Object Sorting System

Submitted in partial fulfillment of the requirements

Of the degree of

BACHELOR OF ENGINEERING

In

COMPUTER ENGINEERING

By

Group No: 42

Roll No.	Name
162023	Pavandeep Kaur Chadha
162046	Sakshi Gupta
162071	Rahul Khatija

Supervisor:

DR. ARCHANA B. PATANKAR

(Professor, Department of Computer Engineering, TSEC)

Computer Engineering Department

Thadomal Shahani Engineering College

University of Mumbai

2018-2019

CERTIFICATE

This is to certify that the project entitled **"Object Sorting System"** is a bonafide work of

Roll No.	Name
162023	Pavandeep Kaur Chadha
162046	Sakshi Gupta
162071	Rahul Khatija

Submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"BACHELOR OF ENGINEERING"** in **"COMPUTER ENGINEERING"**.

Dr. Archana B. Patankar
Supervisor/Guide

Dr. Tanuja Sarode
Head of Department

Dr.G.T.Thampi
Principal

Project Report Approval for B.E

Project report entitled "**Object Sorting System"** by

Roll No.	Name
162023	Pavandeep Kaur Chadha
162046	Sakshi Gupta
162071	Rahul Khatija

is approved for the degree of ***"BACHELOR OF ENGINEERING" in "COMPUTER ENGINEERING"***

Examiners

1.----------

2.----------

Date:

Place:

Declaration

We declare that this written submission represents my ideas in my own words and where other's ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will because for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

1) _____

(Signature)

Pavandeep Kaur Chadha -162023

2) _____

(Signature)

Sakshi Gupta - 162046

3) _____

(Signature)

Rahul Khatija - 162071

Date:

ABSTRACT

Image processing in today's world grabs massive attentions as it leads to possibilities of broaden application in many fields of high technology. The real challenge is how to improve existing sorting system in the modular processing system which consists of four integrated stations of identification, processing, selection and sorting with a new image processing feature.

This project presents a color and shape sorting system solution with the application of image processing. Image processing procedure senses the objects in an image captured in real-time by a camera and then identifies color and information out of it. It is a project based on Image processing and robotics used to identify and sort objects based on their color and shape. This system provides a simplified and efficient method to sort objects by capturing an image of the object to be analyzed in Python 3.7.1 software and use efficient algorithms to identify the color and shape of the object. A camera is used to obtain an image of the target object which is used for analysis. A robotic arm, which is controlled by an Arduino Uno microcontroller, is used for the 'pick up and drop' operation to sort the object and place it in the desired position.

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Automation has become a very important aspect in our everyday life. A process automation or automation system (PAS) is used to automatically control a process such as chemical, oil refineries, paper and other factories. Today's world is the world of technology as well as science. Due to automation, life has become fast and luxurious. As the technology is growing most people are adopting the new technologies rather than using the old. The progress in technology is making people more demanding towards the things they use and consume, this is the reason why everything is automated.

Robotic automation is implemented in various industries. It is very important for industrial environments because it reduces the error capacity of the system and also saves time as compared to human beings. Intelligent machines are very important in today's world. There is a great need for robotic systems which can work without human intervention. The sorting of objects using robotic arm is extensively used in the industries. It helps to reduce need of labors, increases

production and stipends of industry. In the complicated and unbearable sorting and the work which can't be done by human hands it is the most productive method.

Similarly, the influence and impacts of digital image on modern society are also huge. Image processing has become such a crucial component in modern science and technology that many tasks would not be attempted without it. It is used in computer vision, robotics, medical imaging, microscopy and many other fields. The field of robotics is blooming with a fast speed in recent years and many progressive technologies are coming with their own advancement. Robots due to its easy manner of operation are used in domestic, industrial and military purposes. [1]

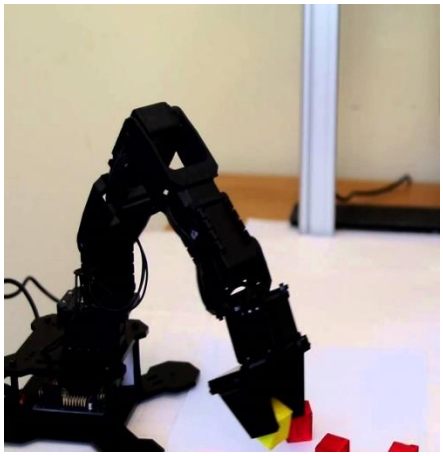

Figure 1.1 Automated sorting system

1.2 Aim & Objectives

The primary objectives of this project are:

1. To provide mechanization to the process using robotic arm.
2. To obtain the image of the target object.
3. To design simplified and efficient algorithms for the analysis of the various properties of the object.
4. To sort objects based on their properties of shape and color.

The robotic arm used in this project work is used to sort the object depending upon the corresponding color and shape of the object identified by analysis of the image captured by the camera, of the object.

This robotic arm is controlled by the Arduino Uno which programs the stepper motors used in the robotic arm. The software devilment is a vital task in the proposed project development. The

software is entirely coded in Python 3.7.1 to identify the color and shape of the object and place the object in its corresponding bin. Thus, by using an automated system, the time required for the sorting process is reduced to great extent due to which the proposed system is fast, accurate, economical, robust and cost efficient.

1.3 Scope

In today's era, small scale and large-scale industries faces common hindrance like shortage of time and workers which leads to inefficient manufacture. A proper result for the above problem can be achieved using robotics. Furthermore, for meticulous result, image processing methods can be beneficial.

Image processing systems built around image recognition and analysis are already an essential component in automated production. Throughout all steps of production, from the inspection of raw materials and production monitoring (i.e. flaw detection) to final inspections, quality assurance and sorting, they are an indispensable part of achieving high efficiency and quality standards.

Image processing can play a decisive role here in determining specific information. One important point is that cameras continue to grow smaller and more affordable, even as performance improves. Where complex systems were once required, today's small, efficient systems can produce the same results or better. This technological progress, taken together with the possibilities of ever-expanding networking, opens up the potential for new applications in the industry.

