Quality Control System

Bachelors of Engineering in Electrical (Telecomm)

BY

NC Zaki Nasir
NC Usama Hamid
NC Usama Sarwar

BATCH

(BETE-49)

SUPERVISOR

Dr. Mir Yasir Umair

Abstract

Quality Control is a major part of any production line and pharmaceutical companies are no exception to this. An automated Quality Control System is necessary to make any production line economically feasible, as manual labor is highly cost ineffective. Creating an automated Quality Control System to detect faulty medicine packaging is the primary objective of this project. The developed system will be responsible for detecting sub-standard medicine packaging and removing them from the production line.

We have tried to solve this problem by creating a Quality Control System that detects faulty medicine packaging through Image Processing Techniques. The system will detect any leaflet of medicine that does not meet the set requirements and a mechanical pusher will remove that package from the production line.

This report contains description about the project along with design details; both software and hardware, project analysis and evaluation, results of rigorous testing and their analysis, problems faced while development of the project, a demonstration outline and recommendations for future work.

Project Supervisor: Dr. Mir Yasir Umair   Dated: May 17, 2016

Copyright Statement

Dedication

Dedicated to our

Friends for their endless support.

Faculty for their efforts and dedication.

And family for their faith in us.

Acknowledgement

First of all, we would like to thank ALLAH (Subhanahuwata'ala) who enlightened us with the knowledge that we needed for our project and gave us strength to accomplish our task.

We acknowledge, with deep gratitude and appreciation, the inspiration, encouragement, valuable and continuous guidance given to us by Dr. Mir Yasir Umair in regard to our project.

Special thanks to our family members for bearing with us through the tough routine and for continuously encouraging us.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

MATLAB:	Matrix Laboratory
RGB:	Red Green Blue
HSB:	Hue Saturation Brightness
PN:	Part Number
R:	Radius
D:	Diameter
RAM:	Random Access Memory
USB:	Universal Serial Bus
HD:	High Definition
HDMI:	High-Definition Multimedia Interface
I/O:	Input/output
RF:	Radio Frequency
TBD:	To be Defined

1. Introduction

The Medicine Industry has a vital role to play in the society. This industry is responsible for providing the public with means to address health problems. The nature of their work demands a great need for the quality assurance of the products.

Currently most industries are using manual labor to ensure that defected medicine packaging is not passed onto the market. There are usually 4 to 6 persons per single production line, responsible for maintaining standardized quality. The need to automate this process was felt.

1.1. Background/Motivation

Numerous problems were found to be associated with manual packaging. Firstly, it is not very cost effective especially when compared with an automated system. The companies could reduce the pressure exerted onto their finances by reducing the pay roll. Companies need only pay one time buying fee for an automated system rather than catering for wages on a monthly basis.

Secondly, the efficiency of such a method is questionable. Humans are very much prone to error, a fact which has been highlighted by science and rectified by the advancement in technology. The efficiency is guaranteed to improve using an automated system.

Then there are concerns regarding hygiene. Having people handle the medicine tablets before they are completely packaged is a risky method. If the hygiene of such employees is not kept under strict scrutiny, there is a danger of the medicine getting contaminated which would defeat the primary purpose of the medicine industry.

1.2. Project Description

This project is mainly based on image processing and signal processing; so this project aims to integrate theoretical knowledge with practicality to gain further insight in the field of electrical engineering and refine our skills in the field of wireless communication and signal processing for practical lives ahead.

This project should correctly identify faulty packaging in a given production line. After correctly identifying the faulty product, the project should remove that particular product from the production line in order to prevent it from being completely packaged and passed onto the market. In case a faulty packaging is detected by the processing, a mechanical pusher installed at the end of the production line is responsible for removing the faulty medicine packaging from the production line.

1.3. Salient Features

The system will comprise of three main parts:

- Capturing Image of the leaflet
- Use of Image Processing Techniques to identify if the leaflet is faulty
- Removal of faulty leaflet using a mechanical pusher

A high resolution camera will be used to capture images of each passing medicine leaflet. The leaflets will be placed onto a moving conveyer belt. Whenever a leaflet will enter the area being focused by the camera, an image will be captured by the camera and sent to the Raspberry Pi. Using appropriate image processing techniques, faults in the packaging, if they exist, will be identified. A mechanical Pusher will be responsible for pushing the faulty packaging off the conveyor belt. Accepted leaflets will be passed on for final packaging by the conveyer belt.

1.4. Scope, Objectives and Deliverables

This project should act as a quality control system for pharmaceutical industries. Medicine leaflets are packaged through an automated process which is prone to error. This project should first detect any faulty medicine packaging and then remove it from the production line.

The processor used for this purpose is a raspberry pi board. The coding is being done in Python using the OpenCV(Open source computer vision Library). Hough Circle transform will be applied to detect the circles. In case a faulty package is detected, a signal will be sent by the raspberry pi to the mechanical pusher to remove the packaging from the production line.

This project will help us polish the knowledge gained through the course of this bachelor’s degree to practical use and polish our skills especially in the field of Image and Signal Processing, Wireless Communication, Embedded System Design, Networking and Programming.

1.5. Organization of document

This document is divided into four main parts:

- First part introduces the project.
- Second part tells about previous work done related to this field, and how previously done work will be integrated with this project.
- Third part is about design, specification and algorithm.
- Fourth part is about recommendations for future work and conclusion.
- Final part is about references and bibliography.

1.6. Sequential Diagram

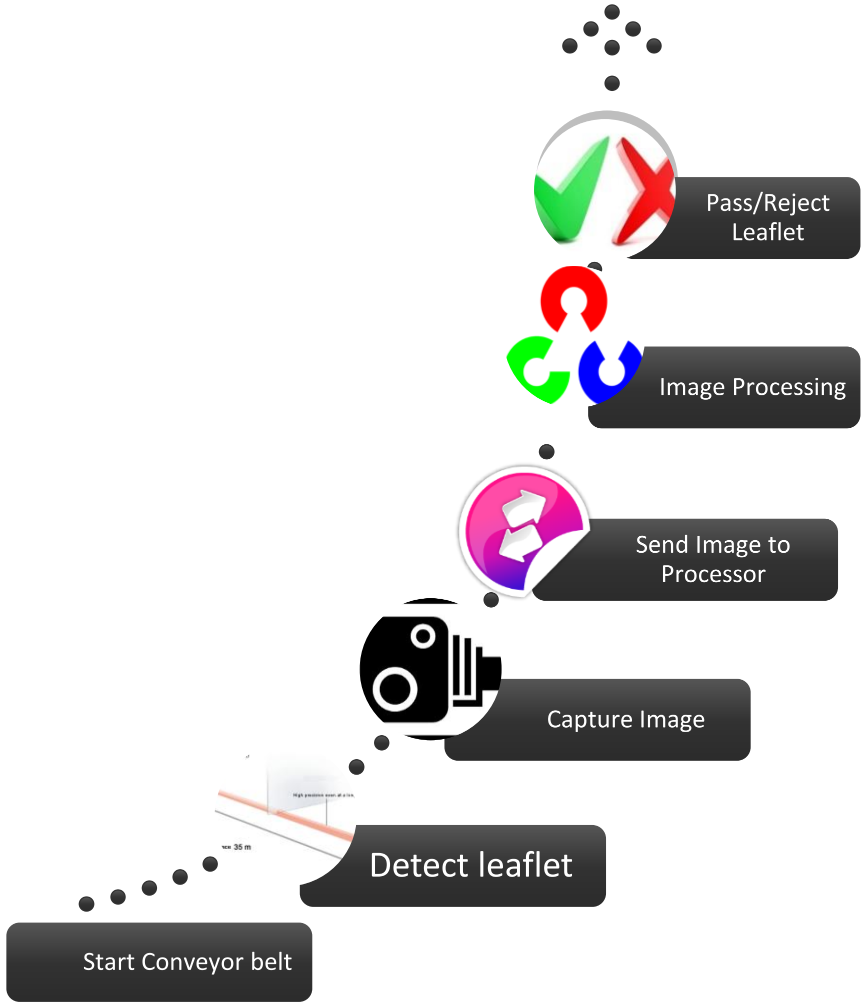


Figure 1. Sequential Diagram

2. Background Study

Koneru Lakshmaiah Education Foundation in India published a research paper that focuses on "the design and selection of the proper conveyor belt". This paper provides to design the conveyor system used for which includes belt speed, belt width, motor selection, belt specification, shaft diameter, pulley, gear box selection, with the help of standard model calculation. [1]

