SciEn
Journal of Engineering Advancements Vol. 04(03) 2023, pp 70-79 https://doi.org/10.38032/jea.2023.03.003

Automated Object Sorting System with Real-Time Image Processing and Robotic Gripper Mechanism Control

Safqut Sanwar and Md. Imteaz Ahmed*

Department of Mechanical Engineering, Chittagong University of Engineering & Technology, Chittagong, Bangladesh

Received: June 09, 2023, Revised: July 05, 2023, Accepted: August 05, 2023, Available Online: September 04, 2023

ABSTRACT

This work represents an industrial sorting system where image processing is accompanied using a pick-and-place robotic gripper. The sorting of objects is done based on their shape and color. Here the color and shape of different objects are identified using image processing. For this, a webcam is used to capture images of the object in real time and then process them via a digital computer. Python programming language is used for image processing in this work. After successfully identifying the color and shape of an object, the object is picked and placed at the desired position using the robotic gripper. Controlling the gripper mechanism is also executed using the Python programming language. It is controlled using the Arduino Uno microcontroller and a few DC servo motors. The gripper can move from 0° to 180°. The objects are brought in front of the camera using a belt conveyor system. After the complete fabrication and assembly, 4 objects of different shapes and colors are used to sort objects at 4 different angles. The objects are picked from 90° and is sorted in either 0°, 45°, 135°, or 180° position. This research work not only gives information about robotics but also can help industries sort complex objects automatically without any human interaction.

Keywords: Image Processing, Arduino UNO, Micro-controller, Object Sorting, Robotic Gripper, Python.

1 Introduction

Using a digital computer and algorithms to process a digital image is known as digital image processing. It is more advantageous than analog image processing because a wider range of algorithms can be applied here. From pattern recognition to feature extraction, many tasks can be done using digital image processing [1]. Image processing is used to not only identify the color but also the shape of the objects in an image. In the modern world, robotics, and automation are making production processes faster and easier and saving both the cost and time of production. Industries around the world are using automation for manufacturing, packaging, and many other applications. Object-sorting robotic arms are a major part of industries. There are thousands of production plants in the world where object sorting is necessary and a vital part of the production process. Sorting objects based on different parameters (i.e., shape and color) is a very important task in the production process [2]. Object sorting robots that operate with the help of image processing can save labor costs as well as increase efficiency in modern-day production. But for this, the program has to be trained more and more so that more complex objects can be sorted easily. This research work is about sorting objects based on shape and color by merging two major tasks together which are the recognition of objects using image processing and sorting objects using a robotic gripper [3].

Image is the best way to perceive one's surroundings. Just like human beings, computer vision, and image analysis can help a machine to understand an image by perceiving it electronically. Image processing is done mainly for two reasons: Analyzing the image to give certain information to the operator (Human). For example, using images from a CCTV camera to identify the face, attributes, etc. of a person. Analyzing the image and then saving and using the information to control or operate another machine automatically. For example, satellites take images in space in a different RGB combination and then the image processing programs convert them to make them understandable [1]. The whole image processing system consists of a few different components such as a sensor (camera), digital computer, display, storage, etc. Image processing uses thousands of images of the same object to make the machine learn and understand the shape, size, and functions of an object. That is why large storage is a very important factor here. A good video camera can help to capture better images and then the computer will process the image using certain algorithms and machine codes. The display is for viewing the acquired images for processing and for viewing the results after processing [4]. The image processing system performs certain operations on an image. These can be simultaneous or individual. The operations are,

Enhancement: Image processing can enhance an image by changing the color grading of the image. An image with low pixels can be enhanced by increasing the pixels using image processing.

Transformation: An image can be transformed into different data by using image processing. Every color has a different hue, saturation, and brightness values which can be separated using image processing.

Compression: If necessary, an image can be easily compressed by using image processing. This is just the opposite of enhancing where the pixels and colors are changed or reduced to reduce the size of the image.

Restoration: An image is nothing but a set of data. If data is lost it can be restored. So, image processing can help to restore lost data. An image is made of thousands of small pixels and image processing can be used to correct these pixels for restoration.

Description: As mentioned above, an image is nothing but a set of data. There are some numerical values in an image,

*Corresponding Author Email Address: u1603050@student.cuet.ac.bd Published by: SciEn Publishing Group

these values can be hue, saturation, brightness, or any other values. Image processing helps to describe different objects based on these values.

Segmentation: An image can be divided into a few segments so that each segment can be used for one particular task. Image processing divides images into necessary segments to separate different areas of an image and this makes it easier for the program to process the segmented part rather than collecting the whole image.

Interpretation: Image processing helps to explain an image based on different numerical values, data, colors, and many other aspects. The action of explaining a particular parameter using image processing is called interpretation in this case.

Recognition: Image processing can be used to recognize anything present in an image. It can be a person or an object. Whichever it is, the shape, color, size, attribute everything can be recognized using image processing.

These operations help both humans and other machines to find out the details in an image and then use that for different applications [1].

The sorting of the objects based on shape and color is done using a robotic gripper. An industrial robotic arm was first introduced in 1961 by a company named Unimate. It was eventually developed into the PUMA (Programmable Universal Machine for Assembly) arm. Nowadays industries are relying more and more on robotic arms for assembling, sorting, delivering, and other purposes [5]. There are various types of robotic arms, and they all have different applications. For example,

Gantry Robotic Arm – This type of arm uses a Cartesian coordinate system to move and so it is also known as a Cartesian Robotic Arm. The Gantry Robotic Arm is used to perform assembly operations, arc welding, handling machine tools, etc.

Cylindrical Robotic Arm – The axes of this type of robot are from a cylindrical coordinate system. It is mostly used for spot welding and die-casting operations.

