

BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**🙡🕮🙣**



**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN**

**Đề tài: Nghiên cứu thiết kế phần mềm nhận dạng, giám sát sản phẩm qua camera.**

SV thực hiện : **Nguyễn Huy Hoàng**

Phạm Quang Bền

Trần Minh Hiếu

Lớp,khoa: TĐH3 K11 – Khoa Điện

GV hướng dẫn: Th.s Nguyễn Đăng Khang

**Hà nội,** 05/2020

# 

BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**🙡🕮🙣**



**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN**

**Đề tài: Nghiên cứu thiết kế phần mềm nhận dạng, giám sát sản phẩm qua camera.**

SV thực hiện : **Nguyễn Huy Hoàng;** Giới tính: Nam; Dân tộc: Kinh

Phạm Quang Bền;Giới tính: Nam; Dân tộc: Kinh

Trần Minh Hiếu; Giới tính: Nam; Dân tộc: Kinh

Lớp,khoa: TĐH3 K11 – Khoa Điện. Năm thứ: 4

Ngành học: Điều khiển và tự động hóa

GV hướng dẫn: Th.s Nguyễn Đăng Khang

**Hà nội,** 05/2020

# Mục lục

[Danh mục hình vẽ 6](#_Toc41807044)

[Danh mục bảng biểu 8](#_Toc41807045)

[Danh mục những từ viết tắt 8](#_Toc41807046)

[LỜI NÓI ĐẦU 9](#_Toc41807047)

[MỞ ĐẦU 10](#_Toc41807048)

[CHƯƠNG 1. TỔNG QUAN VỀ HỆ THỐNG NHẬN DẠNG SẢN PHẨM QUA CAMERA 12](#_Toc41807049)

[1.1. Các phương pháp nhận dạng sản phẩm 12](#_Toc41807050)

[1.2. Tổng quan về xử lý ảnh nhận dạng sản phẩm 13](#_Toc41807051)

[CHƯƠNG 2. GIỚI THIỆU THIẾT BỊ CAMERA VÀ PHẦN MỀM PYTHON NHẬN DẠNG SẢN PHẨM 35](#_Toc41807052)

[2.1. Giới thiệu về Camera HIKVISION sử dụng trong nhận diện sản phẩm 35](#_Toc41807053)

[2.2. các thông số kỹ thuật của camera Hikvision 35](#_Toc41807054)

[2.3. Các ứng dụng của camera Hikvision 37](#_Toc41807055)

[2.4. Phần mềm Python 37](#_Toc41807056)

[CHƯƠNG 3. THIẾT KẾ PHẦN MỀM GIÁM SÁT NHẬN DẠNG SẢN PHẨM QUA CAMERA. 42](#_Toc41807057)

[3.1. Yêu cầu công nghệ 42](#_Toc41807058)

[3.2. Phần mềm và thuật toán nhận dạng 42](#_Toc41807059)

[3.3. Viết phần mềm 52](#_Toc41807060)

[CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU VÀ PHÂN TÍCH 53](#_Toc41807061)

[4.1. Kết quả nghiên cứu 53](#_Toc41807062)

[4.2. Kết luận 58](#_Toc41807063)

[4.3. Hướng phát triển 58](#_Toc41807064)

[Tài liệu tham khảo 60](#_Toc41807065)

# Danh mục hình vẽ

[Hình 1.1: Cảm biến màu sắc và cảm biến siêu âm 12](#_Toc41809742)

[Hình 1.2:Cấu trúc hệ thống giám sát nhận dạng sản phẩm 12](#_Toc41809743)

[Hình 1.3:Lân cận 4 và lân cận 8 14](#_Toc41809744)

[Hình 1.4:Hình tách biên 16](#_Toc41809745)

[Hình 1.5:Bộ lọc Gaussian 18](#_Toc41809746)

[Hình 1.6: Đặc điểm hình chữ nhật 22](#_Toc41809747)

[Hình 1.7:Đặc điểm hình vuông 23](#_Toc41809748)

[Hình 1.8: Đặc điểm hình tam giác 24](#_Toc41809749)

[Hình 1.9:Sơ đồ các bước thực hiện phân loại sản phẩm 24](#_Toc41809750)

[Hình 1.10:Xác định biên 27](#_Toc41809751)

[Hình 1.11:Nhận dạng đường biên 27](#_Toc41809752)

[Hình 1.12:Thuật toán SAD phân vùng 32](#_Toc41809753)

[Hình 1.13: Mô hình các thông số của camera 33](#_Toc41809754)

[Hình 2.1:Hình dạng camera 35](#_Toc41809755)

[Hình 2.2: Giao diện phần mềm python 38](#_Toc41809756)

[Hình 3.1:Ảnh gốc 42](#_Toc41809757)

[Hình 3.2:cấu trúc hệ thống giám sát nhận dạng sản phẩm qua camera 42](#_Toc41809758)

[Hình 3.3:Qúa trình xử lý ảnh 43](#_Toc41809759)

[Hình 3.4:Chuyển đôi từ ảnh RGB sang ảnh xám 46](#_Toc41809760)

[Hình 3.5:chuyển đổi ảnh RGB sang ảnh đen trắng 46](#_Toc41809761)

[Hình 3.6:Lọc nhiễu kiểu Opening 47](#_Toc41809762)

[Hình 3.7:Lọc nhiễu kiểu closing 47](#_Toc41809763)

[Hình 3.8:Vẽ contours quanh hình chữ nhật 48](#_Toc41809764)

[Hình 3.9:Vẽ đường tròn nhỏ nhất 48](#_Toc41809765)

[Hình 3.10:Vẽ đường tròn biết tâm và bán kính 49](#_Toc41809766)

[Hình 3.11:Xóa nhiễu ảnh 49](#_Toc41809767)

[Hình 3.12:thuật toán tổng quát 50](#_Toc41809768)

[Hình 3.13:thuật toán chương trình con di chuyển vật thể 51](#_Toc41809769)

[Hình 3.14:Ảnh giao diện phần mềm 52](#_Toc41809770)

[Hình 4.1:Giao diện truy cập phần mềm pyton 53](#_Toc41809771)

[Hình 4.2:Cửa sổ tìm file ảnh 54](#_Toc41809772)

[Hình 4.3:Hiển thị ảnh gốc 54](#_Toc41809773)

[Hình 4.4:hiển thị số lượng vật thể màu đỏ 55](#_Toc41809774)

[Hình 4.5: xác định số lượng vật thể xanh dương 56](#_Toc41809775)

[Hình 4.6:Đếm số lượng vật thể màu xanh lá cây 56](#_Toc41809776)

[Hình 4.7:Giao diện khi loại bỏ một vật thể số 1 57](#_Toc41809777)

[Hình 4.8:Giao diện khi di chuyển vật thể màu đỏ 58](#_Toc41809778)

[Hình 4.9:Ảnh đếm số lượng tôm giống truyền thống bằng cốc 59](#_Toc41809779)

[Hình 4.10:Ảnh chụp tôm giống 60](#_Toc41809780)

[Hình 4.11:Ảnh gốc tôm giống và ảnh nhận dạng tôm giống qua phần mềm 60](#_Toc41809781)

# Danh mục bảng biểu

[Bảng 1.1: Đặc điểm hình dạng sản phẩm 21](#_Toc41641268)

[Bảng 2.1:Đặc tính kỹ thuật 33](#_Toc41641269)

# Danh mục những từ viết tắt

* AI: Artificial Intelligence
* SAD: Sum of absolute diffirence
* SSD: Sum of squared diffirence
* NCC: Normalized cross correlation
* OpenCV: Open Source Computer Vision Library

# LỜI NÓI ĐẦU

Như Lê-Nin từng nói:”học, học nữa, học mãi” – có lẽ sự học là một quá trình lâu dài và gắn liền với mỗi con người. Xung quanh chúng ta còn rất nhiều điều cần tìm hiểu, còn nhiều thứ đang đợi ta khám phá. Những phát minh, sáng chế của con người đã thay đổi rất lớn cách sống, cách sinh hoạt của con người. Tất cả những thứ đó đều từ nghiên cứu khoa học, sáng tạo của bộ óc con người mà ra. Nghiên cứu khoa học thật sự quan trọng và cần thiết, nó thúc đẩy nền văn minh của con người.

Việc nghiên cứu khoa học trong lĩnh vực xử lý ảnh là một việc làm hết sức cần thiết và có ý nghĩa to lớn. Với những nghiên cứu đi đôi với ứng dụng tốt sẽ làm nên những thay đổi cho cuộc sống của con người, đẩy mạnh và bắt kịp quá trình thông tin hóa đang diễn ra hiện nay của thế giới.

Từ đó, nhóm chúng em đã đi vào tìm hiểu và thực hiện đề tài “Nghiên cứu thiết kế phần mềm nhận dạng, giám sát sản phẩm qua camera”.

Chúng em xin cảm ơn thầy **Nguyễn Đăng Khang** cũng như các thầy cô giáo của khoa đã tạo điều kiện,giúp đỡ để nhóm em có thể hoàn thành đề tài một cách tốt nhất. Bên cạnh đó do còn chưa có nhiều kinh nghiệm thực tế nên đề tài khó tránh khỏi những sai sót.Vậy em rất mong nhận được sự đóng góp ý kiến của các thầy cô giáo cũng như các bạn sinh viên.

Nội dung ngoài việc tham khảo tài liệu, nội dung hoàn toàn mang quan điểm của cá nhân, viết theo văn phong cá nhân. Do đó sẽ không thể tránh khỏi những sai sót. Rất mong thầy và các bạn thông cảm.

Chúng em xin chân thành cảm ơn

.

# MỞ ĐẦU

Trên thế giới , cùng với sự phát triển ngày càng mạnh mẽ của khoa học kĩ thuật trong một vài thập kỷ gần đây, xử lý ảnh tuy là một ngành khoa học còn tương đối mới mẻ so với nhiều ngành khoa học khác nhưng hiện nay đang là một trong những lĩnh vực phát triển rất nhanh và thu hút sự quan tâm đặc biệt từ các nhà khoa học, thúc đẩy các trung tâm nghiên cứu, ứng dụng về lĩnh vực hấp dẫn này.

Xử lý ảnh đóng vai trò quan trọng trong nhiều ứng dụng thực tế về khoa học kĩ thuật cũng như trong cuộc sống.

Trong nước, qua tìm hiểu có thể nhận thấy rằng việc nghiên cứu khoa học về lĩnh vực xử lý ảnh được ứng dụng rất rộng rãi trong đời sống như photoshop, nén ảnh, nén video, nhận dạng biển số xe, nhận dạng khuôn mặt, ảnh y tế...

Xử lý ảnh là một lĩnh vực đang được quan tâm và đã trở thành môn học chuyên ngành của sinh viên ngành tự động hóa, công nghệ thông tin trong nhiều trường đại học trên cả nước.

Trong nền công nghiệp 4.0 việc ứng dụng công nghệ thông tin trong lĩnh vực tự động hóa là xu thế tất yếu. Tuy một số hãng chế tạo các cảm biến chuyên dụng nhận dạng sản phẩm theo màu sắc hoặc kích thước, tuy nhiên tính mở cho các ứng dụng khác bị hạn chế, giá thành cao. Việc nhận dạng sản phẩm theo màu sắc, kích thước, hình dáng có ý nghĩa thực tiễn cao, linh hoạt, mềm dẻo. Vì vậy việc thiết kế một phần mềm có khả năng nhận dạng được sản phẩm theo màu sắc, kích thước, hình dáng là rất cần thiết trong thực tế.

Nắm bắt được tầm quan trọng này, cùng với sự đam mê nghiên cứu khoa học, nhóm bốn sinh viên lớp Tự động hóa khóa 11, Khoa Điện, Trường Đại học Công nghiệp Hà nội đã đi vào tìm hiểu và thực hiện đề tài “Nghiên cứu thiết kế phần mềm nhận dạng, giám sát sản phẩm qua camera”.

Mục tiêu đề tài thiết kế được phẩn mềm là nhận dạng được kích thước và màu sắc của sản phẩm thông qua việc xử lý ảnh qua camera bằng phần mềm nhận dạng, ứng dụng được vào thực tế trong nhiều lĩnh vực như đời sống, sản xuất...

Phương pháp nghiên cứu: Nhóm nghiên cứu đã sử dụng các phương pháp nghiên cứu như: nghiên cứu lý thuyết, phân tích, tổng hợp và cuối cùng sử dụng phương pháp mô phỏng thực nghiệm để kiểm chứng kết quả.

Đối tượng, phạm vi nghiên cứu: Đối tượng nghiên cứu của đề tài là các sản phẩm công nghiệp hoặc các hình ảnh vật thể, giới hạn trong phạm vi nghiên cứu lý thuyết kết hợp mô phỏng thực nghiệm.

# TỔNG QUAN VỀ HỆ THỐNG NHẬN DẠNG SẢN PHẨM QUA CAMERA

## Các phương pháp nhận dạng sản phẩm

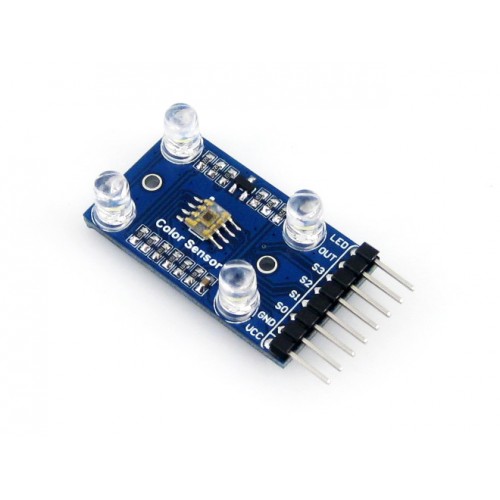
* Các tiêu chí nhận dạng sản phẩm:
  + Nhận dạng qua màu sắc
  + Nhận dạng qua kích thước, hình dạng

- Các phương pháp nhận dạng sản phẩm:

* Nhận dạng dùng cảm biến: cảm biến màu sắc, CB quang . Một số hãng chế tạo các cảm biến chuyên dụng nhận dạng sản phẩm theo màu sắc hoặc kích thước, tuy nhiên tính mở cho các ứng dụng khác bị hạn chế, giá thành cao.







.

.

.

.

Hình. .

Hình 1.1: Cảm biến màu sắc và cảm biến siêu âm

* Nhận dạng sản phẩm dùng camera: Đặc điểm phương pháp này có tính mở rất cao, đa dạng có khả năng nhận dạng được nhiều loại sản phẩm khác nhau, có khả năng áp dụng các công nghệ AI để nhận dạng, tuy nhiên cần kết hợp phần mềm phân tích, xử lý ảnh.







Vật thể, sản phẩm

Hình 1.2:Cấu trúc hệ thống giám sát nhận dạng sản phẩm

## Tổng quan về xử lý ảnh nhận dạng sản phẩm

### Giới thiệu xử lý ảnh

Xử lý ảnh không còn là đề tài quá mới, nó được áp dụng từ trong các hoạt động thường ngày cho đến việc nâng cao sản xuất. Nó giúp ích cho cá nhân hay gia đình, mà còn ứng dụng cả trong Chính trị, Y tế, Giáo dục… Xử lý tín hiệu là một môn học trong kỹ thuật điện tử, viễn thông và trong toán học. Liên quan đến nghiên cứu và xử lý tín hiệu kỹ thuật số và analog, giải quyết các vấn đề về lưu trữ, các thành phần bộ lọc, các hoạt động khác trên tín hiệu. Các tín hiệu này bao gồm truyền dẫn tín hiệu, âm thanh hoặc giọng nói, hình ảnh, và các tín hiệu khác… Trong số các phương pháp xử lý tín hiệu kể trên, lĩnh vực giải quyết với các loại tín hiệu mà đầu vào là một hình ảnh và đầu ra cũng là một hình ảnh, sản phẩm đầu ra được thực hiện trong một quá trình xử lý. Đó chính là quá trình xử lý ảnh. Nó có thể được chia thành xử lý hình ảnh tương tự và xử lý hình ảnh kỹ thuật số.