A research paper published by M. Khojastehnazhand, M. Omid and A. Tabatabaeefar about "Development of a lemon sorting system based on color and size" talks about the classification of citrus such as oranges lemons and tangerines. There are two inspection stages of a sorting system; external inspection and internal inspection. The external inspection is accomplished by the process of thermal imaging while the internal inspection requires special sensors for moisture and acid content. In this paper, an efficient algorithm for grading lemon fruits is developed and implemented in visual basic environment. The system consists of two CCD cameras, two capture cards, an appropriate lighting system, a personal computer and other mechanical parts. The algorithm initially extracts the fruit from the background. The samples of different grades of lemon are situated in front of the cameras and are calibrated off-line. Then information on the HSI color values and estimated volumes of fruits are extracted and saved in a database. By comparing the information during sorting phase with the available information inside the database, the final grade of the passing fruits is determined. This algorithm can be easily adapted for grading and/or inspection of other agricultural products such as cucumber and eggplant. RGB color space is the most popular color model. Some algorithms of fruit vision are based on this color model (Steinmetz et al., 1996; Leemans et al., 1998; Paulus et al., 1997). When color is presented with R, G and B, the amount of information is tripled. Thus more algorithms could be developed to comprehensively utilize the color information in an image. However, RGB system is sensitive to lighting or other conditions. Another color model is HSI (Hue, Saturation and Intensity). Here, however, HIS is preferred, as in this system Hue value is comparatively stable. The color of fruit is determined by calculating average Hue (H) value for the fruit. [2]

In another paper object sorting using robotic Pusher based on color detection is designed and implemented. Existing sorting method uses a set of different capacitive, inductive, and optical sensors to differentiate object color. In the proposed system a mechatronics color sorting system is developed with the image processing technique. Image processing technique senses the objects captured in real-time by a webcam and then identifies color and information out of it. In the proposed system a mechatronics color sorting system is developed with the image processing technique. So, the proposed system will eliminate the monotonous work done by human and provides greater accuracy and speed in the work. [3]

From Khulna University, G. M. Atiqur Rahamanand Md. Mobarak Hossain published a research paper "Automatic Defect Detection and Classification Technique from Image: A special case using ceramic tiles". This paper presents proposes an automated defect detection and classification technique that can have ensured the better quality of tiles in manufacturing process as well as production rate. This image processing operation is carried out in the following manner:

Step 1: Performing some image preprocessing operations.

Step 2: Applying the proposed defect detection process.

Step 3: Classifying the defect using all proposed algorithms.

The method, used, plays an important role in ceramic tiles industries to detect the defects and to control the quality of ceramic tiles. This automated classification method helps us to acquire knowledge about the pattern of defect within a very short period of time and also to decide about the recovery process so that the defected tiles may not be mixed with the fresh tiles. [4]

Another research paper was published by H. Elbehiery, A. Hefnawy, and M. Elewa on "Surface Defects Detection for Ceramic Tiles Using Image Processing and Morphological Techniques". Their work was a quality control enhancement by integrating a visual control stage using image processing and morphological operation techniques before the packing operation to improve the homogeneity of batches received by final users. The ceramic tiles were captured through the online camera held on the line production at the industry. The image captured was converted to another kinds of images (Binary, and Gray scale) to be suitable for the various defect detection algorithms used for the different types of defects. The paper discussed formal mathematical properties like Edge detection, Morphology operations, Noise reduction, smoothing process, Histogram equalization, and intensity adjustment. The effects of unequal lighting and of the space sensitivity of the TV camera CCD were corrected analyzing a sample tile made of white Plexiglas. [5]

Yet another paper, by Aiyush Suhasaria and Khanindra Pathak, deals with separation of gangue materials from an ore. To separate the ore from the waste and gangue materials Image Processing techniques are applied using live video of the ore and gangue mixture on the conveyor. The optical data captured by camera (sensor) is transferred via an optical data transferring system to a processor. The image processing techniques are applied to differentiate the ore and gangue material. The output signal is used to actuate appropriate mechanisms to direct the feed towards appropriate outgoing line. Thus the system reduces the ore dilution and help in enhancing the ore grade. The video feed generated is converted to Hue Saturation Value (HSV) format and appropriate threshold is applied to differentiate ore from the impurities. Detection by setting appropriate threshold in HSV format takes care of the ambient lighting conditions. This paper presents a laboratory scale development and testing of this technique and reports the output of the software developed in C for this purpose. [6]

2.1. Use of Background Study

These documents are about different techniques of image processing and design of a conveyer belt. Each document focuses on a special aspect this project. Positives and negatives are discussed and then the most suitable approach or algorithm is suggested. These background studies not only highlight a lot of problems that occurred for the previous developers but they suggest what improvements can be done in the future.

3. Design and Development

Development and designing is one of the most important features of Final Year Project. This is the part where raw knowledge is taken from various sources and relevant information is extracted to suit the project's requirements.

3.1. Development

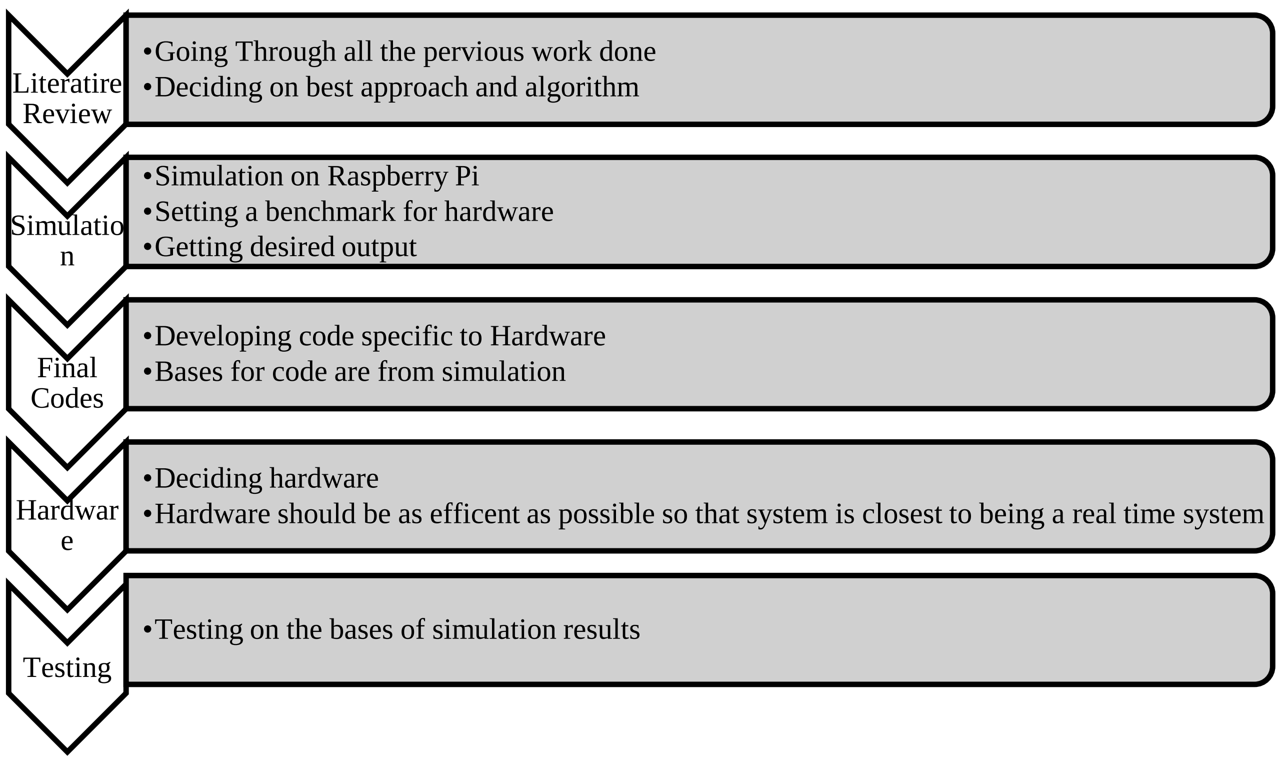


Figure 2. Development Flow Chart

3.2. Design Detail

3.2.1. Conveyor Belt

For any processing to take place moving object must be placed on a conveyor belt. Conveyor belt is a mode of transportation of materials from one location to another. In this case, Conveyor belt is used to transport Pharmaceutical tablets over some distance so that they can be inspected. The designing of Conveyor belt includes belt speed, belt width, motor selection, belt specification, with the help of standard model calculation.

3.2.2. Detection

In computer vision, the processes pertaining to object detection and segmentation are considered vital. Various applications including image search, scene understanding etc are based on the aforementioned processes.

Hough transformation algorithm will be used for the detection of circular tablets, which will enable the system to identify the faulty leaflets.

3.2.3. Conversion from RGB to Gray Scale

First, RGB image is converted to a Greyscale, so that Hough Transform Circle technique maybe applied. Before the conversion, filtering operation is applied for Noise removal purposes. Edge detection operation will then follow (Canny Edge Detection Process)

RGB can be converted to grayscale using the following methods.

Gray RGB color code has equal red, green and blue values:

$R = G = B$

For each image pixel with red, green and blue values of (R, G, B):

$R' = G' = B' = (R+G+B) / 3 = 0.333R + 0.333G + 0.333B$

This formula can be changed with different weights for each R/G/B values.