SCARA Robotic Arm – Two parallel joints have rotary motion in this type of robot. This motion provides plane compliance. The SCARA robotic arms are used to pick and place objects.

Cobot Robotic Arm – The rounded edges and the lightweight construction of these robots make them safe for industrial applications. The Cobot Arms are usually isolated from contact with humans.

Polar Robotic Arm – Instead of using a cylindrical or Cartesian coordinate system, these robotic arms use a polar coordinate system. These are suitable for gas welding, fettling machines, etc.

Articulated Robotic Arm – A robotic arm with a minimum of three rotary joints is known as Articulated Robotic Arms. Spray painting, welding, diecasting, etc. are some of the main functions of these robotic arms [6].

Robotic arms are often described by their degrees of freedom or DoF in short. DoF defines the state of an object or machine by its independent parameters. There are many types of DoF for robotic arms (i.e., 4-DoF, 6-DoF, etc.). However, this research work will use a robotic arm of 4-DoF with a simple gripper mechanism to pick up an object and place it. The goal is to see how efficiently it can place an object after its image is processed by the computer using certain algorithms [7].

2 Literature Review

Previously many works have been done on automatic object sorting using image processing, sensors, processors, robotic arms, etc. In those works, objects of different colors or shapes or other parameters were detected and then the objects were sorted based on those parameters. Some of the highlighted works are mentioned below.

Joy [8] made an object separating a robotic arm based on the color of the object. An ARM7-2138 microcontroller was used to control the arm, and the object was of three different colors. The color detection was done using the light intensity to frequency converter method. Moreover, the robotic arm was controlled using DC servo motors. For light-to-frequency conversion, the TCS3200 programmable convert was used. Nicolaus et al. [9] used a 6-DoF robotic arm to sort objects based on color as well. Here they used two cameras one of which was used to search for objects while the other one was sending digital feedback to the computer and robotic arm. They used MATLAB for programming the codes for image processing and the robotic arm. Jia et al. [10] used image processing to detect the color of the objects and then used the Phantom X Reactor robotic arm to pick and place the object. The arm was controlled using the ArbotiX-M microcontroller. They used OpenCV rather than MATLAB as it is more convenient. Also, they used the C++ programming language to write the codes for image processing and controlling the robot. Shah and Pandey [11] made an automatic sorting robotic arm at low cost using an Arduino UNO3 microcontroller; VGA camera; barcode and QR code scanner; Tactile, RGB, and color sensors. But they didn't use any image processing here. It was a simple low-cost automatic robotic arm system that could detect obstacles and objects through the sensors and then sort them according to the given commands. Sampreeth et al. [12] used image processing and controlling a robotic arm to sort objects based on shape and color. However, they used the ARM7 microcontroller for controlling the robotic arm. Moreover, they used coding on MATLAB for image processing and microcontroller commands. Kale and Kulkarni [13] proposed an object-sorting system using a robotic arm. The proposal was to use a microcontroller to control the arm and a camera to capture the moving object's color and shape. The object would move on a conveyor belt and the belt would be controlled with a motor drive. For image processing they used MATLAB, and they used the ATmega328 CMOS 8-bit microcontroller. Amin et al. [14] created a mechanism to detect and distinguish the color of an object and then place it in a designated position. They used a robotic arm for the pick and place mechanism, but they did not use any image processing. Instead, they used the TCS 3200 color sensor to detect the color of the object. The robotic arm was controlled using a PIC microcontroller (18F452). The robotic arm could rotate up to 360° angle. Mohammed et al. [15] also used a color sensor (TCS230) and a PIC 16F628A microcontroller to sort objects based on color. A conveyor belt was used to move the objects and the color sensor was placed above the objects that could sense the color and then transfer the object to a predetermined position. Another conveyor belt was used to hold the different boxes where the objects would drop according to color. There was no robotic arm used in this research work. Also, the objects were sorted only based on the colors. Bhaskar et al. [16] made a sorting system based on color using a robotic arm. They used the SSC-32 servo control board to control the robotic arm. They used a webcam to capture the images of the objects and then identify their colors. For color detection, they used an application called Lab Windows/CVI from national instruments. After identifying the color, the computer sent a signal to the robotic arm and the arm placed the objects based on the color. Deshmukh et al. [17] used the PIC 16F877A microcontroller to control a conveyor belt system for sorting objects based on color. Did not use any camera, but the TSOP 1738 color sensor, which detected the colors of the different objects and then the conveyor belt sorted the objects according to their color. A robotic arm was also not used in this case. This research work also did not use the shape of the objects for sorting. Shirgave et al. [18] used an Arduino UNO controller, servo motors, and the TCS3200 color sensor to build a color-sorting robot. The robot was not a moving robot that could rotate its arm at a certain angle to place an object based on color. It was an efficient robot but it could not sort the object based on any other criteria except for the color of the object. Suryawanshi et al. [19] made a robotic arm using the Arduino microcontroller for sorting objects based on color. However, they did not use any camera to capture images of the object for color detection. They used the TCS34725 color sensor to detect the color of the objects. To make the robotic arm move they used DC servo motors, and the motors were driven using the L293D motor driver. Patel et al. [20] made an object-sorting machine based on IOT and color. They used the Arduino UNO microcontroller to control a robotic arm for picking and placing the object. For color detection, they used the TSC3200 color sensor. They also use a GSM module in their research. However, their goal was to sort objects based only on color. Jakkan et al. [21] Used both Arduino UNO and ATmega328 microcontrollers to control a robotic arm that was able to pick and place objects based on their colors. The colors of the objects were identified using the TCS3200 color sensor instead of image processing. They also used a Wi-Fi module (ESP8266) and a mobile device in the research. But they did not show the objects based on any other criteria. Patil et al. [22] reviewed a paper where they introduced a model that could sort objects based on color. They used C++ programming language to create codes for an Arduino microcontroller to control a conveyor belt. The conveyor belt sorted objects based on colors and if the color was not specified by the color sensor, the object went to a separate box. They did not use any robotic arm to sort the objects in this case. Sireesha et al. [23] created an automatic object sorting system that was able to sort waste objects based on thickness and color. They created a system without any robotic arm, but an Arduino microcontroller was used to move a part of the system to a certain angle to place the object. Instead of using an image processing system they simply used a color sensor (TCS3200) which could identify only a certain number of colors. Ghosh et al. [24] made a product-organizing tool based on color. They also used an Arduino UNO microcontroller and TCS3200 and TCS230 color sensors to identify the colors of the objects placed in the system and control the servo motors to control the parts of the objects to a certain position. However, they did not use any image processing or robotic arm for this. Also, the objects were supported only based on color and no other features. Simran et al. [25] developed a system for automatically sorting objects using a conveyor belt. The objects were identified using a photoelectric sensor (beam-type) in this research. The different identified objects were then placed in different positions using the conveyor belt. A programmable logic controller (PLC) helped to control the conveyor belt. The programming was done using the MicroWIN software. This research did not use any method to identify the shape of the object. Shen and Hassan [26] created a robot that could sort objects based on their colors. The robot was controlled using an Arduino UNO microcontroller. They used the TCS3200D color sensor to identify the colors of the objects. After identifying the color, the robot could place the objects into their designated color station in a short time. However, the robot could not sort the objects based on their shapes. Zulfiqar et al. [27] used a programmable logic controller to control a conveyor belt for sorting objects based on their material type. To identify the material, they used a proximity sensor. It detected wood, metal, and plastic and then placed the objects in the required positions. However their system could not detect the objects based on their color. Also, they only used censors, and there was no use of any image processing system. Jahan et al. [28] made an object-sorting machine based on color. They used the Arduino microcontroller to control a belt conveyor system to sort the objects. For color detection, they used the TSC 34725 RGB color sensor. They also included a feature where white objects would make the belt system move backward. However, their goal was to sort objects based only on color.