The scope of this project is limited to the identification and sorting of an object based on its properties of color and shape. The image of the object will be obtained using a camera which will be analyzed using Python 3.7.1 algorithms and the object will be sorted using a robotic arm which will be controlled, using an Arduino Uno microcontroller. The range of colors and shapes that can be identified and sorted by the system can be varied depending on the current needs of the particular industry in which the system is being operated in. [2]

Chapter 2

Review of Literature

2.1 Domain Explanation

Image Processing

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually Image Processing system includes treating images as two-dimensional signals while applying already set signal processing methods to them.

It is among rapidly growing technologies today, with its applications in various aspects of a business. Image Processing forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps:

1. Importing the image with optical scanner or by digital photography.
2. Analyzing and manipulating the image which includes data compression and image enhancement and spotting patterns that are not to human eyes like satellite photographs.
3. Output is the last stage in which result can be altered image or report that is based on image analysis.

Purpose of Image Processing

The purpose of image processing is divided into 5 groups. They are:

1. Visualization – Observe the objects that are not visible.
2. Image sharpening and restoration – To create a better image.
3. Image retrieval - Seek for the image of interest.
4. Measurement of pattern – Measures various objects in an image.
5. Image Recognition – Distinguish the objects in an image.

Types

The two types of methods used for Image Processing are Analog and Digital Image Processing.

1. Analog or visual techniques of image processing can be used for the hard copies like print outs and photographs.
2. Digital Processing techniques help in manipulation of the digital images by using computers. [5]

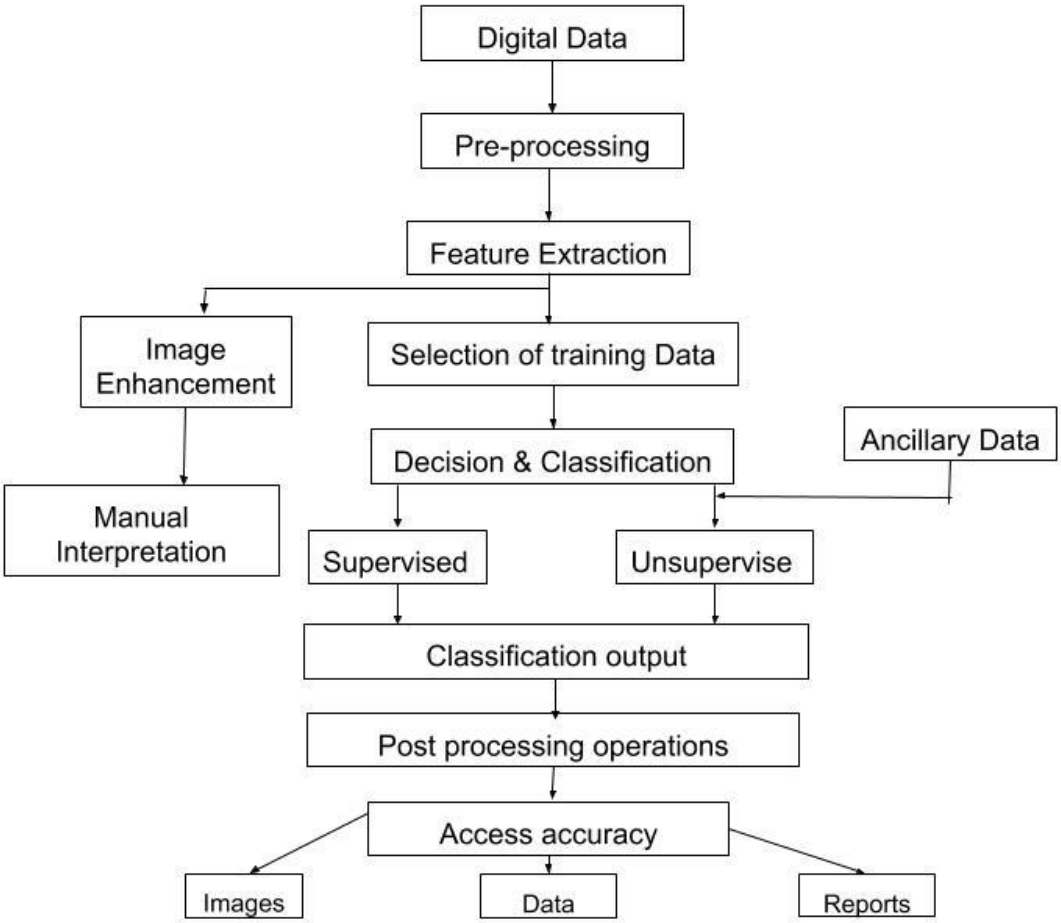

Figure 2.1 Steps in Image Processing

File Formats

1. **TIFF (also known as TIF)**

TIFF stands for Tagged Image File Format. TIFF images create very large file sizes. TIFF images are uncompressed and thus contain a lot of detailed image data (which is why the files are so big) TIFFs are also extremely flexible in terms of color (they can be grayscale, or CMYK for print, or RGB for web) and content (layers, image tags).

TIFF is the most common file type used in photo software (such as Photoshop), as well as page layout software (such as Quark and InDesign), again because a TIFF contains a lot of image data.

2. **JPEG (also known as JPG)**

JPEG stands for Joint Photographic Experts Group, which created this standard for this type of image formatting. JPEG files are images that have been compressed to store a lot of information in a small-size file. Most digital cameras store photos in JPEG format, because then you can take more photos on one camera card than you can with other formats. A JPEG is compressed in a way that loses some of the image detail during the compression in order to make the file small (and thus called “lossy” compression).JPEG files are usually used for photographs on the web, because they create a small file that is easily loaded on a web page and also looks good.

3. **GIF**

GIF stands for Graphic Interchange Format. This format compresses images but, as different from JPEG, the compression is lossless (no detail is lost in the compression, but the file can’t be made as small as a JPEG). GIFs also have an extremely limited color range suitable for the web but not for printing. This format is never used for photography, because of the limited number of colors. GIFs can also be used for animations.