$R' = G' = B' = 0.2126R + 0.7152G + 0.0722B$

Or

$R' = G' = B' = 0.299R + 0.587G + 0.114B$

Figure 3. Conversion to Gray Scale

3.3. Hough Circle Transform

Due to their versatility, geometric circles are considered pivotal in image processing techniques. The Hough Circle Transform can be used to obtain circular edges from any given image.

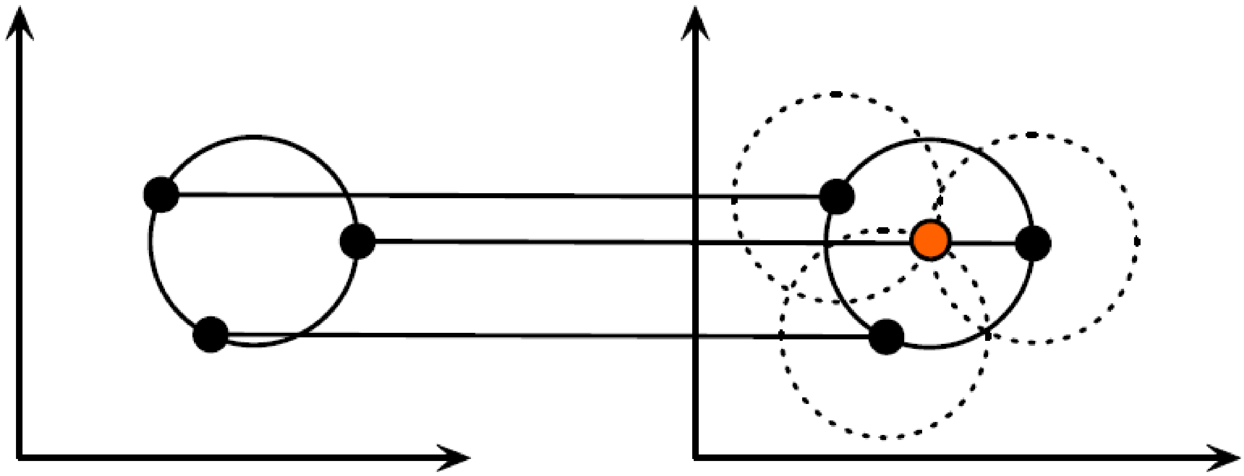
The Hough transform has the ability to determine the parameters of a circle provided the points lying on the perimeter have been determined. Mathematically a circle is described by the parametric equations below.

$$\begin{aligned} x &= a + R\cos(\theta) \\ y &= b + R\sin(\theta) \end{aligned}$$

3.3.1. Search with Fixed R

If the circles in an image are of known radius R, then the search can be reduced to 2D. The objective is to find the (a, b) coordinates of the centers.

$$\begin{aligned} x &= a + R\cos(\theta) \\ y &= b + R\sin(\theta) \end{aligned}$$

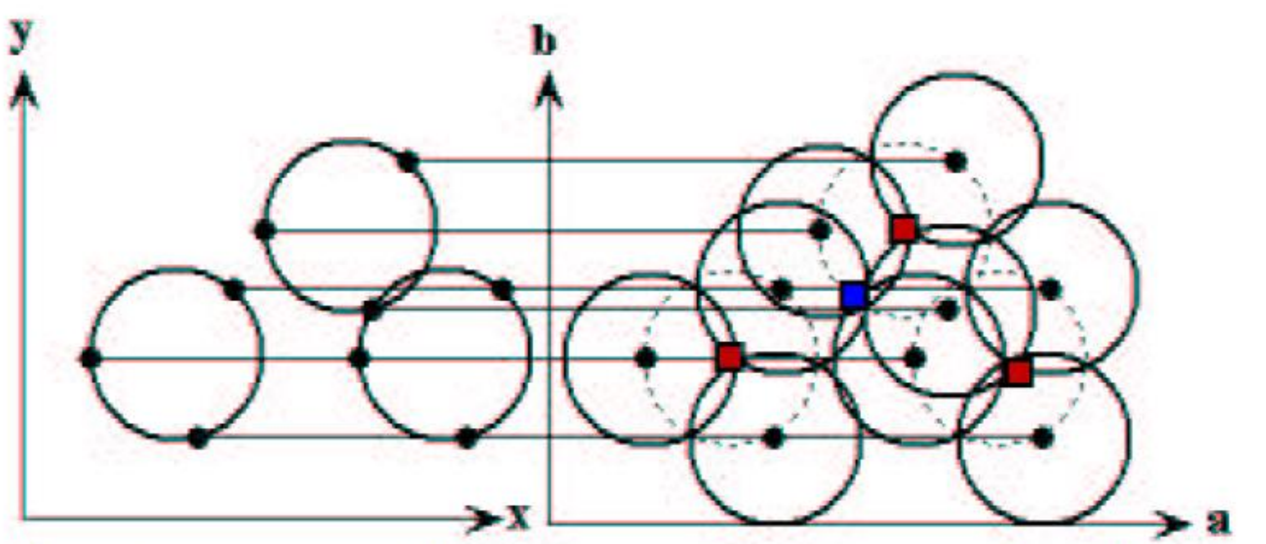


Each point in geometric space (left) generates a circle in parameter space (right). The circles in parameter space intersect at the (a, b) that is the center in geometric space.

Figure 4. Highlighting single boundary

3.3.2. Multiple Circles with known Radius

Using the technique under consideration, various circles with the same radius can be found. The center points are represented as red cells in the image. Overlapping of circles can cause false circles to be detected as well which have been marked by the blue box.



Each point in geometric space (left) generates a circle in parameter space (right). The circles in parameter space intersect at the (a, b) that is the center in geometric space.

Figure 5. Highlighting multiple boundary

3.4. OpenCV

OpenCV (Open Source Computer Vision) is a library of programming functions for real time computer vision. It has the following advantages

- **Speed**

Matlab is built on Java, and Java is built upon C. So when you run a Matlab program, your computer is busy trying to interpret all that Matlab code. Then it turns it into Java, and then finally executes the code. OpenCV, on the other hand, is basically a library of functions written in C/C++. You are closer to directly provide machine language code to the computer to get executed. So ultimately you get more image processing done for your computers processing cycles, and not more interpreting. As a result of this, programs written in OpenCV run much faster than similar programs written in Matlab.

- **Resources needed**

Due to the high level nature of Matlab, it uses a lot of your systems resources. Matlab code requires over a gigabyte memory of RAM to run through video. In comparison, typical OpenCV programs only require 70mb of RAM to run in real-time.

- **Portability**

MATLAB and OpenCV run equally well on Windows, Linux and MacOS. However, when it comes to OpenCV, any device that can run C, can, in all probability, run OpenCV. So in our case using Raspberry Pi as our basic image processing system supports the using of OpenCV.

4. Hardware Design

4.1. Camera

Camera Interfacing is one of the most vital part of this project. In this section, a Camera is integrated/installed to a processor, (in this case, the Raspberry Pi).

The process of connecting a small camera to the main Broadcom BCM2835 processor can be accomplished with the help of the Mobile Industry Processor Interface (MIPI) and the Camera Serial Interface associated with the Raspberry Pi. This provides an electrical bus connection between both the devices.

The Camera Board can be plugged straight into the CSI connector of the Raspberry Pi. The Raspberry Pi camera has the ability to take pictures at 5MP and record 1080p videos at 30 frames per second. A 15 pin Ribbon cable is used to connect the camera board onto the Raspberry Pi via a 15 pin MIPI Camera Serial Interface (CSI) which is used specifically for camera interfacing. Extremely high data rates are supported by the Camera Serial Interface (CSI) data bus and it is responsible for transferring pixel data to the BCM2835 processor.