Most of the previous studies either sorted the robot based on shape or color except for Sampreeth et al. [12]. However, some of the research didn't even use image processing. Also, one of the types of research used two cameras which is less convenient than using a single camera. Only a few of the research used OpenCV and Python coding, which is more efficient than coding on MATLAB. Finally, the ARM7 microcontroller used in most of the previous research is more complex than Arduino. This research work utilized Python programming language for coding the image processing work, and for giving a command to the microcontroller of the robotic arm. At first, the whole system is designed using SOLIDWORKS. Then each of the parts is fabricated and then parts are assembled to merge the whole system. Instead of the ARM7 microcontroller, Arduino is used which is simpler and less costly. Also, the objects are not sorted only based on the color or the shape but on both parameters. From all the previous studies a conclusion came to that image processing can be a better and more efficient way to identify an object's shape and color than using any sensor. This is why the goal of this research is to utilize the benefits of image processing and merge it with robotics to create an automatic object-sorting system that will eventually help a lot of industries.

Table 1 Comparative View of the Previous Studies

Author(s)	Summary	Parameter	Detection Process	Sorting Process	Limitations
Joy [8]	Object sorting robotic arm based on color	Color	Light to frequency converter	Robotic arm	Only color detection
Nicolaus et al. [9]	Development of an autonomous ball picking robot	Color	Image processing	Robotic arm	Only color detection
Jia et al. [10]	Real-time color-based sorting	Color	Image processing	Robotic arm	Only color detection
Shah and Pandey [11]	Concept for Automated Sorting Robotic Arm	Undefined	Tactile Sensor	Robotic arm	No particular parameter
Sampreeth et al. [12]	Object sorting robot using image processing	Color and Edge	Image Processing	Robotic arm	No particular shape detection
Kale and Kulkarni [13]	Object sorting system using robotic arm	Color and Shape	Image Processing	Robotic arm	Proposal only
Amin et al. [14]	Detection and distinction of colors using color sorting robot	Color	RGB color sensor	Robotic arm	Only color detection
Kunhimohammed et al. [15]	Automatic color sorting machine using color sensor	Color	TCS230 Color Sensor	Belt conveyor	Only color detection
Bhaskar et al. [16]	Colour sorting system with robotic arm	Color	Image processing	Robotic arm	Only color detection
Deshmukh et al. [17]	Automatic conveyor color sorting	Color	Color sensor	Belt conveyor	Only color detection
Shirgave et al. [18]	Color sorting robot	Color	TCS3200 color sensor	Slot moving mechanism	Only color detection
Suryawanshi et al. [19]	Color sorting robotic arm	Color	TCS34725FN color sensor	Robotic arm	Only color detection
Patel et al. [20]	IOT color-based object sorting machine	Color	TCS3200 color sensor	Robotic arm	Only color detection
Jakkan et al. [21]	Color based product sorting machine	Color	TCS3200 color sensor	Robotic arm	Only color detection
Patil et al. [22]	Review on colored object sorting system	Color	TCS3200 color sensor	Belt conveyor	Only color detection
Sireesha et al. [23]	Automated sorting system based on color and thickness	Color and Thickness	TCS3200 color sensor and ultrasonic sensor	Slot moving mechanism	No particular shape detection
Ghosh et al. [24]	Color-based product organization tool	Color	TCS3200 color sensor	Slot moving mechanism	Only color detection
Simran et al. [25]	Development of automatic sorting belt conveyor	Height	Photoelectric sensor	Belt conveyor	Only height detection
Shen and Hassan [26]	Design and development of color sorting robot	Color	TCS3200 color sensor	Driving robot	Only color detection
Zulfiqar et al. [27]	PLC Based Automated Object Sorting System	Material	Proximity sensors	Belt conveyor	Only material type detection
Nuva et al. [28]	Color and weight-based sorting system on belt conveyor	Color and Weight	TCS 34725 RGB color sensor and load cell	Belt conveyor	No particular shape detection