4. **PNG**

PNG stands for Portable Network Graphics. It was created as an open format to replace GIF, because the patent for GIF was owned by one company and nobody else wanted to pay licensing fees. It also allows for a full range of color and better compression. It’s used almost exclusively for web images, never for print images. For photographs, PNG is not as good as JPEG, because it creates a larger file. But for images with some text, or line art, it’s better, because the images look less “bitmappy.”

5. **Raw image files**

Raw image files contain data from a digital camera (usually). The files are called raw because they haven’t been processed and therefore can’t be edited or printed yet. There are a lot of different raw formats–each camera company often has its own proprietary format. Raw files usually contain a

vast amount of data that is uncompressed. Because of this, the size of a raw file is extremely large. Usually they are converted to TIFF before editing and color-correcting. [6]

File Format	Advantages	Disadvantages
JPEG (.jpg)	Permits setting compression level; small image size	Low image quality at higher degrees of compression
GIF (.gif)	Good internet browser support; Good for displaying animations	Larger file sizes; limited color support (256 colors or 256 shades of gray)
PNG (.png)	Good internet browser support; Good image quality	Limited acceptance with earlier web browsers
TIFF (.tif)	High image quality	Large file size

Figure 2.2 Types of file formats

Tools Used in Image Processing

MATLAB

Figure 2.3 Logo of MATLAB

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to *learn* and *apply* specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others. [4]

OPENCV-Python

Figure 2.4 Logo of OpenCV

OpenCV was started at Intel in 1999 by **Gary Bradsky** and the first release came out in 2000. Currently OpenCV supports a wide variety of programming languages like C++, Python, Java etc. and is available on different platforms including Windows, Linux, OS X, Android, iOS etc. OpenCV-Python is the Python API of OpenCV. It combines the best qualities of OpenCV C++ API and Python language.

Python is a general-purpose programming language started by **Guido van Rossum**, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability.

Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation.

And the support of Numpy makes the task easier. NumPy is a highly optimized library for numerical operations. It gives a MATLAB-style syntax. All the OpenCV array structures are converted to-and-from NumPy arrays. So whatever operations you can do in NumPy, you can combine it with OpenCV. Besides that, several other libraries like SciPy, Matplotlib which supports NumPy can be used with this.

Applications

1. Remote Sensing
2. Moving object tracking
3. Biomedical Imaging techniques
4. Intelligent Transportation Systems

2.2 Existing Solution

The goal is to construct a productive, microcontroller-based project that get right sorting of items and put it down at opportune spot to upgrade the profitability, minimizing the expense of the items and diminishing human oversights.

In the existing system, a TCS34725 color sensor is interfaced with the Raspberry Pi A+ board on its 2nd I2C interface channel, running on Linux OS. The values are read using Python script by calling the Python I2C libraries. The color sensor requires a light source, white SMD LED to illuminate the sample. The circuit is realized using a set of comparators – 8x LM324 for each axis, and 2x CMOS Decade counter CD4017 to generate scanning in XY axis. The circuitry is controlled by Microcontroller (AT89S52) and is interfaced by its UART to the Raspberry Pi A+ processor board. The communication is handled in high level by the PySerial libraries. Based upon the detection, the robotic arm moves to the specified location, releases the object and comes back to the original position. [3]

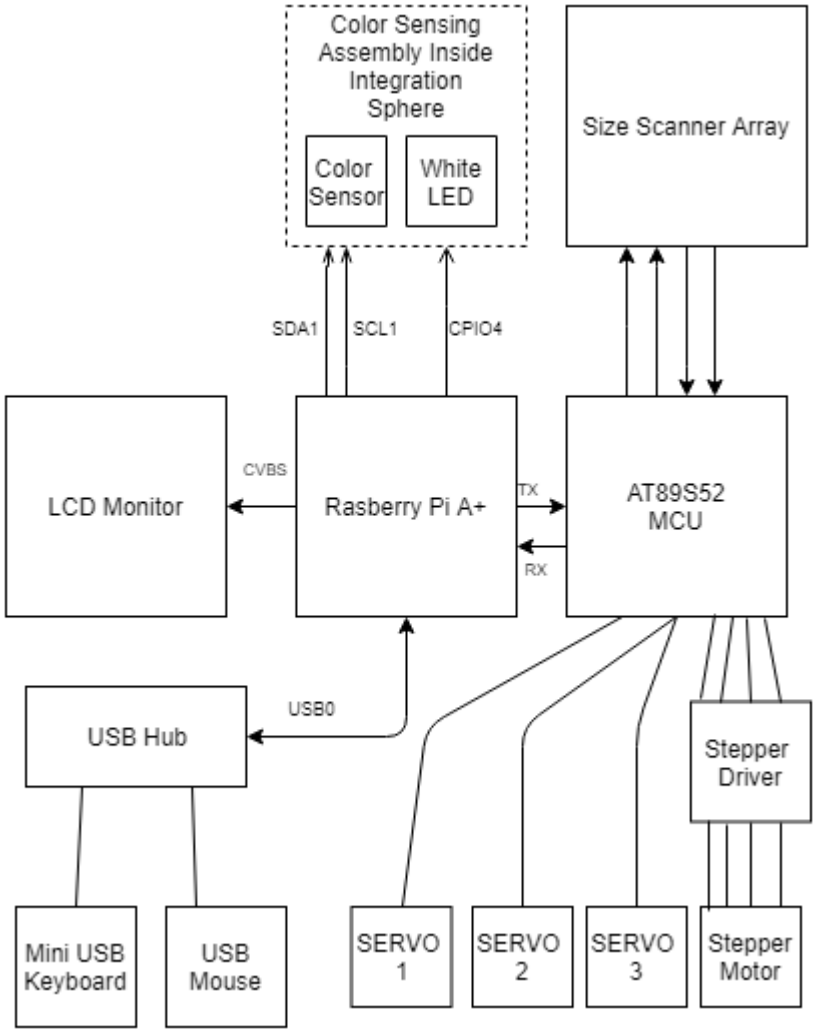

Figure 2.5 Block diagram of existing system

Limitations of the existing system:

- It requires that the object should be opaque.
- It sorts objects only on the basis of a single parameter i.e. color.
- The system is operated using Raspberry Pi A+, TCS34725 color sensor and Arm Processor 1176 which are expensive.
- It needs the object under consideration to be externally illuminated.

