Object Sorting Robotic Arm Based on Image Processing

Submitted in partial fulfillment of the requirements
of the degree of

Bachelor of Engineering

in

Electronics & Telecommunication

By

Haju Saima Dilawar (16DET84)
Ansari Usama Raees Ahmed, (16DET77)
Borkar Mohd Taha A.Aziz (16DET80)

Under the guidance of

Banda Nawaz Kotiyal

Department of Electronics and Telecommunication Engineering

Anjuman-I-Islam's Kalsekar Technical Campus
Sector 16, New Panvel, Navi Mumbai
University of Mumbai

2018-19

CERTIFICATE

Department of Electronics and Telecommunication Engineering
Anjuman-I-Islam's Kalsekar Technical Campus
Sector 16, New Panvel , Navi Mumbai
University of Mumbai

This is to certify that the project entitled **"Object Sorting Robotic Arm Based On Image Processing"** is a bonafide work of **Miss. Haju Saima Dilawar (16DET84), Ansari Usama Raees Ahmed (16DET77), Borkar Mohd Taha A.Aziz (16DET80)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of Bachelor of Engineering in Department of Electronics and Telecommunication Engineering.

Supervisor   Examiner

Head of Department   Director

Project Report Approval for Bachelor of Engineering

This project entitled **"Object Sorting Robotic Arm Based On Image Processing"** by **Miss. Haju Saima Dilawar (16DET84), Ansari Usama Raees Ahmed (16DET77), Borkar Mohd Taha A.Aziz (16DET80)** is approved for the degree of **Bachelor of Engineering Electronics and Telecommunication** .

Examiner

....

Supervisor

....

Date;

Place:

Declaration

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause 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.

………………
Saima Haju
(16DET84)

……………….
Ansari Usama
(16DET77)

…………………….
Borkar Mohd Taha
(16DET80)

Date:

Acknowledgments

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

We are highly indebted to **Mr. Banda Nawaz Kotiyal** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express my gratitude towards my parents & Staff of Anjuman-I-Islam's Kalsekar Technical Campus for their kind co-operation and encouragement which help me in completion of this project.

We My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

Haju Saima Dilawar…………………(16DET84)

Ansari Usama Raees Ahmed………..(16DET77)

Borkar Mohd Taha A.Aziz………….(16DET80)

ABSTRACT

Our project is about to create a Robotic Arm with Real-Time Image Processing using Raspberry Pi which can either be automated or can be operated manually. In the present era, we are making a robot capable of detecting and placing, the pre-specified object, at the Pre-specified location. The code for detection of colour of an object has been written in Python. For hardware implementation, we have used Raspberry Pi 3B, which has Raspbian OS based on Debian which is Linux OS.

The robotic arm detects the pre-specified objects and segregates them based on colour (RGB). The program includes modules for controlling the robotic arm, capturing the object image and identifying the colour of the object. Web control have been used to control motor remotely and perform all task automatically using Raspberry Pi.

ACKNOWLEDGEMENT

We are extremely fortunate to be involved in an exciting and challenging project like **“Object Sorting Based Robotic Arm Based on Image Processing”** It has enriched our life, giving us an opportunity to work in field of programming and automation. This project increased our thinking and understanding capability and after the completion of this project, we experience the feeling of achievement and satisfaction.

We should like to express our greatest gratitude and respect to our guide **Prof. Banda Nawaz Kotiyal** For his excellent guidance, valuable suggestion and endless support. He has not only been wonderful guide but also a genuine person. We consider ourselves extremely lucky to be able to work under guidance of such a good personality. Also we would like to thanks our **HOD Prof. SHAIK AFZAL** for guiding us to improve our project. Actually, he is one such genuine person for whom our words will not be enough to express.

It was impossible for us to complete our project without their help. We are also grateful to our Director **Mr. ABDUL RAZZAK HONNUTAGI** for their encouragement. We like to express our grateful thanks to our classmates, all staffs and faculty members of electronics and telecommunication engineering department who willingly rendered us their unselfish help and support Last but not least, we want to convey our heartiest thanks to our parents for their immeasurable love, support and encouragement.