3 Methodology

This research work is about sorting objects based on their shape and color using image processing. Here a webcam is used to scan the image of objects of different shapes and colors. After scanning the image, the results are sent to the Arduino microprocessor that controls the robotic arm. The robotic arm then picks up the object and places it in the desired place. For example, if the object has a blue color rectangular shape, it is placed in a position. If it has a red color and rectangular shape, it is placed in a different position. Similarly, if it has a blue color and circular shape, it is placed in a third different position, and so on. The objects pass by the camera with the help of a belt conveyor. The camera is placed on one side of the conveyor. The robotic arm is on the end part of the belt conveyor. As soon as the object comes in front of the camera, it captures the image, and the computer identifies it using image processing and then sends a signal to the arm through the Arduino microcontroller for picking and sorting.

3.1 Fabrication and Assembly

Designing a part, component, or setup is one thing, but fabrication is totally different. In this research work, SOLIDWORKS was used to design the main component or the setup of the research. However, these components were fabricated after the design to make the vision of the setup a reality. This chapter describes the fabrication and assembly process of the whole setup.

This research contains a few major components and each of the components was designed, and the models were used to fabricate them. After that, those components were assembled.

1) Fabrication of the Robotic Gripper: The robotic gripper was designed using SOLIDWORKS and it was a typical model of the arm. After designing the arm, it was time to make a list of the components that were needed to fabricate the arm. The components or parts used to fabricate the arm were bolts, nuts, servo motors, 3D-printed arm parts, and wires. At first, the individual parts of the arm were 3D printed. Then the parts were connected to each other using bolts and nuts so that the parts could move but stay connected to each other. Then 4 servo motors were connected to the parts. The function of the motors is as follows. 1 is used to control the base. 1 is used to control the gripper. The other 2 motors on the left and the right side are used to make the gripper move forward-backward and up-down. All the servo motors are connected to wires. There are 3 wires of a servo motor, brown, red, and yellow. The red wire is connected to the voltage source, the yellow wire is connected to the PWM (pulse width modulation) pins of Arduino Uno, and the brown wire is for ground connection. Fig. 1 shows the robotic (arm) after it is fabricated.

Fig. 1 The fabricated robotic gripper

The robotic gripper has a total of 4 degrees of freedom (DOF). Here's an explanation of the DOF for the robotic gripper:

Base Rotation (1 DOF): The gripper arm has a servo motor mounted at the base, which allows it to rotate. This rotational movement provides the arm with the ability to change its orientation within a 180° range. This DOF enables the arm to cover a wide area without physically moving its position.

Arm Forward and Backward Movement (2 DOF): Two servo motors are employed to control the main body of the arm. These motors enable the arm to move forward and backward, extending or retracting the arm's length from its base. This movement provides the gripper with the ability to reach objects at different distances, extending its operational range.

Claw Up and Down Movement (1 DOF): One servo motor is dedicated to controlling the movement of the claw mechanism. This motor allows the claw to move up and down, enabling it to grip objects of varying heights. This DOF is crucial for the gripper to adapt to objects with different vertical positions.

By combining the 1 DOF for base rotation, the 2 DOF for arm movement, and the 1 DOF for claw movement, we have a total of 4 DOF for the robotic gripper. These degrees of freedom provide the gripper with the necessary flexibility and range of motion to perform object-sorting tasks effectively.

2) Fabrication of Detecting Object Shape and Color: The camera or webcam that is used to capture the images for shape and color detection is Fifine K420. This is connected to the computer using the USB port and it captures and sends data to the computer for image processing. In this research work, the object's shape, and color both are detected using a single camera. An algorithm in Python had to be developed to detect both at the same time. The working mechanism of the algorithm is very simple. At first, the code runs and detects the color of the object. As soon as it detects the color, it then operates to detect the shape of the object. The simple "if" and "elif" statements are used to guide the code to pick the color and then the shape. The color is converted from RGB (Red, Green, Blue) value to HSV (Hue, Saturation, Value) so that they can be captured through the HSV values of the object. This is more precise as different colors have different saturation levels. However, after detecting the color, the code detects the shape of the object. For this, another set of codes is run. Here the higher and lower HSV are set using the trackbar that is also generated when the code runs. This is how the camera is used to detect the color and shape of the object. Here are some key points on image processing or OpenCV used in this work:

Image Acquisition: OpenCV's VideoCapture function is utilized to acquire video frames from a specified video source (in this case, the default camera, identified as index 0). The acquired frames serve as the input for subsequent image-processing operations.

Color Space Conversion: The cv2.cvtColor function from OpenCV is employed to convert the acquired frames from the default BGR color space to the HSV color space. This conversion enables better color-based object detection and analysis.

Thresholding: Within the HSV color space, OpenCV's cv2.inRange function is used to perform thresholding. This function creates a binary mask by selecting pixels within a specific range of hue, saturation, and value (HSV) values. Thresholding helps isolate objects of interest based on their color characteristics.

Contour Detection: OpenCV's cv2.findContours function is employed to detect contours within the binary mask obtained through thresholding. Contours are important for shape analysis and identification of objects within an image. The cv2.RETR_TREE and cv2.CHAIN_APPROX_SIMPLE arguments specify the contour retrieval mode and approximation method, respectively.

Shape Approximation: The cv2.approxPolyDP function from OpenCV is utilized to approximate the contours and simplify their representation. This function uses the Ramer-Douglas-Peucker algorithm to reduce the number of points in a contour while preserving its overall shape.