Để có 1 bức ảnh số ta có quá trình thực hiện như sau: Chụp ảnh từ một máy ảnh là một quá trình vật lý. Ánh sáng mặt trời sử dụng như một nguồn năng lượng. Một dãy cảm biến được sử dụng cho việc thu lại của hình ảnh. Vì vậy, khi ánh sáng mặt trời rơi trên người đối tượng, sau đó số lượng ánh sáng phản xạ của đối tượng được cảm nhận từ các cảm biến, và một tín hiệu điện áp liên tục được tạo ra bởi số lượng dữ liệu cảm biến đó. Để tạo ra một hình ảnh kỹ thuật số, chúng ta cần phải chuyển đổi dữ liệu này thành một dạng kỹ thuật số. Điều này liên quan đến việc lấy mẫu và lượng tử hóa. Kết quả của việc lấy mẫu và lượng tử hóa sau một quá trình xử lý là một hình ảnh kỹ thuật số.

### Những vấn đề trong xử lý ảnh

#### Điểm ảnh

Là đơn vị cơ bản nhất để tạo nên một bức ảnh kỹ thuật số, địa chỉ của điểm ảnh được xem như là một tọa độ (x,y) nào đó. Một bức ảnh kỹ thuật số - có thể được tạo ra bằng cách chụp hoặc bằng một phương pháp đồ họa nào khác - được tạo nên từ hàng ngàn hoặc hàng triệu pixel riêng lẻ. Bức ảnh càng chứa nhiều pixel thì càng chi tiết. Một triệu pixel thì tương đương với 1 megapixel.

#### Ảnh số

Ảnh số là tập hợp hữu hạn các điểm ảnh với mức xám phù hợp dùng để mô tả ảnh gần với ảnh thật. Số điểm ảnh xác định độ phân giải của ảnh. Ảnh có độ phân giải càng cao thì càng thể hiện rõ nét các đặt điểm của tấm hình càng làm cho tấm ảnh trở nên thực và sắc nét hơn. Một hình ảnh là một tín hiệu hai chiều. Nó được xác định bởi hàm toán học f(x, y) trong đó x và y là hai tọa độ theo chiều ngang và chiều dọc. Các giá trị của f(x, y) tại bất kỳ điểm nào là cung cấp các giá trị điểm ảnh (pixel ) tại điểm đó của một hình ảnh.

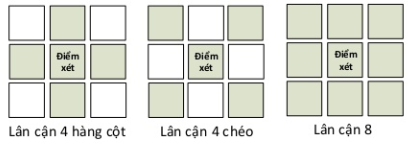
#### Phân loại

ảnh mức xám của điểm ảnh là cường độ sáng, gán bằng một giá trị tại điểm đó. Các mức ảnh xám thông thường: 16, 32, 64, 128, 256. Mức được sử dụng thông dụng nhất là 256, tức là dùng 1byte để biểu diễn mức xám.

Ảnh nhị phân: Là ảnh có 2 mức trắng và đen. Chỉ có 2 giá trị 0 và 1 và chỉ sử dụng 1 bit dữ liệu trên 1 điểm ảnh. Ảnh đen trắng: Là ảnh có hai màu đen, trắng (không chứa màu khác) với mức xám ở các điểm ảnh có thể khác nhau.

Ảnh màu: Là ảnh kết hợp của 3 màu cơ bản lại với nhau để tạo ra một thế giới màu sinh động. Người ta thường dùng 3byte để mô tả mức màu, tức là có khoảng 16,7 triệu mức màu. Quan hệ giữa các điểm ảnh Lân cận điểm ảnh: được nói một cách hài hước như là hàng xóm của các điểm ảnh. Có 2 loại lân cận cơ bản là lân cận 4 và lân cận 8.

6 Điểm xét Điểm xét Điểm xét Lân cận 4 chéoLân cận 4 hàng cột Lân cận 8



Hình .:Lân cận 4 và lân cận 8

Bốn điểm ảnh lân cận 4 theo cột và hàng với tọa độ lần lượt là (x+1, y), (x-1, y), (x,y+1), (x, y-1) ký hiệu là tập N4(p). Bốn điểm ảnh lân cận 4 theo đường chéo có tọa độ lần lượt là (x+1, y+1), (x+1, y+1), (x-1, y+1), (x-1, y-1) ký hiệu là tập ND(p). Tập 8 điểm ảnh lân cận 8 là hợp của 2 tập trên: N8(p) = N4(p) + ND(p). Liên kết ảnh [3][4]: Các mối liên kết của ảnh được xem như là mối liên kiết của 2 điểm ảnh gần nhau, có 3 loại liên kết: liên kết 4, liên kết 8, lên kết m(liên kết hỗn hợp). Trong ảnh đa mức xám, ta có thể đặt V chứa nhiều giá trị như V={tập con}. Cho p có tọa độ (x, y). Liên kết 4: hai điểm ảnh p và q có giá trị thuộc về tập V được gọi là liên kết 4 của nhau nếu q thuộc về tập N4(p). Liên kết 8: hai điểm ảnh p và q có giá trị thuộc về tập V được gọi là liên kết 8 của nhau nếu q thuộc về tập N8(p). Liên kết m: hai điểm ảnh p và q có giá trị thuộc về tập V được gọi là Liên kết M của nhau nếu thõa 1 trong 2 điều kiện sau: q thuộc về tập N4(p). q thuộc về tập ND(p) và giao của hai tập N4(p), N4(q) không chứa điểm ảnh nào có giá trị thuộc V.

#### Lọc nhiễu

Ảnh thu nhận được thường sẽ bị nhiễu nên cần phải loại bỏ nhiễu. Các toán tử không gian dùng trong kỹ thuật tăng cường ảnh được phân nhóm theo công dụng: làm trơn nhiễu, nổi biên. Để làm trơn nhiễu hay tách nhiễu [3-4], người ta sử dụng các bộ lọc tuyến tính (lọc trung bình, thông thấp) hoặc lọc phi tuyến (trung vị, giả trung vị, lọc đồng hình). Từ bản chất của nhiễu (thường tương ứng với tần số cao) và từ cơ sở lý thuyết lọc là: bộ lọc chỉ cho tín hiệu có tần số nào đó thông qua, để lọc nhiễu người ta thường dùng lọc thông thấp (theo quan điểm tần số không gian) hay lấy tổ hợp tuyến tính để san bằng (lọc trung bình). Để làm nổi cạnh (ứng với tần số cao), người ta dùng các bộ lọc thông cao, lọc Laplace. Phương pháp lọc nhiễu. Chia làm 2 loại: lọc tuyến tính, lọc phi tuyến.

#### Làm trơn nhiễu bằng lọc tuyến tính

Khi chụp ảnh có thể xuất hiện nhiều loại nhiễu vào quá trình xử lý ảnh, nên ta cần phải lọc nhiễu . Gồm các phương pháp cơ bản lọc trung bình, lọc thông thấp… Lọc trung bình: Với lọc trung bình, mỗi điểm ảnh được thay thế bằng trung bình trọng số của các điểm lân cận. Lọc trung bình có trọng số chính là thực hiện chập ảnh đầu vào với nhân chập H. Nhân chập H có dạng:

H= (1.1)

Lọc thông thấp: Lọc thông thấp thường được sử dụng để làm trơn nhiễu. Về nguyên lý của bộ lọc thông thấp giống như đã trình bày trên.

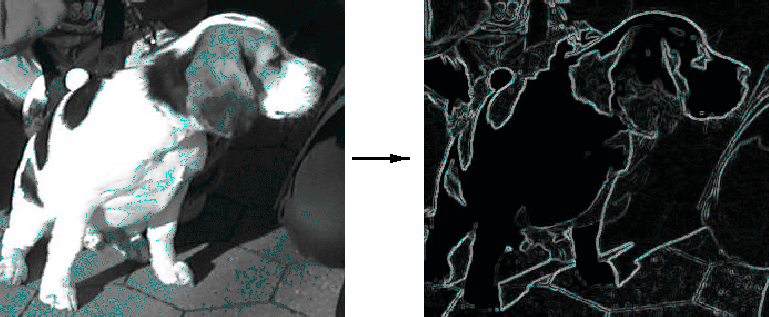
H= (1.2)

#### Làm trơn nhiễu bằng lọc phi tuyến

Các bộ lọc phi tuyến cũng hay được dùng trong kỹ thuật tăng cường ảnh. Một số phương pháp lọc cơ bản bộ lọc trung vị, lọc ngoài… Với lọc trung vị, điểm ảnh đầu vào sẽ được thay thế bởi trung vị các điểm ảnh còn lọc giả trung vị sẽ dùng trung bình cộng của hai giá trị “trung vị” (trung bình cộng của max và min). Lọc trung vị: Kỹ thuật này đòi hỏi giá trị các điểm ảnh trong cửa sổ phải xếp theo thứ tự tăng hay giảm dần so với giá trị trung vị. Kích thước cửa số thường được chọn sao cho số điểm ảnh trong cửa sổ là lẻ.

#### Phương pháp phát hiện biên

Biên là một trong những vấn đền ta cần quan tâm trong xử lý ảnh. Vì ở giai đoạn phân đoạn ảnh chủ yếu dựa vào biên .



Hình .:Hình tách biên

* Điểm biên: Một điểm ảnh được coi là điểm biên nếu có sự thay đổi nhanh hoặc đột ngột về mức xám. Ví dụ trong ảnh nhị phân, điểm đen gọi là điểm biên nếu lân cận nó có ít nhất một điểm trắng.
* Đường biên: tập hợp các điểm biên liên tiếp tạo thành một đường biên.

Ý nghĩa của đường biên trong xử lý: ý nghĩa đầu tiên của đường biên là một loại đặc trưng cục bộ tiêu biểu trong phân tích, nhận dạng ảnh. Thứ hai, người ta sử dụng biên làm phân cách các vùng xám cách biệt. Ngược lại, người ta cũng sử dụng các vùng ảnh để tìm đường phân cách. Tầm quan trọng của biên: để thấy rõ tầm quan trọng của biên, xét ví dụ sau: khi người họa sỹ muốn vẽ một danh nhân, họa sỹ chỉ cần vẽ vài đường nứt tốc họa mà không cần vẽ một cách đầy đủ.

Như vậy, phát hiện biên một cách lý tưởng là phát hiện được tất cả các đường biên trong các đối tượng. Định nghĩa toán học của biên ở trên là cơ sở cho các kỹ thuật phát hiện biên. Điều quan trọng là sự biến thiên giữa các điểm ảnh thường nhỏ, trong khi đó biến thiên độ sáng của điểm biên thường là khá lớn khi qua biên. Xuất phát cơ sở này người ta thường sử dụng hai phương pháp phát hiện biên như sau:

* Tách biên theo đạo hàm bậc một : Có 2 phương pháp cơ bản là: một là tạo gradient của hai hướng và trực giao trong ảnh, hai là dùng tập đạo hàm có hướng.
* Tách biên theo đạo hàm bậc hai : được thực hiện trên một số dạng vi phân bậc 2 để làm xuất hiện biên. Có hai dạng của phương pháp đạo hàm bậc hai đã được nghiên cứu là: phương pháp Lapplace và đạo hàm trực tiếp.

Bộ tách biên Canny : phương pháp phát hiện này được sử dụng phổ biến vì nó có nhiều ưu điểm hơn các phương pháp khác.

Các bước thực hiện:

Làm phẳng dùng bộ lọc Gauss.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 4 | 7 | 4 | 1 |
| 4 | 16 | 26 | 16 | 4 |
| 7 | 26 | 41 | 26 | 7 |
| 4 | 16 | 26 | 16 | 4 |
| 1 | 4 | 7 | 4 | 1 |

Hình .:Bộ lọc Gaussian

Sau đó, Gradient cục bộ của biên độ và hướng được tính. Tìm điểm ảnh có biên độ lớn nhất dùng kỹ thuật nonmaximal suppression.

Các điểm ảnh đỉnh tìm được chia làm hai ngưỡng T1 và T2, T1 < T2. Các điểm ảnh đỉnh có giá trị lớn hơn T2 được gọi là Strong và nằm trong khoảng T1 và T2 được gọi là Weak. Liên kiết các điểm ảnh Weak có 8 kết nối đến điểm ảnh Strong.

Phương pháp gradient: Gradient là một vector có các thành phần biểu thị tốc độ thay đổi giá trị của điểm ảnh theo 2 hướng x và y, hay có thể nói là nó đại diện cho sự thay đổi về hướng và độ lớn của một vùng ảnh. Ảnh được làm mịn sau đó được lọc bằng hạt nhân Sobel theo cả hướng ngang và dọc để lấy đạo hàm đầu tiên theo hướng nằm ngang (Gx) và hướng dọc (Gy). Từ hai hình ảnh này, chúng ta có thể tìm gradient và hướng cạnh cho mỗi pixel như sau:

Edge\_Gradient (G) = (1.3)

Angle(Ø) = tan-1 (1.4)

#### Phân đoạn ảnh

Phân đoạn ảnh là bước then chốt trong xử lý ảnh. Giai đoạn này nhằm phân tích ảnh thành các vùng có cùng tính chất nào đó dựa theo biên hay các vùng liên thông. Tiêu chuẩn để xác định các vùng liên thông có thể là cùng mức xám, cùng màu hay cùng độ nhóm.

Quá trình phân đoạn ảnh nhằm tách đối tượng cần khảo sát ra khỏi phần nội dung còn lại của ảnh, hay phân chia các đối tượng trong ảnh thành những đối tượng riêng biệt, như vậy quá trình phân đoạn ảnh là quá trình giảm bớt số lượng thông tin trong ảnh và chỉ giữ lại những thông tin cần thiết cho ứng dụng. Do đó phân đoạn ảnh là quá trình loại bỏ các đối tượng không quan tâm trong ảnh.

Có nhiều phương pháp phân đoạn ảnh khác nhau. Trong đó quá trình phân đoạn ảnh sử dụng một ngưỡng giá trị xám để phân đoạn ảnh ra thành các đối tượng và nền là phương pháp đơn giản nhất. Lúc này các điểm ở bên dưới ngưỡng giá trị xám thuộc về nền còn những điểm ảnh ở bên trên ngưỡng giá trị xám thuộc về đối tượng. Phương pháp phân đoạn ảnh này hiệu quả lớn đối với ảnh nhị phân, văn bản in hay đồ họa...

Dựa vào đặc tính vật lý của vùng ảnh, các kỹ thuật phân đoạn vùng có thể được chia làm 3 loại:

* Các kỹ thuật cục bộ: dựa trên các đặc tính cục bộ của các điểm ảnh và các lân cận của nó.
* Các kỹ thuật tổng thể: phân đoạn một ảnh dựa trên cơ sở của thông tin lấy từ tổng thể như sử dụng biểu đồ mức xám histogram.
* Các kỹ thuật chia, nối và phát triển: dựa trên các khái niệm tương đồng về hình dạng và tính đồng nhất. Hai vùng có thể được nối lại với nhau và liền kề bên nhau. Các vùng không đồng nhất có thể được chia thành các vùng nhỏ. Một vùng có thể được phát triển bằng các nối các điểm ảnh sao cho nó đồng nhất với nhau.

#### Các phần mềm hỗ trợ xử lý ảnh

Hiện nay xử lý ảnh được giảng dạy trường đại học và ứng dụng vào thực tế rất nhiều như các phần mềm chỉnh sửa hình ảnh hay nhận biết khuôn mặt. Chính vì thế có rất nhiều

công cụ để chúng ta lập trình ứng dụng vào thực tế, Như phải kể đến Matlap, hay ngôn ngữ Python…

### Các hình dạng cơ bản của sản phẩm

Trong cuộc sống hiện nay, nhu cầu về kiểu dáng sản phẩm ngày càng được chú trọng, để đáp ứng nhu cầu thiết yếu đó nên trên thị trường ra nhiều sản phẩm có hình dạng và kiểu dáng đa dạng.

Ở đây ta sẽ chia sản phẩm ra 2 nhóm, cách chia này dựa vào sự phổ biến của hình dạng sản phẩm ở ngoài cuộc sống, nhóm hình dạng chính (hình chữ nhật, hình vuông, hình tam giác) và các hình dạng còn lại ta chia thành 1 nhóm.

Khi kết quả phân loại sản phẩm rơi vào một trong những hình nhóm hình này thì kết quả sau nhận dạng chỉ ra đúng tên hình dạng của sản phẩm.

