

1

## ĐẠI HỌC BÁCH KHOA HÀ NỘI

## TRƯỜNG CƠ KHÍ

**~~~~~~\*~~~~~~**

BÀI TẬP LỚN MÔN HỌC

XỬ LÝ ẢNH TRONG CƠ ĐIỆN TỬ

**ĐỀ TÀI*:* ỨNG DỤNG XỬ LÍ ẢNH TRONG PHÂN LOẠI NẮP CHAI**

|  |  |  |
| --- | --- | --- |
| **Giáo viên hướng dẫn:** | **TS. Nguyễn Thành Hùng** |  |
| **Danh sách sinh viên** | **Lớp** | **Mã số sinh viên** |
| 1. **Bùi Phạm Sơn Hà** 2. **Phạm Duy Tường** 3. **Nguyễn Long Nhật** | **Cơ điện tử 01-K63**  **Cơ điện tử 01-K63**  **Cơ điện tử 02-K63** | **20184655**  **20184666**  **20184570** |
|  |  |  |
| **Mã lớp** | **124349** |  |

|  |  |
| --- | --- |
| **Nhiệm vụ** | **Người thực hiện** |
| Phân tích, lựa chọn đề tài | Nguyễn Long Nhật, Bùi Phạm Sơn Hà, Phạm Duy Tường |
| Code cho vi điều khiển và thiết kế giao diện app | Nguyễn Long Nhật, Bùi Phạm Sơn Hà, Phạm Duy Tường |
| Thiết kế cơ khí | Nguyễn Long Nhật, Bùi Phạm Sơn Hà, Phạm Duy Tường |
| Làm báo cáo | Nguyễn Long Nhật, Bùi Phạm Sơn Hà, Phạm Duy Tường |

NHIỆM VỤ CỦA THÀNH VIÊN TRONG ĐỀ TÀI

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

Ngày tháng năm

Giảng viên hướng dẫn

(kí tên)

**MỤC LỤC**

[CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI 5](#_bookmark2)

[1.1. Tổng quan đề tài 5](#_bookmark3)

[1.2. Mục tiêu 5](#_bookmark3)

[1.3. Nội dung bài tập lớn 5](#_bookmark3)

[1.4. Giới hạn 6](#_bookmark3)

[CHƯƠNG 2.CƠ SỞ LÝ THUYẾT…. 6](#_bookmark5)

* 1. [Thư viện OPENCV 6](#_bookmark6)
  2. [Các vấn đề trong xử lý ảnh 6](#_bookmark8)

2.2.1. [Ảnh tĩnh và ảnh động 6](#_bookmark8)

2.2.2. [Các Bước cơ bản trong việc xử lý ảnh 7](#_bookmark8)

* 1. [Các thuật toán áp dụng vào đề tài 7](#_bookmark8)

2.3.1. [Lọc ảnh 7](#_bookmark8)

2.3.2. [Chuyển đổi các hệ màu trong OpenCV 8](#_bookmark8)

2.3.3. [Thuật toán phát hiện đường thẳng với Hough Circle 8](#_bookmark8)

* 1. [Các thiết bị 10](#_bookmark8)

2.4.1. [Arduino 10](#_bookmark8)

2.4.2. [Mạch cầu L298N 10](#_bookmark8)

2.4.3. [Camera 11](#_bookmark8)

[2.4.4. Hình ảnh về mô hình hệ thống phân loại 12](#_bookmark8)

[CHƯƠNG 3: THỰC HIỆN ĐỀ TÀI 13](#_bookmark13)

[3.1. .Môi trường thực hiện đề tài 13](#_bookmark8)

[3.2. Chụp ảnh từ camera 13](#_bookmark8)

[3.2.2. Nhận dạng màu sắc nắp chai 14](#_bookmark8)

[3.2.3. Sử dụng HoughCircle tìm các đường tròn 15](#_bookmark8)

[3.2.4. Lập trình giao diện điều khiển các thiết bị qua Arduino 18](#_bookmark8)

[CHƯƠNG 4: CÁC KẾT QUẢ ĐẠT ĐƯỢC 20](#_bookmark15)

[CHƯƠNG 5: KẾT LUẬN 21](#_bookmark15)

**CHƯƠNG I. GIỚI THIỆU ĐỀ TÀI**

* 1. **Tổng quan đề tài**

**Hệ thống phân loại** là một giải pháp công nghiệp thay thế con người thực hiện các khâu **phân loại sản phẩm**, từ thực hiện bằng tay chuyển đổi sang sử dụng các **hệ thống** tự động hóa để **phân** chia **sản phẩm** theo từng đặc tính mà người sử dụng quy định.