INDEX

LIST OF FIGURES

CHAPTER NO: 01

INTRODUCTION

CHAPTER NO: 01

INTRODUCTION

1.1: Project Overview

In today's scenario, the robot with high accuracy, high output, and no error is in demand, the precise work or repetitive work is better done with robots, for the robot the sensor or camera is common sense for the machine-like image processing to detect and identify an object and its characteristic which helps to perform a required task.

A Robotic arm can decide the object based on colour like Red, Green, and Blue using camera and image processing in raspberry pi.

The main aim of the project is to make a robot that has the capability of pick a pre-specified object and placing it in separate divisions based on colour. Raspberry Pi has found its way in many useful and changeable implementations in robotic systems. Raspberry Pi does not implement any usual motor control peripherals and it is available at low cost. The python code has been formulated for creating a robotic arm with image processing and local web page with slider to adjust servo motor position manually.

For the industrial purpose, the robotic arm can help to separate and segregate the object based on colour with a good frequency.

CHAPTER NO: 02

HARDWARE SPECIFICATION

CHAPTER NO: 02

HARDWARE SPECIFICATION

2.1: Rasberry pi 3B:

Fig: 01

2.1.1: Processors

The Broadcom BCM2835 SoC used in the first generation Raspberry Pi includes a 700 MHz ARM11 76JZF-S processor, Video Core IV graphics processing unit(GPU), and RAM. It has a level 1 (L1) cache of 16 KB and a level 2 (L2) cache of 128 KB. The level 2 cache is used primarily by the GPU. The SoC is stacked underneath the RAM chip, so only its edge is visible. The 1176JZ (F)-S is the same CPU used in the original iPhone, although at a higher clock rate, and mated with a much faster GPU.

The earlier V1.1 model of the Raspberry Pi 2 used a Broadcom BCM2836 SoC with a 900 MHz 32-bit, quad-core ARM Cortex-A7 processor, with 256 KB shared L2 cache. The Raspberry Pi 2 V1.2 was upgraded to a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, the same SoC which is used on the Raspberry Pi 3, but under clocked (by default) to the same 900 MHz CPU clock speed as the V1.1. The BCM2836 SoC is no longer in production as of late 2016.

The Raspberry Pi 3+ uses a Broadcom BCM2837B0 SoC with a 1.4 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache.

2.1.2: RAM

On the older beta Model B boards, 128 MB was allocated by default to the GPU, leaving 128 MB for the CPU. On the first 256 MB release Model B (and Model A), three different splits were possible. The default split was 192 MB (RAM for CPU), which should be sufficient for standalone 1080p video decoding, or for simple 3D, but probably not for both together. 224 MB was for Linux only, with only a 1080p frame buffer, and was likely to fail for any video or 3D. 128 MB was for heavy 3D, possibly also with video decoding (e.g. XBMC). Comparatively the Nokia 701 uses 128 MB for the Broadcom Video Core IV.

For the later Model B with 512 MB RAM, new standard memory split files (arm256_start.elf, arm384_start.elf, arm496_start.elf) were initially released for 256 MB, 384 MB and 496 MB CPU RAM (and 256 MB, 128 MB and 16 MB video RAM) respectively. But a week or so later the RPF released a new version of starts. if that could read a new entry in config.txt (gp mem=*xx*) and could dynamically assign an amount of RAM (from 16 to 256 MB in 8 MB steps) to the GPU, so the older method of memory splits became obsolete, and a single start. If worked the same for 256 MB and 512 MB Raspberry Pi.

The Raspberry Pi 2 and the Raspberry Pi 3 have 1 GB of RAM. The Raspberry Pi Zero and Zero W have 512 MB of RAM.

2.1.3: Special-purpose features

The Pi Zero can be used as a USB device or "USB gadget", plugged into another computer via a USB port on another machine. It can be configured in multiple ways, for example to show up as a serial device or an ethernet device. Although originally requiring software patches, this was added into the mainline Raspbian distribution in May 2016.

The Pi 3 can boot from USB, such as from a flash drive. Because of firmware limitations in other models, the Pi 3 is the only board that can do this.

