Research Article
Volume 11:3, 2022

Advances in Robotics & Automation

ISSN: 2168-9695

Open Access

Automated Inspection by Vision Assisted Robot Arm using LabVIEW and Raspberry Pi

Kota Saisunil* and Y Ashok Kumar Reddy

Department of Mechanical Engineering, NBKRIST, Vidyanagar, Andhra Pradesh, India

Abstract

The goal of the proposed work is to develop the Automated inspection system for sorting of circular and non-circular objects in the conveyor system. There is a robotic manipulator at the end of conveyor system to perform this sorting operation. The manipulator and conveyor's control software were developed by LabVIEW and Raspberry pi and these two are interfaced by using MQTT protocol. We also used OpenCV algorithm for object detection and tracking. In addition to that, a vision system was developed at the middle of the conveyor system and integrated with the manipulator system. This whole process is initiated, monitored and stopped using LabVIEW. The developed vision system allowed detection of object within the manipulator's work-space. It is observed that, the performance of the automated system is very high as compared to manual inspection system.

Keywords: Automated inspection system • LabVIEW • Raspberry pi • Mqtt • Opencv

Introduction

In today's manufacturing world quality has a top most priority. Automated inspection control has become very popular, which reduces human errors and maximizes quality of the product. Using Vision system with robotic arm is beneficial in many industrial applications, especially in automated object picking and placing tasks [1]. We are using OpenCV (Open Source Computer Vision Library) it is an open source computer vision and machine learning software library [2,3]. The library has more than 2500 optimized algorithms, which includes a comprehension set of both classical and state-of-art computer vision and machine learning algorithms. These algorithms can be used to identify object, track camera movements and track moving objects etc., the manipulators control software were developed using Raspberry pi and the real-time monitoring of the whole system is carried out by using LabVIEW. The Lab VIEW and Raspberry pi interfacing is done through MQTT (Message Que Telemetry Protocol). In this we are using Raspberry Pi, which is an open source Linux based board. Raspberry Pi has found its way in major in number of useful & versatile applications in robotic system's [4]. Finally in this paper, the automated inspection system was developed for automated Sorting operation of circular and Non-circular objects, the manipulators working errors are observed and discussed in results column.

Materials and Methods

System description

Automated inspection system: Automated inspection is defined as the automation of one or more steps involved in the inspection procedure. Automated or semi-automated inspection can be implemented in the number of alternative ways. Sorting of circular and non-circular objects using automated inspection system is main goal of this proposed paper (Table 1).

Raspberry Pi 3 Model B: Figure 1 shows the Raspberry Pi 3 Model B [5,6] which is the third generation Raspberry Pi. This powerful credit-card sized single board computer can be used for many applications and supersedes the original Raspberry Pi Model B+ and Raspberry Pi 2 Model B.

Whilst maintaining the popular board format the Raspberry Pi 3 Model B brings you a more powerful processor, 10x faster than the first generation Raspberry Pi.

Pi camera: Pi camera [7] is capable of capturing still images as well as high definition video. Stills are captured at a resolution of 2592 ×1944, while video is supported at 1080p at 30 FPS, 720p at 60 FPS and 640 × 480 at 60 or 90 FPS. The camera is supported in the latest version of Raspbian, Raspberry Pi's preferred operating system.

LabVIEW: LabVIEW offers a graphical programming approach that helps you visualize every aspect of your application, including hardware configuration, measurement data, and debugging. This visualization makes it simple to integrate measurement hardware from any vendor, represent complex logic on the diagram, develop data analysis algorithms, and design custom engineering user interfaces.

In this paper real-time monitoring is carried out by using LabVIEW. This whole process is initiated, monitored and stopped using LabVIEW.

Robotic manipulator: Figure 2 shows the Robotic Manipulator which

Table1. The list of components with specifications used for the system.