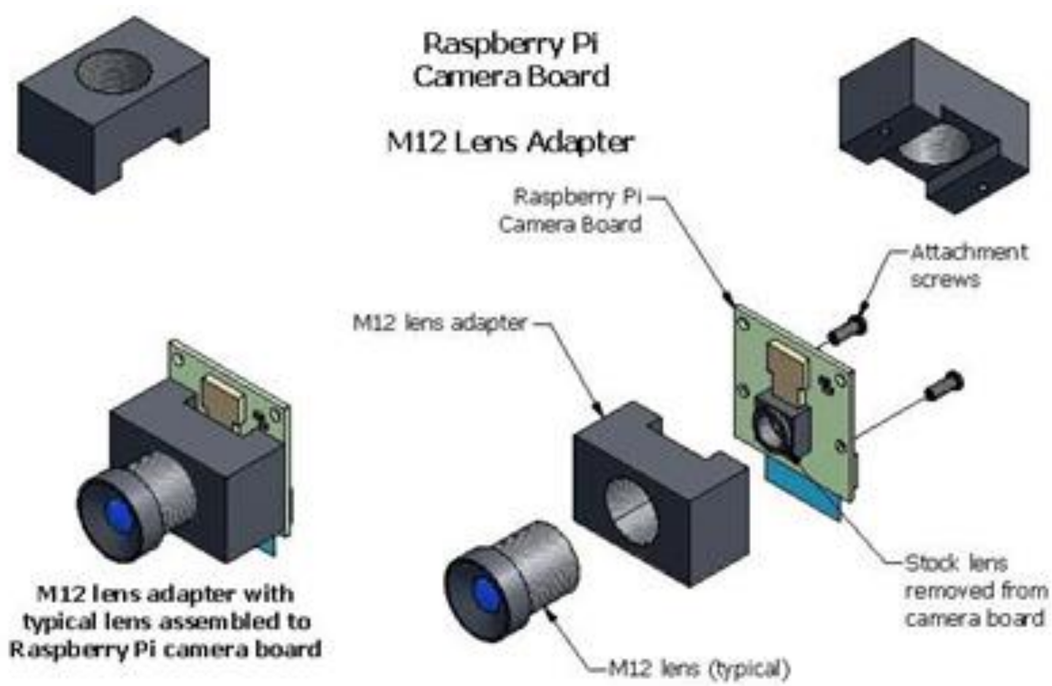


Figure 6. Components of Raspberry Pi

4.1.1. Specifications

Net price	25 $
Size	around 25 x 20 x 9 mm
Weight	3 g
Still resolution	5 Megapixels
Video modes	1080p30, 720p60 and 640x480p60/90
Linux integration	V4L2 driver available
C programming API	OpenMAX IL and others available
Sensor	OmniVision OV5647
Sensor resolution	2592 x 1944 pixels
Sensor image area	3.76 x 2.74 mm
Pixel size	1.4 µm x 1.4 µm
Optical size	1/4"
Full-frame SLR lens equivalent	35 mm
S/N ratio	36 dB
Dynamic range	67 dB @ 8x gain
Densitivity	680 mV/lux-sec
Dark current	16 mV/sec @ 60 C
Well capacity	4.3 Ke-
Fixed Focus	1 m to infinity
Focal length	3.60 mm +/- 0.01
Horizontal field of view	53.50 +/- 0.13 degrees
Vertical field of view	41.41 +/- 0.11 degress
Focal ratio (F-Stop)	2.9

Table 1. Raspberry Pi Camera Specification

4.2. Raspberry pi 2 Board:

The Raspberry Pi is an economical small sized computing device that plugs into any HDMI display device. It has four USB ports and supports the standard mouse and keyboard. It has a powerful processor and is capable of running simultaneous instances of heavy applications and softwares. It can perform almost all the functions of a standard desktop computer including programming codes in different languages, accessing the internet and playing HD videos in any format and making spreadsheets in excel along with various other tasks.

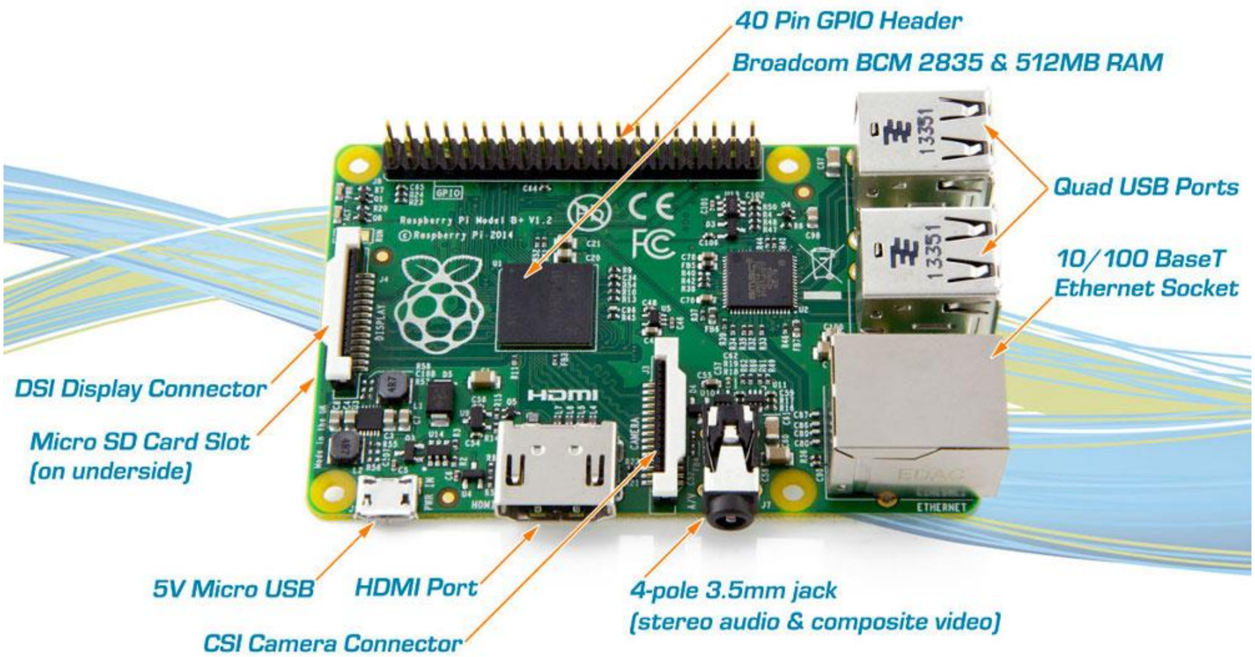


Figure 7. Raspberry Pi 2 Model B+ Port Label

4.2.1. Specifications:

- SoC: Broadcom BCM2836 (CPU, GPU, DSP, SDRAM)
- CPU: 900 MHz quad-core ARM Cortex A7 (ARMv7 instruction set)
- GPU: Broadcom Video Core IV @ 250 MHz
- More GPU info: OpenGL ES 2.0 (24 GFLOPS); 1080p30 MPEG-2 and VC-1 decoder (with license); 1080p30 h.264/MPEG-4 AVC high-profile decoder and encoder
- Memory: 1 GB (shared with GPU)
- USB ports: 4
- Video input: 15-pin MIPI camera interface (CSI) connector
- Video outputs: HDMI, composite video (PAL and NTSC) via 3.5 mm jack
- Audio input: I^2S

- Audio outputs: Analog via 3.5 mm jack; digital via HDMI and I^2S
- Storage: MicroSD
- Network: 10/100Mbps Ethernet
- Peripherals: 17 GPIO plus specific functions, and HAT ID bus
- Power rating: 800 mA (4.0 W)
- Power source: 5 V via MicroUSB or GPIO header
- Size: 85.60mm × 56.5mm
- Weight: 45g (1.6 oz)

4.3. Mechanical Pusher

4.3.1. Elements:

1. Disk
2. Stepper Motor
3. Motor Driver (L298n)
4. Plastic rod
5. Plastic Pusher

4.3.2. Construction:

A disk is used which a movable joint has attached to it. The movable joint has a rod connected at the end that will be used to push the leaflets off of the conveyer belt. Now this disk will be connected with the stepper motor. The disk moves back and forth 180 degrees. Stepper motor is then connected with the motor driver (L298n). A plastic rod is the connected with the disk that will move as the disk rotates 180 degrees. The corresponding movement of the rod and the disk will push the leaflets off of the conveyer belt.

4.3.3. Disk:

A circular disk is used which is 12 inches in diameter, 2mm in width and weighs approximately 150 grams.

4.3.4. Stepper motor:

A stepper motor is a modern day electromechanical device that takes electrical pulses as an input and translates them into corresponding mechanical movements. By manipulating the sequence of the electrical pulses applied at the pins of the motor which are responsible for the mechanical outputs, the desired mechanical movements to accomplish the set objectives can be achieved.

4.3.4.1. Specifications:

5-16 Volts.

3 Amps current.

5 KG torque.

250 Grams weight.

Min resolution 7.2 degrees.

4.3.4.2. Application:

The mechanical pusher will receive a signal from the raspberry pi whether or not the leaflet has to be pushed off or not. If in case a faulty leaflet has been detected, the stepping motor will move forward 180 degrees to push off the leaflet. Once the leaflet has been pushed off, the motor will then move backwards 180 degrees.