2.1.4: Performance

While operating at 700 MHz by default, the first generation Raspberry Pi provided a real-world performance roughly equivalent to 0.041 GFLOPS. On the CPU level the performance is similar to a 300 MHz Pentium II of 1997–99. The GPU provides 1 Gpixel/s or 1.5 Gtexel/s of graphics processing or 24 GFLOPS of general purpose computing performance. The graphical capabilities of the Raspberry Pi are roughly equivalent to the performance of the Xbox of 2001.

Raspberry Pi 2 V1.1 included a quad-core Cortex-A7 CPU running at 900 MHz and 1 GB RAM. It was described as 4–6 times more powerful than its predecessor. The GPU was identical to the original. In parallelised benchmarks, the Raspberry Pi 2 V1.1 could be up to 14 times faster than a Raspberry Pi 1 Model B+.

The Raspberry Pi 3, with a quad-core ARM Cortex-A53 processor, is described as having ten times the performance of a Raspberry Pi 1. This was suggested to be highly dependent upon task threading and instruction set use. Benchmarks showed the Raspberry Pi 3 to be approximately 80% faster than the Raspberry Pi 2 in parallelised tasks.

2.1.5: Supply

The raspberry pi is low power controller which uses a micro USB connection to power itself. The recommended amount of current is between 700mA for a Raspberry Pi Model A, and up to 2.5A for a Raspberry Pi 3 Model B+.

2.2: USB webcam

In this project, we are using the USB webcam to monitoring the real-time object image on the Raspberry Pi. It’s able to deliver a clear 3-megapixel resolution image or 720p HD video recording at 30 frames/sec. The quality and frames/sec of the image helps to enhance and process the image to detect the colour of the object with maximum accuracy.

Fig: 02

2.3: MG996R High Torque Metal Gear Dual Ball Bearing Servos

This High-Torque MG996R Digital Servo features metal gearing resulting in extra high 10kg stalling torque in a tiny package. The MG996R is essentially an upgraded version of the famous MG995 servo, and features upgraded shock-proofing and a redesigned PCB and IC control system that make it much more accurate than its predecessor. The gearing and motor have also been upgraded to improve dead band- with and centering. The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cir- rus, Blue Bird, Blue Arrow, Corona, Berg, Spectrum and Hitec. This high-torque standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback gear box, especially since it will fit in small places. The MG996R Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast.

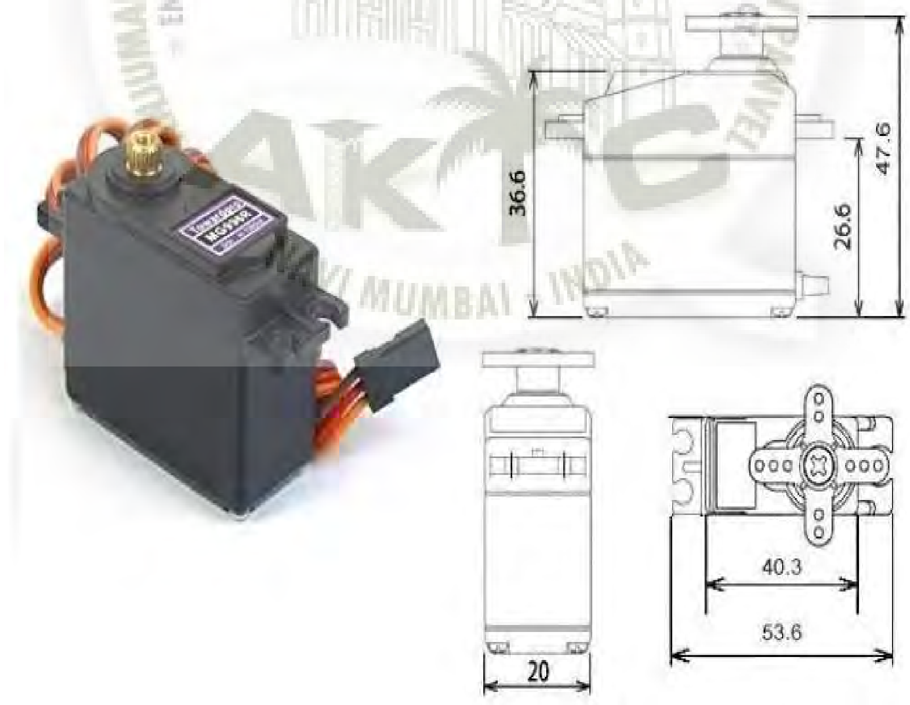

Fig: 03

2.3.1: Specifications

Weight: 55 g

Dimension: 40.7 x 19.7 x 42.9 mm approx.