S. No	Name of the Component	Quantity	Characteristics
1)	Raspberry Pi 3 model – B board	1	1 GB memory
2)	PI camera board module	1	5 MP
3)	Power supply (SMPS)	1	12V, 5 Amps
4)	IR sensors	2	Range 10-15 cm
5)	DC gear motor (Conveyor)	1	12V, 10 rpm
6)	DC gear motor (Robot base)	1	12V,30 rpm
7)	DC motor (Robot arm and gripper)	2	12V,60 rpm
8)	L293D DC motor driver	2	16 pin, each can control 2 motors
9)	ADXL345	1	3-Axis accelerometer
10)	Conveyor	1	45 cm

***Address for Correspondence:** Kota Saisunil, Department of Mechanical Engineering, NBKRIST, Vidyanagar, Andhra Pradesh, India, Tel: +91 9849518448; E-mail: kotasaisunil@gmail.com*

Received 04 March, 2022, Manuscript No: ara-22-57829; **Editor assigned:** 06 March, 2022, PreQC No: P-57829; **Reviewed:** 18 March, 2022, QC No: Q-57829; **Revised:** 23 March, 2022, Manuscript No: R-57829; **Published:** 30 March, 2022, DOI: 10.4172/ara.2022.11.199

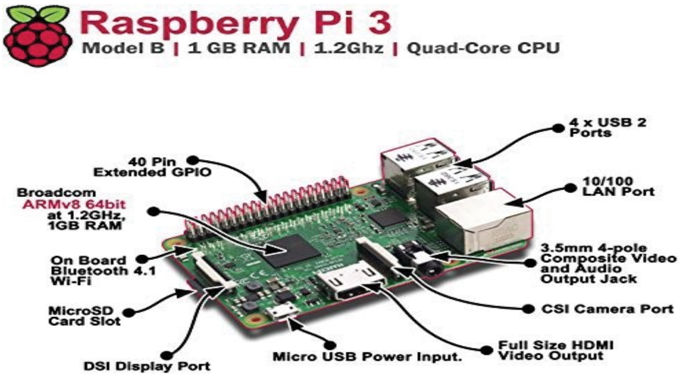


Figure 1. Raspberry Pi 3 Model B board.

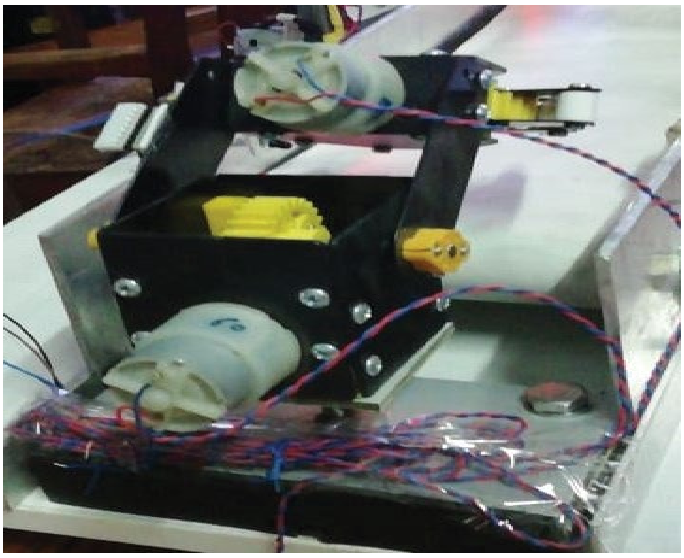


Figure 2. Robotic manipulator.

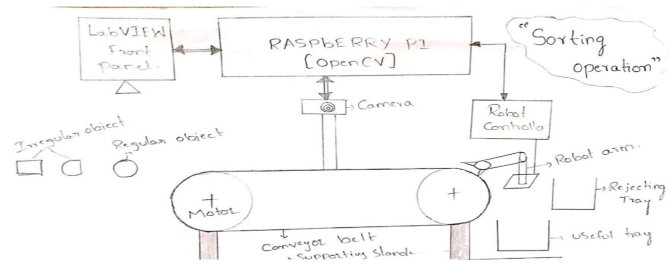


Figure 3. Block diagram.

is just like a human arm which is made of one DC Gear motor (12 v, 30 rpm) and two DC motors (12 v, 60 rpm), which will have different configuration of mechanical motion; most of the outer phase is made of plastic parts. The features of robot are functional gripper, base, base rotation, elbow, and wrist motion. These 3 motors operate the grab, release, rotate, lower and lift.

Maximum lift: 100 g, Weight: 687 g

Block diagram: Figure 3 shows the complete block diagram of how the whole system looks like. We will see the implementation and working of this inspection system below. The list of components with specifications used for this system is shown below.