2.3 Hardware and Software Requirements

2.3.1 Hardware Requirements

1. Camera
2. Arduino Uno
3. Servo Motor
4. Switched-Mode Power Supply

2.3.2 Software Requirements

1. Python 3.7.1

Chapter 3

Analysis

3.1 Functional Requirements

- **Robotic Arm Mechanism**

  The system provides pick up and drop operation to aid in the automation of the system and consequently decrease the human involvement in the industrial process.

- **Identifying properties of the object**

  The microcontroller of the system analyses and identifies the shape and color of the object using the input received from the camera.

- **Sorting Mechanism**

  Python 3.7.1 is used for processing the captured image and based on the results the object is placed in the corresponding bin using the robotic arm.

3.2 Non-Functional Requirements

- **Scalability**

  The system will be scalable with an increase in the diversity of the objects.

- **Usability**

  The system will be easy to operate.

- **Portability**

  The relatively small size of the system makes it portable.

3.3 Proposed System

In this project, we propose a robotic sorting system which can sort objects based on their shape and color, a robotic arm is used to pick and place the object in their predefined place based on their shape and color parameters. The system can be used in small scale and large-scale industries to sort the products based on the various parameters. For example, to sort the bottles or candies of various sizes such as medicine and candy industry. Three servo motors are used in the construction of robotic arm. Two servo motor is used to move robotic arm in up and down direction and one servo motor is used for the gripper to pick a particular color object. One stepper motor is used for moving robotic arm in a circular direction for placing the picked object in a particular color box. The camera is used to sense particular color for picking and placing process. Image sensing using Python 3.7.1 is used in a system for detecting the color of the object. Arduino is the central processing unit, used to control all the functions of other blocks in this robot system. The microcontroller takes or read data from color from Python 3.7.1 software and controls all the other functions of the system by manipulating these data. Microcontroller control the motor on the robotic arm to pick a particular color object, as per the signal from image processing Arduino can understand the color of the object, it controls the arm motor to move towards the specified location,

again control the gripper motor to release the object into that particular location. Stepper motor is used for angular movement of the robotic arm. Stepper motor is a machine to convert pulse to angle displacement.

This system can differentiate between a wide range of colors and also between objects with and without edges which are square and circular objects. After processing, the output is displayed on a screen.

Chapter 4

Design

4.1. Design Consideration

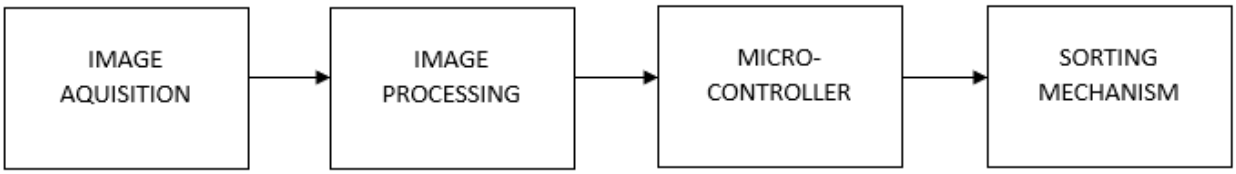

Figure 4.1 Flow of the System

The proposed system works in following three steps:

i) Image Acquisition

ii) Image Processing

iii) Sorting Mechanism

Image Acquisition

The image of the object is captured by the camera and is sent to the Python 3.7.1 workspace. The input image from the camera cannot be given for processing directly. Pre-processing is done on the image such as thresholding. Then object image is converted in binary format and smoothened to remove noise. This final threshold image of object is now ready for processing.

Image Processing

The objects are sorted on the basis of color and shape. For shape recognition captured image is converted to gray from RGB. Then thresholding is done followed by inverting the image. Boundaries Concentrate are found and lastly shape is identified using shape properties.

To identify the color, firstly the image is converted into gray format and then thresholding is done. After thresholding color components are extracted and the image is converted into black and white format which is called as binary format. Remove all those pixels less than 300px. Label all the connected components in the image. Find region properties & bounding box and the color is identified.

Sorting Mechanism

The sorting mechanism consist of a robotic arm. After identifying the shape and color of the object, command will be sent to direct the motor of a robotic arm through com port of the computer. According to the shape and color the robotic arm places the objects.

4.2. Design Details

System Operation

Object sorting is done using two ways: Color based sorting and shape-based sorting.

Color Based Sorting

After image acquisition thresholding is done. Then red, green and blue color pixels are calculated. The image is converted into binary. Remove all those pixels less than 300px. Label all the connected components in the image. Find region properties & bounding box and identify red or green or blue object.