Stall torque: 9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)

Operating speed: 0.17 s/60^{o} (4.8 V), 0.14 s/60^{o} (6 V)

Operating voltage: 4.8 V a 7.2 V

Running Current 500 mA –

Stall Current 2.5 A (6V)

Dead band width: 5 s

Stable and shock proof double ball bearing design

Temperature range: 0 ^{o}C – 900 mA (6V) 55 ^{o}C

2.4: Ultrasonic Ranging Module HC - SR04

2.4.1: Product features

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

(1) Using IO trigger for at least 10us high level signal.

(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.

(3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time velocity× of sound (340M/S) / 2.

2.4.2: Wire connecting direct as following

5V Supply.

Trigger Pulse Input.

Echo Pulse Output

0VGround.

2.4.3: Electric Parameter

Voltage DC 5 V Working

Current 15mA Working

Frequency 40Hz

Max Range 4m

Min Range 2cm Measuring Angle

15 degree

Trigger Input Signal 10uS TTL pulse

Echo Output Signal Input TTL lever signal and the range in

proportion Dimension 45*20*15mm

Fig: 04

2.5: DC Gear Motor

2.5.1: Operating Precautions:

DC motors are compact and display high output, and their speed is easy to control. They may be driven by battery or any other power supply and are therefore also easy to use. However, inappropriate power supply may lead to burnout or abnormal brush wear. Problems with power supply, installation, and general precautions and problems with a motor installed in-circuit will be described.

Applied voltage: Be sure to use a motor at its rated voltage (+IUVI), and avoid any surge voltage. We can specially manufacture motors designed with an electrical path protecting the motor from surges and reversed polarity. Please contact us for details.

Applying non-rated supply voltages: Applying a voltage higher than the mo- tor's rating results in a temperature increase, leading to heat damage or lowered service life. Scoring of the commutator surface by sparks and mechanical brush wear arising from vibration may also occur. Applying a voltage lower than the mo- tors rating may eventually result in the motor failing to start. This is due to the build up of carbon powder on the commutator. Motors are manufactured for use within +IOVo of their rated specifications.

Brush wear promoted by power supply ripples: Brush wear may be mechan- ical wear due to brush and commutator abrasion or electrical wear due to sparking between the brush and commutator, the latter being the most common. Brush wear is therefore greatly affected by ripples in the power supply voltage, and use of general regulated DC is recommended. However, when rectifying AC for use by a motor, be sure to use full-wave rectification with a capacitor or similar element in a smoothing circuit.

Ambient conditions: The service life of a DC motor is dependant upon its rectifying action. Care must be taken to ensure good commutation, as dust, oil, gas, water, etc. Water, etc, on the commutator surface results in poor rectification and increases brush wear.

Changing the brush position: The brushes are generally fixed in position such that rotational speed and current characteristics are maintained equivalent in both clockwise and counter-clockwise directions. These are basically determined based on the position of the magnetic poles. Rotating the motor after not carefully relo- cating parts such as the brush holder (for fixing the brushes) or rear cover results in misalignment of the brushes and magnets. This will produce change in the above characteristics in the rotational direction or cause poor rectification, leading normal brush wear. Therefore, changing of the brush positioning is to be avoided.

Installed orientation: Motors are generally designed for use with a horizontal output shaft. Special consideration must be given to components including bear- ings and grease washers when intended for an upward- or downward-facing output shaft. Please contact us for details. Further, avoid installing a motor in a manner in which grease from the gear head would tend to enter the motor (e.g., with an upward-facing output shaft).