Hình chữ nhật: Đặc điểm nhận dạng: Có 4 đỉnh, 4 cạnh, 4 góc vuông, 2 cạnh đối diện đều bằng nhau, khác với so với hình vuông là 2 cạnh kề không bằng nhau…

Hình vuông: Đặc điểm nhận dạng: Có 4 đỉnh, 4 cạnh, 4 góc vuông, 2 cạnh đối diện đều bằng nhau, 2 cạnh kề bằng nhau…

Hình tam giác: Đặc điểm nhận dạng: có 3 đỉnh và 3 cạnh là hình tam giác…

### Phương pháp nhận dạng sản phẩm theo màu sắc

* Phương pháp được chia thành 4 phần:
* Input từ camera
* Nhận dạng các vật có màu sắc từ camera
* Nhị phân hóa hình ảnh
* Xuất ra kết quả

**a. Input video từ camera**

Video được input vào từ những camera kỹ thuật số.

Các vật có màu sắc được đưa vào camera

Bước tiền xử lý được dùng để làm tăng tốc độ xử lý của video làm tăng tốc độ tương phản và giảm nhiễu trong video

**b. Nhận các vật có màu sắc**

Camera sẽ nhận các vật được người dùng đưa vào. Chương trình bắt đầu phân tích với dãy thuật lả dùng ảnh xám trừ ảnh gốc trừ những vùng có màu sắc đỏ, xanh lá, xanh dương.

**c. Nhị phân hóa**

Các hình ảnh trong video được lọc bằng bộ lọc để tạo ra một hình ảnh bằng cách tính toán mức ngưỡng

Tìm các giá trị nhỏ nhất và lớn nhất của pixel từ video

Quá trình này chuyển tất cả pixel của video sang dạng 0-1 và làm cho các bước xử lý sau dễ dàng hơn

Khi đó các màu đỏ, xanh lá, xanh dương được rang giá trị ở mức 1. Còn tất cả các màu khác ở mức 0.

Chương trình sẽ xuất ra các giá trị ở mức 1 được phân tích ra màn hình.

**d. Xuất kết quả**

Nhận dạng và cho ra kết quả các vật có màu sắc như: đỏ, xanh lá và xanh dương từ các vật bất kì mang những màu sắc trên.

### Phương pháp nhận dạng sản phẩm theo hình dạng

#### Phương pháp

Nhận dạng ảnh là giai đoạn quan trọng cuối của một hệ thống xử lý ảnh. Nhận dạng là quá trình phân loại các đối tượng được biểu diễn theo một mô hình nào đó. Ảnh được chụp sẽ được phân tích thành các đặc trưng riêng biệt, với những đặc trưng đó ta đem đi tìm hiểu và nhận dạng.

Mô hình của tôi nhận dạng 3 hình cơ bản là: hình vuông, hình chữ nhật, hình tam giác.

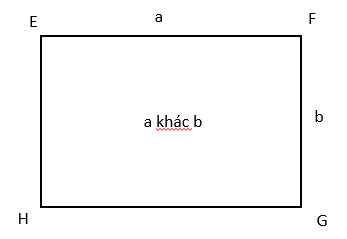
Như đề cập ở trên các các hình dạng của sản phẩm có các đặc điểm đặc trưng. Tuy nhiên, thì vẫn có một số đặc điểm giống nhau, đặc điểm khác nhau, cùng với đó có thêm một số đặc điểm bị dư thừa trong việc so sánh với các sản phẩm khác tức là một số đặc điểm không liên quan đến nhau. Nếu nhận dạng tất cả các đặc điểm của các hình thì điều đó rất mất thời gian cùng với tiêu tốn tài nguyên dữ liệu lớn. Chính vì vậy ở đề tài này, ta cần chọn lọc ra từng đặc điểm riêng biệt của từng hình mà các hình không có và loại bỏ các đặc điểm cần thiết.

Như vậy, để nhận dạng được hình dạng của sản phẩm cần xác định các đặc điểm, đặc trưng riêng biệt của của từng hình dạng.

Ở đây chỉ có 3 hình dạng sản phẩm (hình chữ nhật, hình vuông, hình tam giác) cần phân loại, chính vì vậy sẽ có một số đặc điểm hình dạng không cần xét tới.

**a. Sản phẩm hình chữ nhật**

Hình chữ nhật là hình có 4 đỉnh, 4 cạnh, 4 góc vuông, 2 cạnh đối diện bằng nhau,... Từ đặc điểm đó ta đi so sánh với 2 hình còn lại và phân tích thấy: Đặc điểm thứ nhất có 4 đỉnh, nhưng sản phẩm hình vuông cũng có 4 đỉnh. Chính vì vậy ta có đặc điểm thứ hai để so sánh sự khác biệt với hình vuông, đó là 2 cạnh kề không bằng nhau với hình chữ nhật, với hình vuông thì chúng bằng nhau.

Như vậy, đặc điểm nhận dạng sản phẩm hình chữ nhật, gồm 4 đỉnh và 2 cạnh kề không bằng nhau.

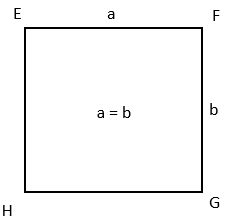
Hình 1.6: Đặc điểm hình chữ nhật

**b.Sản phẩm hình vuông**

Đặc điểm nhận dạng: Có 4 đỉnh, 4 cạnh, 4 góc vuông, 2 cạnh đối diện đều bằng nhau, 2 cạnh kề bằng nhau…

Từ đặc điểm đó ta đi so sánh với 2 hình còn lại và phân tích thấy: Đặc điểm thứ nhất có 4 đỉnh, nhưng sản phẩm hình chữ nhật cũng có 4 đỉnh như đã đề cập ở trên. Chính vì vậy ta có đặc điểm thứ hai để so sánh sự khác biệt với hình chữ nhật, đó là 2 cạnh bằng nhau.

Như vậy, đặc điểm nhận dạng sản phẩm hình vuông, gồm 4 đỉnh và 2 cạnh kề bằng nhau.

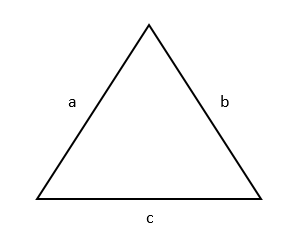


Hình 1.7:Đặc điểm hình vuông

**c. Sản phẩm hình tam giác**

Đặc điểm nhận dạng: có 3 đỉnh và 3 cạnh là hình tam giác...

Từ đặc điểm đó ta đi so sánh với 2 hình còn lại và phân tích thấy: Chỉ có hình tam giác có 3 đỉnh, và đó chính là đặc điểm đặc trưng riêng biệt của tam giác.

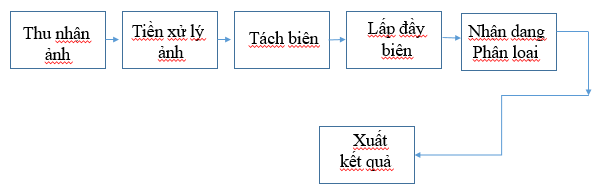


Hình 1.8: Đặc điểm hình tam giác

Bảng 1.1: Đặc điểm hình dạng sản phẩm

|  |  |  |  |
| --- | --- | --- | --- |
| Đặc điểm | Hình vuông | Hình chữ nhật | Hình tam giác |
| Đỉnh | 4 | 4 | 3 |
| Cạnh | 2 cạnh kề bằng nhau | 2 cạnh kề không bằng nhau | Có 3 cạnh |

#### Các bước thực hiện phân loại sản phẩm theo hình dạng



Hình 1.9:Sơ đồ các bước thực hiện phân loại sản phẩm

**a.Thu nhận ảnh**

Thu thập ảnh là bước mà ảnh có thể nhận qua camera hoặc có thể lấy từ bộ nhớ của hệ thống.

**b.Tiền xử lý ảnh**

Là bước chỉnh sửa chất lượng ảnh, như việc lọc nhiễu hay tăng cường độ sáng, để nâng cao chất lượng ảnh và chuyển sang ảnh xám để dễ dạng nhận dạng ảnh.

Chuyển đổi ảnh xám: Giả sử, hình ảnh của được lưu trữ dưới dạng RGB (Red-Green- Blue). Điều này có nghĩa là có ba ma trận xám tương ứng cho màu Red, Green, Blue. Công việc là tìm cách tổng hợp ba ma trận này về thành một ma trận duy nhất. Một trong số các công thức phổ biến để thực hiện việc đó là:

Y = 0.2126R + 0.7152G + 0.0722B (2.5)

Trong đó:

Y: ma trận xám cần tìm

R: ma trận xám đỏ của ảnh

G: ma trận xám lục của ảnh

B: ma trận xám lam của ảnh

Lọc trung bình: Giả sử có một ảnh đầu vào với I(x,y) là giá trị điểm ảnh tại một điểm (x,y) và một ngưỡng θ.

Bước 1: Tính tổng các thành phần trong ma trận lọc (Kernel).

Bước 2: Chia lấy trung bình của tổng các thành phần trong ma trận được tính ở trên với số lượng các phần tử của cửa sổ lọc ra một giá trị Itb(x, y).

Bước 3: Hiệu chỉnh:

* + Nếu I(x, y) - Itb(x,y) > θ thì I(x,y) = Itb(x,y).
  + Nếu I(x, y) - Itb(x,y) <= θ thì I(x,y)=I(x,y).

Chú ý: θ là một giá trị cho trước và có thể có hoặc không tùy thuộc vào mục đích.

**c.Tách biên**

Tách biên là quá trình loại bỏ dữ liệu không cần thiết để giảm thiểu tài nguyên sử dụng và dễ dàng tính toán.

Tách biên được chia thành các bước:

* Giảm nhiễu
* Xác định đường biên
* Loại bỏ dữ liệu không phải đường biên

Giảm nhiễu:

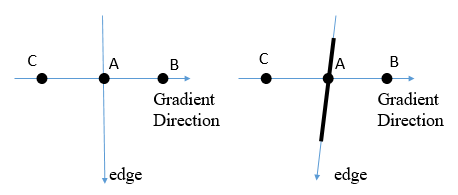
Vì phát hiện cạnh, đường biên dễ bị nhiễu trong ảnh, bước đầu tiên là loại bỏ nhiễu trong ảnh bằng bộ lọc Gaussian.

Xác định đường biên:

Ảnh được làm mịn bằng bộ lọc tuyến tính nhằm mục đích loại bỏ những chi tiết nhỏ ra khỏi ảnh trước khi tiến hành tách các thành phần lớn hơn ra khỏi, ảnh sau đó được lọc bằng hạt nhân Sobel tìm gradient và hướng cạnh cho mỗi pixel. Hướng dốc luôn vuông góc với các cạnh. Nó được làm tròn thành một trong bốn góc đại diện cho hướng dọc, ngang và hai đường chéo.

Sau khi nhận được độ lớn và hướng gradient, việc quét toàn bộ hình ảnh được thực hiện để xóa bất kỳ pixel không mong muốn nào có thể không tạo thành cạnh. Đối với điều này, tại mỗi điểm ảnh, pixel được kiểm tra nếu nó là một cực đại cục bộ trong vùng lân cận theo hướng gradient.

Kiểm tra hình ảnh dưới đây:

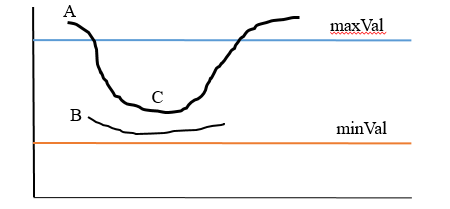


Hình .:Xác định biên

Điểm A nằm trên cạnh theo hướng thẳng đứng. Hướng dốc là bình thường cho cạnh. Điểm B và C theo hướng dốc. Vì vậy, điểm A được so sánh với điểm B và C để xem nó có là một cực đại hay không, nếu có thì nó thuộc đường biên.

Loại bỏ dữ liệu không phải đường biên:

Giai đoạn này quyết định đó là tất cả các cạnh, biên có thực sự là cạnh hay không. Đối với điều này, chúng ta cần hai giá trị ngưỡng, minVal và maxVal. Bất kỳ cạnh nào có gradient cường độ lớn hơn maxVal đều chắc chắn là các cạnh và các cạnh dưới minVal chắc chắn là không có cạnh, do đó bị loại bỏ. Những biên nằm giữa hai ngưỡng này được phân loại các cạnh liên tục hay không liên tục, nếu chúng được kết nối với các pixel "chắc chắn", chúng được coi là một phần của các cạnh. Nếu không, chúng cũng bị loại bỏ.



Hình 1.11:Nhận dạng đường biên

Cạnh A nằm trên giá trị maxVal, do đó được coi là đường biên. Mặc dù cạnh C là dưới maxVal, nó được kết nối với cạnh A, do đó cũng được coi đường biên hợp lệ và chúng ta có được đường cong đầy đủ đó. Nhưng cạnh B, mặc dù nó ở trên minVal và nằm trong cùng một vùng với cạnh của C, nó không được kết nối với bất kỳ đường biên nào trên giá trị maxval nên không phải cạnh do đó nó bị loại bỏ.

**d.Lấp đầy biên**

Là quá trình lấp đầy các đường biên kín, không bị gián đoạn hay đứt khúc, để tạo ra được ảnh nhị phân có giá trị 0 và 1. Mức 1 là giá trị của hình dạng lấp đầy biên, và bên ngoài biên được coi là giá trị 0. Như hình bên trên, đầu tiên phải sơn lại các vùng ở ngoài biên từ ảnh ban đầu ảnh a tức là đổi giá trị 0 bên ngoài thành giá trị 1 thành ảnh b. Sau đó đảo giá trị ảnh ngược lại của ảnh b ta được ảnh c.

Cuối cùng ta dùng phép toán OR để giữa ảnh a và ảnh c thì ta có kết quả.

**e. Nhận dạng và phân loại sản phẩm**

Ta nhận dạng sản phẩm dựa vào các đặc điểm riêng biệt của từng hình, thì sau đó ta đã có được các đặc điểm về đỉnh, đặc điểm về cạnh. Đó là nhận dạng sản phẩm.

Sau khi có được các đặc điểm riêng biệt của từng hình như trên. Ta có phân loại từng hình dạng sản phẩm một nhờ vào việc so sánh các đặc điểm riêng biệt của chúng. Ta lần lượt đi so sánh từng hình một, từ hình vuông, đến hình chữ nhật, đến hình tam giác. Nếu các đặc điểm trùng khớp với đặc điểm được đề ra trước ở hình dạng nào thì ta kết luận ngay đó là sản phẩm có hình dạng đó. Ngược lại nếu không có đặc điểm nào khớp ta kết luận đó là một sản phẩm nào đó khác mà không phải sản phẩm đã định sẵn.

### Giới thiệu hiệu chỉnh stereo camera

Hiệu chỉnh stereo camera là phương pháp tính toán và thực nghiệm nhằm tìm ra các tham số của camera để tái tạo không gian 3D của một cảnh, một vật thể nào đó trong thực tế bằng những ảnh mà camera đó ghi lại được.

### Stereo camera

Stereo vision là kỹ thuật sử dụng hai camera để đo khoảng cách giữa hai đối tượng. Thông thường sử dụng hai camera cùng loại và thông số được đặt trên đường thẳng hoặc ngang.

Cấu hình chuẩn của hệ thống 2 camera Sử dụng camera stereo này, chúng ta có thể thu được hình ảnh của đối tượng tại hai vị trí khác nhau: ảnh bên trái và ảnh bên phải của đối tượng (sự chênh lệch). Các ảnh của các camera được phân tích để tìm các điểm chung. Sử dụng quy tắc tam giác đồng dạng và độ lệch của các điểm chung để xác định khoảng cách (độ sâu) so với camera.

Hình mô tả hệ thống tọa độ của một stereo camera, có thể được tính như trong hai phương trình sau:

= (1.6)

Z= = (1.7)

Trong đó:

* + d = xl-xr là sự chênh lệch.
  + xl là tọa độ x của camera bên trái.
  + xr là tọa độ x của camera bên phải.
  + b là chiều dài cơ sở của hai camera.
  + f là tiêu cự của máy ảnh.
  + z là khoảng cách từ đường cơ sở đến đối tượng.

Tính toán bản đồ chênh lệch:

Là một trong những vấn đề quan trọng trong thị giác máy tính 3D. Một số lượng lớn các thuật toán đã được đề xuất để giải quyết vấn đề này. Một trong những phương pháp tương đối mới là “Tính toán bản đồ độ sâu” từ hình ảnh stereo. Đối với các cặp hình ảnh stereo đã được chỉnh sửa epipolar, mỗi điểm trong hình ảnh bên trái nằm trên đường nằm ngang (đường epipolar) có thể có điểm ảnh tương ứng trong hình ảnh bên phải. Cách tiếp cận này được sử dụng để làm giảm không gian tìm kiếm chiều sâu bản đồ thuật toán tính toán. Chiều sâu của một điểm ảnh là khoảng cách điểm không gian tương ứng tới trung tâm máy ảnh. Để ước tính bản đồ độ sâu và phát hiện các đối tượng 3D, các điểm ảnh tương ứng trong những ảnh trái và ảnh phải cần được phát hiện.