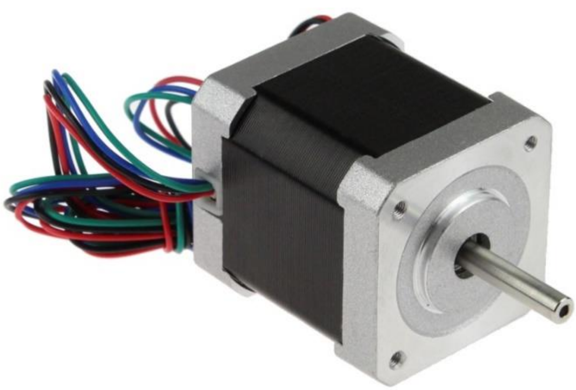


Figure 8. Stepper Motor

4.3.4.3. Motor Driver (L298n):

The LN298 is a high voltage, high current, dual full-bridge motor driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors

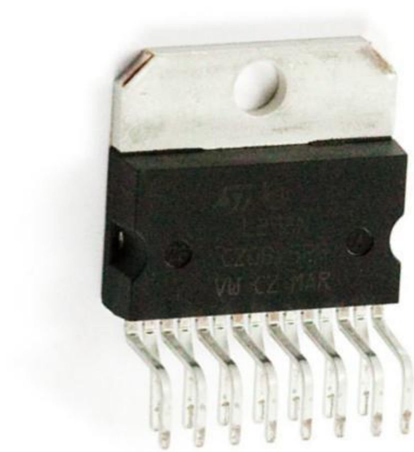


Figure 9. Motor Driver (L298n)

4.3.4.4. Features:

- Operating supply voltage of up to 46V
- 4.5-7VDC logic supply voltage
- Total DC current of up to 4A
- Low saturation voltage
- Over-temperature protection
- Logical ‘0’ input up to 1.5V (high-noise immunity)

4.3.5. Plastic Rod and Pusher:

A plastic rod is connected with the disk, which is 24 inches in length and 0.2mm in diameter. A plastic pusher is then connected at the end of the rod that will be used to act as the final force to push the leaflet off the conveyer belt.

5. Software Design

5.1. Technique:

Canny edge detection technique will be used during the detection process. The Process of Canny edge detection algorithm can be broken down to 5 different steps:

1. Gaussian filter is applied for smoothing and noise removal purposes
2. Intensity gradients of image are calculated
3. Detection of spurious edges and false circles using non maximum suppression
4. Prospective edges in the image are determined by application of double threshold
5. Edge tracking through hysteresis.

5.2. Code and Explanation:

5.2.1. Detection of faulty leaflets:

Coding was done in python language using Open CV. The code is relatively simple. After importing all the necessary code packages and creating necessary variables for image manipulation, the program prompts for an input image. Then a copy if the image is created. The copy is then converted into greyscale (to allow for the application of appropriate image processing techniques). Hough circles technique is then called upon and applied on the greyscale image for circle detection purposes. A variable by the name of ‘circles’ stores the result. If ‘circles’ is not equal to ‘none’ then, the Raspberry Pi turns on the stepper motor via motor driver which in turn controls the movement of the mechanical arm to push off the faulty leaflet. Simultaneously, the program also highlights the circumference of the detected circles and marks their centers with rectangular spots. After completing this procedure the output image is displayed.

5.2.2. Code

```
# first import the necessary  packages

importnumpy as np

importargparse

import cv2


# construct the argument parser and parse the arguments

ap = argparse.ArgumentParser()

ap.add_argument("-i", "--image", required = True, help = "Path to the image")

args = vars(ap.parse_args())


# load the image, clone it for output, and then convert it to grayscale

image = cv2.imread(args["image"])

output = image.copy()

gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)


# detect circles in the image

circles = cv2.HoughCircles(gray, cv2.HOUGH_GRADIENT, 1.2, 100)


# ensure at least some circles were found
```

```
if circles is not None:

    /* call the function to signal the stepper motor(see the sub-section below) */

      # convert the (x, y) coordinates and radius of the circles to integers

      circles = np.round(circles[0, :]).astype("int")


      # loop over the (x, y) coordinates and radius of the circles

      for (x, y, r) in circles:

      # draw the circle in the output image, then draw a rectangle corresponding to the center to
      circles

              cv2.circle(output, (x, y), r, (0, 255, 0), 4)

              cv2.rectangle(output, (x - 5, y - 5), (x + 5, y + 5), (0, 128, 255), -1)

      # show the output image

      cv2.imshow("output", np.hstack([image, output]))

      cv2.waitKey(0)
```

5.2.3. Stepper Motor Code:

Once the function to the stepper motor is called in the program, the Raspberry Pi starts signaling the stepper motor. What basically happened is that the Raspberry Pi configures the appropriate GPIO pins on the stepper motor. These particular pins correspond to the two coils attached to the stepper motor which will enable Raspberry Pi to achieve the desired motion. Two pins correspond to each of the two coils, one for positive end and the other for the negative end.

The ‘forward’ sub-function enables and disable these for pins in a series of four actions. In each action, the pins are given a value of 0 or a 1 as per the requirement. 0 denotes negative end of the coil and 1 denotes the pin which is the positive end. By switching the negative and the positive ends in a particular order, the stepper motor achieves a rotatory motion. This sub-function the forward rotation of the disc. The 'backwards' sub-function in the program uses the same four steps but in the reverse order, allow the control for the backward motion of the disc.

Now, once the motor function is called upon and the appropriate pins have been set, the forward and backward sub-functions will be called in order causing the disc to move 180 degrees forward, in order to push off the leaflet, and then 180 degrees backwards to come back to its mean position.

```
importRPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BCM)

enable_pin = 18


//pins to be used

coil_A_1_pin = 4

coil_A_2_pin = 17

coil_B_1_pin = 23

coil_B_2_pin = 24


GPIO.setup(enable_pin, GPIO.OUT)

GPIO.setup(coil_A_1_pin, GPIO.OUT)

GPIO.setup(coil_A_2_pin, GPIO.OUT)

GPIO.setup(coil_B_1_pin, GPIO.OUT)
```

```
GPIO.setup(coil_B_2_pin, GPIO.OUT)

GPIO.output(enable_pin, 1)


def forward(delay, steps):

fori in range(0, steps):

setStep(1, 0, 1, 0)

tim e.sleep(delay)

setStep(0, 1, 1, 0)

tim e.sleep(delay)

setStep(0, 1, 0, 1)

tim e.sleep(delay)

setStep(1, 0, 0, 1)

tim e.sleep(delay)


def backwards(delay, steps):

fori in range(0, steps):

setStep(1, 0, 0, 1)

tim e.sleep(delay)

setStep(0, 1, 0, 1)
```

```
time.sleep(delay)

setStep(0, 1, 1, 0)

time.sleep(delay)

setStep(1, 0, 1, 0)

time. sleep(delay)


defsetStep(w1, w2, w3, w4):

GPIO.output(coil_A_1_pin, w1)

GPIO.output(coil_A_2_pin, w2)

GPIO.output(coil_B_1_pin, w3)

GPIO.output(coil_B_2_pin, w4)


while True:

delay = raw_input("Delay between steps(milliseconds)?")

steps= raw_input("How many stepsforward? ")

forward(int(delay) / 1000.0, int(steps))

steps= raw_input("How many stepsbackwards? ")

backwards(int(delay) / 1000.0, int(steps))
```

5.3. Project Analysis and Evaluation

Initially, a random input picture was selected upon which the code was tested. The picture consisted of 8 circles: one large, 6 medium sized and one small (see figure below). Thresholding was set such that the Raspberry Pi should ignore all the circles which were smaller in size than the medium sized circles. The result was:

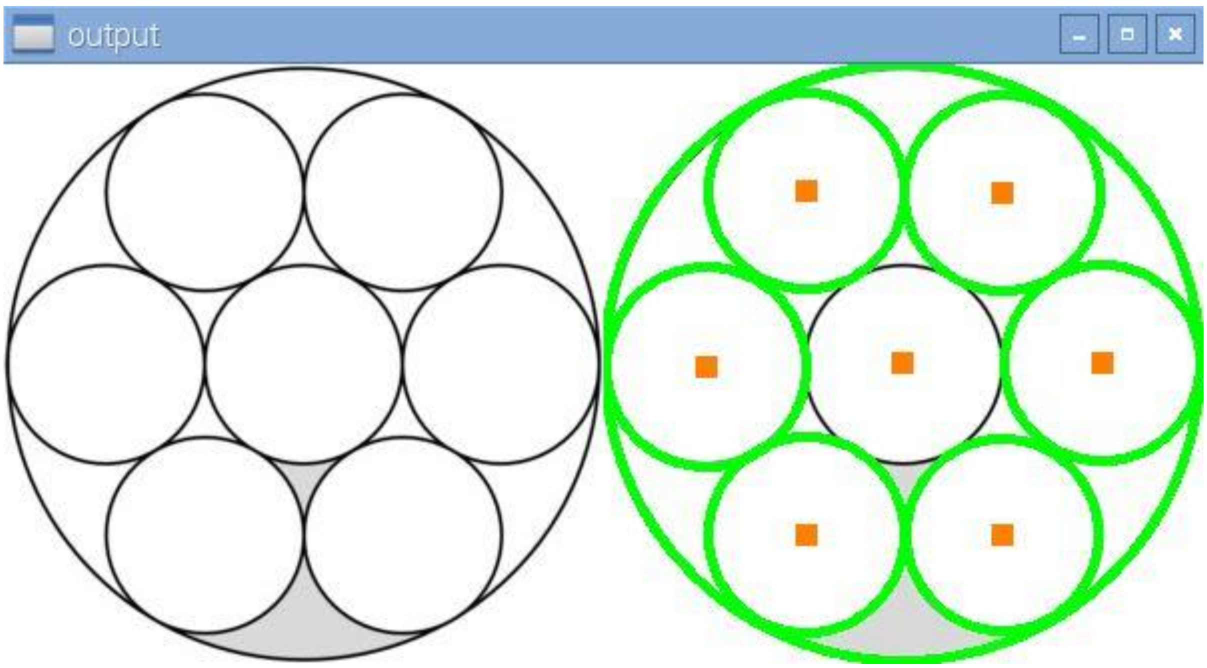


Figure 10. Simulation Result (e.g. 1)

Choosing an appropriate threshold value allowed us to avoid the detection of "false circles" in order to increase the output efficiency to a maximum possible value. (This value was set based on the regular size of the medicine upon which the procedure is to be applied.) After successfully achieving the set objective, the next step was to use a picture of a medicinal leaflet of tablets as the input to the program. After a few trials and modifications made based on those trials, the following results were achieved.