Hệ thống phân loại sản phẩm là hệ thống điều khiển tự động hoặc bán tự động nhằm chia sản phẩm ra các nhóm có cùng thuộc tính với nhau để thực hiện đóng gói hay loại bỏ sản phẩm hỏng.

Có nhiều cách phân loại hệ thống phân loại sản phẩm. Ví dụ:

•Dựa trên phương thức điều khiển chia ra hệ thống tự động hay bán tự động, có sự tham gia của con người hay không mức độ đến đâu, điều khiển bằng PLC, vi xử lí.

•Theo màu sắc: màu sắc sẽ được cảm biến màu nhận biết chuyển sang tín hiệu điện rồi qua bộ chuyển đổi ADC về bộ xử lí.

•Theo trọng lượng, kích thước bên ngoài.

Ngoài ra còn nhiều cách phân loại khác tùy vào yêu cầu và sự khác biệt của phôi với nhau.

Ứng dụng của hệ thống phân loại sản phẩm trong thực tế (Máy tách màu)

**1.2 Mục tiêu**

Nhận diện nắp chai (hình tròn) từ đó phân loại chúng theo màu sắc. Dựa trên ngôn ngữ C++ với thư viện chính là OpenCV thực hiện trên Arduino.

**1.3 Nội dung bài tập lớn**

Đề tài của bài tập lớn có những nội dung sau:

* Nội dung 1: Tìm hiểu về Arduino và lập trình Visual Studio
* Nội dung 2: Tổng quan về xử lý ảnh về các thuật toán sử dụng
* Nội dung 3: Viết chương trình và chạy thử nghiệm

**1.4 Giới hạn**

Hình ảnh từ Camera không được ổn định. Ở điều kiện ánh sáng khác nhau, khả năng nhận diện màu cũng kém đi.

**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

**2.1 Thư viện OpenCV**

OpenCV (Open Computer Vision) là một thư viện mã nguồn mở hàng đầu cho xử lý về thị giác máy tính, machine learning, xử lý ảnh. OpenCV đươc viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực. Opencv có các interface cho C/C++, Python Java vì vậy hỗ trợ được cho Window, Linux, MacOs lẫn Android, iOS.

Ứng dụng của thư viện OpenCV:

* Nhận dạng ảnh
* Xử lý hình ảnh
* Phục hồi hình ảnh/video
* Thực tế ảo
* Các ứng dụng khác

**2.2 Các vấn đề trong xử lý ảnh**

**2.2.1 Ảnh tĩnh và ảnh động**

Ảnh tĩnh (Still Image): Biểu diễn bởi hàm độ chói của các biến toạ độ trong mặt phẳng ảnh I(x,y).

Ảnh nhị phân: 1 bit/pixel

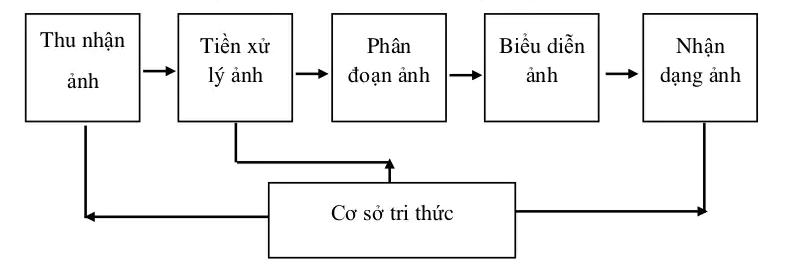
• Ảnh xám: 8 bits/pixel

• Ảnh màu: 16-24 bit/pixel

• RGB, YUV, HSL, YCbCr

Chuỗi ảnh (Sequence of Images): Hàm độ chói của các biến tọa độ mặt phẳng và biến thời gian I(x,y,t). Chuỗi các ảnh (khung hình), quan hệ thời gian giữa các khung hình biểu diễn ảnh động.