Thuật toán là phần rất quan trong trong hệ thị giác máy tính 3D. Tuy nhiên, hiện nay có rất nhiều thuật toán đã được phát triển và hàng năm thì cũng có nhiều thuật toán mới được đề xuất. Việc phân loại tất cả các thuật toán là một điều rất khó vì hầu hết các nhà nghiên cứu chỉ đưa ra kết quả định tính hoặc là kết quả cuối cùng về hiệu quả của các thuật toán của họ. Các thuật toán truyền thống của thị giác máy tính 3D là SAD (sum of absolute diffirence), SSD (sum of squared diffirence) và NCC (normalized cross correlation). Các hàm giá trị phù hợp được kết hợp trên các vùng cung cấp. Các vùng cung cấp thường nhận được như các cửa sổ được kết hợp có thể là hình vuông hoặc hình chữ nhật, kích thước cố định hoặc thích nghi. Việc xác định giá trị của kết hợp các hàm giá trị phù hợp, dẫn đến nền tảng của hầu hết các thuật toán thị giác nổi. Các thuật toán truyền thống có thể được biểu diễn toán học như sau:

1) Phương pháp tổng bình phương khác biệt (SSD – Sum of Squared Differences):

SSD(x,y,d) = 2  (1.8)

Phương pháp này tính tổng của các bình phương của các hiệu nên xuất hiện thêm phép nhân trước khi tính tổng. Do đó độ phức tạp tăng lên đáng kể.

2) Phương pháp tổng sự khác biệt tuyệt đối (SAD - Sum of Absolute Differences):

SAD(x,y,d) =  (1.9)

Theo công thức trên, việc tính giá trị chênh lệch của điểm (x,y) được thực hiện bằng phép tính hiệu trong cửa sổ W. Do đó thuật toán đơn giản. Cửa sổ W và phạm vi chênh lệch d càng lớn thì số phép tính tăng. Thông thường có thể thực tính SAD với cửa sổ 3x3, hoặc 5x5 hoặc 7x7 hoặc 11x11. Phạm vi chênh lệch d thường chọn dưới 120 với các giá trị tham khảo như 16, 50, 128.

3) Phương pháp dựa trên sự tương quan chéo chuẩn (NCC)

NCC(x,y,d)= (1.10)

Trong đó Il , Ir là những giá trị cường độ trong hình ảnh trái và phải, (x, y) là tọa độ của điểm ảnh, d là giá trị sai lệch được xem xét và W là cửa sổ phức hợp.

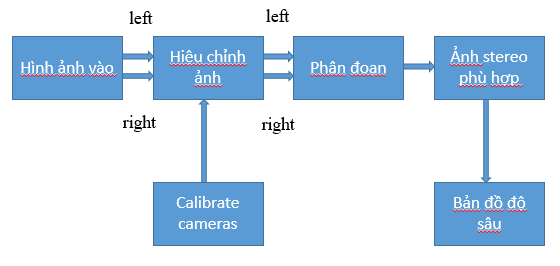
Việc lựa chọn các giá trị sai lệch thích hợp cho mỗi điểm ảnh được thực hiện sau đó. Phương pháp này có độ phức tạp cao nhất vì xuất hiện các phép toán tiêu tốn nhiều tài nguyên của máy tính như phép bình phương, khai căn. Do đó việc ứng dụng của thuật toán này trong các hệ thống thời gian thực là không khả thi tức là, cho mỗi điểm ảnh (x, y) và cho giá trị không đổi có sự khác nhau d, chi phí tối thiểu được lựa chọn.

D(x, y) = argminSAD(x,y,d) (1.11)

Phương trình trên được dùng cho phương pháp SAD. Tuy nhiên, trong nhiều trường hợp lựa chọn khác biệt là một quá trình lặp đi lặp lại, vì độ chênh lệch của mỗi điểm ảnh là tùy thuộc vào sự chênh lệch của các điểm ảnh lân cận. Các thuật toán phân vùng lai là một sự kết hợp với phương pháp SAD được áp dụng phù hợp với hình ảnh stereo để tinh chỉnh các bản đồ độ sâu cuối cùng.

Các thuật toán phù hợp với âm thanh stereo dựa trên phương pháp mà không SAD phân đoạn và cách tiếp cận lai các thuật toán phù hợp với hình ảnh stereo dựa trên SAD lai có vẻ là thuật toán hiệu quả hơn để sản xuất bản đồ chênh lệch sạch hơn với các khu vực đồng nhất. Mặt khác, các thuật toán phù hợp với hình ảnh stereo dựa trên phương pháp SAD sản xuất một bản đồ chiều sâu rõ ràng về hiện trường. Hơn nữa, nó tạo ra một chất lượng cao hơn và ít lỗi gây ra bởi tắc phân đoạn hình ảnh.

Thuật toán bao gồm các bước như hình 1.12 :



Hình 1.12:Thuật toán SAD phân vùng

Bước 1: là quá trình hiệu chuẩn ảnh. Mục đích của quá trình này là xác định được các thông số bên trong và bên ngoài của camera :

* Các thông số bên trong như là tiêu cự, điểm trung tâm, hiệu chỉnh vuông góc alpha và độ méo ...
* Các thông số bên ngoài như là các phép xoay và các phép chuyển.

Bước 2: là quá trình hiệu chỉnh ảnh. Quá trình hiệu chỉnh ảnh là quá trình mà hai ảnh của cảnh cố định chuyển đổi đơn ứng, có nghĩa rằng các đường tâm cực đồng tâm trùng nhau và song song với trục tâm quang.

Bước 3: là quá trình tìm điểm ảnh tương đồng giữa ảnh mục tiêu và ảnh tham chiếu tức là tìm bản đồ chênh lệch.

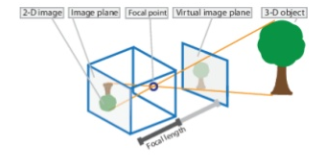
Bước 4: là sử dụng quy tắc tam giác đồng dạng và độ lệch của các điểm chung để xác định khoảng cách so với camera., có nghĩa là tìm được bản đồ độ sâu.

### Phương pháp hiệu chỉnh chuẩn

Phương pháp hiệu chỉnh chuẩn là công việc ước tính các thông số của thấu kính, cảm biến hình ảnh của máy ảnh. Có thể sử dụng các thông số này để cân chỉnh lại hiện tượng biến dạng, đo kích thước một vật ngoài thế giới thực, tìm ra vị trí của camera khi chụp một bức hình.

Những công việc đó cũng được sử dụng trong các ứng dụng như thị giác máy tính để phát hiện và đo đối tượng, trong robotics, hệ thống điều hướng, xây dựng ảnh 3D.

Các thông số camera bao gồm các thông số nội, thông số ngoại, các hệ số biến dạng. Để ước tính các thông số camera, bạn cần có những điểm 3D thực và điểm ảnh 2D tương ứng. Bạn có thể có những điểm trên bằng cách sử dụng nhiều hình chụp của một mẫu để cân chỉnh ảnh, chẳng hạn là một bàn cờ. Sử dụng các ảnh đó, bạn có thể giải ra được thông số camera. Mô hình camera pinhole là một mô hình camera đơn giản. Bạn có thể bắt gặp mô hình camera này được cài đặt đơn giản khi người ta muốn xem hiện tượng nhật thực. Mô hình gồm một hộp đen kín, được đục một lỗ nhỏ gọi là lỗ kim. Thuật ngữ gọi pinhole là Focal Point. Khoảng cách từ Focal Point tới màn nhận ảnh là Focal length. Độ mở của pinhole gọi là Aperture khẩu độ.



Hình 1.13: Mô hình các thông số của camera

Các thông số ảnh được biểu diễn bởi ma trận 3×4 được gọi là ma trận camera. Ma trận camera này ánh xạ cảnh thế giới thực 3D vào mặt phẳng ảnh. Thuật toán calibration tính ra ma trận camera sử dụng các thông số nội và thông số ngoại. Thông số ngoại của camera đại diện cho vị trí của camera trong thế giới 3D. Thông số nội của camera đại diện cho tâm quang và tiêu cự của camera.

Trong đó:

* + (X, Y, Z) là tọa độ của điểm ảnh trong không gian ba chiều
  + (u, v) là tọa độ của điểm ảnh hai chiều
  + (cx, cy) là điểm chính, thường là tâm ảnh fx, fy là độ dài tiêu cự biểu diễn ở đơn vị pixel. Ma trận các thông số nội tại không phụ thuộc vào cảnh chụp. Vì vậy, một khi ước tính, nó có thể được sử dụng lại miễn là độ dài tiêu cự được cố định (trong trường hợp ống kính zoom). Ma trận joint rotation-translation được gọi là ma trận của các tham số bên ngoài. Nó được sử dụng để mô tả chuyển động của camera xung quanh một cảnh tĩnh, hoặc ngược lại, chuyển động không thay đổi hình dáng của một đối tượng ở phía trước của một máy ảnh tĩnh. Tức là, [R|t] dịch tọa độ của một điểm (X, Y, Z) đến một hệ tọa độ, cố định đối với máy ảnh.

Các ống kính thật thường có một số biến dạng, phần lớn là biến dạng xoay và biến dạng tiếp xúc. Các hệ số biến dạng không phụ thuộc vào cảnh. Do đó, chúng cũng thuộc về các thông số nội. Và chúng vẫn giữ nguyên bất kể độ phân giải hình ảnh.

Ví dụ: nếu máy ảnh đã được hiệu chỉnh trên các hình ảnh có độ phân giải 320 x 240, hoàn toàn có thể sử dụng các hệ số biến dạng tương tự cho các hình ảnh 640 x 480 từ cùng một máy ảnh trong khi fx, fy cx và cy cần phải được tính lại.

Các function dưới đây sử dụng mô hình trên để thực hiện những việc sau:

* + Dựng điểm 3D từ ảnh 2D với các thông số nội và ngoại.
  + Tính các thông số ngoại với thông số nội, một vài điểm 3D và các phép chiếu của chúng được cho trước.
  + Ước lượng các thông số nội và ngoại của máy ảnh từ nhiều khung cảnh của vật.
  + Ước lượng vị trí tương đối và định hướng của stereo camera “heads” và tính toán sự chuyển đổi chỉnh lưu làm cho trục quang của máy ảnh song song.

# GIỚI THIỆU THIẾT BỊ CAMERA VÀ PHẦN MỀM PYTHON NHẬN DẠNG SẢN PHẨM

## Giới thiệu về Camera HIKVISION sử dụng trong nhận diện sản phẩm



Hình 2.1:Hình dạng camera

Sản phẩm [**camera Hikvision**](https://camerasaigon24h.com/gioi-thieu-ve-camera-HIKVISON-hang-thuong-hieu_cv50) được nghiên cứu và phát triển bởi hơn 4.200 kỹ sư, và được sản xuất với công nghệ khép kín từ khâu thiết kế - sản xuất để luôn đảm bảo tính sáng tạo và công nghệ mới. Mọi sản phẩm đều được kiểm nghiệm trong các môi trường đa dạng khác nhau trên toàn thế giới. Đáp ứng toàn diện các nhu cầu liên quan đến hệ thống camera giám sát. Chất lượng cao, nhiều tính năng vượt trội, mẫu mã đẹp và giá cực kì cạnh tranh

## các thông số kỹ thuật của [camera Hikvision](https://camerasaigon24h.com/gioi-thieu-ve-camera-HIKVISON-hang-thuong-hieu_cv50)

Sử dụng công nghệ mới HD-TVI cho hình ảnh sắc nét.

Công nghệ HD-TVI (High Definition Transport Video Interface): HD-TVI truyền tải hình ảnh, dữ liệu đảm bảo chất lượng hình ảnh, Video rõ ràng, không bị trễ hình qua cáp đồng trục.

Cảm biến hình ảnh: CMOS.

Độ phân giải: 2.0 Megapixel (1920 x 1080).

Ống kính: 2.8mm, 3.6mm, 6mm (tùy chọn).

Tầm quan sát hồng ngoại: 20 mét.

Chức năng quan sát Ngày và Đêm.

Chức năng hồng ngoại thông minh Smart IR.

Tiêu chuẩn chống thấm nước: IP66 (thích hợp sử dụng trong nhà và ngoài trời).

Nguồn điện cung cấp: 12VDC ± 15%.

Kích thước: Φ70 x 149.5mm.

Bảng 2.1:Đặc tính kỹ thuật

|  |  |
| --- | --- |
| **Model** | **DS-2CE16D0T-IR** |
| Signal System | PAL/ NTSC |
| Effective Pixels | 1920(H) x 1080(V) |
| Min. illumination | 0.01 Lux @ (F1.2,AGC ON),0 Lux with IR |
| Shutter Time | 1/25(1/30)s to 1/50,000s |
| Lens | 2.8mm, 3.6mm, 6mm optional. Angle of View: 103° (2.8mm), 82.2° (3.6mm), 54° (6mm) |
| Lens Mount | M12 |
| Day & Night | ICR |
| Angle Adjustment | Pan: 0 - 360°, Tilt: 0 - 180°, Rotation: 0 - 360° |
| Synchronization | Internal synchronization |
| Video Frame Rate | 1080p@25fps/1080p@30fps |
| HD Video Output | 1 Analog HD output |
| S/N Ratio | >62dB |
| Power Supply | 12VDC±15% |
| Power Consumption | Max. 4W |
| Protection Level | IP66 |
| IR Range | Up to 20m |
| Dimensions | Φ70 x 149.5mm |
| Weight | 400g |

## Các ứng dụng của [camera Hikvision](https://camerasaigon24h.com/gioi-thieu-ve-camera-HIKVISON-hang-thuong-hieu_cv50)

Phát hiện chuyển động, phát hiện mất tín hiệu, và phát hiện có sự tác động từ bên ngoài lên camera (đặc biệt với Camera IP sẽ có thêm nhiều tính năng báo động cao cấp khác) giúp cảnh báo tới Người sử dụng khi có người xâm nhập các vùng nhạy cảm và quan trọng, camera bị mất tín hiệu hay có sự tác động lên camera. Bên cạnh đó chế độ che vùng riêng tư giúp Người sử dụng đặt các vùng mà camera và đầu ghi không ghi được, đảm bảo sự riêng tư cần thiết.

Các chế độ ghi hình: Liên tục, theo lịch trình đặt sẵn, theo chuyển động, và theo sự báo động trên camera cho phép người sử dụng chủ động trong quá trình lưu trữ hình ảnh: Kéo dài thời gian lưu trữ, dễ dàng tìm kiếm sự kiện khi muốn xem lại.

Người sử dụng được cấp 1 tên miền chính hãng (hik-online.com) trọn đời sản phẩm khi sử dụng đầu ghi và [**Camera IP Hivision**](https://camerasaigon24h.com/gioi-thieu-ve-camera-HIKVISON-hang-thuong-hieu_cv50) để xem và quản lý hệ thống camera từ xa qua mạng internet hay 3G.

Người sử dụng có thể xem từ xa hệ thống camera của mình bằng 3 cách: Trình duyệt Web (Internet explore, Fire Fox, Chrome...), phần mềm cài trên máy tính, và phần mềm cài trên Điện thoại và máy tính bảng.

Phần mềm quản lý hữu hiệu cho phép Người sử dụng có thể quản lý tập trung và kết nối dễ dàng một hoặc nhiều hệ thống camera tại cùng 1 địa điểm lắp đặt hay nhiều địa điểm khác nhau.

## Phần mềm Python

### Giới thiệu chung về python

Python là một ngôn ngữ lập trình được sử dụng phổ biến ngày nay từ trong môi trường học đường cho tới các dự án lớn . Ngôn ngữ phát triển nhiều loại ứng dụng, phần mềm khác nhau như các chương trình chạy trên desktop, server, lập trình các ứng dụng web... Ngoài ra Python cũng là ngôn ngữ ưa thích trong xây dựng các chương trình trí tuệ nhân tạo trong đó bao gồm machine learning. Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng sau này, nó đã chạy trên mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix.