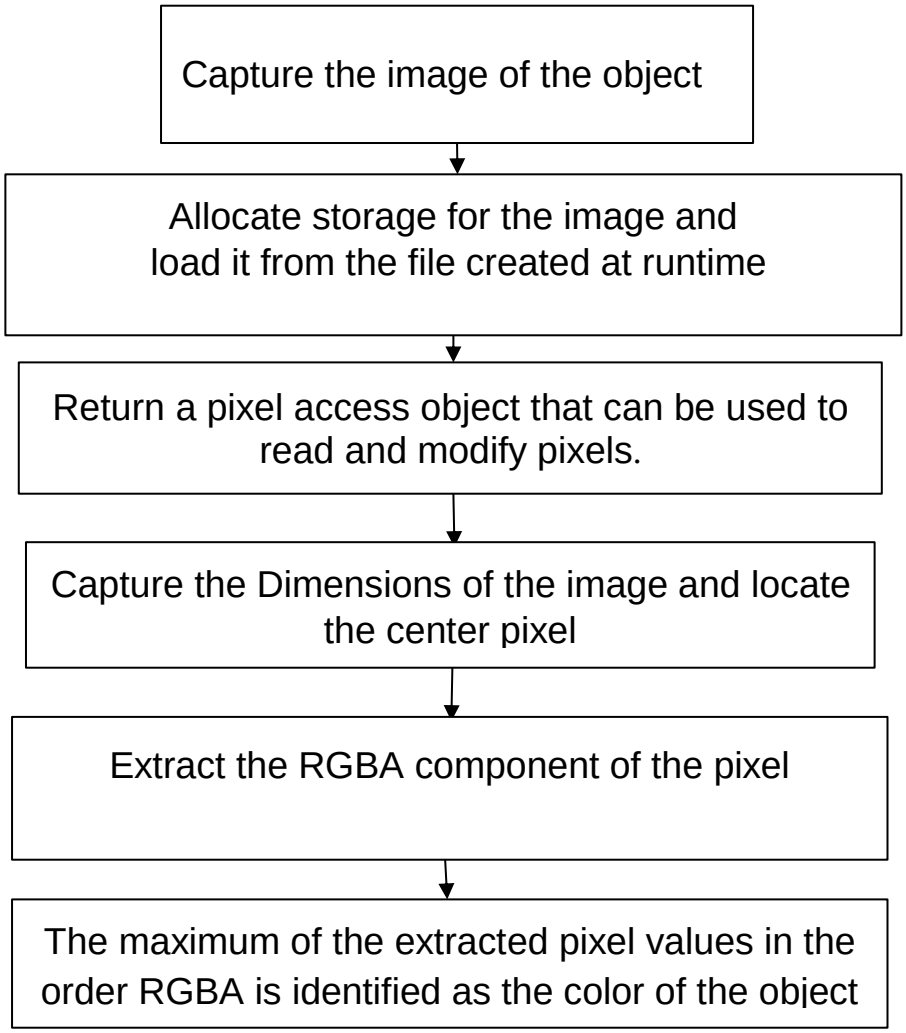

Figure 4.2 Color-based sorting Flowchart

Shape Based Sorting

Acquire the image. Convert that image from rgb to gray. Threshold the image. Convert it to the Binary Image and then invert it. Find the boundaries concentrate. Determine Shapes properties. Classify Shapes according to properties.

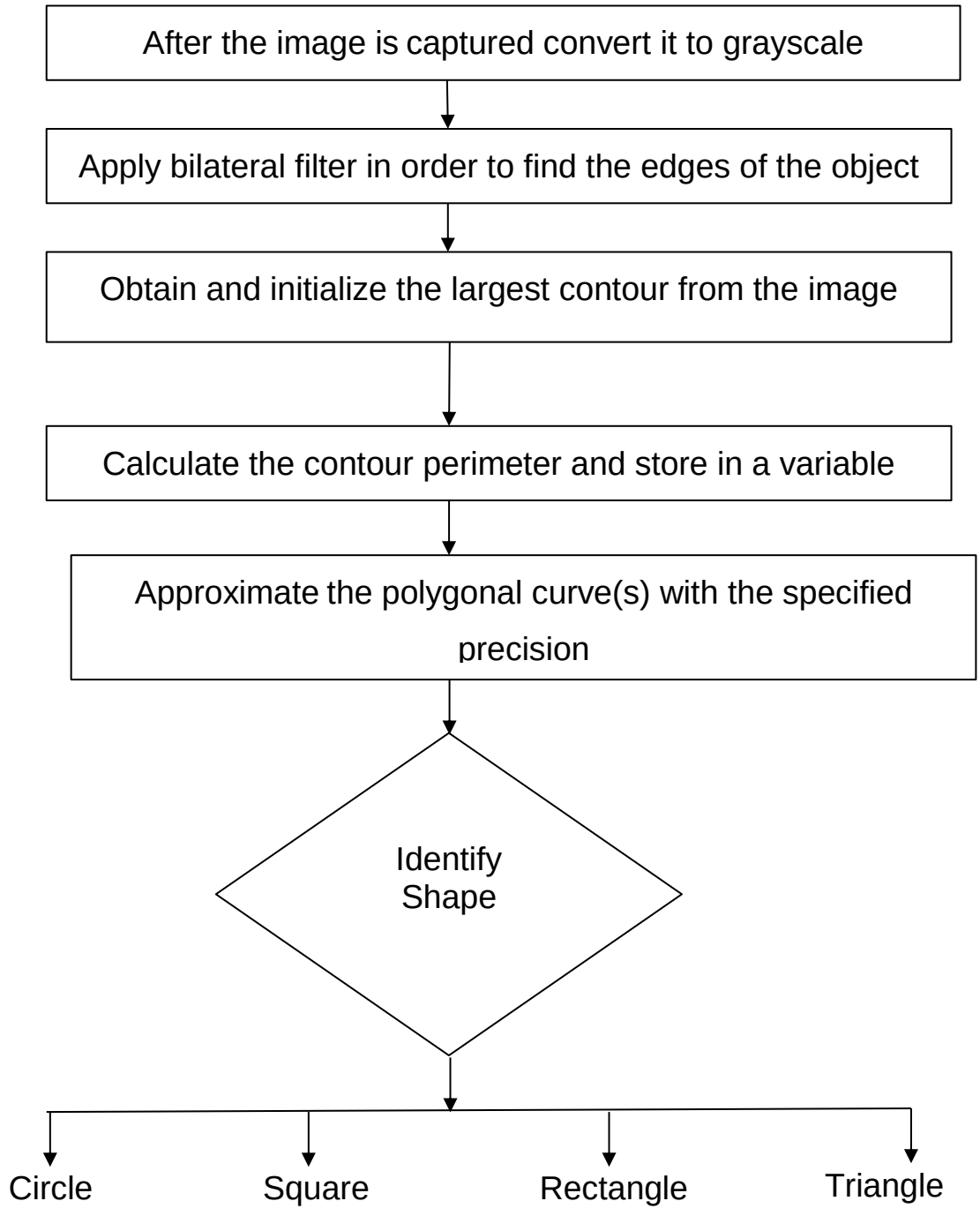

Figure 4.3 Shape-based sorting Flowchart

Flow of the System

1. Initialize Microcontroller, camera, motors.
2. The camera captures the image of the object.
3. After the image is captured, image processing is performed in Python 3.7.1.
4. It gives the information about its shape & color.
5. After this information is sent to the roboarm using PySerial, the roboarm grips that object and places it in the corresponding bin.