Noise generation: Electrical noise is generated as a result of sparks from commutation between the brushes and commutator. Please contact us for assistance with lowering noise.

Gear heads for intermittent drive: The gearhead is assembled with a fixed shaft about which a gear revolves and transmits power. It is not suited to continuous drive. You should maintain the duty ratio between ON and OFF states at no more than 50with the maximum ON state not exceeding 5 seconds.

Motor and gear head combination: When combining a gear head with a pinion shaft, gently fit the gear head on turning it right and left, being careful that the pinion and the gear in the gear head do not strongly clash with each other. Using force will cause noise-producing scratches in the pinion and the gear. Scratches are Failures by a decreased service life and are the cause of unforeseen accidents.

Load variation: Even with torque below the rated load, a motor will more damage than might be imagined if there is frequent load variation. Exercise caution with operating conditions and load restrictions.

Insulation resistance: The insulation resistance of a brush motor will naturally continue to decrease as its running time increases. The figures for resistance given in the catalog are for a new motor.

Service life : Service life depends greatly on operating conditions and environment.

Fig: 05

CHAPTER NO: 03

Making and Assembling

CHAPTER NO: 03

Making and Assembling

3.1: Robotic Arm

A mechanical structure with electric motors and end effectors. Look like the human hand to grip the object and place it in the predefined position. The robotic arm attempt to reproduce movement similar to a human arm. It has the base to rotate the arm from 0 degrees to 180 degrees. Like the human hand, arm consists shoulder, elbow, wrist and mechanical gripper to hold the object. The arm can be made up of plastics-based 3D printing, wood, scrap materials, acrylic sheet, Aluminum and more. The arm using in our project has a low-cost arm that has a good quality acrylic sheet. A term that's used for a mechanical system which is the number of in- dependent parameters that define its configuration called DOF (degrees of freedom). when a robot arm is designed the DOF is considered and it is related to roll, yaw, and pitch. The arm has 7 DOF to perform pick and place operation with maximum accuracy.

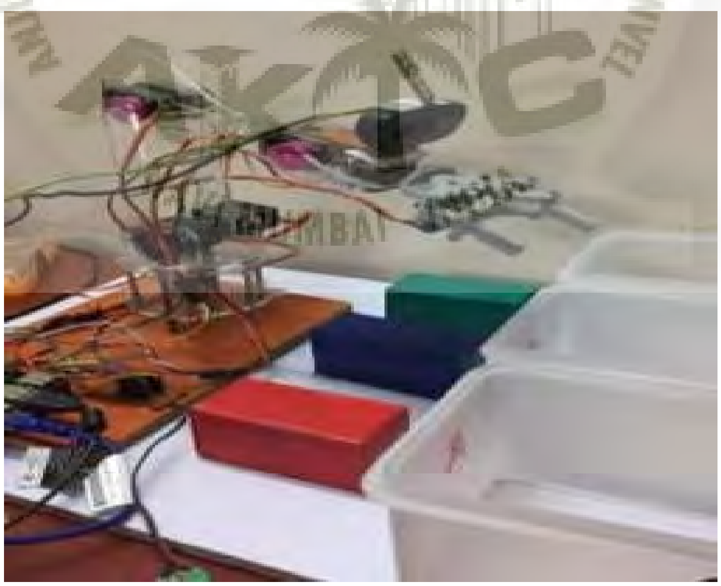

Fig: 06

Maximum lift: 200g.

Dimensions: 9” L* 8” W*15” H Weight: 550g

CHAPTER NO: 04

SOFTWARE SPECIFICATIONS

CHAPTER NO: 04

SOFTWARE SPECIFICATIONS

4.1: SOFTWARE

The system software is the Debian Raspbian OS that is stored on the Micro SD card. The algorithm is programmed with Python, after importing the OpenCV li- brary for image processing. OpenCV outdoes the use of MATLAB in our application as OpenCV is an open source real-time image processing unlike MATLAB that is li- censed and has constraints for use in Raspberry Pi. Also in terms of speed, OpenCV is found to process faster as it is a library of functions that is written in C, close to the machine language unlike MATLAB that is built on Java. While OpenCV supports various programming languages like Python or C++, Python is viewed as a relatively better option. This is mainly because of its simplicity and code read- ability. Python can also be easily extended with C/C++. With this it is possible to write computationally intensive code in C/C++ and create Python wrappers that can be used as Python modules. This is advantageous because the code is as fast as the original C/C++ code and it is easier to code in Python than C/C++. In this way, the OpenCV-Python which is a Python wrapper for the original OpenCV C++ implementation is more convenient to use.