5.3.1. Simulation:

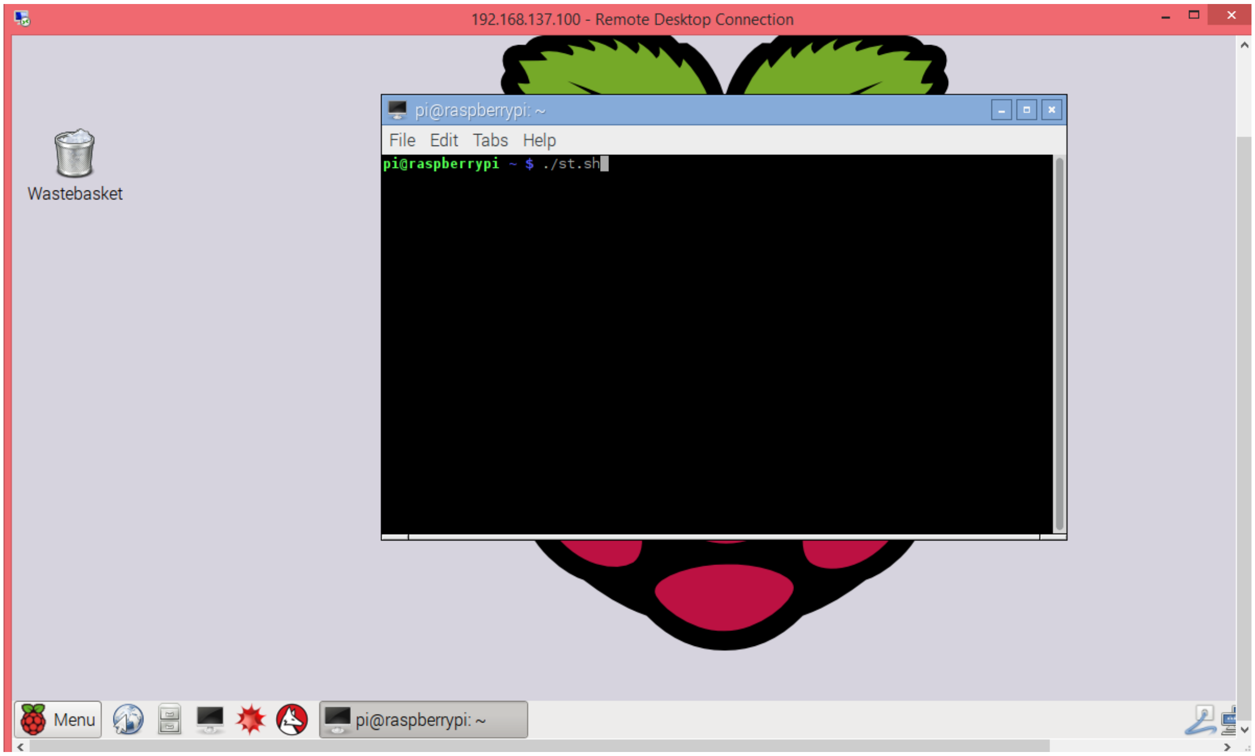
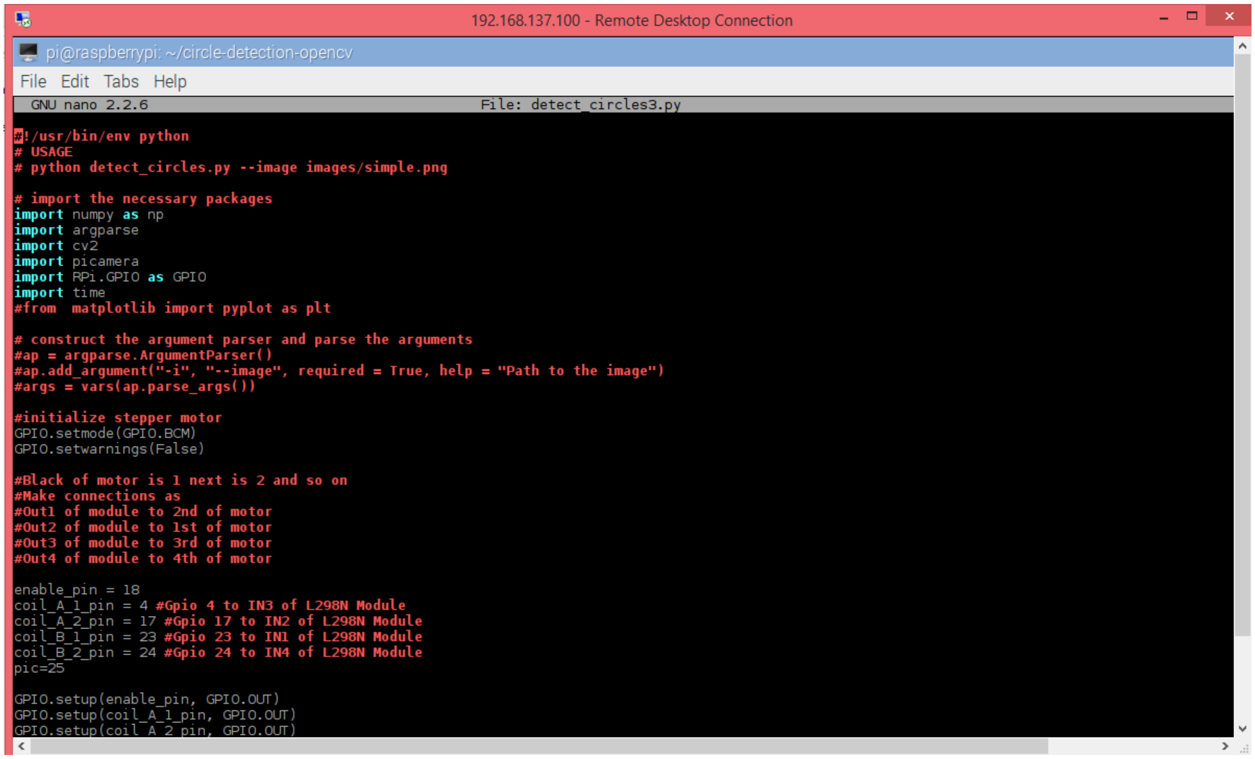


Figure 11. Function Call in Command Prompt (Raspberry Pi)

The raspberry pi was connected to the laptop using 'Remote Desktop Connection'. Appropriate IP's were given through the connection center which enabled us to access the pi through the laptop. Alternatively, an HDMI cable can be connected with the pi to display the output on any HDMI enabled LCD screen.




```
#!/usr/bin/env python
# USAGE
# python detect_circles.py --image images/simple.png

# import the necessary packages
import numpy as np
import argparse
import cv2
import picamera
import RPi.GPIO as GPIO
import time
#from  matplotlib import pyplot as plt

# construct the argument parser and parse the arguments
#ap = argparse.ArgumentParser()
#ap.add_argument("-i", "--image", required = True, help = "Path to the image")
#args = vars(ap.parse_args())

#initialize stepper motor
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

#Black of motor is 1 next is 2 and so on
#Make connections as
#Out1 of module to 2nd of motor
#Out2 of module to 1st of motor
#Out3 of module to 3rd of motor
#Out4 of module to 4th of motor

enable_pin = 18
coil_A_1_pin = 4 #Gpio 4 to IN3 of L298N Module
coil_A_2_pin = 17 #Gpio 17 to IN2 of L298N Module
coil_B_1_pin = 23 #Gpio 23 to IN1 of L298N Module
coil_B_2_pin = 24 #Gpio 24 to IN4 of L298N Module
pic=25

GPIO.setup(enable_pin, GPIO.OUT)
GPIO.setup(coil_A_1_pin, GPIO.OUT)
GPIO.setup(coil_A_2_pin, GPIO.OUT)
```

Figure 12. Hough Circle Python Code

The code was run on the command terminal. The threshold values and circle detection segments can be seen in the image.