Arduino Uno

Arduino Uno is an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board. Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs.

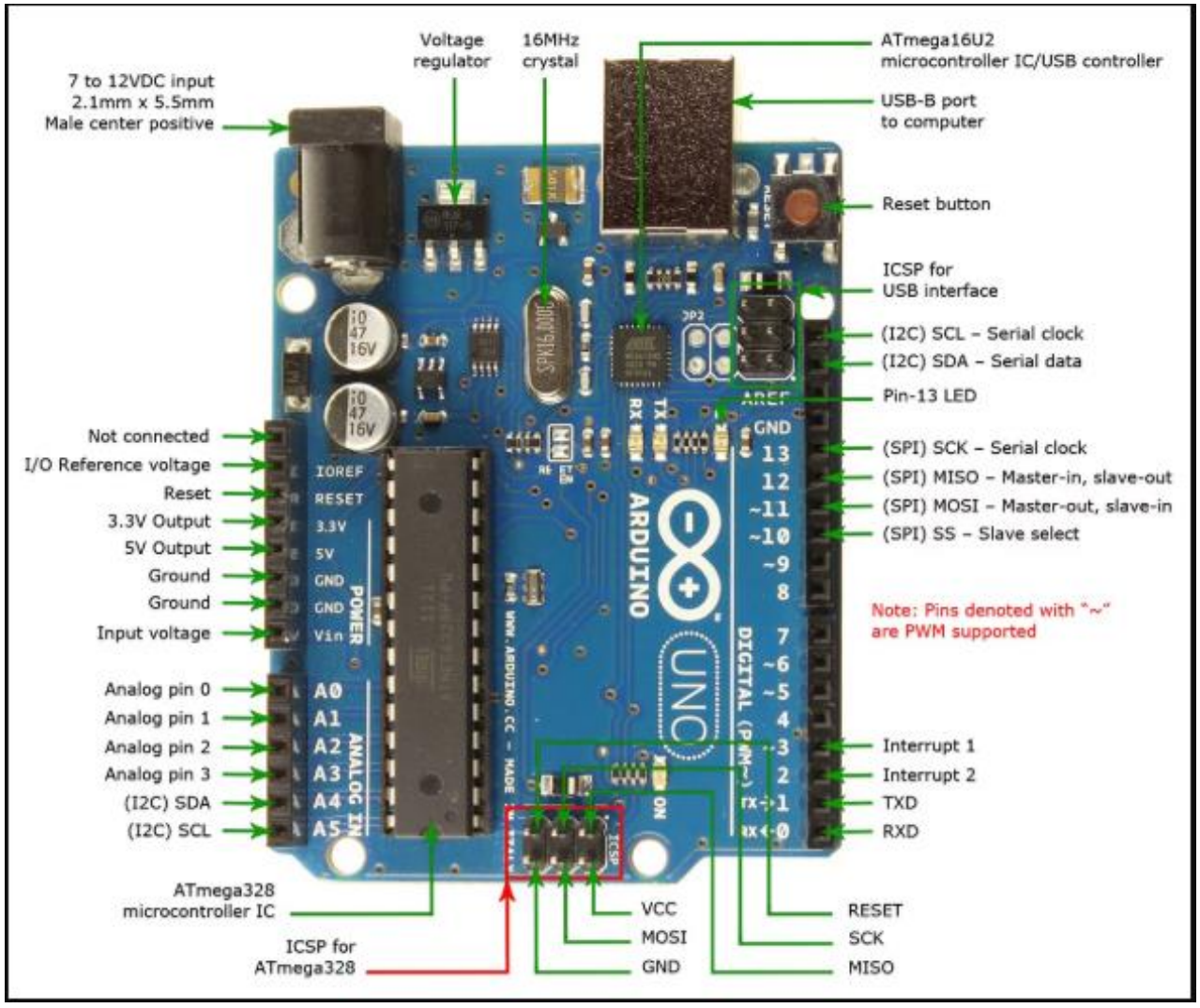

Figure 4.4 Arduino Uno

Chapter 5

Implementation

5.1 Implementation Details

Step I Object Placement

An object of specific shape and color is placed on a predefined spot.

Step II Image Capture

After Step 1, real time image of the object is taken with the help of camera. This image should be good quality and which is send to PC with Python 3.7.1 software.

Step III Image Processing

The image captured by the camera is then transferred to the PC in which different image processing algorithms are applied on it.

Important terms related to image processing:

Pixel: Pixel is the building blocks of an image. In other words, a pixel is the smallest possible image that can be detected on your screen.

RGB Image: An image is composed of the three primary colors, Red, Green and Blue. Hence is called as RGB image.

RGB value: All colors which we see around us can be made by adding red, blue and green components in varying proportions. Hence, any color of the world can uniquely be described by its RGB value which stands for Red, Blue and Green values.

Binary Image: An image that consists of mainly black and white pixels.

Gray scale Image: It contains intensity values ranging from a minimum (absolute black) to a maximum (absolute white) and in between varying shades of gray. Typically, this range is between 0 and 255.

Firstly, the image, that is captured by the camera is send to the PC via microcontroller, is an RGB image. This RGB values are converted into gray scale values. Then this gray scale image is converted into binary image using thresholding.

Thresholding is the simplest method of image segmentation. From a grayscale image, thresholding can be used to create binary images. Thresholding replace each pixel in an image with a black pixel if the image intensity 'im' is less than some fixed constant 'th' (that is, im<th) or a white pixel if the image intensity is greater than that constant.

This binary image is then inverted i.e. the black pixels are converted into white pixels and the white pixels are converted into black pixels for simplification. Then find the region properties, bounding box and shape properties to identify the shape and color of the object.

Step IV Instructions to Robotic Arm

It is basically 3 axes' robotic arm. This is used to send pick and drop the object to its respective bin based on its color and shape. This is achieved due to the instructions given to the motors present in the robotic arm by the PC via microcontroller using PySerial.