Object Recognition and Classification: Based on the number of sides in the approximated contour, the objects are classified as either rectangles or circles. This classification enables the identification of different shapes based on their contour characteristics.

3) Fabrication of the Belt Conveyor: The belt conveyor was designed using SOLIDWORKS and it was a typical model of the conveyor system. After designing the belt conveyor, it was time to make a list of the components that were needed to fabricate the system. The components or parts used to fabricate the belt conveyor were ball bearing, conveyor belt, roller pipes, DC motor, bolts and nuts, switches, AC power adapter, and aluminum frame. At first, the roller pipes were attached to the bearings and one roller pipe was connected to the DC motor. Then the belt was mounted over the rollers tightly to achieve enough friction that would move the belt when the rollers rotated. Then the rollers were mounted on the aluminum frame using nuts and bolts. The DC motor can be controlled forward and reverse using a switch. The switches were connected to the AC power adapter so that the voltage level of the motor could be limited. Fig. 2 shows the belt conveyor after it is fabricated.

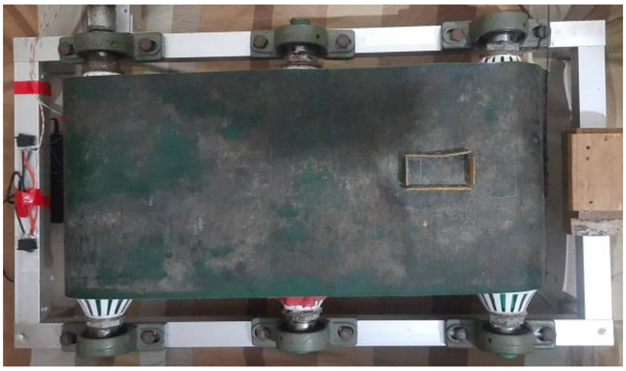

Fig. 2 The fabricated belt conveyor

4) Complete Assembly of the Work: After fabricating all the major components, the complete system was assembled. The robotic gripper was attached to one end of the belt conveyor frame. The servo motors of the robotic gripper were then connected to the Arduino Uno microcontroller. A breadboard was used to distribute the ground and power connection. The microcontroller was connected to the digital computer using a USB cable. Another USB port of the computer was used to connect the webcam to the system. The webcam was placed on one side of the belt conveyor. An AC power adapter was used to supply power to the DC motor of the belt conveyor. The motor and the adapter were connected by 2 switches one of which made the system rotate forward and the other one made it rotate backward. Below is Fig. 3 which shows the fabrication of the complete assembly.

Fig. 3 The actual setup of the system

5) Circuit Design and Diagram of the System:

The circuit design of the robotic arm system incorporates several key components, including servo motors, a 12V DC motor, an Arduino Uno microcontroller, and a 12V 5A AC power adapter. The circuit is designed to enable precise control and manipulation of the robotic arm for object sorting based on shape and color. The novelty of the circuit lies in its integration of different components to achieve the desired functionality. The Arduino Uno microcontroller serves as the central control unit, coordinating the actions of the servo motors and the DC motor. Fig. 4 shows the circuit diagram of the system.

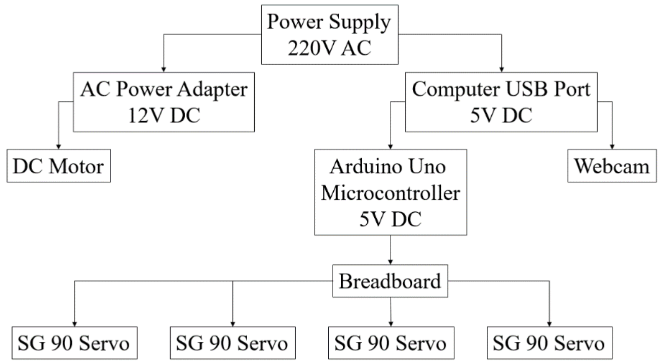

Fig. 4 Circuit diagram of the system

It receives input from the video stream captured by the camera and processes the frames using the OpenCV library. The servo motors, with their specified torque, operating speed, and range of rotation, provide the necessary actuation for the arm's base, main body, and claw. These motors allow the arm to rotate, move forward and backward, lift objects, and grip them securely. The specific angles and durations for the servo motor movements are programmed in the code to perform the desired sorting actions. The 12V DC motor, with its gear ratio and speed specifications, is likely employed to drive specific mechanisms or perform additional functions in the arm assembly.

While the exact details of its usage are not explicitly mentioned in the provided information, it could be utilized for tasks such as base rotation or additional arm movements. The 12V 5A AC power adapter is responsible for providing the necessary power to drive the entire system. It supplies the required voltage and current to ensure reliable and consistent operation of the motors and the microcontroller. The logic flow of the circuit involves capturing video frames, processing them using OpenCV for color and shape analysis, and based on the identified objects, controlling the servo motors to perform the appropriate sorting actions.

The code implements various image processing techniques, such as color space conversion, thresholding, contour detection, and shape approximation, to accurately identify objects and classify them based on their shape and color. The novelty of the circuit design lies in its integration of image processing algorithms with the control of a multi-axis robotic arm. By combining the power of OpenCV for real-time object detection and analysis with the precise control of servo motors, the system enables automated sorting based on shape and color.

6) Flowchart of the Sorting Process: Fig. 5 shows flow chart of the sorting process.