**2.2.2 Các bước cơ bản trong việc xử lý ảnh**



**2.3 Các thuật toán áp dụng vào đề tài**

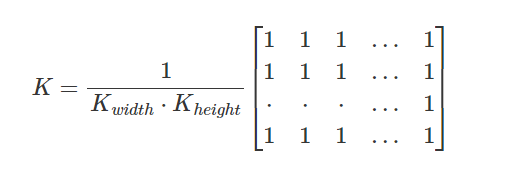
**2.3.1 Lọc ảnh**

Nguyên tắc chung của các phương pháp lọc là cho ma trận ảnh nhân với một ma trận lọc (Kernel). Ma trận lọc (Kernel) còn có thể được gọi là cửa số chập (trong phép nhân chập), cửa sổ lọc, mặt nạ,…

**1 số bộ lọc ảnh làm mịn ảnh :**

a) Bộ lọc trung bình

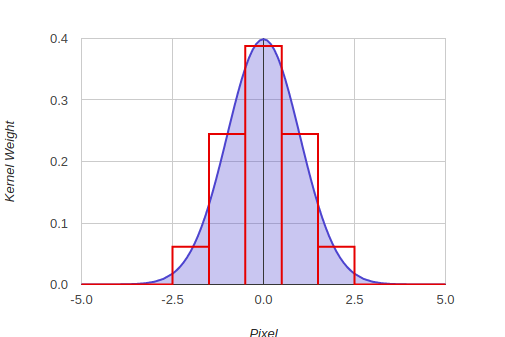
Đây là bộ lọc đơn giản nhất. Nó được xây dựng dựa trên ý tưởng tính giá trị một điểm ảnh bằng trung bình cộng các điểm ảnh xung quanh nó.

Cách lọc này thường được áp dụng cho làm trơn ảnh vẫn muốn giữ lại biên không bị mờ.

Ma trận lọc có dạng

b)Bộ lọc Gauss (Gaussian Filter)

Bộ lọc Gauss được cho là bộ lọc hữu ích nhất, được thực hiện bằng cách nhân chập ảnh đầu vào với một ma trận lọc Gauss sau đó cộng chúng lại để tạo thành ảnh đầu ra.

Ý tưởng chung là giá trị mỗi điểm ảnh sẽ phụ thuộc nhiều vào các điểm ảnh ở gần hơn là các điểm ảnh ở xa. Trọng số của sự phụ thuộc được lấy theo hàm Gauss (cũng được sử dụng trong quy luật phân phối chuẩn).

**2.3.2 Chuyển đổi các hệ màu trong OpenCV**

Không gian màu (Colorspaces) là một mô hình toán học biểu diễn các màu sắc trong thực tế dưới dạng số học. Có rất nhiều không gian màu: BGR, BGRA, HSV, CMYK,… Trong đó mỗi chữ cái đại diện cho 1 kênh màu (channel): B: Blue, G: Green, R: Red, H: Hue, S: Saturation, V: Value, A: Alpha.

OpenCV hỗ trợ phép chuyển đổi ảnh giữa một số cặp không gian màu bằng hàm cvtColor():

Thông thường ảnh được biểu diễn trong không gian màu BGR, nhưng không gian màu HSV lại dễ dàng xử lý hơn. Dãy màu của kênh Hue là “bảy sắc cầu vồng” từ đỏ đến tím rồi lại về đỏ, nên có thể xác định được ngay nó là màu gì. Do đó trong hầu hết bài toán đều phải tiến hành chuyển ảnh ban đầu sang không gian màu HSV trước khi xử lý những bước tiếp theo.

**2.3.3 Thuật toán phát hiện đường thẳng với Hough Circle**

Ta quy định những vật có hình tròn mới là nắp chai còn lại ta cho là sản phẩm khác.