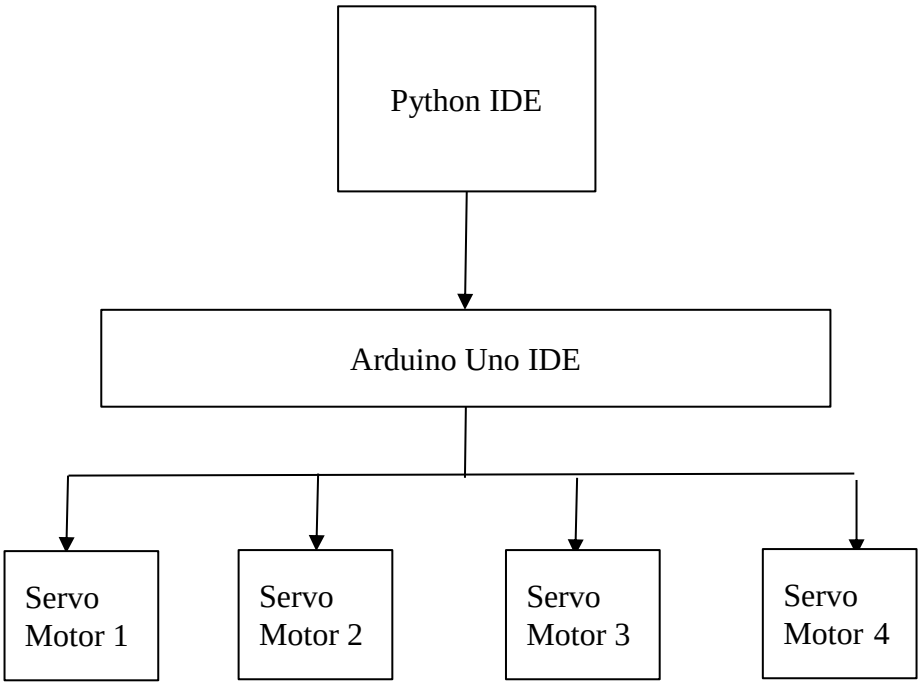

Figure 5.1 Working of the System

Algorithm for Color Detection

1. Start
2. Capture the image of the object from the camera
3. Allocate storage for the image and load it from the file created at runtime
4. Return a pixel access object that can be used to read and modify pixels. The access object behaves like a 2-dimensional array
5. Capture the Dimensions of the image and locate the center pixel
6. Using the access object extract the RGBA (Red, Green, Blue and Alpha values) component of the pixel
7. The maximum of the extracted pixel values in the order RGBA is identified as the color of the object
8. Stop

Algorithm for Shape Detection

1. Start
2. After the image is captured convert it to grayscale
3. Blur the image in order to find the edges of the object
4. Find contours in the edged image, keep only the largest ones, and initialize the screen contour
5. Loop over the contours and approximate them
6. Calculate the contour perimeter and store in a variable peri
7. Approximate the polygonal curve(s) with the specified precision: 0.04 * peri. Store it in a variable called approx
8. Based on the length of approx, the shapes are categorized into 4 sided (Square/Rectangle) or 3 sided (Triangle)
9. To distinguish between Square and Rectangle calculate the up-right bounding rectangle of a point set , i.e. approx
10. Let (x,y) be the top-left coordinate of the rectangle and (w,h) be its width and height. Find the ratio of width to height. If the ratio is greater than 0.95 it is classified as a Square, else if the ratio is less than 1.05 the object is Rectangle. If any of the criteria's do not match, the default shape is given as Circle

Python to Arduino communication

On the Computer side of things, we used a Python module called PySerial. PySerial is a library which provides support for serial connections ("RS-232") with Python over a variety of different devices: old-style serial ports, Bluetooth dongles, infra-red ports, and so on. It also supports remote serial ports via RFC 2217 (since V2.5).

Features:

1. Same class based interface on all supported platforms.
2. Access to the port settings through Python properties.
3. Support for different byte sizes, stop bits, parity and flow control with RTS/CTS and/or Xon/Xoff.
4. Working with or without receive timeout.

5.2 Result and Evaluation

The object sorting system based on Arduino Uno microcontroller is successfully implemented. It classifies the object placed at a predefined position based on its color and shape. The system limits itself to 4 shapes: circle, square, rectangle and triangle and 3 colors: Red, Blue and green. Based on the input object, the image is captured and then processed to determine its shape and color using Python. The resultant data is sent to the Arduino Uno Microcontroller on its serial port using PySerial. Based on the data received by the microcontroller, it commands and controls the drivers of the robotic arm to pick, rotate to a particular angle and drop the object in its respective bin.
For example,

Object	**Degree of Rotation**	
Red Rectangle	40’	
Blue Square	80’	
Green Triangle	120’	
Red Circle	160’	etc.

