., Akter, S., Salma, U.: Computer vision based industrial robotic arm for sorting objects by color and height. J. Eng. Adv. **1**, 116–122 (2020)
Sivapriyan, R., Ajay, K.V., Ashwath, K.N.: Arduino-Nano based low cost power converter learning kit. In: International Conference on Inventive Systems and Control, pp. 133–137 (2020)
Sudhoff, S.: Power magnetic devices: a multi-objective design approach. In: Power Magnetic Devices: A Multi-Objective Design Approach (2014)
Wang, Z., Fan, Y., Zhang, H.: Lane-line detection algorithm for complex road based on OpenCV. In: Advanced Information Management, Communicates, Electronic and Automation Control Conference, pp. 1404–1407 (2019)
Wang, J., Long, L.: Raspberry Pi 4B-based design of home intelligent access control system. In: Asia-Pacific Conference on Image Processing, Electronics and Computers, pp. 959–962 (2022)
Wettinger, J., Andrikopoulos, V., Leymann, F., Strauch, S.: Middleware-oriented deployment automation for cloud applications. IEEE Trans. Cloud Comput. **6**, 1054–1066 (2018)
Zhang, W., Zhang, C., Li, C., Zhang, H.: Object color recognition and sorting robot based on OpenCV and machine vision. In: International Conference on Mechanical and Intelligent Manufacturing Technologies, pp. 125–129 (2020)

Md Fahim Shahoriar Titu obtained his Bachelor of Science in Computer Science and Engineering (BSCSE) from North South University, Dhaka, Bangladesh. He is interested in various fields such as metamaterial, machine learning (ML), artificial intelligence, big data analytics, cloud computing, natural language processing, cyber security, computer vision and robotics.

S. M. Rezwanul Haque is pursuing a Bachelor of Science in Computer Science and Engineering (BSCSE) at North South University in Dhaka, Bangladesh. He is currently working as an IT-System Engineer at Curtex-IT, Dhaka, Bangladesh. His main areas of interest include metamaterials, artificial intelligence, machine learning, natural language processing, and cloud computing.

Rifad Islam received a Bachelor of Science in Computer Science Engineering (BSCSE) from North South University in Dhaka, Bangladesh. His main areas of interest include artificial intelligence, machine learning, data analytics, NLP, cloud computing, cyber security, and web networks.

Akram Hossain received a B.Sc in Computer Science and Engineering (BSCSE) at North South University in Dhaka, Bangladesh. He is working on a startup in Internet Networking in Dhaka, Bangladesh. His main area of focus is creating a vast network of Internet Networking throughout the rural areas of Bangladesh.



Mohammad Abdul Qayum is currently working as an Assistant Professor in the Electrical and Computer Engineering department of North South University (NSU), Dhaka, Bangladesh. He completed his B.Sc. degree in Electrical and Electronic Engineering (EEE) from Bangladesh University of Engineering and Technology (BUET). He obtained his MSc in Electrical Engineering from Oklahoma State University, USA. He received his Ph.D. in Electrical Engineering from New Mexico State University, USA. His Ph.D. work was on Hybrid Transactional Memory for large scale graph applications. After completion of his Ph.D., he joined as a post-doctoral fellow in New Mexico State University where he worked on OS-friendly microarchitecture. Before joining NSU he worked in HIVE blockchain, Minnesota State University, Mankato, Banglalink (Global Telecom Holding). His current research area is blockchain, IoT and robotics.

Riasat Khan received his B.Sc. in Electrical and Electronic Engineering from the Islamic University of Technology, Bangladesh, in 2010. He continued his academic pursuits, obtaining both M.Sc. and Ph.D. degrees in Electrical Engineering from New Mexico State University, Las Cruces, USA, in 2018. Currently, he serves as an Assistant Professor in the Department of Electrical and Computer Engineering at North South University, Dhaka, Bangladesh. His research interests encompass machine learning, computational bioelectromagnetics, model order reduction, and power electronics.