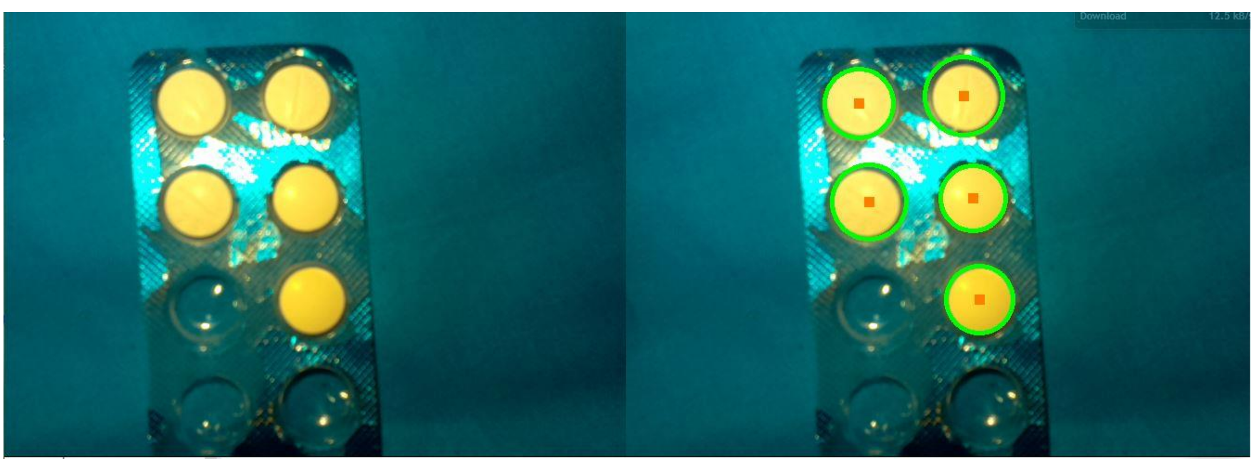


Figure 13. Simulation Result (e.g. 2)

As can clearly be seen, the program was successfully able to detect through the Raspberry Pi, only those cavities which contained the actual medicine. Based on this result, the Raspberry Pi would send a signal to the mechanical arm via the stepper motor causing it to initiate the roll off sequence in order to push the faulty leaflet out of the assembly line. The tablets used in this case were yellow in colour, which shows that this system can be used for medicine tablets of any colour.

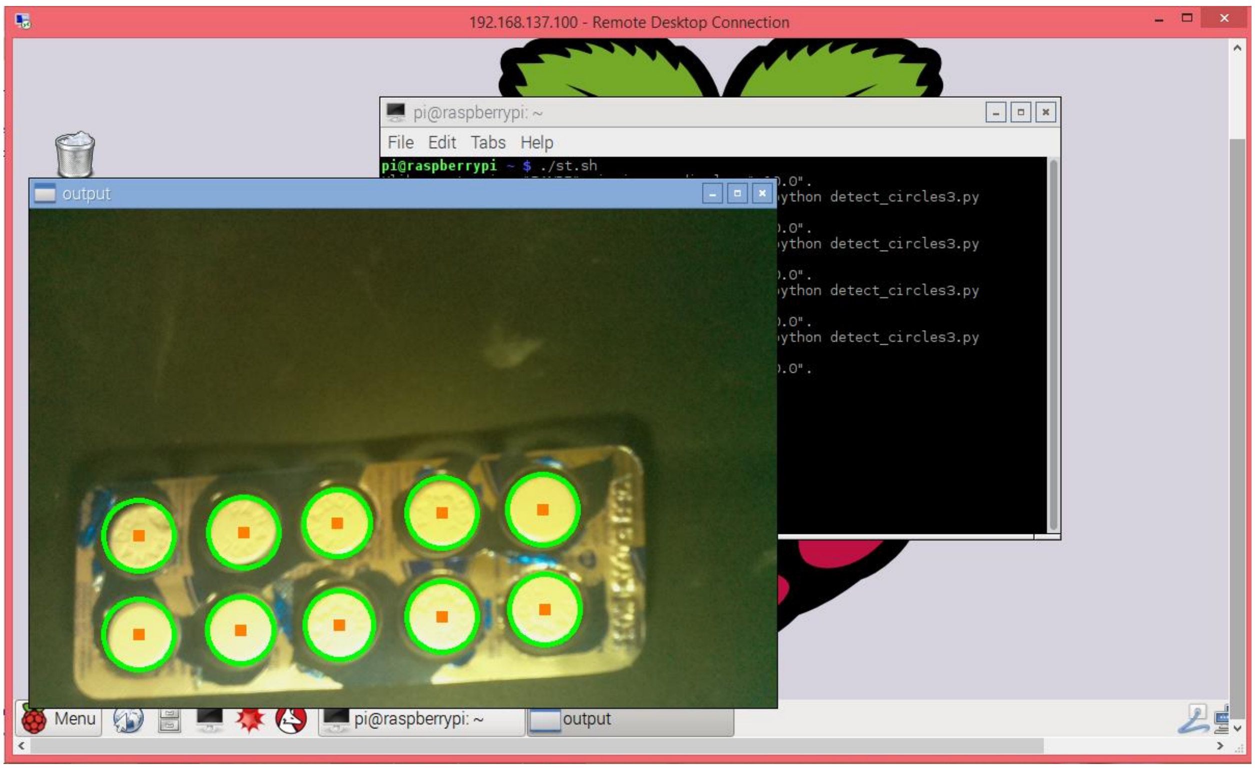


Figure 14. Output Image of Leaflet

The simulation results show that all ten circles have been detected by the raspberry pi. As the leaflet meets all the packaging standards, this leaflet was not pushed off by the conveyer belt.

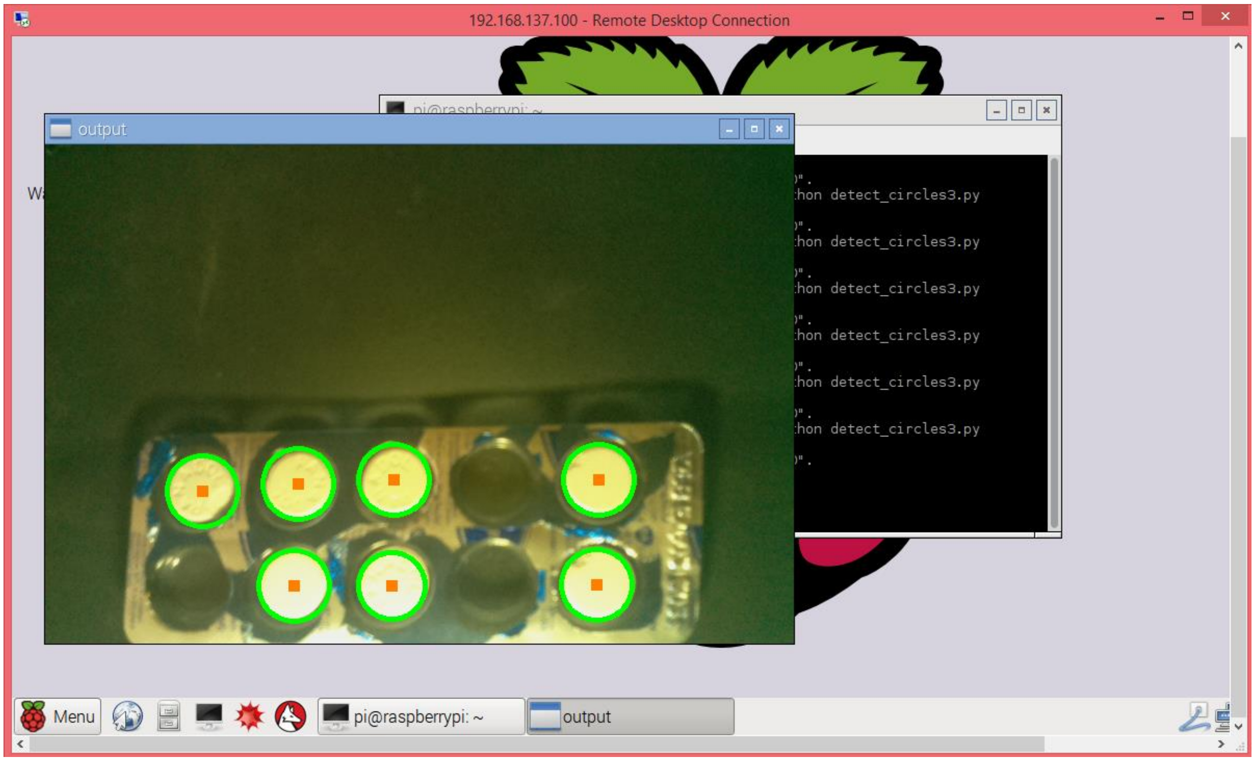


Figure 15. Output Image of Faulty Leaflet

The simulation results in this case show that three tablets are missing and they haven't been detected by the camera. This packaging doesn't meet the packaging requirements and will be pushed off of the conveyer belt by the mechanical arm.