Python do Guido van Rossum tạo ra năm 1990. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

Python là ngôn ngữ có hình thức đơn giản, cú pháp ngắn gọn, sử dụng một số lượng ít các từ khoá, do đó Python là một ngôn ngữ dễ học đối với người mới bắt đầu tìm hiểu.

Python là ngôn ngữ có mã lệnh (source code hay đơn giản là code) không mấy phức tạp. cả trường hợp bạn chưa biết gì về Python bạn cũng có thể suy đoán được ý nghĩa của từng dòng lệnh trong source code.



Hình 2.2: Giao diện phần mềm python

Python có nhiều ứng dụng trên nhiều nền tảng, chương trình phần mềm viết bằng ngôn ngữ Python có thể được chạy trên nhiều nền tảng hệ điều hành khác nhau bao gồm Windows, Mac OSX và Linux.

### Tính năng chính của Python

**Ngôn ngữ lập trình đơn giản, dễ học:** Python có cú pháp rất đơn giản, rõ ràng. Nó dễ đọc và viết hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như C++, Java, C#. Python làm cho việc lập trình trở nên thú vị, cho phép bạn tập trung vào những giải pháp chứ không phải cú pháp.

**Miễn phí, mã nguồn mở:** Bạn có thể tự do sử dụng và phân phối Python, thậm chí là dùng cho mục đích thương mại. Vì là mã nguồn mở, bạn không những có thể sử dụng các phần mềm, chương trình được viết trong Python mà còn có thể thay đổi mã nguồn của nó. Python có một cộng đồng rộng lớn, không ngừng cải thiện nó mỗi lần cập nhật.

**Khả năng di chuyển:** Các chương trình Python có thể di chuyển từ nền tảng này sang nền tảng khác và chạy nó mà không có bất kỳ thay đổi nào. Nó chạy liền mạch trên hầu hết tất cả các nền tảng như Windows, macOS, Linux.

**Khả năng mở rộng và có thể nhúng:** Giả sử một ứng dụng đòi hỏi sự phức tạp rất lớn, bạn có thể dễ dàng kết hợp các phần code bằng C, [C++](https://quantrimang.com/cplusplus) và những ngôn ngữ khác (có thể gọi được từ C) vào code Python. Điều này sẽ cung cấp cho ứng dụng của bạn những tính năng tốt hơn cũng như khả năng scripting mà những ngôn ngữ lập trình khác khó có thể làm được.

**Ngôn ngữ thông dịch cấp cao:** Không giống như [C](https://quantrimang.com/lap-trinh-c)/C++, với Python, bạn không phải lo lắng những nhiệm vụ khó khăn như quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa,... Khi chạy code Python, nó sẽ tự động chuyển đổi code sang ngôn ngữ máy tính có thể hiểu. Bạn không cần lo lắng về bất kỳ hoạt động ở cấp thấp nào.

**Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến:** Python có một số lượng lớn thư viện tiêu chuẩn giúp cho công việc lập trình của bạn trở nên dễ thở hơn rất nhiều, đơn giản vì không phải tự viết tất cả code. Ví dụ: Bạn cần kết nối [cơ sở dữ liệu](https://quantrimang.com/co-so-du-lieu) MySQL trên Web server? Bạn có thể nhập thư viện MySQLdb và sử dụng nó. Những thư viện này được kiểm tra kỹ lưỡng và được sử dụng bởi hàng trăm người. Vì vậy, bạn có thể chắc chắn rằng nó sẽ không làm hỏng code hay ứng dụng của mình.

**Hướng đối tượng:** Mọi thứ trong Python đều là hướng đối tượng. [Lập trình hướng đối tượng](https://quantrimang.com/steve-jobs-dinh-nghia-lap-trinh-huong-doi-tuong-khien-ca-the-gioi-than-phuc-131900) (OOP) giúp giải quyết những vấn đề phức tạp một cách trực quan. Với OOP, bạn có thể phân chia những vấn đề phức tạp thành những tập nhỏ hơn bằng cách tạo ra các đối tượng.

### Các thành phần cơ bản trong Python

Cấu trúc rẽ nhánh: Cấu trúc if dùng để thực thi có điều kiện một khối cụ thể.

**Cấu trúc lặp, gồm lệnh while và vòng lặp for:**

While chạy một khối cụ thể cho đến khi điều kiện lặp có giá trị là sai thì dừng. Vòng lặp for, lặp qua từng phần tử, mỗi phần tử sẽ được đưa vào biến cục bộ để sử dụng với từng khối trong vòng lặp.

**Class:**

Dùng để khai báo lớp (sử dụng trong lập trình hướng đối tượng) và lệnh def dùng để định nghĩa hàm.

**Kiểu dữ liệu.**

Sử dụng Python, thuận lợi cho người mới sử dụng đó là không cần phải khai báo biến. Biến được xem là đã khai báo nếu nó được gán một giá trị lần đầu tiên. Căn cứ vào mỗi lần gán, Python sẽ tự động xác định kiểu dữ liệu của biến. Python có một số kiểu dữ liệu cơ bản sau:

* + Int, long là kiểu số nguyên. Độ dài của kiểu số nguyên là tùy ý, chỉ bị giới hạn bởi bộ nhớ máy tính.
  + float: số thực.
  + Complex: số phức.
  + Str: chuỗi ký tự…

### Thư viện OpenCV

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở, nó là miễn phí cho những ai bắt đầu tiếp cận với các học thuật. OpenCV được ứng dụng trong nhiều lĩnh vực như cho thị giác máy tính hay xử lý ảnh và máy học. Thư viện được lập trình trên các ngôn ngữ cấp cao: C++, C, Python, hay Java và hỗ trợ trên các nền tảng Window, Linux, Mac OS, iOS và Android. OpenCV đã được tạo ra tại Intel vào năm 1999 bởi Gary Bradsky, và ra mắt vào năm 2000. Opencv có rất nhiều ứng dụng: Nhận dạng ảnh, xử lý hình ảnh, phục hồi hình ảnh/video, thực tế ảo...

Ở đề tài này thư viện OpenCV được chạy trên ngôn ngữ Python. OpenCV được dùng làm thư viện chính để xử lý hình ảnh đầu vào và sau đó đi nhận dạng ảnh.

OpenCV Là một thư viện mở nên sử dụng các thuật toán một cách miễn phí, cùng với việc chúng ta cũng có thể đóng góp thêm các thuật toán giúp Thư viện thêm ngày càng phát triển.

**Các tính năng của thư viện OpenCV:**

* + Đối với hình ảnh, chúng ta có thể đọc và lưu hay ghi chúng.
  + Về Video cũng tương tự như hình ảnh cũng có đọc và ghi. • Xử lý hình ảnh có thể lọc nhiễu cho ảnh, hay chuyển đổi ảnh.
  + Thực hiện nhận dạng đặc điểm của hình dạng trong ảnh. Phát hiện các đối tượng xác định được xác định trước như khuôn mặt, mắt, xe trong video hoặc hình ảnh.
  + Phân tích video, ... ước lượng chuyển động của nó, trừ nền ra và theo dõi các đối tượng trong video.

# THIẾT KẾ PHẦN MỀM GIÁM SÁT NHẬN DẠNG SẢN PHẨM QUA CAMERA.

## Yêu cầu công nghệ

Từ các hình ảnh hoặc video sản phẩm công nghiệp hoặc các vật thể có hình dáng kích thước màu sắc khác nhau, cần giám sát nhận dạng các vật thể đó để phục vụ các mục đích tiếp theo như phân loại sản phẩm, kiểm điếm sản phẩm….



Hình .:Ảnh gốc

Từ yêu cầu bài toán trên nhóm nghiên cứu thiết kế phần mềm nhận dạng sản phẩm qua ảnh từ camera dung ngôn ngữ của phần mềm Python.







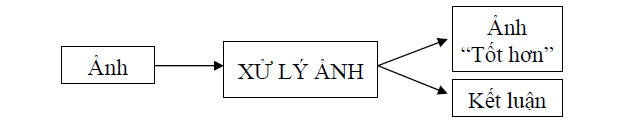
Vật thể, sản phẩm

Hình 3.2:cấu trúc hệ thống giám sát nhận dạng sản phẩm qua camera

## Phần mềm và thuật toán nhận dạng

### Thư viện ngôn ngữ phần mềm python

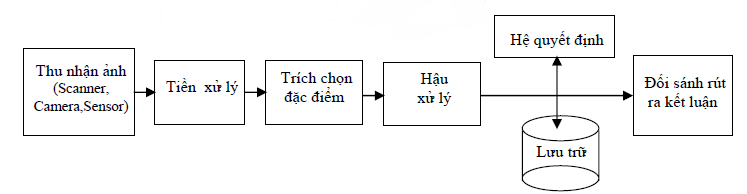
Quá trình xử lý ảnh được xem như là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh “tốt hơn” hoặc một kết luận.



Hình .:Qúa trình xử lý ảnh

Ảnh có thể xem là tập hợp các điểm ảnh và mỗi điểm ảnh được xem như là đặc trưng cường độ sáng hay một dấu hiệu nào đó tại một vị trí nào đó của đối tượng trong không gian và nó có thể xem như một hàm n biến P (c1, c2 ... cn). Do đó, ảnh trong xử lý ảnh có thể xem như ảnh n chiều.

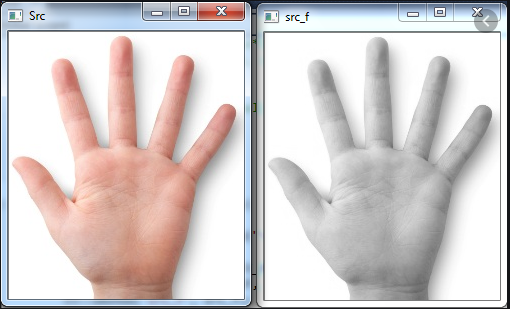
Sơ đồ tổng quát của một hệ thống xử lý ảnh:



Hình 3. 3 Các bước cơ bản trong hệ thống xử lý ảnh

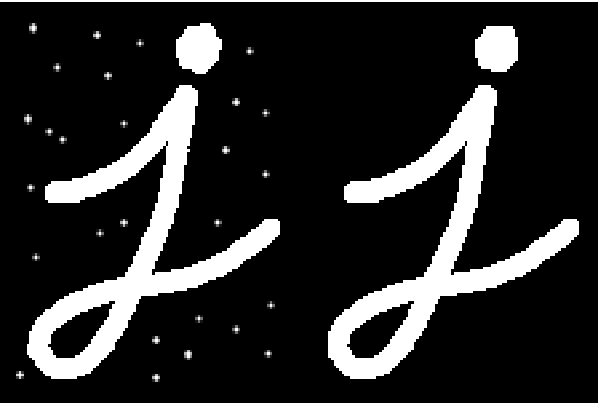
Trong quá trình tìm hiểu và nghiên cứu chúng em có tìm hiểu các thuật toán trong thư viện xử lý ảnh Opencv bằng ngôn ngữ Python, sử dụng phần mềm lập trình là Visual Studio Code, thư viện thiết kế giao diện Pyqt5.

* Thư viện Pyqy5,
  + QPushbutton: PushButton thường được sử dụng để kích hoạt chương trình thực hiện điều gì đó với người dùng chỉ cần nhấn một nút. Điều này có thể bắt đầu tải hoặc xóa một tập tin.
    - Cấu trúc lệnh: pushbutton = QPushButton(label)
      * ‘label’ có thể được bỏ qua nếu không yêu cầu
    - Các phương thức sử dụng trong bài:
      * pushbutton.setText() : hiển thị nội dung trên button .
      * pushbutton.setIcon() : thêm một icon cho nút ấn
      * pushbutton.setIconSize(): chỉnh sử kích thước icon
      * pushbutton.clicked.connect() : kết nối phương thức thực hiện khi pushbutton được click
  + QToolButton : Toolbutton được sử dụng để chọn đường dẫn cho ảnh muốn hiển thị, xử lý.
    - Cấu trúc lệnh : toolbutton = QToolButton()
    - Các phương thức được sử dụng trong bài:
      * toolbutton.setIcon(icon) : tạo icon cho nút ấn
      * toolbutton.setGeometry() : thiết lập vị trí và kích thước cho toolbutton.
      * toolbutton.setIcon (): thêm một icon cho toolbutton
      * toolbutton.setIconSize() : chỉnh kích thước cho icon
      * toolbutton.clicked.connect() : kết nối phương thức được thực hiện khi toolbutton được click
  + QgroupBox : GroupBox cung cấp một cách gọn gàng để nhóm các mục, với khung chứa có nhãn tiêu đề và khung viền
    - Cấu trúc lệnh : groupbox = QGroupBox(title)
    - Các phương thức được sử dụng trong bài:
      * Groupbox.setGeometry() : thiết lập vị trí và kích thước cho Groupbox.
      * Groupbox.setTitle() : tạo tiêu đề cho Groupbox
  + Qlabel : hiển thị văn bản text hoặc hiển thị ảnh
    - Cấu trúc lệnh : label = QLabel(text)
    - Các phương thức được sử dụng trong bài :
      * label.setText(text) : hiển thị văn bản trên label
      * label.setGeometry() : thiết lập vị trí và kích thước cho label
      * label.mousePressEvent = getPos : lấy giá trị tọa độ pixel chuột click vào vị trí trong label
      * label.setPixmap () : hiển thị ảnh
  + Qcombobox : hiển thị, thêm các lựa chọn dữ liệu
    - Cấu trúc lệnh : combobox = QComboBox()
    - Các phương thức được sử dụng trong bài:
      * combobox.setGeometry() : thiết lập vị trí và kích thước Combobox
      * combobox.addItem(text) : thêm lựa chọn dạng text cho box
      * combobox.currentText() : lấy dự liệu của combobox hiện đang hiển thị
  + QFiledialog : lấy đường dẫn file được chọn
    - Cấu trúc lệnh : filedialog = QFileDialog()
    - Các phương thức được sử dụng trong bài :
      * QfileDialog.getOpenFileName () : lấy giá trị đường dẫn được chọn
      * QfileDialog.Options() : điều chỉnh các chế độ lựa chọn, loại đường dẫn cho file
  + Qspinbox: cho phép nhập dữ liệu và hiển thị dữ liệu dạng số, có thể điều chỉnh được dữ liệu qua giao diện
    - Cấu trúc lệnh : spinbox = QSpinBox()
    - Các lệnh được sử dụng trong bài :
      * spinbox.setGeometry() : thiết lập vị trí và kích thước cho spinbox
      * spinbox.setValue(value) : thiết lập giá trị hiển thị cho spinbox
* Thư viện lập trình Opencv :
  + cv2.imread() : đọc ảnh
  + cv2.cvtColor (input\_image, flag): chuyển đổi các dạng ảnh (RGB image, Gray image, Binary image,…). Ex : chuyển ảnh RGB sang ảnh xám ( gray image)

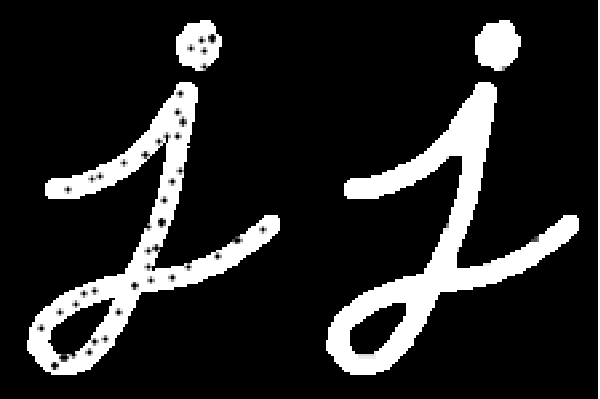
Hình .:Chuyển đôi từ ảnh RGB sang ảnh xám

* + cv2.threshold() : chuyển đổi dạng ảnh xám sang ảnh đen trắng theo ngưỡng sáng được chọn

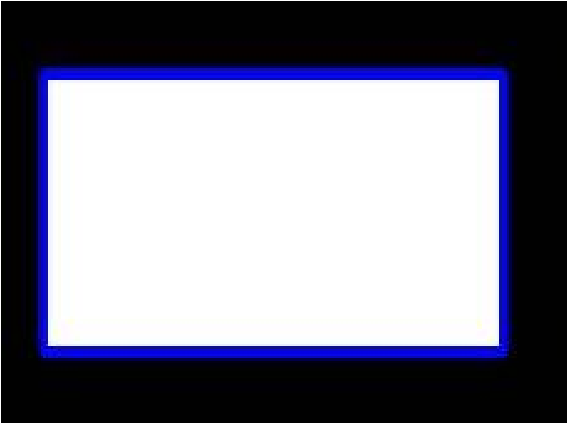
Hình .:chuyển đổi ảnh RGB sang ảnh đen trắng

* + cv2.mophologyEx() : lọc ảnh binary, loại nhiễu
    - Opening:

Hình .:Lọc nhiễu kiểu Opening

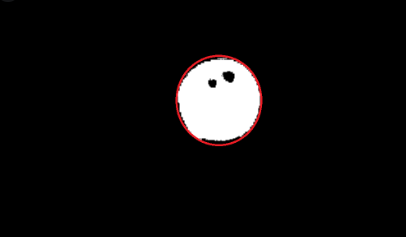
* + - Closing :

Hình .:Lọc nhiễu kiểu closing

* + cv2.findcontours() : tìm các đường biên của đối tượng trong ảnh
  + cv2.drawcontours() : vẽ contours tìm được từ hàm findcontours()

Hình .:Vẽ contours quanh hình chữ nhật

* + cv2. minEnclosingCircle (contour) : lấy giá trị tâm và bán kính đường tròn nhỏ nhất quanh đường biên của đối tượng .