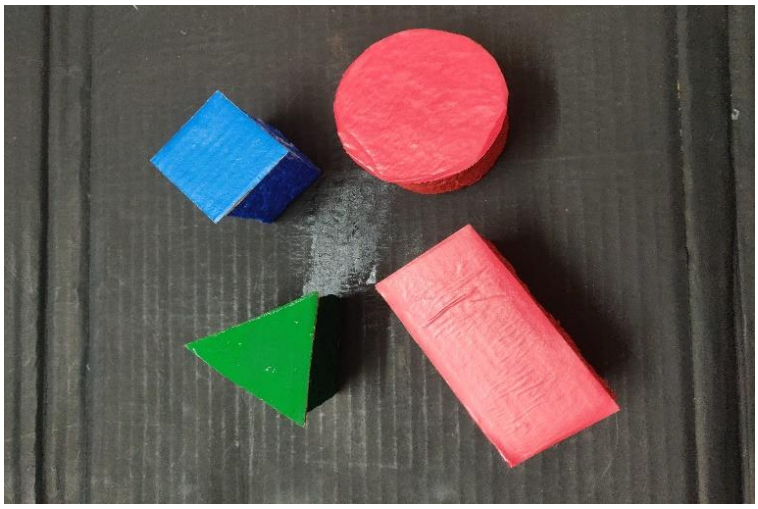

Figure 5.2 Objects to be sorted by the system

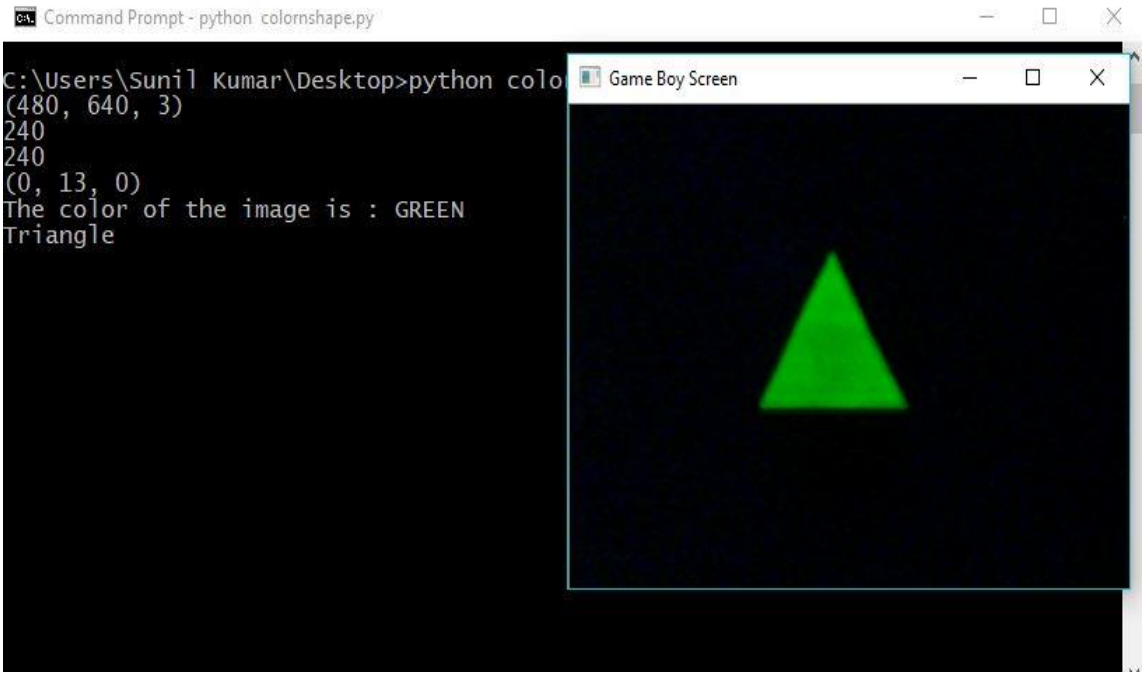

Figure 5.3 Result after processing image using Python

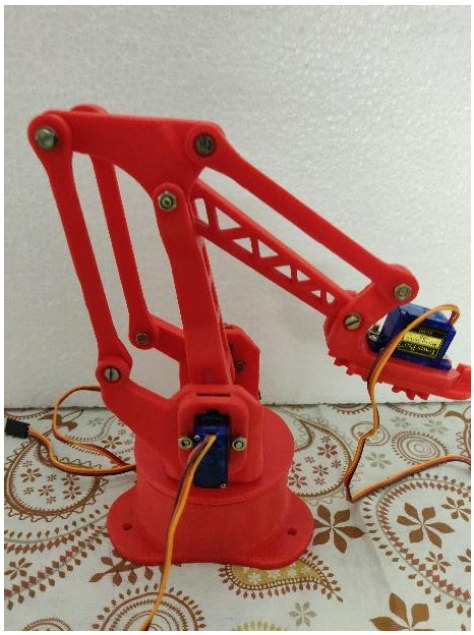

Figure 3.4 Robotic arm

Chapter 6

Conclusion and Future Work

We developed a color and shape-based object sorting system using image processing. It is a user-friendly model which uses robotic arm mechanism for sorting and a camera for taking images of the various objects. The project deals with an object handling system and will aim in classifying the objects by color and size and placing the objects in its respective pre-programmed place. Thereby eliminating the monotonous work done by humans, achieving accuracy and speed in the work.

Over the period of nine months, we studied and learnt the basic of necessary, efficient hardware and software and used this knowledge to implement the system. The system is a sample version, so for a large scale manufacture the number of robotic arms and cameras can be attributed. Moreover, modified model of robotic arm can be used to pick large and heavy objects and sort them effectively. The clarity and capturing capability of the imaging device can be improved. A more expensive and stronger motor with better torque capacity can be used to pick-up more heavier objects. The size of the arm can be increased for use in real life applications. The system is restricted to basic shapes and only Red, Green and Blue objects and can be enhanced to work for a wide variety of objects.

References

1. Rafael C. Gonzalez and Richard E. Woods, “Digital Image Processing”, Third Edition, Vol.3, Pearson Publication.
2. Snehal Shirgave, Aishwrya Salunkhe, Khadija Shirgave, S. Y. Upadhye “Color Sorting Robot” International Journal of Advanced Research in Computer and Communication Engineering ISO 3297:2007 Certified Vol. 6, Issue 3, March 2017
3. Tuong Phuoc Tho, Nguyen Truong Thinh “Design and development of the sorting system based on robot” Published in: 2015 15th International Conference on Control, Automation and Systems (ICCAS)
4. www.mathworks.com
5. https://www.tutorialspoint.com/dip/
6. http://imageprocessingplace.com/root_files_V3/tutorials.htm

Acknowledgement

We would like to express our sincere thanks and gratitude to **Dr. G.T. Thampi**, the principal of Thadomal Shahani Engineering College, Bandra (W) for providing the facilities and infrastructure for the project.

Also, we would like to express our gratitude to Dr. Tanuja Sarode, the Head of our Department for giving us an opportunity to work on this project.

We would like to further extend our thanks to Dr. Archana Patankar, our project supervisor, guide, for her constant guidance at every step of implementation.

1) Pavandeep Kaur Chadha - 162023

2) Sakshi Gupta - 162046

3) Rahul Khatija - 162071