Chuyển đổi hàm Hough Circle:

Ta có phương trình đường tròn được xác định bởi:

Diagram, schematic

Description automatically generated

Trong đó, *(u,v)* là tâm đường tròn, *R* là bán kính đường tròn, *θ* là góc có giá trị từ 0 tới  
360 độ. Một đường tròn sẽ hoàn toàn được xác định nếu ta biết được bộ ba thông số  
(*u,v,R*). Từ phương trình trên ta có thể chuyển đổi tương đương

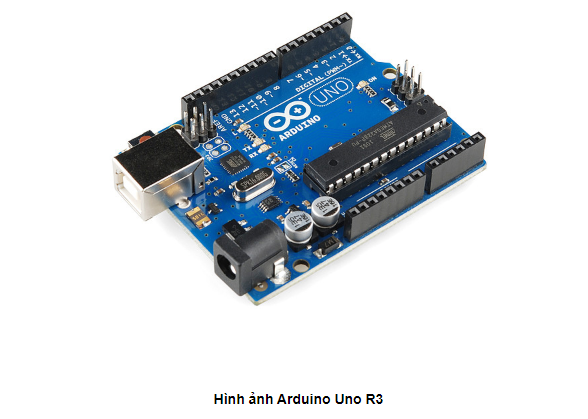
Text, letter

Description automatically generated  
Ta xét với trường hợp đã biết trước giá trị của R. Khi đó, với mỗi điểm ảnh (x,y) ta sẽ xác  
định được một giá trị (u,v) và lưu nó vào một mảng. Tâm của đường tròn sẽ là giá trị xuấ  
hiện trong mảng với tần suất lớn nhất. Trong trường hợp R chưa biết, ta tăng giá trị của Rtừ một ngưỡng min tới ngưỡng max nào đó và tiến hành như với trường hợp đã biết trước  
giá trị R

**2.4. Các thiết bị**

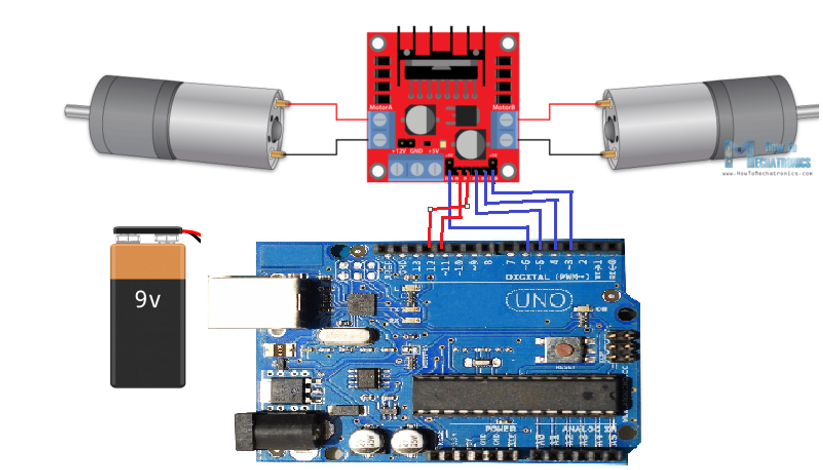
**2.4.1. Arduino**

Arduino Uno là một bo mạch vi điều khiển dựa trên chip Atmega328P. Uno có 14 chân I/O digital ( trong đó có 6 chân xuất xung PWM), 6 chân Input analog, 1 thạch anh 16MHz, 1 cổng USB, 1 jack nguồn DC, 1 nút reset.



**2.4.2. Mạch cầu L298N**

Module điều khiển động cơ (Motor Driver) sử dụng chip cầu H , L298N giúp điều khiển tốc độ và chiều quay của động cơ DC một cách dễ dàng, ngoài ra module L298N còn điều khiển được 1 động cơ bước lưỡng cực. Mạch cầu H của IC L298N có thể hoạt động ở điện áp từ 5V đến 12V.