Hình .:Vẽ đường tròn nhỏ nhất

* + cv2.circle() : vẽ đường tròn khi biết tâm và bán kính, có thể lựa chọn thêm màu của đường tròn và độ dày của đường tròn

Hình .:Vẽ đường tròn biết tâm và bán kính

* + cv2.inpaint() : tự động làm phù hợp về màu sắc của đối tượng được chọn với màu của các pixel xung quanh ( thay đổi màu sắc đối tượng theo các pixel xung quanh => sử dụng cho việc xóa đối tượng và giữ lại nền)

Hình .:Xóa nhiễu ảnh

* + cv2.imwrite() : lưu lại ảnh với tên và đường dẫn file ảnh đã được cung cấp, nếu không có đường dẫn, file ảnh sẽ được lưu tại thư mục chứa file .py của chương trình chứa câu lệnh này.

**3.2 Thuật toán và viết phần mềm nhận dạng sản phẩm**

- Thuật toán tổng quát :

Kiểm tra vật thể trên ảnh

Đọc fille ảnh

Khoanh vùng, xác định vị trí ảnh

Xác định vị trí tọa độ hình dạng, kích thước màu sắc vật thể

Vẽ lại vật thể

Hình .:thuật toán tổng quát

-Thuật toán chương trình con di chuyển vật thể:

Kiểm tra vật thể

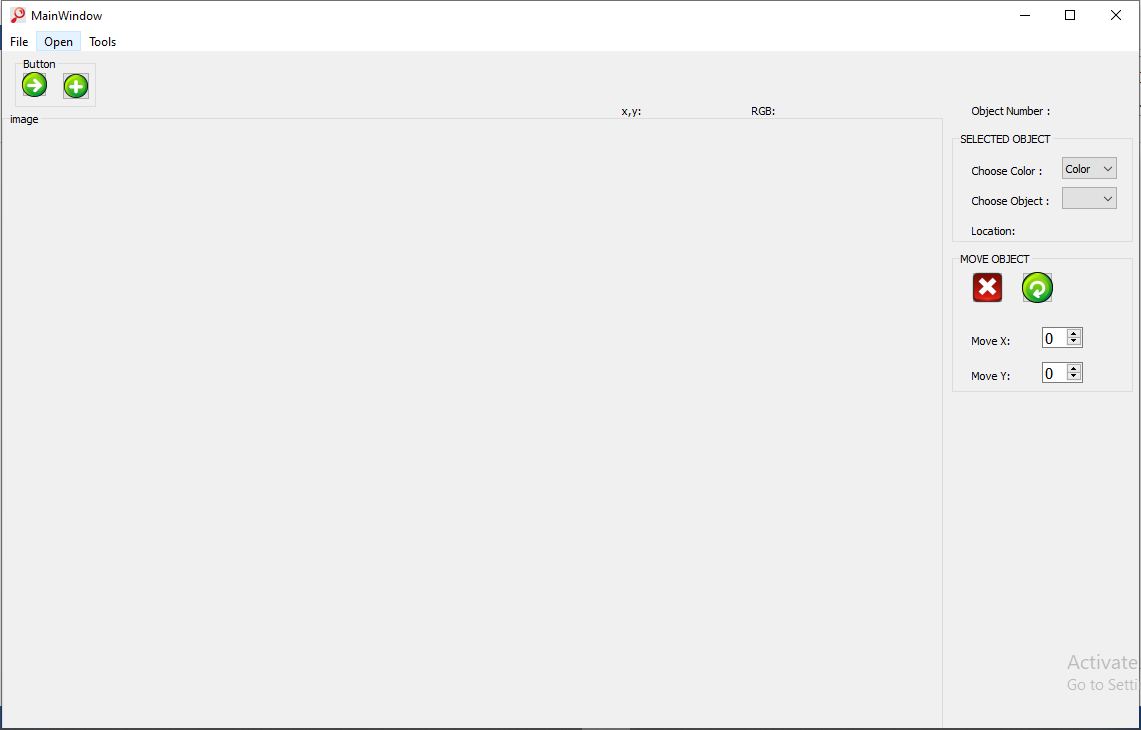
Hiển thị ảnh giao diện mới

Xác định vật thể cần di chuyển

Di chuyển vật thể

Hình .:thuật toán chương trình con di chuyển vật thể

## Viết phần mềm

* Giao diện phần mềm:

Hình .:Ảnh giao diện phần mềm

* Chương trình:

Chương trình Operate.py: (xem phụ lục)

Chương trình Setui.py: (xem phụ lục)

Kết quả phần mềm giám sát nhận dạng sản phẩn qua camera được thể hiện ở chương 4.

# KẾT QUẢ NGHIÊN CỨU VÀ PHÂN TÍCH

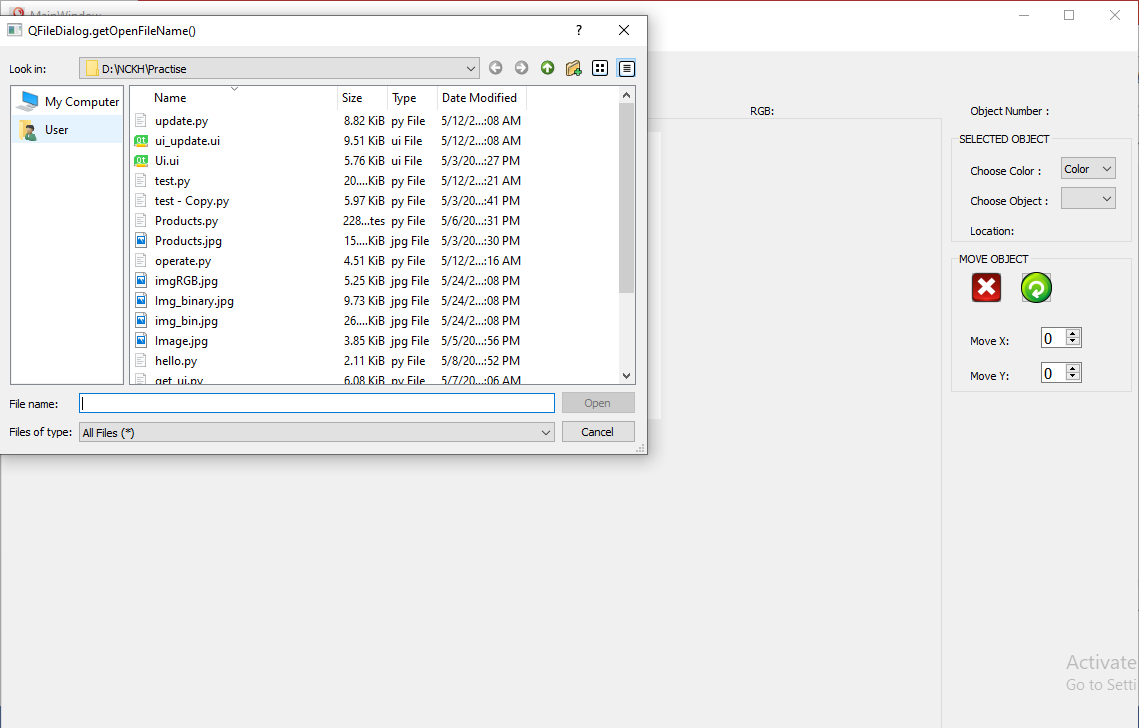
## Kết quả nghiên cứu



Hình .:Giao diện truy cập phần mềm pyton

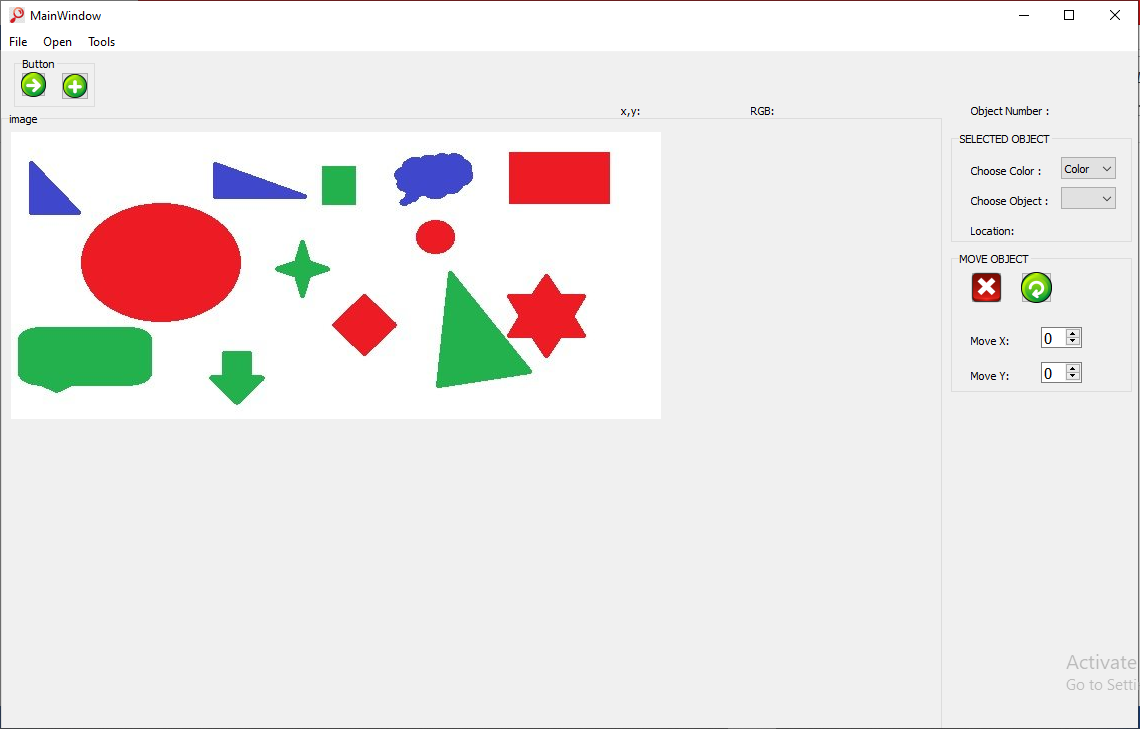
### Giao diện thiết kế

* Lấy ảnh và hiển thị ảnh ( sử dụng nút bấm D:\NCKH\Practise\icon\Button-Add-icon.png ) :
  + Cửa sổ tìm kiếm ảnh sẽ được hiện lên. Chú ý là ta cần chọn đúng file ảnh (thường có đuôi là .jpg, png…)



Hình .:Cửa sổ tìm file ảnh

* + Sau khi click phím open, file ảnh bạn chọn sẽ được hiển thị lên vùng hiển thị ảnh như sau:



Hình .:Hiển thị ảnh gốc

* Chọn màu muốn phân tích tại phần ‘choose color’ của group ‘choose object’. Có ba màu có thể lựa chọn: Green, Blue, Red
  + Sau khi chọn xong, ta có thể trích ra những đối tượng có màu tương ứng xanh lục, xanh dương, đỏ và sẽ được khoanh xung quanh đối tượng có màu đó bằng cách nhất nút ấn D:\NCKH\Practise\icon\Button-Next-icon (1).png .

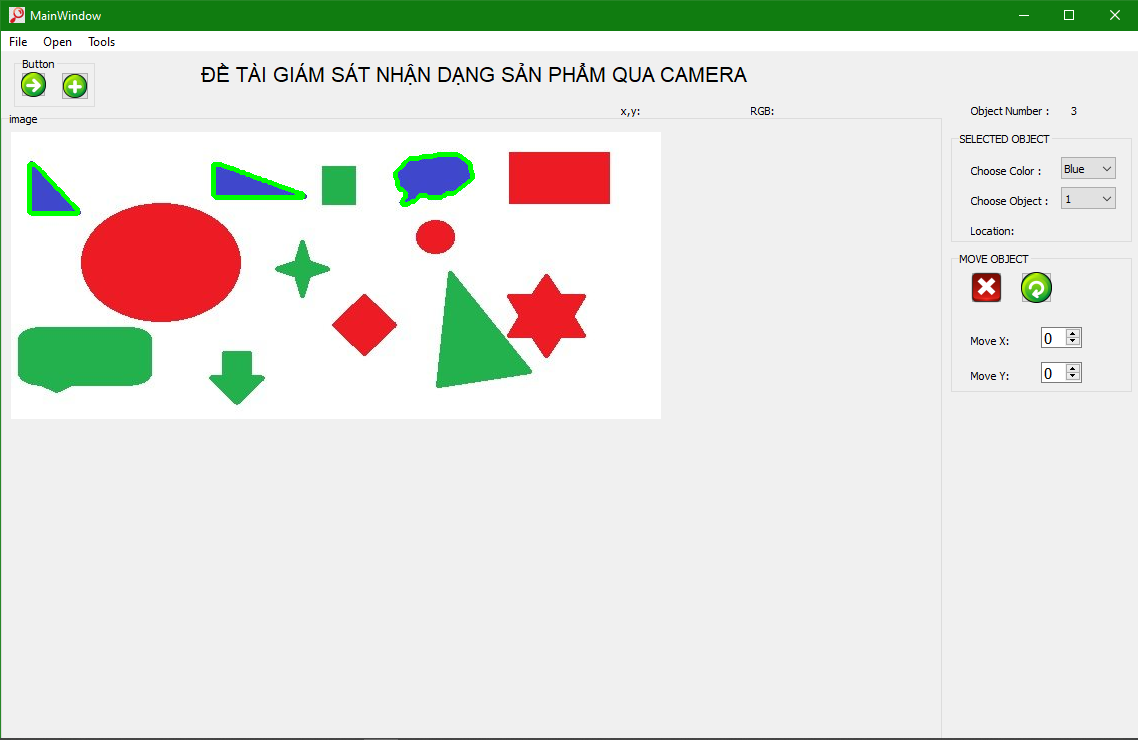
**Ví dụ ta chọn vật thể màu đỏ và đếm số lượng:**



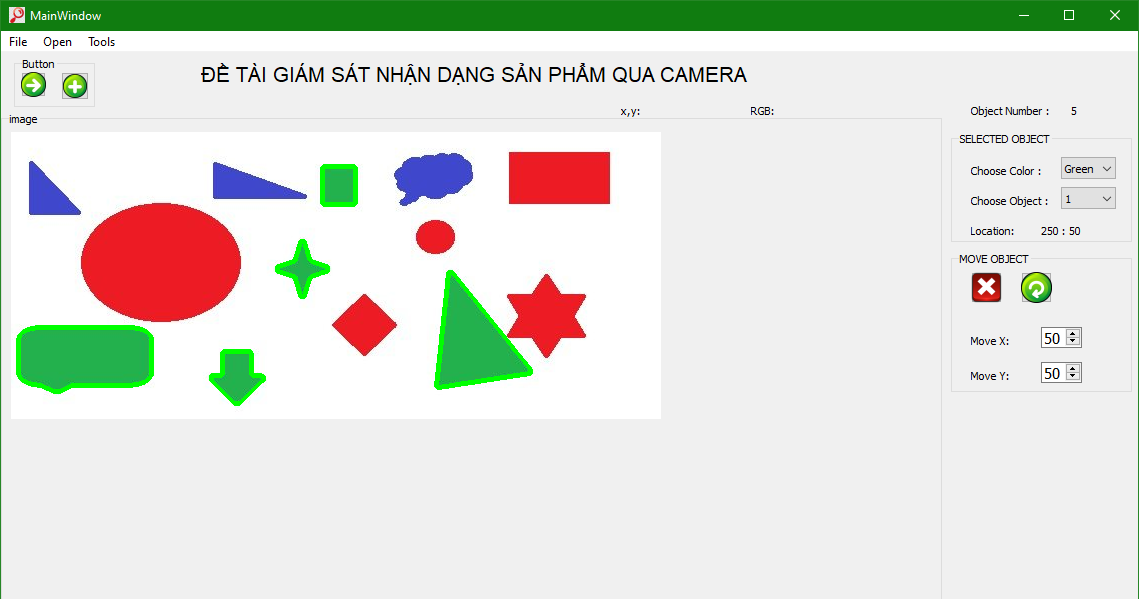
Hình .:hiển thị số lượng vật thể màu đỏ

* Các đối tượng có màu đỏ đã được bao quanh bởi đường màu xanh
* Số lượng vật thể được hiển thị lên trong mục phần Object Number. Cụ thể ở đây ta có 5 đối tượng màu đỏ.
* Tiếp theo ta chọn cụ thể vật thể muốn xóa hoặc di chuyển vị trí trọng mục ‘Choose Object’. Vật thể có điểm pixel cao nhất (giá trị theo trục OY lớn nhất) đạt giá trị lớn nhất (pixel cao nhất của vật màu đỏ đó thấp nhất so với các pixel cao nhất của các vật màu đỏ khác) thì có hệ số nhỏ nhất (1), lần lượt tới các vật thể còn lại có màu đỏ.