The tablet moves on the conveyer belt and as soon as it passes through the sensor, a signal is sent to the raspberry pi which in turn signals the camera to take a picture. LED's are installed besides the camera to enhance the lighting conditions. Once the picture is taken, it is displayed on the screen and a signal is sent to the raspberry pi to either push off the leaflet or let it pass.

5.3.2. Problems Faced

While testing of this project following were the major problems faced:

- The environment is light sensitive. The light that shines on the camera while taking the picture has to be of a specific intensity, otherwise the algorithm might face problems in detecting the circle.
- The mechanical pusher had to be synchronized with the production line so that it moved exactly when the tablet was in front of the pusher. Otherwise the tablet would not have been pushed off completely.
- The thresholding had to be adjusted so that no false circles are detected.
- The issue of friction between the production line and the mechanical pusher had to be dealt with. The production line had to be smooth enough to facilitate the free movement of the mechanical pusher over the production line.

6. Recommendation for Future Work

This technology is still being modified regularly and the following improvements can be made in the future. The Medicine Industry has a vital role to play in the society. This industry is responsible for providing the public with means to address health problems. The nature of their work demands a great need for the quality assurance of the products

- This technique is shape limited. Future work can be done to develop a code that can detect tablets of any shape, not just circles.
- This system can be made more compact, robust and efficient.
- The mechanical pusher can be replaced with a mechanical arm to pick off the faulty packaging off the production line.
- This system can be modified for other types of Quality Control Systems

7. Conclusion

The Medicine Industry has a vital role to play in the society. This industry is responsible for providing the public with means to address health problems. The nature of their work demands a great need for the quality assurance of the products

7.1. Overview

The basic purpose behind undertaking this project was to solve this problem by creating a Quality Control System that detects faulty medicine packaging through Image Processing Techniques. The system will detect any leaflet of medicine that does not meet the set requirements and a mechanical pusher will remove that package from the production line. Most of the set objectives were achieved by the end of this project. The system is able to detect faulty medicine packaging through Image Processing Techniques, will detect any leaflet of medicine that does not meet the set requirements and will remove that package from the production line.

7.2. Object Achieved

Most of the set objectives were achieved by the end of this project. The system is able to detect faulty medicine packaging through Image Processing Techniques, will detect any leaflet of medicine that does not meet the set requirements and will remove that package from the production line.

7.3. Achievements

This project correctly identifies faulty packaging in a given production line. After correctly identifying the faulty product, the project removes that particular product from the production line in order to prevent it from being completely packaged and passed onto the market. In case a faulty packaging is detected by the processing, a mechanical pusher installed at the end of the production line removes the faulty medicine packaging from the production line.

8. Bibliography

[1] DESIGN AND SELECTING THE PROPER CONVEYOR-BELT by Konakalla Naga Sri Ananth1, Vaitla Rakesh2 ANDPothamsettyKasi Visweswarao3

http://www.technicaljournalsonline.com/ijeat/VOL%20IV/IJAET%20VOL%20IV%20ISSUE%20II%20APRIL%20JUNE%202013/Vol%20IV%20Issue%20II%20Article%2012.pdf

[2] Development of a lemon sorting system based on color and size by M. Khojastehnazhand, M. Omid* AND A. Tabatabaeefar.

http://www.academicjournals.org/article/article1380110185_Khojastehnazhand%20et%20al.pdf

[3] Automatic color object sorting system by Shubhangi Wanve1, B.G.Gawalwad2, 1E&TC Department, SVIT Chincholi, Nasik, shubhangi16_wanve@yahoo.co.in 2 E&TC Department, SVIT Chincholi, Nasik, balaji_gawalwad@rediffmail.com

http://www.ijmter.com/published_special_issues/07-02-2015/automatic-color-object-sorting-system.pdf

[4] AUTOMATIC DEFECT DETECTION ANDCLASSIFICATION TECHNIQUE FROMIMAGE: A SPECIAL CASE USING CERAMIC TILES by

G. M. Atiq ur Rahaman

Computer Science and Engineering Discipline

Khulna University

Khulna 9208, Bangladesh

atiq99@hotmail.com and

Md. Mobarak Hossain

Computer Science and Engineering Department

Asian University of Bangladesh

Dhaka, Bangladesh

mobarak10jan@yahoo.com

https://www.google.com.pk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&sqi=2&ved=0CCAQFjAA&url=http%3A%2F%2Farxiv.org%2Fpdf%2F0906.3770&ei=t7YJVc6UAYvdUfemgJAL&usg=AFQjCNFT-X8AJZ7nFxf45CvpMbARoZGuPA&sig2=o5wVM7eg_6p9Fzno9j2FMA&bvm=bv.88198703,d.d24

[5] Surface Defects Detection for Ceramic Tiles Using Image Processing and Morphological Techniques by H. Elbehiery, A. Hefnawy, and M. Elewa
https://www.google.com.pk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&sqi=2&ved=0CCkQFjAB&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.135.1354%26rep%3Drep1%26type%3Dpdf&ei=t7YJVc6UAYvdUfemgJAL&usg=AFQjCNFjFdtc29KFXxN4oowGJrD9R2-byQ&sig2=R8ZiYpn6q9BncSgnu33r4g&bvm=bv.88198703,d.d24

[6] Application of Color Based Ore Sorting through Image Processing by Aiyush Suhasaria1 and Khanindra Pathak2
http://www.infomine.com/publications/docs/Suhasaria2012.pdf

[7] V. (Ed.). (n.d.). OpenCV vs. MATLAB. Retrieved November 25, 2015, from http://vaaiibhav.me/opencv-vs-matlab/

[8] Thakkar, K. (2012, November 21). *Me*. Retrieved November 25, 2015, from https://karanjthakkar.wordpress.com/tag/opencv-vs-matlab/

[9] Thakkar, K. (2012, November 21). Retrieved November 25, 2015, from https://karanjthakkar.wordpress.com/tag/opencv-vs-matlab/

9. Appendix A

Quality Control System

Extended Title: Automated Quality Control System for Detection of Faulty Medicine Packaging.
Brief Description of The Project / Thesis with Salient Specs: This project will basically detect faulty medicine leaflets. Once a leaflet, on the conveyor belt is directly below the camera, a signal will be sent to the micro-controller to put the conveyer belt to a halt. An image of the leaflet will then be captured by the camera and if the leaflet is determined to be faulty a signal will be sent to the mechanical pusher to remove that particular leaflet from the conveyor belt.
Scope of Work: Algorithms will be used to detect faulty packaging system. Image processing will play a major role as it will be used to maintain a standard of quality at the production line.
Academic Objectives: Put our theoretical knowledge to practical use. Polish our skill in the field of image processing and embedded system design.
Application / End Goal Objectives: To successfully detect a faulty leaflet, remove it from the production line, send acceptance/rejection data to the processing board.
Previous Work Done on The Subject: This technology is already implemented in other countries such as USA, UK and UAE. BESE - 14 has also done some work on Quality Control.
Material Resources Required: Raspberry Camera Raspberry Pi 2 Model B+ Conveyor belt Mechanical Pusher
No of Students Required: 3
Special Skills Required : Signal processing, Image processing, Programming, Networking and Designing of hardware

10. Appendix B

Timeline

S. No.	Date	Task	Status
1	April '15	Project Synopsis	Completed
2	June '15	Project Defence	Completed
3	June – Sept '15	Literature Research Finalization of approach used Assembling resources	Completed
4	Sept – Oct '15	Algorithm testing Completing the functioning of Conveyor Belt Learning Image Processing Techniques	Completed
5	Oct – Jan '16	Finalizing the implementation of code Interfacing the Camera and Conveyor belt	Completed
6	Jan – Feb '16	Debugging the code	Completed
7	Mar – May '16	Testing of the Final Hardware, Development of pusher Final Project Display	Completed

Table 2. Timeline

11. Appendix C

Cost Breakdown

Items	Cost
Rasoberry Pi	5,500
Raspberry Pi Camera	3,500
HDMI cable	800
Circuit Components	1,500
Stepper Motor and Mechanical Pusher	2,000
AC Adapters and Driving Motors	2,000
Lights	100
Total	**15,400**

Table 3. Cost Breakdown

12. Appendix D

Demonstration Outline

Since this is an application based project and to get the best possible results this need to work in a closed environment. Therefore to demonstrate this project system will be set up indoor, where one group member will place the pharmaceutical tablets on the conveyor belt and real-time processing will be shown.

13. Appendix E