Automated inspection implementation

Implementation of this Automated inspection system is carried out in two stages:

1. Hardware stage

2. Software stage

Hardware stage: In this hardware stage, Fabrication and working of this system is shown in detail:

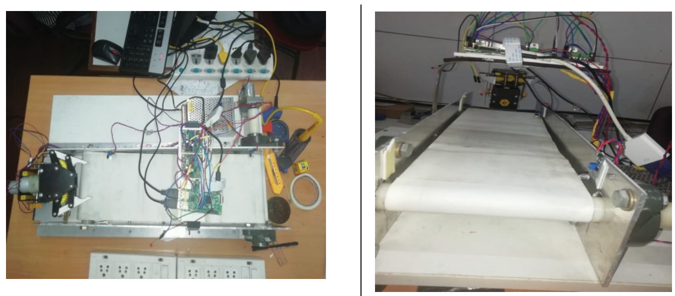
The components specifications are as shown above. The Working of this Automated Inspection system is as shown below:

- The power supply is given to the SMPS, Raspberry Pi Board and Desktop as well.
- For Initial programming USB Mouse, keyboard and LAN cable is connected to the Raspberry Pi board.
- After Running the Program. The object is placed in the starting position at IR sensor1, now the conveyor motor starts running.
- Assume it as a CIRCULAR object, the object travels in the conveyor. The Pi camera detects the Circular cross section and sends the signal to the robot arm.
- After the object reaches the IR sensor2, now the conveyor motor stops and robot arm starts working, the robot arm at the end of conveyor picks the object and place it in Left side tray.
- In the similar manner, if we place the Non-circular object in the conveyor. The Pi camera detects it as Non-circular and sends the signal to the robot arm.

The top view and front view of the automated inspection system are as shown in the above Figure 4.

Software stage: In this paper we are mainly using two codes for background, namely: LabVIEW code [8,9] written in Graphical programming language and Raspberry pi code written in Python.

LabVIEW code: Figure 5 shows the Front panel. In the front panel we can observe, the following:



(a) (b)

Figure 4. (a) Top view of system and (b) front view of system.

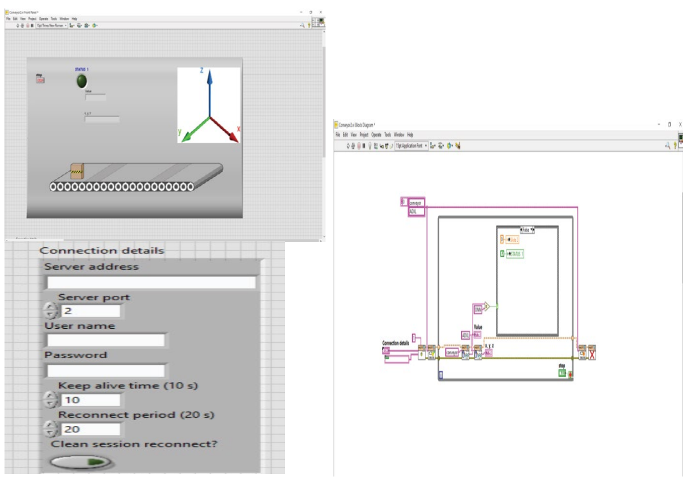


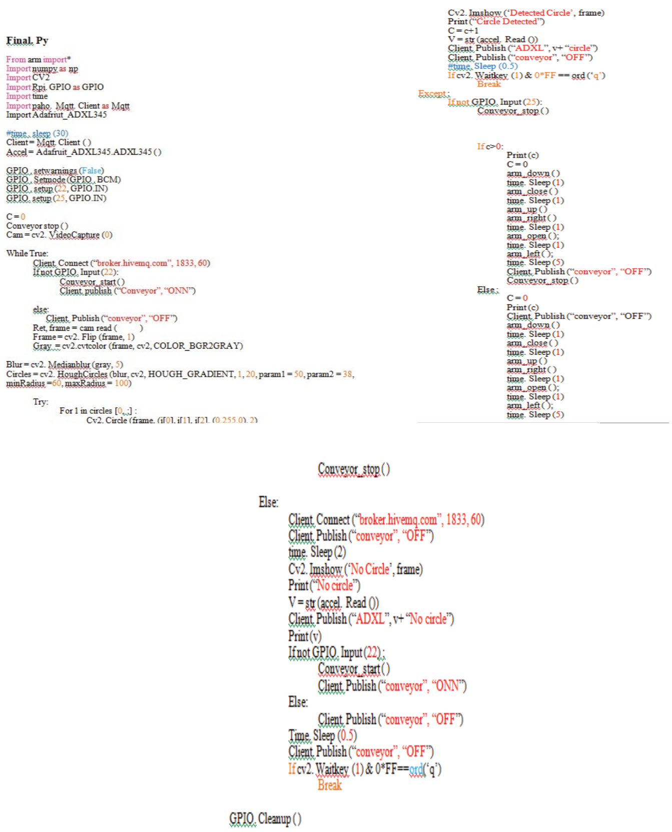
Figure 5. Front panel.