**Ví dụ ta chọn vật thể màu xanh dương và đếm số lượng:**

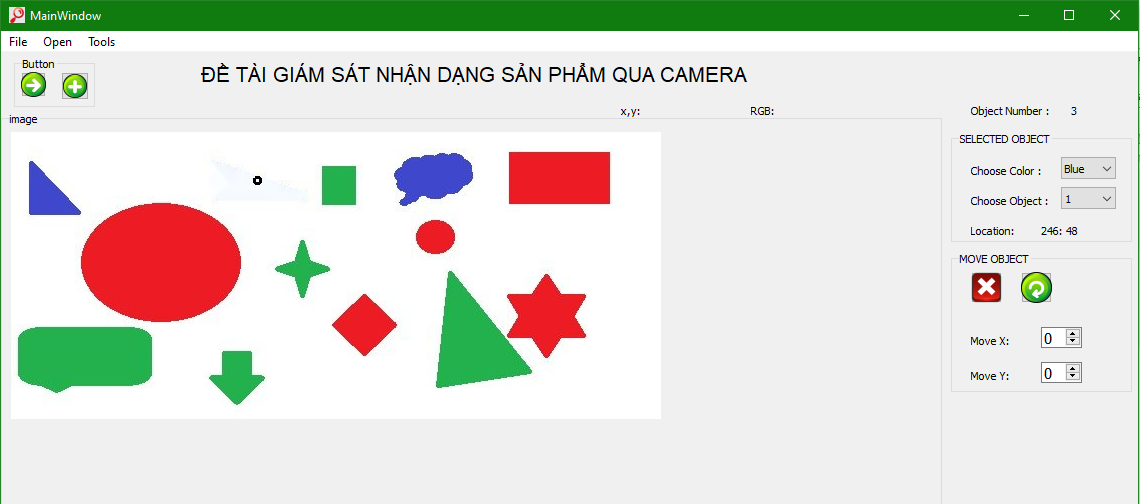
Hình .: xác định số lượng vật thể xanh dương

**Ví dụ ta chọn vật thể màu xanh lá cây và đếm số lượng:**



Hình .:Đếm số lượng vật thể màu xanh lá cây

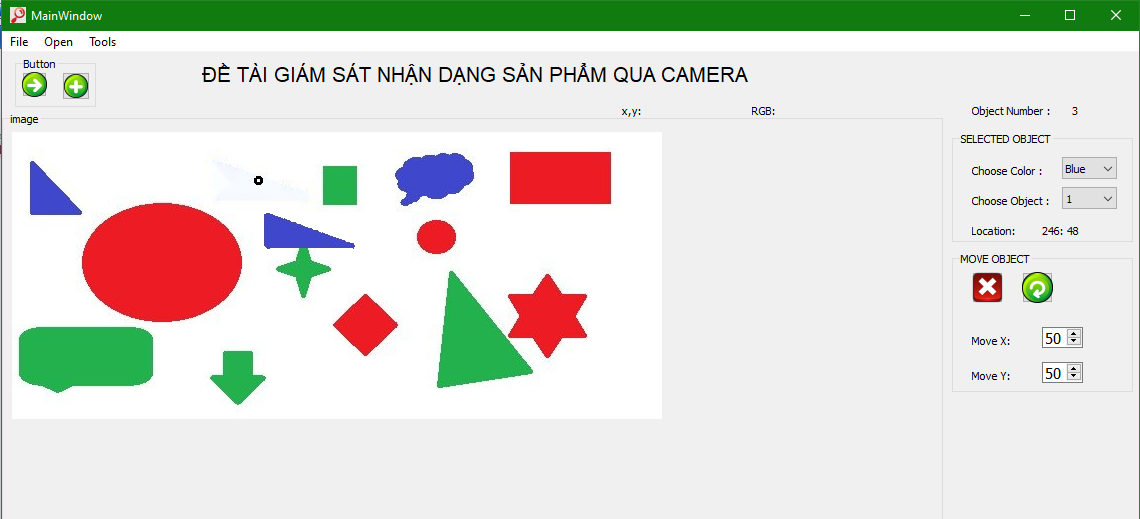
**Ví dụ ta muốn xóa vật thể màu xanh lá cây đối tượng số 1:**



Hình .:Giao diện khi loại bỏ một vật thể số 1

* Đối tượng số 1 ( là đối tượng có điểm pixel cao nhất đạt giá trị theo trục OY nhỏ nhất ) đã bị xóa khi ta nhấn phím D:\NCKH\Practise\icon\Button-Delete-icon.png. Đồng thời tọa độ x, y của đối tượng được hiển thị lên trong phần ‘ Location’
* Di chuyển vật thể: Để di chuyển vật thể khi đã biết ‘Location’, ta cần xác định vecto di chuyển của đối tượng [MoveX, MoveY]. Ta có thể nhấp chuột vào vị trí muốn chuyển tới để lấy tọa độ điểm đó (x, y), sau đó trừ tương ứng giá trị tọa độ của đối tượng bị xóa sẽ tìm được vecto di chuyển của đối tượng. Sau khi điền giá trị [MoveX,MoveY], ta ấn phím D:\NCKH\Practise\icon\Button-Reload-icon.png để bắt đầu di chuyển vật thể.

**Ví dụ ta muốn di chuyển đối tượng màu đỏ trên theo vecto [50 50]:**

Hình .:Giao diện khi di chuyển vật thể màu đỏ

*\* Như vậy phần mềm đã thực hiện được chức năng nhận dạng và giám sát vật thể (hoặc sản phẩm theo màu sắc và hình dạng). Tương tự muốn nhận dạng file ảnh khác ta thực hiện truy cập lại file ảnh được chụp qua camera, phần mềm sẽ lọc và xuất ra các vật thể, sản phẩm*.

## Kết luận

Sau thời gian nghiên cứu, nhóm tác giả đã thiết kế được phần mềm có khả năng nhận diện vật thể qua màu sắc, hình dạng, có xác định được chính xác tọa độ để có thể sử dụng kết nối các tay máy hoặc rô bốt để lấy chính xác các vật thể đó.

Về ứng dụng: Ta có thể ứng dụng phần mềm cho phân loại sản phẩm trong công nghiệp bằng màu sắc hoặc hình dạng, kích thước, giám sát sản phẩm được qua camera, hoặc ứng dụng trong lĩnh vực phân loại sản phẩm trong nông nghiệp như phân loại cà chua, hạt điều, cà phê, đếm tôm giống…

## Hướng phát triển

Thiết kế thêm phần truyền thông kết nối với PLC, vi xử lý, rô bốt công nghiệp, AVG… để thực hiện công việc phân loại sản phẩm theo yêu cầu cho trước.

Áp dụng các thuật toán nhận dạng xử lý ảnh hiện đại như dùng thuật toán noron, AI để nâng cao chất lượng xử lý nhận dạng ảnh chính xác ngay cả khi c ảnh thu được bị mờ.

Ứng dụng phát triển khác của đề tài: tương tự như vậy phần mềm có thể phát triển để ứng dụng nhận dạng các sản phẩm vật thể khác như nhận dạng hạt điều để phân loại hạt điều, nhận dạng màu sắc cà chua để phân loại cà chua, nhận dạng hình ảnh tôm giống để đếm số lượng tôm giống như sau:

Ví dụ: Nhận dạng tôm giống qua camera dung để đếm số lượng tôm giống.

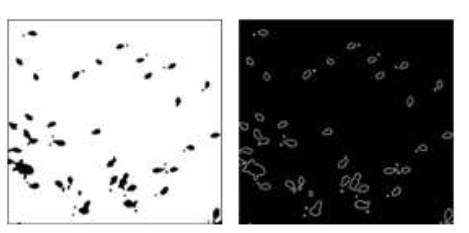


Hình .:Ảnh đếm số lượng tôm giống truyền thống bằng cốc



Hình .:Ảnh chụp tôm giống

Nhận dạng tôm giống từ 7 đến 13 ngày tuổi để đếm tôm giống qua camera.



Hình .:Ảnh gốc tôm giống và ảnh nhận dạng tôm giống qua phần mềm

Qua phần mềm xử lý ảnh ta thu được ảnh nhận dạng tôm giống, từ đó sẽ xuất ra số lượng tôm giống trên ảnh.

# Tài liệu tham khảo

[1]. TS. Đỗ Đăng Toàn,TS. Phạm Việt Bình(2007), *Giáo trình Xử lý ảnh***,** NXB KH&KT.

[2]. TS.Nguyễn Thanh Hải, Giáo Trình Xử Lý Ảnh, NXB Đại Học Quốc Gia Tp.Hồ Chí Minh.

[3]. Phạm Minh Hà, 2016, Kỹ thuật mạch điện tử, NXB KHKT.

[4]. Đặng Lê Khoa,2014, Truyền thông không dây, NXB KHKT.

[5]. Phạm Thượng Hàn, 2009, *Xử lý số tín hiệu và ứng dụng*, NXB Giáo dục.

[6]. Oppenheim A.V.Willsky A.S, 2007, *Signals and Systems*, Prentice Hall of India.

**Phụ lục**

Chương trình (code) của phần mềm:

* + ***Operate.py***

*import cv2 as cv*

*import numpy as np*

*from matplotlib import pyplot as plt*

*import math*

*from PyQt5 import QtCore, QtGui, QtWidgets*

*import operate*

*import sys*

*class Operate ():*

*def DelObj(self,imgRGB,num,r1,g1,b1,r2,g2,b2):*

*img= cv.cvtColor(imgRGB,cv.COLOR\_BGR2GRAY)*

*img1= img.copy()*

*# \_,img\_bin= cv.threshold(img,200,255,cv.THRESH\_BINARY\_INV)*

*img\_gray = cv.cvtColor(imgRGB,cv.COLOR\_BGR2GRAY)*

*\_,img\_bin= cv.threshold(img\_gray,127,255,cv.THRESH\_BINARY)*

*h,w = img\_gray.shape*

*for i in range(h):*

*for j in range(w):*

*if imgRGB[i,j][0]>b1 and imgRGB[i,j][0]<b2 and imgRGB[i,j][1]>g1 and imgRGB[i,j][1]<g2 and imgRGB[i,j][2]>r1 and imgRGB[i,j][2]<r2:*

*img\_bin[i,j]=255*

*else : img\_bin[i,j]=0*

*kernel2 = np.ones((5,5),np.uint8)*

*img\_bin = cv.morphologyEx(img\_bin, cv.MORPH\_CLOSE, kernel2)*

*cv.imwrite('Img\_binary.jpg',img\_bin)*

*kernel2 = np.ones((5,5),np.uint8)*

*opening = cv.morphologyEx(img\_bin, cv.MORPH\_OPEN, kernel2)*

*img\_bin = cv.morphologyEx(opening, cv.MORPH\_CLOSE, kernel2)*

*contours,hierarchy = cv.findContours(img\_bin, 1, 2)*

*print('length of contours',len(contours))*

*contour=contours[num]*

*(x,y),radius = cv.minEnclosingCircle(contour)*

*center = (int(x),int(y))*

*radius = int(radius)*

*img = cv.circle(img,center,radius,(0,255,0),2)*

*(x,y),radius = cv.minEnclosingCircle(contour)*

*center = (int(x),int(y))*

*radius = int(radius)*

*print('x,y,radius: ',int(x),int(y),radius)*

*# print(img.shape)*

*h,w= img.shape*

*for i in range(h):*

*for j in range(w):*

*if pow(i-y,2)+pow(j-x,2)< pow(radius+5,2):*

*pass*

*else: img[i,j]=255*

*\_,img\_bin= cv.threshold(img,200,255,cv.THRESH\_BINARY\_INV)*

*dst = cv.inpaint(imgRGB,img\_bin,3,cv.INPAINT\_TELEA)*

*# cv.circle(img\_original,(i[0],i[1]),2,(0,0,255),7)*

*dst = cv.circle(dst,center,3,(0,0,0),2)*

*cv.imwrite('img\_bin.jpg',dst)*

*# dst = cv.inpaint(img,img\_bin,3,cv.INPAINT\_TELEA)*

*cv.imwrite('imgRGB.jpg',img\_bin)*

*return dst,len(contours),int(x),int(y),radius*

*def GetLengthImgae(self,imgRGB,r1,g1,b1,r2,g2,b2):*

*img= cv.cvtColor(imgRGB,cv.COLOR\_BGR2GRAY)*

*img1= img.copy()*

*img\_gray = cv.cvtColor(imgRGB,cv.COLOR\_BGR2GRAY)*

*\_,img\_bin= cv.threshold(img\_gray,127,255,*

*cv.THRESH\_BINARY)*

*h,w = img\_gray.shape*

*for i in range(h):*

*for j in range(w):*

*if (imgRGB[i,j][0]>b1 and imgRGB[i,j][0]<b2 and*

*imgRGB[i,j][1]>g1 and imgRGB[i,j][1]<g2 and imgRGB[i,j][2]>r1 and*

*imgRGB[i,j][2]<r2):*

*img\_bin[i,j]=255*

*else : img\_bin[i,j]=0*

*kernel2 = np.ones((5,5),np.uint8)*

*img\_bin = cv.morphologyEx(img\_bin, cv.MORPH\_CLOSE,*

*kernel2)*

*cv.imwrite('Img\_binary.jpg',img\_bin)*

*kernel2 = np.ones((5,5),np.uint8)*

*opening = cv.morphologyEx(img\_bin, cv.MORPH\_OPEN,*

*kernel2)*

*img\_bin = cv.morphologyEx(opening, cv.MORPH\_CLOSE,*

*kernel2)*

*contours,hierarchy = cv.findContours(img\_bin, 1, 2)*

*cv.drawContours(imgRGB, contours, -1, (0,255,0), 3)*

*return imgRGB, len(contours)*

*def hello(self,imgRGB,num,r1,g1,b1,r2,g2,b2):*

*img,length,x,y,radius= Operate.DelObj(self,imgRGB,*

*num,r1,g1,b1,r2,g2,b2)*

*return img,x,y*

*def MovOb(self,imgRGB,num,pixel\_num1,pixel\_num2,r1,g1,*

*b1,r2,g2,b2):*

*img\_del,\_,x,y,radius= Operate.DelObj(self,imgRGB,*

*num,r1,g1,b1,r2,g2,b2)*

*h,w ,\_= imgRGB.shape*

*for i in range(h):*

*for j in range(w):*

*if imgRGB[i,j][0]>b1 and imgRGB[i,j][0]<b2*

*and imgRGB[i,j][1]>g1 and imgRGB[i,j][1]<g2 and imgRGB[i,j][2]>r1 and imgRGB[i,j][2]<r2 and pow(i-y,2)+ pow(j-x,2) < pow(radius+2,2):*