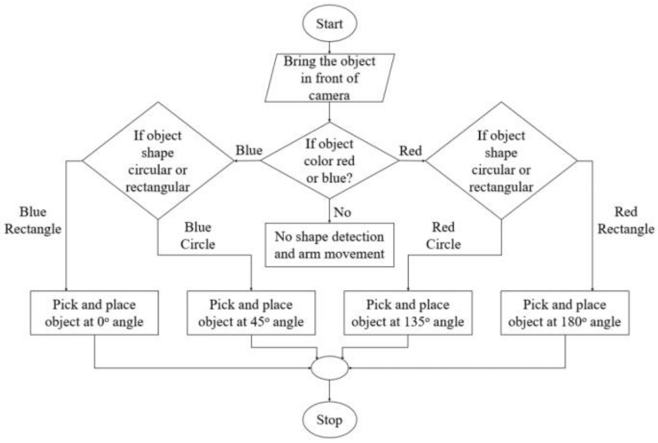

Fig. 5 Flow chart of the sorting process

The algorithm used in the provided code combines computer vision techniques with Arduino control to create a real-time object detection and sorting system. Here's an explanation of why this specific algorithm was chosen over others:

Color-based object detection: The algorithm leverages the HSV color space to detect objects based on their color. HSV representation is often preferred for color-based object detection as it separates the color information (hue) from brightness and saturation. This makes it more robust to changes in lighting conditions and shadows compared to other color spaces like RGB or grayscale.

Shape approximation: The algorithm utilizes contour detection and shape approximation techniques to determine the shape of the detected objects. By approximating the contours, it can classify objects as either rectangles or circles based on the number of vertices. This approach provides a simple and effective way to categorize shapes in real time.

Real-time processing: The algorithm is designed to process video frames in real time, allowing for live object detection and sorting. By continuously capturing frames from the camera, it can detect and respond to objects on the fly, making it suitable for interactive applications and dynamic environments.

Integration with Arduino: The algorithm integrates with an Arduino board to control a robotic gripper. By sending signals to the Arduino, it can manipulate the gripper based on the detected shape and color. This combination of computer vision and hardware control enables the system to perform physical sorting tasks based on visual cues.

Simplicity and versatility: The chosen algorithm strikes a balance between simplicity and versatility. It uses straightforward techniques like color thresholding, contour detection, and shape approximation, making it relatively easy to understand and implement. Moreover, by adjusting the color thresholds and shape criteria, the algorithm can be adapted to detect and sort objects of different colors and shapes, providing flexibility for various sorting scenarios.

While there are other object detection algorithms and frameworks available (such as deep learning-based approaches like YOLO or Faster R-CNN), they may require more computational resources, training data, and specialized hardware. The chosen algorithm, on the other hand, offers a simpler and more accessible solution for real-time object detection and sorting, combining traditional computer vision techniques with Arduino control.

3.2 Working Principal of the System

Here is how the whole system works to automatically sort objects based on shape and color:

1) At first, the code is run with PyCharm IDE, and the camera opens a Window on the desktop that shows an undefined color.
2) Then the object is brought in front of the webcam using the belt conveyor. Here the object is placed on top of the belt and then the forward switch is pressed.
3) As soon as the object comes in front of the camera, the code enables the camera to automatically detect the color of the object.
4) Then another Window opens by the code which enables the camera to detect the shape of the object.
5) After detecting the color and the shape, the code sends a signal to the Arduino Uno microcontroller to operate the servo motors of the robotic gripper.
6) Based on the shape and the color, the robotic gripper then places the object in either one of the 4 co-ordinates:
    a) If the color is blue and the shape is rectangular, the object is placed at a 0° angle.
    b) If the color is blue and the shape is circular, the object is placed at a 45° angle.
    c) If the color is red and the shape is rectangular, the object is placed at a 180° angle.
    d) If the color is red and the shape is circular, the object is placed at a 135° angle.

This is how the whole system works to automatically sort objects of different shapes and colors. Arduino Uno is used as a microcontroller board for the object sorting system. While Arduino Uno is not typically considered an industrial-grade system, it has been used in this context as a model or prototype to demonstrate the concept and functionality of the sorting system. The choice of Arduino Uno in this implementation is primarily driven by its accessibility, ease of use, and widespread adoption in the maker and prototyping communities. Arduino boards, including Arduino Uno, offer a user-friendly development environment, a range of input/output pins, and extensive community support.

These factors make Arduino Uno an ideal choice for quickly developing and testing proof-of-concept systems, such as the object sorting system described in the code. However, in an actual industrial setting, where robustness, scalability, and reliability are crucial, a more advanced and powerful microcontroller or industrial-grade system would be required. These systems often offer features like higher processing power, expanded input/output capabilities, industrial communication protocols, and rugged designs to withstand harsh industrial environments. The implementation described in the code and the provided paper data serve as a starting point or an initial prototype to showcase the feasibility of the object sorting concept.

When transitioning this system to an industrial setting, it would be necessary to select an appropriate microcontroller or industrial automation platform that meets the specific requirements and demands of industrial applications. Using Arduino Uno in the initial development stage, allows for rapid prototyping, iterative testing, and validation of the concept before investing in more specialized and industrial-grade hardware. This approach helps to reduce development time, cost, and risks associated with implementing a complex system directly in an industrial environment.

4 Results and Discussions

Results are the outcomes that are perceived after running a system. Here the system built into this research work automatically sorts objects based on shape and color. After testing several times, the results helped to identify how accurate and precise the sorting system was. Here, two shapes and two colors were used to test the accuracy and precision of the system. In this chapter, the accuracy and precision of the system are discussed via a few tables. Finally, a discussion is included based on the results that are found.