- Object movement on the conveyor.
- Status of the Automated Inspection System.
- Robot Arm Movements (in all X, Y, Z co-ordinates).

Raspberry Pi code: Figure 6 shows the Raspberry pi code which is written in python language.

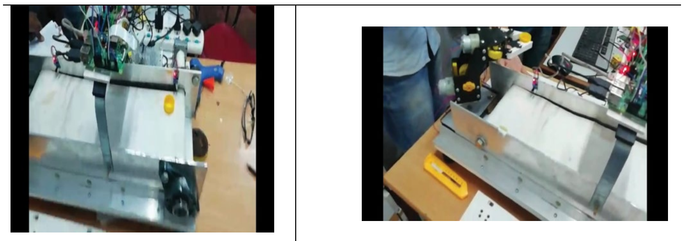
Results

Figures 7 and 8 shows the experimentation on automated inspection of circular and non-circular objects by Vision assisted robot arm using LabVIEW and Raspberry Pi, these are the results obtained.



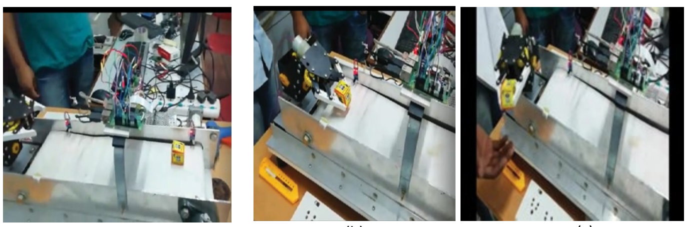
```
Final. Py

From arm import*
Import numpy as np
Import CV2
Import Rpi. GPIO as GPIO
Import time
Import paho. Mqtt. Client as Mqtt
Import Adafriut_ADXL345

#time.. sleep (30)
Client= Mqtt. Client ( )
Accel= Adafruit_ADXL345.ADXL345 ( )

GPIO.. setwarnings (False)
GPIO.. Setmode (GPIO.. BCM)
GPIO.. setup (22, GPIO.IN)
GPIO. setup (25, GPIO.IN)

C=0
Conveyor stop ( )
Cam = cv2. VideoCapture (0)

While True:
        Client. Connect ("broker.hivemq.com", 1833, 60)
        If not GPIO. Input (22):
                Conveyor_start ( )
                Client. publish ("Conveyor", "ONN")

        else:
            Client. Publish ("conveyor", "OFF")
        Ret, frame = cam read (        )
        Frame= cv2. Flip (frame, 1)
        Gray. = cv2.cvtcolor (frame, cv2, COLOR_BGR2GRAY)

Blur = cv2. Medianblur (gray, 5)
Circles = cv2. HoughCircles (blur, cv2, HOUGH_GRADIENT, 1, 20, param1 = 50, param2 = 38,
minRadius =60, maxRadius = 100)

        Try:
                For 1 in circles [0,,:] :
                        Cv2. Circle (frame, (i[0], i[1], i[2], (0,255,0), 2)
        Cv2. Imshow ('Detected Circle', frame)
        Print ("Circle Detected")
        C = c+1
        V = str (accel. Read ())
        Client. Publish ("ADXL", v+ "circle")
        Client. Publish ("conveyor", "OFF")
        #time. Sleep (0.5)
        If cv2. Waitkey (1) & 0*FF == ord ('q')
                Break
Except :
        If not GPIO. Input (25):
                Conveyor_stop ( )

                If c>0:
                        Print (c)
                        C = 0
                        arm_down ( )
                        time. Sleep (1)
                        arm_close ( )
                        time. Sleep (1)
                        arm_up ( )
                        arm_right ( )
                        time. Sleep (1)
                        arm_open ( );
                        time. Sleep (1)
                        arm_left ( );
                        time. Sleep (5)
                        Client. Publish ("conveyor", "OFF")
                        Conveyor_stop ( )
                Else :
                        C = 0
                        Print (c)
                        Client. Publish ("conveyor", "OFF")
                        arm_down ( )
                        time. Sleep (1)
                        arm_close ( )
                        time. Sleep (1)
                        arm_up ( )
                        arm_right ( )
                        time. Sleep (1)
                        arm_open ( );
                        time. Sleep (1)
                        arm_left ( );
                        time. Sleep (5)

                        Conveyor_stop ( )

        Else:
                Client. Connect ("broker.hivemq.com", 1833, 60)
                Client. Publish ("conveyor", "OFF")
                time. Sleep (2)
                Cv2. Imshow ('No Circle', frame)
                Print ("No circle")
                V = str (accel. Read ())
                Client. Publish ("ADXL", v+ "No circle")
                Print (v)
                If not GPIO. Input (22):
                        Conveyor_start ( )
                        Client. Publish ("conveyor", "ONN")
                Else:
                        Client. Publish ("conveyor", "OFF")
                Time. Sleep (0.5)
                Client. Publish ("conveyor", "OFF")
                If cv2. Waitkey (1) & 0*FF==ord('q')
                        Break

GPIO. Cleanup ( )
```