*try:*

*img\_del[i+pixel\_num1,j+pixel\_num2]=*

*imgRGB[i,j]*

*except: pass*

*else : pass*

*return img\_del*

*def GetRGB(self, img, x,y):*

*r = img[x,y][2]*

*g = img[x,y][1]*

*b = img[x,y][0]*

*return r,g,b*

* + ***Setui.py :***

*from PyQt5 import QtCore, QtGui, QtWidgets*

*import operate*

*import sys*

*import cv2 as cv*

*import numpy as np*

*from PyQt5.QtCore import \**

*from PyQt5.QtGui import \**

*from PyQt5.QtWidgets import \**

*from PyQt5.QtGui import QPainter, QColor, QPen*

*from PyQt5.QtGui import QIcon*

*class Ui\_MainWindow(object):*

*def setupUi(self, MainWindow):*

*MainWindow.setObjectName("MainWindow")*

*MainWindow.resize(1137, 860)*

*MainWindow.setWindowIcon(QIcon(QPixmap(*

*"icon/finder.jpg")))*

*self.centralwidget = QtWidgets.QWidget(MainWindow)*

*self.centralwidget.setObjectName("centralwidget")*

*self.result = QtWidgets.QPushButton(*

*self.centralwidget)*

*self.result.setGeometry(QtCore.QRect(20, 20, 25, 25))*

*self.result.setText("")*

*self.result.setObjectName("pushButton")*

*self.result.setIcon(QIcon(QPixmap(*

*"icon/Button-Next-icon.png")))*

*self.result.setIconSize(QtCore.QSize(25,25))*

*self.groupBox = QtWidgets.QGroupBox(*

*self.centralwidget)*

*self.groupBox.setGeometry(QtCore.QRect(*

*0, 60, 941, 761))*

*self.groupBox.setObjectName("groupBox")*

*self.display\_image = QtWidgets.QLabel(self.groupBox)*

*self.display\_image.setGeometry(QtCore.QRect(10, 20, 921, 721))*

*self.display\_image.setText("")*

*self.display\_image.setObjectName("label")*

*self.display\_image.setAlignment(QtCore.Qt.*

*AlignLeading|QtCore.Qt.AlignLeft|QtCore.Qt.AlignTop)*

*self.image = QtWidgets.QToolButton(self.centralwidget)*

*self.image.setGeometry(QtCore.QRect(60, 20, 28, 28))*

*self.image.setText("")*

*self.image.setObjectName("toolButton")*

*self.image.setIcon(QIcon(QPixmap(*

*"icon/Button-Add-icon.png")))*

*self.image.setIconSize(QtCore.QSize(25,25))*

*self.groupBox\_2 = QtWidgets.QGroupBox(*

*self.centralwidget)*

*self.groupBox\_2.setGeometry(QtCore.QRect(*

*3, 5, 81, 51))*

*self.groupBox\_2.setObjectName("groupBox\_2")*

*self.label\_2 = QtWidgets.QLabel(self.centralwidget)*

*self.label\_2.setGeometry(QtCore.QRect(*

*970, 50, 81, 16))*

*self.label\_2.setObjectName("label\_2")*

*self.object\_num = QtWidgets.QLabel(*

*self.centralwidget)*

*self.object\_num.setGeometry(QtCore.QRect(*

*1070, 52, 47, 13))*

*self.object\_num.setText("")*

*self.object\_num.setObjectName("object\_num")*

*self.label\_3 = QtWidgets.QLabel(self.centralwidget)*

*self.label\_3.setGeometry(QtCore.QRect(*

*970, 110, 81, 16))*

*self.label\_3.setObjectName("label\_3")*

*self.color = QtWidgets.QComboBox(self.centralwidget)*

*self.color.setGeometry(QtCore.QRect(*

*1060, 105, 55, 22))*

*self.color.setObjectName("color")*

*self.color.addItem("Color")*

*self.color.addItem("Red")*

*self.color.addItem("Green")*

*self.color.addItem("Blue")*

*self.label\_4 = QtWidgets.QLabel(self.centralwidget)*

*self.label\_4.setGeometry(QtCore.QRect(*

*970, 140, 81, 16))*

*self.label\_4.setObjectName("label\_4")*

*self.object\_range = QtWidgets.QComboBox(*

*self.centralwidget)*

*self.object\_range.setGeometry(QtCore.QRect(*

*1060, 135, 55, 22))*

*self.object\_range.setObjectName("object\_range")*

*self.label\_5 = QtWidgets.QLabel(self.centralwidget)*

*self.label\_5.setGeometry(QtCore.QRect(*

*970, 170, 51, 16))*

*self.label\_5.setObjectName("label\_5")*

*self.toa\_do = QtWidgets.QLabel(self.centralwidget)*

*self.toa\_do.setGeometry(QtCore.QRect(*

*1040, 170, 91, 16))*

*self.toa\_do.setText("")*

*self.toa\_do.setObjectName("label\_6")*

*self.dele = QtWidgets.QPushButton(self.centralwidget)*

*self.dele.setGeometry(QtCore.QRect(970, 220, 31, 31))*

*self.dele.setText("")*

*self.dele.setObjectName("dele")*

*self.dele.setIcon(QIcon(QPixmap(*

*"icon/Button-Delete-icon.png")))*

*self.dele.setIconSize(QtCore.QSize(31,31))*

*self.rec = QtWidgets.QPushButton(self.centralwidget)*

*self.rec.setGeometry(QtCore.QRect(1020, 220, 31, 31))*

*self.rec.setText("")*

*self.rec.setObjectName("rec")*

*self.rec.setIcon(QIcon(QPixmap(*

*"icon/Button-Reload-icon.png")))*

*self.rec.setIconSize(QtCore.QSize(31,31))*

*self.label\_7 = QtWidgets.QLabel(self.centralwidget)*

*self.label\_7.setGeometry(QtCore.QRect(*

*970, 280, 51, 16))*

*self.label\_7.setObjectName("label\_7")*

*self.sp1 = QtWidgets.QSpinBox(self.centralwidget)*

*self.sp1.setGeometry(QtCore.QRect(1040, 275, 41, 21))*

*self.sp1.setMaximum(2000)*

*self.sp1.setMinimum(-2000)*

*font = QtGui.QFont()*

*font.setPointSize(11)*

*self.sp1.setFont(font)*

*self.sp1.setObjectName("sp1")*

*self.label\_8 = QtWidgets.QLabel(self.centralwidget)*

*self.label\_8.setGeometry(QtCore.QRect(*

*970, 315, 51, 16))*

*self.label\_8.setObjectName("label\_8")*

*self.sp2 = QtWidgets.QSpinBox(self.centralwidget)*

*self.sp2.setGeometry(QtCore.QRect(1040, 310, 41, 21))*

*self.sp2.setMaximum(2000)*

*self.sp2.setMinimum(-2000)*

*font = QtGui.QFont()*

*font.setPointSize(11)*

*self.sp2.setFont(font)*

*self.sp2.setObjectName("sp2")*

*self.groupBox\_2.raise\_()*

*self.groupBox.raise\_()*

*self.result.raise\_()*

*self.image.raise\_()*

*self.label\_2.raise\_()*

*self.object\_num.raise\_()*

*self.label\_3.raise\_()*

*self.color.raise\_()*

*self.label\_4.raise\_()*

*self.object\_range.raise\_()*

*self.label\_5.raise\_()*

*self.toa\_do.raise\_()*

*self.dele.raise\_()*

*self.rec.raise\_()*

*self.label\_7.raise\_()*

*self.sp1.raise\_()*

*self.label\_8.raise\_()*

*self.sp2.raise\_()*

*self.groupBox\_3 = QtWidgets.QGroupBox(*

*self.centralwidget)*

*self.groupBox\_3.setGeometry(QtCore.QRect(*

*950, 80, 181, 111))*

*self.groupBox\_3.setObjectName("groupBox\_3")*

*self.groupBox\_3.lower()*

*self.groupBox\_4 = QtWidgets.QGroupBox(*

*self.centralwidget)*

*self.groupBox\_4.setGeometry(QtCore.QRect(*

*950, 200, 181, 141))*

*self.groupBox\_4.setObjectName("groupBox\_4")*

*self.groupBox\_4.lower()*

*self.groupBox\_3.autoFillBackground()*

*self.groupBox\_4.autoFillBackground()*

*self.label\_9 = QtWidgets.QLabel(self.centralwidget)*

*self.label\_9.setGeometry(QtCore.QRect(*

*620, 50, 21, 16))*

*self.label\_9.setObjectName("label\_9")*

*self.xy\_value = QtWidgets.QLabel(self.centralwidget)*

*self.xy\_value.setGeometry(QtCore.QRect(*

*650, 50, 71, 16))*

*self.xy\_value.setText("")*

*self.xy\_value.setObjectName("xy\_value")*

*self.RGB\_value = QtWidgets.QLabel(self.centralwidget)*

*self.RGB\_value.setGeometry(QtCore.QRect(*

*790, 50, 101, 16))*

*self.RGB\_value.setText("")*

*self.RGB\_value.setObjectName("RGB\_value")*

*self.label\_12 = QtWidgets.QLabel(self.centralwidget)*

*self.label\_12.setGeometry(QtCore.QRect(*

*750, 50, 31, 16))*

*self.label\_12.setObjectName("label\_12")*

*MainWindow.setCentralWidget(self.centralwidget)*

*self.menubar = QtWidgets.QMenuBar(MainWindow)*

*self.menubar.setGeometry(QtCore.QRect(*

*0, 0, 1137, 21))*

*self.menubar.setObjectName("menubar")*

*self.menuFile = QtWidgets.QMenu(self.menubar)*

*self.menuFile.setObjectName("menuFile")*

*self.menuOpen = QtWidgets.QMenu(self.menubar)*

*self.menuOpen.setObjectName("menuOpen")*

*self.menuTools = QtWidgets.QMenu(self.menubar)*

*self.menuTools.setObjectName("menuTools")*

*MainWindow.setMenuBar(self.menubar)*

*self.statusbar = QtWidgets.QStatusBar(MainWindow)*

*self.statusbar.setObjectName("statusbar")*

*MainWindow.setStatusBar(self.statusbar)*

*self.menubar.addAction(self.menuFile.menuAction())*

*self.menubar.addAction(self.menuOpen.menuAction())*

*self.menubar.addAction(self.menuTools.menuAction())*

*self.dele.clicked.connect(self.DeleteOb)*

*self.result.clicked.connect(self.ShowResult)*

*self.image.clicked.connect(self.GetImg)*

*self.rec.clicked.connect(self.MoveOb)*

*self.display\_image.mousePressEvent = self.getPos*

*# self.display\_image.alignment(QtCore.Qt.AlignTop())*

*self.retranslateUi(MainWindow)*

*QtCore.QMetaObject.connectSlotsByName(MainWindow)*

*MainWindow.show()*

*def retranslateUi(self, MainWindow):*

*\_translate = QtCore.QCoreApplication.translate*

*MainWindow.setWindowTitle(\_translate("MainWindow",*

*"MainWindow"))*

*self.groupBox.setTitle(\_translate("MainWindow",*

*"image"))*

*self.groupBox\_2.setTitle(\_translate("MainWindow",*

*"Button"))*

*self.label\_2.setText(\_translate("MainWindow",*

*"Object Number :"))*

*self.label\_3.setText(\_translate("MainWindow",*

*"Choose Color :"))*

*self.label\_4.setText(\_translate("MainWindow",*

*"Choose Object :"))*

*self.label\_5.setText(\_translate("MainWindow",*

*"Location: "))*

*self.label\_7.setText(\_translate("MainWindow",*

*"Move X:"))*

*self.label\_8.setText(\_translate("MainWindow",*

*"Move Y:"))*

*self.menuFile.setTitle(\_translate("MainWindow",*

*"File"))*

*self.menuOpen.setTitle(\_translate("MainWindow",*

*"Open"))*

*self.menuTools.setTitle(\_translate("MainWindow",*

*"Tools"))*

*self.color.setItemText(0, \_translate("MainWindow",*

*"Color"))*

*self.color.setItemText(1, \_translate("MainWindow",*

*"Red"))*

*self.color.setItemText(2, \_translate("MainWindow",*

*"Green"))*

*self.color.setItemText(3, \_translate("MainWindow",*

*"Blue"))*

*self.groupBox\_3.setTitle(\_translate("MainWindow",*

*"SELECTED OBJECT"))*

*self.groupBox\_4.setTitle(\_translate("MainWindow",*

*"MOVE OBJECT"))*

*self.label\_9.setText(\_translate("MainWindow",*

*"x,y:"))*

*self.label\_12.setText(\_translate("MainWindow",*

*"RGB:"))*

*def getPos(self , event):*

*x = event.pos().x()*

*y = event.pos().y()*

*r,g,b = operate.Operate.GetRGB(self,self.img,y,x)*

*self.xy\_value.setText(str(x)+' , '+str(y))*

*self.RGB\_value.setText(str(r)+' , '+str(g)+' ,*

*'+str(b))*

*print (str(x),str(y))*

*def test(self):*

*print('test OK!')*

*def ShowResult(self):*

*# r1,g1,b1,r2,g2,b2 ,3,180,0,0,255,100,100*

*if self.color.currentText()== 'Red':*

*self.r1=180*

*self.g1=0*

*self.b1=0*

*self.r2=255*

*self.g2=100*

*self.b2=  100*

*elif  self.color.currentText()=='Green':*

*self.r1=0*

*self.g1=120*

*self.b1=20*

*self.r2=100*

*self.g2=220*

*self.b2=  120*

*else:*

*self.r1=0*

*self.g1=20*

*self.b1=150*

*self.r2=120*

*self.g2=120*

*self.b2=  250*

*img= self.img.copy()*

*img,length=operate.Operate.GetLengthImgae(self,img,*

*self.r1,self.g1,self.b1,self.r2,self.g2,self.b2)*

*print('img,length',length)*

*self.object\_num.setText(str(length))*

*self.object\_range.clear()*

*for i in range(length):*

*self.object\_range.addItem(str(i+1))*

*self.ShowImage(img)*

*def DeleteOb(self,ShowResult):*

*num= self.object\_range.currentText()*

*num=int(num)-1*

*image= self.img.copy()*

*image,x,y= operate.Operate.hello(self,image,num,*

*self.r1,self.g1,self.b1,self.r2,self.g2,self.b2)*

*self.toa\_do.setText(str(x)+": "+str(y))*

*self.ShowImage(image)*

*def GetImg(self):*

*print('Get image')*

*appli= App()*

*self.img=cv.imread(appli.ImgPath)*

*print(self.img)*

*self.ShowImage(self.img)*

*def MoveOb(self):*

*num= self.object\_range.currentText()*

*num=int(num)-1*

*img= self.img.copy()*

*pixel\_num1= self.sp2.value()*

*pixel\_num2= self.sp1.value()*

*img\_del=operate.Operate.MovOb(self,img,num,*

*pixel\_num1,pixel\_num2,self.r1,self.g1,self.b1,self.r2,self.*

*g2,self.b2)*

*self.ShowImage(img\_del)*

*def ShowImage(self,img):*

*height, width, channel = img.shape*

*bytesPerLine = 3 \* width*

*img = QImage(img.data, width, height, bytesPerLine, QImage.Format\_BGR888)*

*self.display\_image.setPixmap(QtGui.QPixmap(img))*

*class App(QWidget):*

*def \_\_init\_\_(self):*

*super().\_\_init\_\_()*

*self.title = 'PyQt5 file dialogs - pythonspot.com'*

*self.left = 10*

*self.top = 10*

*self.width = 640*

*self.height = 480*

*self.initUI()*

*def initUI(self):*

*self.setWindowTitle(self.title)*

*self.setGeometry(self.left, self.top, self.width,*

*self.height)*

*self.ImgPath= self.openFileNameDialog()*

*def openFileNameDialog(self):*

*options = QFileDialog.Options()*

*options |= QFileDialog.DontUseNativeDialog*

*fileName, \_ = QFileDialog.getOpenFileName(self,*

*"QFileDialog.getOpenFileName()", "","All Files (\*);;Image Files (\*.jpg \*.png)", options=options)*

*return fileName*

*class start(Ui\_MainWindow,App):*

*app = QtWidgets.QApplication(sys.argv)*

*MainWindow = QtWidgets.QMainWindow()*

*Ui\_MainWindow= Ui\_MainWindow()*

*Ui\_MainWindow.setupUi(MainWindow)*

*sys.exit(app.exec\_())*