**2.4.3 Cảm biến hồng ngoại**

*Thông số kỹ thuật IC so sánh: LM393*

* Điện áp: 3.3V - 6VDC
* Dòng tiêu thụ:
  + Vcc = 3.3V: 23 mA
  + Vcc = 5.0V: 43 mA
* Góc hoạt động: 35°
* Khoảng cách phát hiện: 2 ~ 30 cm
* LED báo nguồn và LED báo tín hiệu ngõ ra
* Mức logic ngõ ra:
  + Mức thấp - 0V: khi có vật cản
  + Mức cao - 5V: khi không có vật cản
* Kích thước: 3.2cm x 1.4cm



**2.4.4 Động cơ servo SR90**

Sử dụng trong việc gạt nắp chai vào đúng vị trí

**2.4.4 Camera**

Sử dụng camera mặt sau 16MP của SamsungA5

Hình ảnh được truyền đến máy tính thông qua kết nối IP được hỗ trợ thông qua phần mềm DroidCam



**2.4.5 Hình ảnh về mô hình hệ thống phân loại sản phẩm theo màu sắc**

Diagram

Description automatically generated with low confidence

**CHƯƠNG 3: THỰC HÀNH ĐỀ TÀI**

**3.1 Môi trường thực hiện đề tài**

Môi trường để ứng dụng thực hiện đề tài được tạo ra là 1 băng tải có nền màu xanh. Ánh sáng được cung cấp từ đèn được bố trí gần camera để hỗ trợ việc chụp ảnh

Graphical user interface, text, application

Description automatically generated

**3.2 Lập trình phát hiện, phân loại nắp chai**

**3.2.1 Chụp ảnh từ camera**

A picture containing graphical user interface

Description automatically generatedKhi vật qua cảm biến hồng ngoại, máy tính tự động chụp lại 1 ảnh chứa vật để máy tính thực hiện các công đoạn nhận diện tiếp theo

**3.2.2 Lọc ảnh nhận diện màu sắc nắp chai**

Chuyển ảnh về dạng HSV rồi lọc màu của Nắp chai qua hàm inrange

Mat HSV;

cvtColor(frame, HSV, COLOR\_BGR2HSV);

imshow("HSV",imgHSV);

Bước 2: Phát hiện đối tượng dựa trên Giá trị phạm vi HSV

// Gam màu HSV của mà đỏ

int iLowH\_R = 160;

int iHighH\_R = 179;

int iLowS\_R = 150;

int iHighS\_R = 255;

int iLowV\_R = 60;

int iHighV\_R = 255;

Mat imgThresholded1;

inRange(imgHSV, Scalar(iLowH\_R, iLowS\_R, iLowV\_R), Scalar(iHighH\_R, iHighS\_R, iHighV\_R), imgThresholded1); //Threshold the image

Icon

Description automatically generated with medium confidenceA picture containing square

Description automatically generated

Hình ảnh khi chuyển hệ màu HSV Hình ảnh khi qua bộ lọc inRange

**3.2.4 Sử dụng hàm Hough Circle tìm đường tròn**

1. *Sử dụng bộ lọc trung vị để lọc cái nhiễu nhỏ và trung bình của ảnh*

medianBlur(imgThresholded1,imgThresholded1, 5)

Icon

Description automatically generated with medium confidenceA picture containing application

Description automatically generated

1. *Tìm và vẽ viền cho vật được phát hiện*

void FilteredObject(Mat& threshold, Mat& cameraFeed, Scalar& colors, string& display) // Hàm tìm và vẽ viền cho vật được phát hiện

{

vector< vector<Point> > contours;

vector<Vec4i> hierarchy;

findContours(threshold, contours, hierarchy, RETR\_TREE, CHAIN\_APPROX\_SIMPLE);

//use moments method to find our filtered object

bool objectFound = false;

if (hierarchy.size() > 0) {

int numObjects = hierarchy.size();

//if number of objects greater than MAX\_NUM\_OBJECTS we have a noisy filter

if (numObjects < 1000) {

for (int index = 0; index >= 0; index = hierarchy[index][0]) {

Moments moment = moments((cv::Mat)contours[index]);

double dM01 = moment.m01;

double dM10 = moment.m10;

double dArea = moment.m00;

if (dArea > 10000) {

int posX = dM10 / dArea;

int posY = dM01 / dArea;

drawContours(cameraFeed, contours, -1, colors, 1.8);

putText(cameraFeed, display, Point(posX, posY + 20), 1, 1, Scalar(0,0,0), 2);