Figure 6. Raspberry Pi code.



(a) (b)

Figure 7. Circular object placed in the conveyor.



(a) (b) (c)

Figure 8. Non-circular object placed in the conveyor .

In the above diagrams we can clearly see different stages of our system in detail. Circular Objects enters the conveyor system and the middle we can clearly see that raspberry pi camera recognizes circular shape and at the end robot picks the circular object and place in the left side tray.

Same as circular shapes, when irregular shapes enters into the conveyor (other than circular), we can clearly see that at the end, the robot arm picks and place it in the right side tray.

Discussion

It is observed that Non-Circular shapes picked by the robot arm, is placed right side to the conveyor System.

At last we can say that, the automated inspection system developed for sorting operation of regular and irregular objects in the conveyor system integrated with vision system is observed with maximum efficiency.

Conclusion

In this project, a description was given to a fully automated and vision assisted robot arm system that was controlled autonomously. The manipulator functioned as expected in simulation tests. Positioning error tests showed that the manipulator's mechanism yielded minor errors considering the fact that it is an open loop control system. The characteristics of the project are observed, and the result compared to manual inspection is stated below:

- Performance of the Automated Inspection System is very high as compared to Manual Inspection System.
- Human errors can be eliminated by employing Automated Inspection system.
- They are able to adapt, quickly and easily, to different products and surfaces.
- They are able to work 24 hours a day.

References

1. Thomas, Ron Oommen, and K. Rajasekaran. "Remote monitoring and control of robotic arm with visual feedback using Raspberry Pi." *Int J Comput Appl* (2014).
2. Tiwari, Rohit, and Dushyant Kumar Singh. "Vehicle control using Raspberry Pi and image processing." Intelligent Communication, Control and Devices (2018).
3. Hemalatha, P, C.K Hemantha Lakshmi, and S.A.K Jilani. "Real time image processing based robotic arm control standalone system using Raspberry pi." - *J Electron Commun* (2015).
4. Pereira, Viren, Vandyk Amsdem Fernandes, and J. Sequeira. "Low cost object sorting robotic arm using Raspberry Pi." - *IEEE Glob Humanit Technol Conf* (2014).
5. "Raspberry Pi forum and discussion page."
6. "Programming the Raspberry Pi: Getting started with Python." Book by Simon Monk.
7. https://wiki.eprolabs.com/index.php?title=Raspberry_Pi_Camera_Module.
8. Travis, Jeffrey and Jim Kring. "LabVIEW for everyone: Graphical programming made easy and fun."
9. www.ni.com/discussion forum.

How to cite this article: Saisunil, Kota and Y Ashok Kumar Reddy. "Automated Inspection by Vision Assisted Robot Arm using LabVIEW and Raspberry Pi." Adv Robot Autom 11 (2022): 199.