4.2: Anaconda IDE

Anaconda is a free and open source distribution of the Python and programming languages for data science and machine learning related applications (large-scale data processing, predictive analytics, scientific computing), that aims to simplify package management and deployment. Package versions are managed by the package management system anaconda. The Anaconda distribution is used by over 6 million users, and it includes more than 250 popular data science packages suitable for Windows, Linux, and Mac OS.

4.3: Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda pack- ages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, in- stall them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

Navigator is automatically included with Anaconda version 4.0.0 or higher.

The following applications are available by default in Navigator :

JupyterLab
Jupyter Notebook
QtConsole
Spyder

4.4: Spyder

Spyder is an open source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, pandas, IPython, SymPy and Cython, as well as other open source software.It is released under the MIT license.

Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python develop- ers and the community.

Spyder is extensible with first- and third-party plugins,includes support for interactive tools for data inspection and embeds Python-specific code quality assurance and introspection instruments, such as Pyflakes, Pylint and Rope. It is available cross-platform through Anaconda, on Windows with WinPython and Python (x,y), on macOS through MacPorts, and on major Linux distributions such as Arch Linux, Debian, Fedora, Gentoo Linux, openSUSE and Ubuntu.

Spyder uses Qt for its GUI, and is designed to use either of the PyQt or PySide Python bindings. QtPy, a thin abstraction layer developed by the Spyder project and later adopted by multiple other packages, provides the flexibility to use either backend.

4.5: Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whites- pace. It provides constructs that enable clear programming on both small and large scales. In July 2018, Van Rossum stepped down as the leader in the language com- munity after 30 years.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the refer-ence implementation of Python, is open source software and has a community-based development model, as do nearly all of Python's other implementations. Python and CPython are managed by the non-profit Python Software Foundation.

4.6: Open CV

Open CV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then It sees (which was later acquired by Intel). The library is cross-platform and free for use under the open-source BSD license.

Open CV supports the deep learning frameworks Tensor Flow, Torch/Py Torch and Caffe.

4.7: Colour detection

Once the object contours are extracted, the mean colour of each object in terms of the R, G, B components must be extracted. Out of the three, the component identified with the highest value shows the dominant colour of the object. Thus every object gets classified as one of these: red, green, blue, black or white coloured object.

However in our work, by converting the RGB image to a HSV scale, with the hue values, the object colour can also be identified as yellow, cyan and magenta, with further precision. Thus converting to the HSV scale gives more precise classification. The object colour can be now broadly classified into black, white, red, yellow, green, cyan, blue and magenta using the Hue values alone.