4.1 Accuracy and Precision

The tables below show the accuracy and the precision of the system for sorting 2 different colors and shapes:

Table 2 Color Calibration of the Camera

Actual Color	Color Detected by Camera			Accuracy	Precision
	1st Value	2nd Value	3rd Value		
Red	Red	Red	Red	100%	High
Blue	Blue	Blue	Blue	100%	High

Table 3 Shape Calibration of the Camera

Actual Shape	Shape Detected by Camera			Accuracy	Precision
	1st Value	2nd Value	3rd Value		
Rectangular	Rectangular	Rectangular	Rectangular	100%	High
Circular	Circular	Circular	Circular	100%	High

Table 4 16 Samples' Sorting Chart on the basis of Color and Shape

Sl. No.	Actual Color and Shape	Detected Color and Shape	Angle of Placement	Accuracy	Precision
1	Blue Rectangle	Blue Rectangle	0°	100%	High
2	Blue Circle	Blue Circle	45°	100%	High
3	Red Rectangle	Red Rectangle	135°	50%	Medium
4	Red Circle	Red Circle	135°	100%	High
5	Blue Rectangle	Blue Rectangle	0°	100%	High
6	Red Rectangle	Red Rectangle	180°	100%	High
7	Red Circle	Red Circle	135°	100%	High
8	Blue Rectangle	Blue Rectangle	45°	50%	Medium
9	Blue Circle	Blue Circle	45°	100%	High
10	Red Circle	Red Circle	135°	100%	High
11	Blue Rectangle	Blue Rectangle	0°	100%	High
12	Blue Circle	Blue Circle	45°	100%	High
13	Red Circle	Red Circle	135°	100%	High
14	Red Rectangle	Red Rectangle	135°	50%	Medium
15	Blue Circle	Blue Circle	45°	100%	High
16	Red Rectangle	Red Rectangle	180°	100%	High

4.2 Discussion on Results

This chapter showed several tests performed on the object to identify the shapes and colors and then sort the objects using a robotic gripper at desired angles. After the tests, the results were put down in a table which finally showed the accuracy and the precision. After running the system several times, it is seen that the accuracy and precision are quite good. If the color and shape detection is done using Python OpenCV, then it is always precise and accurate. Moreover, a camera of higher resolution and better specification is also responsible for the outcome. Finally, if the assembly of the robotic gripper is done correctly and the codes are put in proper order, then the sorting is also precise and accurate. The angle accuracy is quite high with an error of ±5%.

In certain instances, the accuracy of the system may experience a reduction of up to 50% when operating at specific angles. This decrease in accuracy can be attributed to environmental factors, particularly variations in lighting conditions, which can impact the camera's ability to accurately capture the true colors or shapes of objects. Lighting plays a crucial role in computer vision systems as it affects the way objects are perceived by the camera. When the lighting conditions are not optimal or inconsistent, the camera may struggle to accurately distinguish between different colors or accurately detect shapes. This can lead to a decrease in the system's overall accuracy when identifying and classifying objects based on their color or shape characteristics. Lighting conditions can vary depending on the environment in which the system is deployed. Factors such as natural lighting, artificial lighting, shadows, and reflections can all contribute to the fluctuations in lighting conditions. These variations pose a challenge for the camera's image processing algorithms, potentially causing a decline in accuracy at specific angles where the impact of lighting is more pronounced.

5 Conclusions and Future Work

In this era of automation, industries around the world are using the most advanced technologies to reduce the time of production as well as save costs. A great contribution can be made here by automatic object sorting via image processing for shape and color detection. A robotic arm that is capable of automatically sorting objects based on shape and color will cut the cost of labor and increase efficiency in the sorting and packaging of products. So, all in all, an advanced object-sorting robotic arm can be a huge benefit for any industry in the future.

A. Recommendations

There are certain things that are recommended for this research. For example,

- The servo motors should be connected to the correct PWM pin of the Arduino. Because, in the code, the pins are defined, and they control which servo should be rotated. The ground and the power connection should also be checked.
- It is important to check the USB port where the Arduino Uno microcontroller is connected. Because, in Python, the Arduino Uno microcontroller is imported via the PyFirmata library, and it is necessary to define the port in the code.
- When the webcam is connected to the computer, it also needs to be defined correctly in the code. Without properly defining the camera, the code will not open the right camera and the system will not work properly.
- Another important recommendation is to use an adapter to supply power to the DC motor of the belt conveyor. Because there is a voltage limit for the motor excess voltage will burn the coil and the motor will fail.

B. Suggestions for Future Work

Suggestions for future work means suggesting such methods that provide a further implementation of this research work. Nowadays automatic sorting process is not only a low-cost operation method but also plays a vital role in the industrial enhancement of any country. Here, this research has been developed for a color and shape-based sorting system and a simple pick-and-place mechanism of the robotic gripper. But this can be improved by:

- Adding a stronger and more versatile robotic gripper arm.
- Adding weight, height, size, and multiple colors, to the objects.
- Automating the belt conveyor and merging it with the whole system.

However, by improving this research work, it can be implemented in production plants to increase efficiency and save time on production as well as reduce cost.

In this work, a small-scale prototype of an industrial sorting system was used to sort objects based on various shapes and colors. However, this can be implemented in industries by making a system on a bigger scale for which a few modifications will be necessary. First, the arm body in this prototype was made of plastic whereas a more durable but lightweight material (such as aluminum alloys) can be used to make the body of the arm. The servo motors that are used in this work to control the arm are less powerful whereas a more powerful motor can help to lift heavier objects. Furthermore, more complex objects with complex shapes and various colors can be handled by implementing image processing in a more versatile way. In this project, the OpenCV library function of Python was used to identify a color by measuring the hue and other parameters. Moreover, the shapes were identified by calculating the contours where OpenCV was used as well. But in industries, more shapes and colors can be detected by either using this method or by implementing other methods of image processing. Overall, there are definite ways to improve the system and upgrade it to an industrial level by improving the specifications of the components used in this work.

Acknowledgments

The Chittagong University of Engineering & Technology's Mechanical Engineering department provided us with some helpful technology, for which we are appreciative. We want to express our sincere appreciation to everyone who helped us in any manner throughout this research's development. Lastly, we want to thank the Almighty for everything He has made possible.