}

else objectFound = false;

}

}

else putText(cameraFeed, "TOO MUCH NOISE! ADJUST FILTER", Point(0, 50), 1, 2, Scalar(0, 0, 255), 2);

}

}

//Tìm và vẽ viền cho ảnh màu đầu ra

Scalar colors1 = Scalar(220, 240, 60)

Scalar colors2 = Scalar(220, 240, 120);

string name\_red = "RED";

FilteredObject(imgThresholded1, imgOriginal, colors1, name\_red);

1. *Phát hiện đường tròn bao gồm tọa độ tâm và bán kính*

void CircleObject(Mat& threshold, Mat& cameraFeed, Scalar& colors)// tìm đường tròn

{

std::vector<cv::Vec3f> circles;

HoughCircles(threshold, circles, cv::HOUGH\_GRADIENT, 1.5, threshold.rows, 100, 50, 50, 300); // algorithm for detecting circles

for (int i = 0; i < circles.size(); i++) {

std::cout << "Ball position X = " << circles[i][0] << ",\tY = " << circles[i][1] << ",\tRadius = " << circles[i][2] << "\n";

cv::circle(cameraFeed, cv::Point((int)circles[i][0], (int)circles[i][1]), 3, colors, cv::FILLED);

cv::circle(cameraFeed, cv::Point((int)circles[i][0], (int)circles[i][1]), (int)circles[i][2], colors, 2);

putText(cameraFeed, "Detection Circle", cv::Point(20, 20), 1, 1, colors, 2);

putText(cameraFeed, "0" + std::to\_string(i + 1) + "(" + std::to\_string((int)circles[i][0]) + "," + std::to\_string((int)circles[i][1]) + ")", cv::Point(circles[i][0] - 5, circles[i][1] - 20), 1, 1, colors, 2);

putText(cameraFeed, "R=" + std::to\_string((int)circles[i][2]), cv::Point(circles[i][0], circles[i][1] - 35), 1, 1, colors, 2);

}

}

// Gọi hàm với các tham số đã cho

CircleObject(imgThresholded1, imgOriginal, colors2);

Phát hiện và vẽ đường tròn bao gồm tọa độ tâm và bán kính

imshow("imgOriginal", imgOriginal);

Graphical user interface

Description automatically generated

1. *Lưu ảnh vào máy tính mỗi khi máy tính có tín hiệu chụp ảnh*

// Hàm lưu ảnh đã xử lý vào ổ F, thư mục final

void imageWrite(const cv::Mat& image, const std::string filename)

{

// Support for writing JPG

vector<int> compression\_params;

compression\_params.push\_back(IMWRITE\_JPEG\_QUALITY);

compression\_params.push\_back(100);

// This writes to the specified path

std::string path = "F:\\final\\" + filename + ".jpg";

cv::imwrite(path, image, compression\_params);

}

int dem = 0;

imageWrite(imgOriginal, to\_string(dem));

for (int i = 0; i < 1; i++) { dem++; }// gán địa chỉ dạng số tự nhiên và tăng dần khi có thêm ảnh được lưu

Graphical user interface, application

Description automatically generated

1. *Lập trình giao diện điều khiển từ WinForm*

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated

**3.2.6 Điều khiển động cơ servo**

Sau khi lập trình xử lí ảnh và gửi phát tín hiệu lên giao diện người dùng qua cổng Serial, ta sẽ điều khiển động cơ servo để nắp rơi vào đúng vị trí ta mong muốn

**Nếu màu đỏ truyền tín hiệu “r”**

std::string name\_red = "RED";

String^ port\_red = "r";

blur(imgThresholded1, imgThresholded1, cv::Size(3, 3));

FilteredObject(imgThresholded1, imgOriginal, colors1, name\_red);

CircleObject(imgThresholded1, imgOriginal, colors2, count\_red,port\_red);

labelcount\_red->Text = "" + count\_red;

//imshow("Thresholded Image1", imgThresholded1); //show the thresholded image

**Nếu màu cam truyền tín hiệu “o”**

std::string name\_orange = "ORANGE";