CHAPTER NO: 05

PROPOSED SYSTEM

CHAPTER NO: 05

PROPOSED SYSTEM

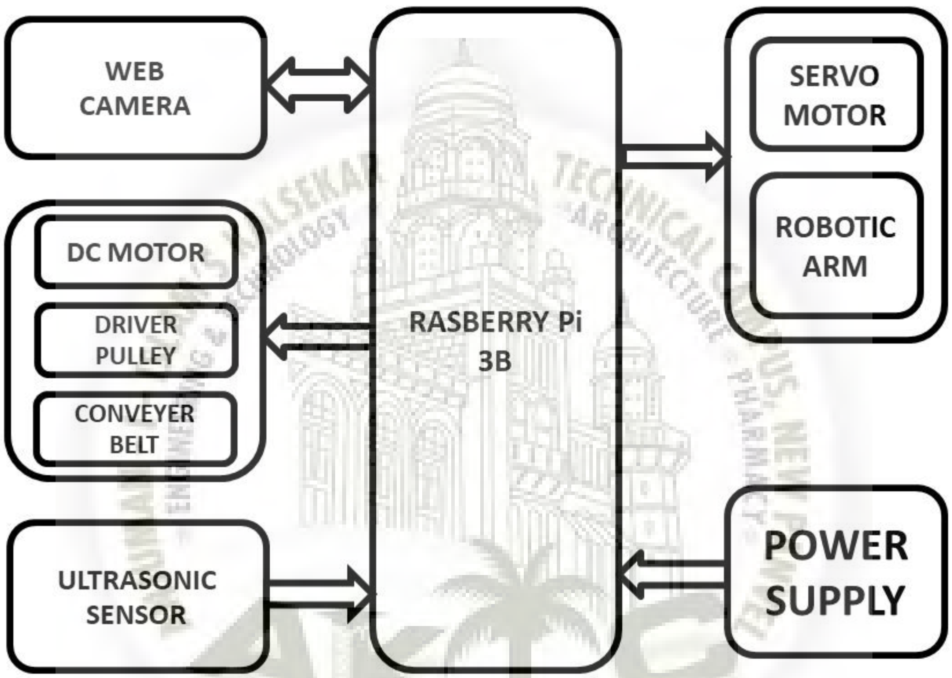

Fig: 07

5.1: Implementation

Raspbian OS based on Linux is used in raspberry pi for processing the hardware. To identification and detection of the object and its colour, the code is written in python. Simple CV libraries have allowed to enhance or process the image called image processing. Which required to identifying the colour of an object and control the robotic arm for pick and place pre-specified object operations. The code has local web page also so that using a slider to control the position of the robotic arm. The implementation process of the proposed work is done in three steps.

1. Image processing techniques, in order to identify the object colour(RGB).

2. Assembling and Controlling of Robotic Arm.

3. Integrate the vision system with a Robotic arm to control for pick and place pre-specified object.

5.2: Image processing techniques, in order to identify the object colour(RGB)

The value stored for each pixel in the image depends on the colour space and colour model being used. The colour model describes how colours are represented by a set of numbers, where each number corresponds to a different colour 'channel'. The colour space is the mapping of the channels of the colour model to absolute reference values. For example, the widely used colour spaces RGB is based on an RGB colour model, where each pixel has values in the Red, Green and Blue colour channels. Each value represents the intensity of that colour relative to the absolute reference colours defined by the RGB colour space. The aim is applying a Colour Key to an image, picking out a specific colour or range of colours in the image and removing them, or replace them with a separate background image.Integrate the vision system with a Robotic arm to control for pick and place the desired object:

The robotic arm can operate as automatic and manual to pick and place the object. The arm and camera are integrated into python code. A local server page has created for manual operation with slider to change the position of the servo motor for undefined instructions. This help for industrial purpose when some correction is required to change the fine position. Python code has instruction for three colours.

i.e. Red, Green, and Blue to identify the object colour and instruct the robotic arm to pick and place pre-specified object. The servo motors can produce the back electromagnetic force(emf) with a high spike of voltage which can damage the controller. In order to save it, either servo motor drivers are used or just diode to stop back emf, because one servo can produce back emf of 100μV and it can multiply by 7 number of servos, but this back emf can stop by using the diode.