References

[1] Gonzalez, R.C. and Woods, R.E. (2018) Digital Image Processing. Uttar Pradesh, India: Pearson.

[2] Groover, M.P. et al. (1986a) Industrial Robotics: Technology, programming and applications. New York: McGraw-Hill.

[3] Siciliano, B., Sciavicco, L. and Villani, L. (2009) Robotics modelling, planning and Control. New York: Springer.

[4] Pratt, W.K. (2007) Digital Image Processing: PIKS scientific inside. Hoboken, NJ: Wiley-Interscience.

[5] Robotics: Modelling, planning and control (2009a). London: Springer.

[6] CORKE, P. (2023) Robotics, vision and Control: Fundamental Algorithms in MATLAB. S.l.: SPRINGER.

[7] CRAIG, J.J. (2005) Introduction to robotics: Mechanics and control. Upper Saddle River: Pearson Prentice Hall.

[8] Joy, A., 2014. Object sorting robotic arm based on colour sensing. *Int. J. Adv. Res. Electr*, *3*(3), pp.7741-7746.

[9] Nicolaus, K., Hooper, J., Wood, R. and Ham, C., 2016, October. Development of an autonomous ball-picking robot. In 2016 International Conference on Collaboration Technologies and Systems (CTS) (pp. 373-378). IEEE.

[10] Jia, Y., Yang, G. and Saniie, J., 2017, May. Real-time color-based sorting robotic arm system. In 2017 IEEE International Conference on Electro Information Technology (EIT) (pp. 354-358). IEEE.

[11] Shah, R. and Pandey, A.B., 2018. Concept for automated sorting robotic arm. *Procedia Manufacturing*, *20*, pp.400-405.

[12] Sampreeth, A., Harsha, A. K., Rao, A., Desai, K. V., Rohith K., Praveena K. S., 2018. Object Sorting Robot Using Image Processing, International Journal of Engineering Research & Technology, 6(13), pp.1-4.

[13] Kale, V.R. and Kulkarni, V.A., 2013. Object sorting system using robotic arm. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, *2*(7), pp.3400-3407.

[14] Amin, U., Ahmad, G., Liaqat, N., Ahmed, M. and Zahoor, S., 2014. Detection & Distinction of Colors using Color Sorting Robotic Arm in a Pick & Place Mechanism. *International Journal of Science and Research*, *3*(6), pp.1164-1168.

[15] Kunhimohammed, C.K., Saifudeen, K.M., Sahna, S., Gokul, M.S. and Abdulla, S.U., 2015. Automatic color sorting machine using TCS230 color sensor and PIC microcontroller, *International Journal of Research and Innovations in Science and Technology*, *2*(2), pp.33-38.

[16] Bhaskar, P. S., Dofe, S. S., Tripahi, S. S. and Banode, H. R., 2016. Colour Sorting System with Robotic Arm, *International Journal for Innovative Research in Science & Technology*, *2*(10), pp.267-272.

[17] Deshmukh, A., Nagane, M. and Awatade, V., 2016. Design & Development of Automatic Conveyor Controlling System for Sorting of Component on Color basis. *Red*, *630*, p.780.

[18] Shirgave, S., Salunkhe, A., Shirgave, K. and Upadhye, S.Y., 2017. Color sorting robot. *International Journal of Advanced Research in Computer and Communication Engineering*, *6*(3), pp.2278-1021.

[19] Suryawanshi, S., Sonone, S., Patil, P. and Parbhane, P., 2017. Color Sorting Robotic Arm. *International Research Journal of Engineering and Technology*, *4*(10), pp.1011-1013.

[20] Patel, H., Joy, R., Macwan, S. and Modi, H., 2018. IOT color based object sorting machine. *International Journal of Applied Engineering Research*, *13*(10), pp.7383-7387.

[21] Jakkan, D.A., Sudhakar, C.B., Vilas, J.S. and Aslam, K.U., 2019. Color based product sorting machine using IoT. *Journal of Embedded Systems and Processing*, *4*(3), pp.25-29.

[22] Patil, P. B., Patil, S. S. and Harugade, M. L., 2019. Review on Colored Object Sorting System Using Arduino UNO. *International Research Journal of Engineering and Technology*, *6*(05), pp.6181-6184.

[23] Sireesha, T., Raghuram, C. H., Surya, K. V., and Venkat, T., 2021. Automated Sorting System Based on Color and Thickness of Waste Management. *International Research Journal of Engineering and Technology*, *8*(2), pp. 229-232,.

[24] Ghosh, A., Mukherjee, D., Das, D., Chakraborty, M., Chandra, S., Mitra, S., Saha, N., and Saha, I., 2019. Colour based Product Organization Tool using Arduino, *International Journal for Research in Applied Science & Engineering Technology*, *7*(5), pp.1895-1898.

[25] Kamboj, D.S. and Diwan, A., 2019. Development of Automatic Sorting Conveyor Belt Using PLC. *International Journal of Mechanical Engineering and Technology*, *10*(8).

[26] Shen, L.J. and Hassan, I., 2015. Design and development of colour sorting robot. *Journal of Engineering Science and Technology*, *10*, pp.71-81.

[27] Zulfiqar, R., Mehdi, B., Ifthikar, R., Khan, T., Zia, R. and Saud, N., 2019, January. PLC based automated object sorting system. In 2019 4th International Electrical Engineering Conference (IEEC 2019) (pp. 1-7).

[28] Nuva, T.J., Ahmed, M.I. and Mahmud, S.S., 2022. Design & Fabrication of Automatic Color & Weight-Based Sorting System on Conveyor Belt. *Journal of Integrated and Advanced Engineering*, *2*(2), pp.147-157.