String^ port\_orange = "o";

blur(imgThresholded2, imgThresholded2, cv::Size(5, 5));

FilteredObject(imgThresholded2, imgOriginal, colors1, name\_orange);

CircleObject(imgThresholded2, imgOriginal, colors2, count\_orange,port\_orange);

labelcount\_orange->Text = "" + count\_orange;

// imshow("Thresholded Image2", imgThresholded2); //show the thresholded image

**Nếu màu xanh truyền tín hiệu “g”**

std::string name\_green = "GREEN";

String^ port\_green = "g";

blur(imgThresholded3, imgThresholded3, cv::Size(5, 5));

FilteredObject(imgThresholded3, imgOriginal, colors1, name\_green);

CircleObject(imgThresholded3, imgOriginal, colors2, count\_green,port\_green);

labelcount\_green->Text = "" + count\_green;

// imshow("Thresholded Image 3", imgThresholded3); //show the

**Adruino điều khiển servo quay góc từ 0 đến 90 tùy theo tín hiệu từ winform truyền xuống**

#include <Servo.h>

Servo myservo\_red;

Servo myservo\_orange;

Servo myservo\_green;

void setup()

{

Serial.begin(9600);

Myservo\_red.attach(9);

Myservo\_orange.attach(8);

Myservo\_green.attach(7);

}

void loop()

{

if (Serial.available())

{

int serialValue = Serial.read();// Đọc giá trị từ cổng Serial

switch (serialValue){

case 'r'://nhận dự liệu từ nút nhấn UNLOCK trên Form1 để mở khóa cửa

myservo\_red.write(0);

delay(5000);

myservo\_red.write(90);

break;

case 'o':

myservo\_orange.write(0);

delay(5000);

myservo\_orange.write(90);

break;

break;

case 'g'

myservo\_orange.write(90);

delay(50);

break;

}

}

delay(200); }

**CHƯƠNG 4: CÁC KẾT QUẢ ĐẠT ĐƯỢC**

Do thời gian tìm hiểu có giới hạn, nên trong quá trình tìm hiểu nhóm em đã gặp phải một số khó khăn nhất định. Nhưng với sự tìm hiểu và sự hướng dẫn nhiệt tìm của thầy giáo đã giúp cho nhóm em đạt được một số kết quả sau:

* + - Biết được tình hình sử dụng, xu hướng phát triển của việc áp dụng xử lý ảnh vào thực tiễn áp dụng thư viện OpenCV.
    - Thiết kế tương đối hoàn thiện mô hình hệ thống phân loại sản phẩm
    - Hiểu được nguyên lý làm việc của các module trong hệ thống và cách ghép nối chúng như thế nào.
    - Tiến hành chạy thử nghiệm và đánh giá kết quả.
    - Rèn luyện kỹ năng làm việc nhóm một cách có hiệu quả .

Bên cạnh những kết quả đạt được thì mô hình vẫn tồn tại nhiều hạn chế về kĩ thuật như khả năng kết nối, xảy ra nhiễu , thiết kế cơ khí vẫn chưa tối ưu, việc tín hiệu truyền đến vẫn gặp vấn đề, khả năng quay góc chính xác của servo cũng như tốc độ của bang tải chưa được ổn định,..

## CHƯƠNG 5: KẾT LUẬN

Sau khi hoàn thành đề tài, sản phẩm đã đạt được những yêu cầu đã đề ra trước đó. Nhưng so với những sản phẩm trên thị trường còn rất nhiều hạn chế, việc hoàn thiện được sản phẩm cần có nhiều thời gian hơn để nghiên cứu. Qua đề tài này em mong muốn sử dụng kiến thức học được trong môn học để thực hiện việc tiếp cận với công nghệ và xu hướng của thế giới.

Dự định sắp tới của nhóm em là hoàn thiện thêm sản phẩm, nâng cấp cho việc phân loại bằng robot gắp thay vì servo, nhận diện được các mẫu sản phẩm khác nhau. Từ đó tạo lên các bước phát triển đột phá trong nhiều lĩnh vực khác nhau của nhân loại.