CHAPTER NO: 06

FLOWCHART

CHAPTER NO: 06

FLOWCHART

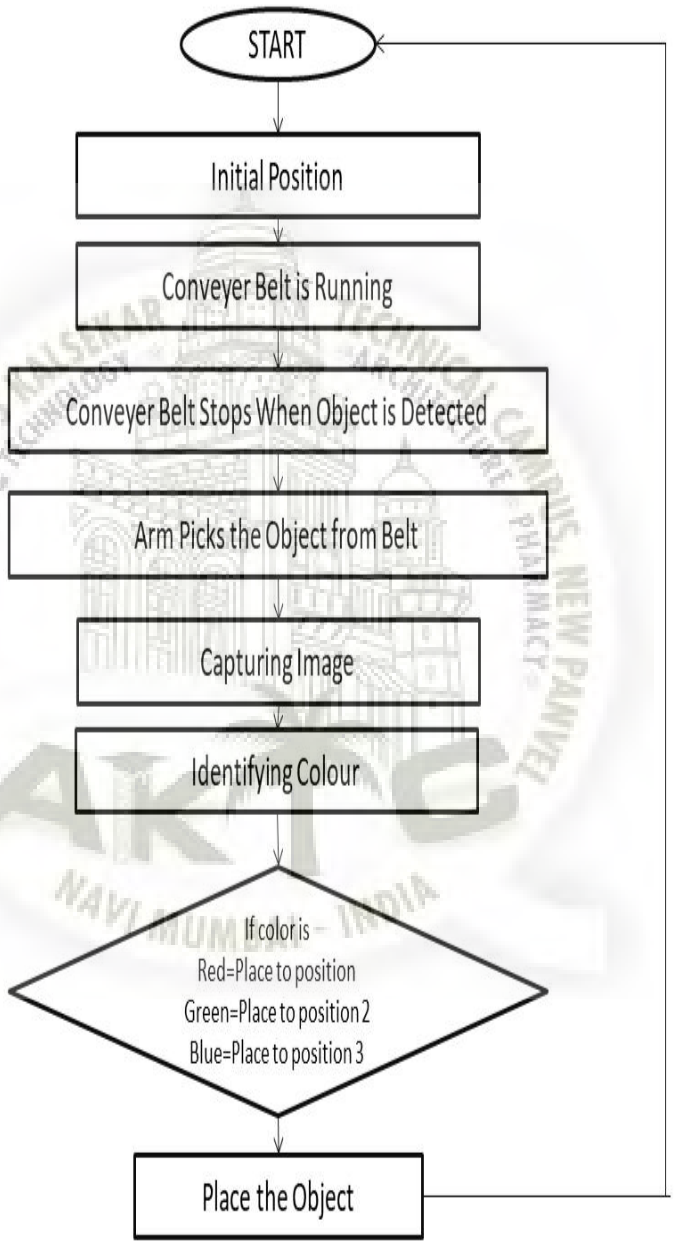

Fig: 08

CHAPTER NO: 07

WORKING

CHAPTER NO: 07

WORKING

1. The object should be placed in between the clamp in front of the camera which will be sorted depending upon the color of the object (red,blue and green can add more colors).

2. The robotic arm will place them at three different angles at 90,180 and 270 degrees. The images will be processed by raspberry pi using opencv and the robotic arm will be controlled by arduino board due to its simplicity.

3. Raspberry pi will detect the color by performing various operations on the images captured and will determine the color of the object.

4. After the color is detected ,rpi will send the information to arduino using 2 bits (10,01,11), 1 is high and 0 is low using gpio pins of raspberry pi

5. The two wires will be connected from gpio pins of rpi to two digital pins of arduino using simple jumper wires(No UART or other communication bus is used instead two simple wires which will either be high or low.

6. The robotic arm will perform operation depending upon the color. 7.Arduino will control three servo motors and motor control clamp.

CHAPTER NO: 08

CONCLUSION

CHAPTER NO: 08

CONCLUSION

This proposed solution gives better results when compared to the earlier existing systems such as efficient image capture, etc. Identification of colored object by the robotic arm can be controlled both automatically and manually for industrial purpose. The arm can segregate among three colors and place it accordingly. The hardware implementation was carried out using the raspberry pi.

-

CHAPTER NO: 09

References

CHAPTER NO: 09

References

1. ROBOTIC ARM WITH REAL-TIME IMAGE PROCESSING USING RASP BERRY PI, BOTHAUTOMATED AND MANUALLY. Dr. A. Brintha Therese1 and Prashant Gupta2.

2. Practical applications for robotic arms using image processing by Mihai Dragusu1, Anca Nicoleta Mihalache1 and Razvan Solea

3. Remote monitoring and control of robotic arm with visual feedabck. By K.Rajasekaran.

4. Computer Vision Based Robotic Hand By Prof. Priti Shende, Varun Sakpal, Rahul Pareek and Deep Bhatt.

5. Arm Controller Based Object Recognition by K.S.Sandhya, Pokuru Divya, Prajakta Saitwa