****

**BÁO CÁO BÀI TẬP**

**TÌM VẬT THỂ TRONG ẢNH**

Sinh viên: Nguyễn Đức Anh

Mã sinh viên : 22022661

Link mã nguồn:

1. **Mô tả yêu cầu bài toán.**

- Yều cầu bài toán khi người dùng click vào 1 ảnh đồ vật hệ thống sẽ xử lý và đánh dấu vào đồ vật đó tương ứng trong bức ảnh lớn.

Ví dụ:



Khi người dùng click chuột vào ảnh que kem thì hệ thống phải đánh dấu vào vào hình ảnh que kem bên trong bức hình lớn.



1. **Chi tiết Usecase.**

- Mục tiêu: Phát hiện vật thể và vẽ bounding box tương ứng.

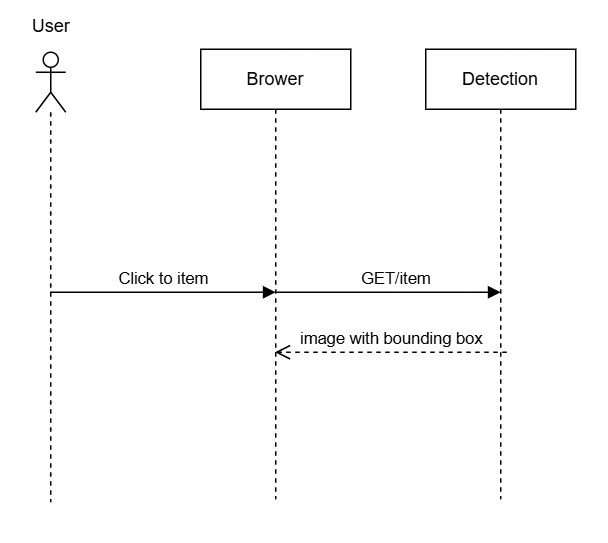
- Điều kiện tiên quyết: Người dùng phải click đúng vào ảnh đồ vật tương ứng.

- Điều kiện hậu: Vẽ bounding box tương ứng với vật thể

- Luồng chính:

1. Người dùng click vào sản phẩm.
2. Máy chủ nhận thông tin về đồ vật người dùng vừa click.
3. Xứ lý ảnh phát hiện vật thể tương ứng
4. Trả về ảnh đã được vẽ bounding box tương ứng.
5. **Thiết kế**

- Sơ đồ thiết kế hệ thống.



1. **Triển khai.**

* Phương pháp để xử lý ảnh và vẽ bounding box.

1. Sau khi nhận được đồ vật mà người dùng vừa click. Gọi hàm *detection().* Với input của hàm này là chuỗi string là tên của đồ vật đó và output là ảnh sau khi đã được vẽ bounding box.

detection(item):

image\_gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

template\_image = get\_image\_from\_path(item)

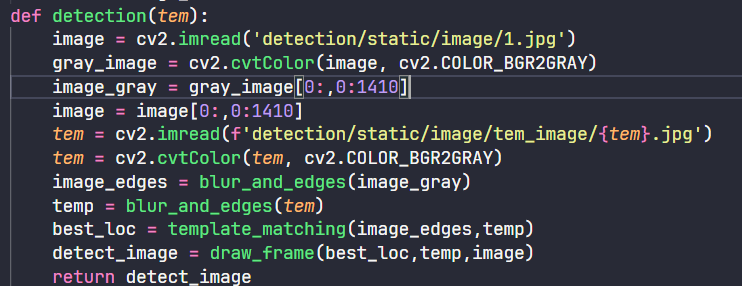
image\_edges = blur\_and\_edges(image\_gray)

template\_edges = blur\_and\_edges(template\_image)

best\_loc = template\_matching(image\_edges,template\_edges)

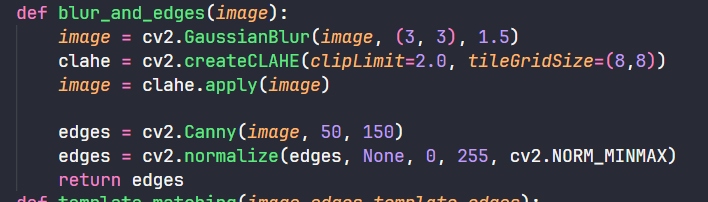
detect\_image = draw\_frame(image,best\_loc)

*return* detect\_image



1. Hàm detection thực hiện đọc ảnh lớn và cách các chi tiết thừa rồi chuyển ảnh lớn về ảnh mức xám. Tiếp tục đọc ảnh template tương ứng với đồ vật đã được truyền vào. Sau đó đưa cả 2 ảnh lớn và ảnh template vào hàm *blur\_and\_edges(image).*

Hàm *blur\_and\_edges(image)*: Nhận input là một ảnh và out put là một ảnh đã được xử lý nhiễu làm mượt ảnh bằng gaussian và phát hiện cạnh bằng thuật toán candy *.*



1. Sau khi đã xử ảnh thực hiện template matching với 2 ảnh đã được xử lý. Sử dụng phương pháp Normalized Cross-Correlation Coefficient tính toán hệ số tương quan giữa ảnh template và ảnh lớn. Kết hợp với Multi-Scale Template Matching thực hiện so khớp với nhiều kích thước khác nhau để trách kích thước của ảnh template và ảnh template trong ảnh lớn không đồng nhất.

*def template\_mactching(image,template):*

*best\_val = 0*

*best\_loc = none*

*scales = np.arange(0.4,2,0.05)*

*for scale in scales:*

*template\_resize = resize(template,scales)*

*result = cv2.matchtemplate(image,template\_resize)*

*min\_val,max\_val,min\_loc,max\_loc = cv2.minMaxLoc(result)*

*if max\_val > best\_val:*

*best\_val = max\_val*

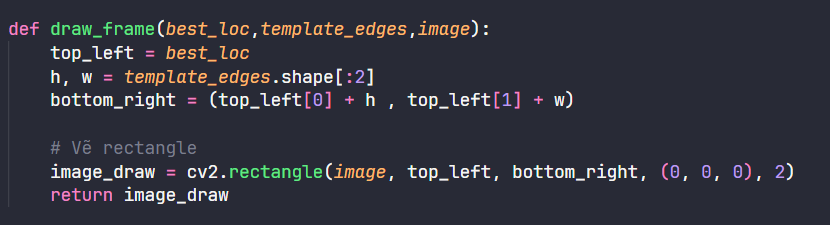
*bes\_loc = max\_loc*

*return best\_loc*

Hàm temaplate\_mactching(image,template): nhận input là ảnh lơn và ảnh template đã được xử lý và output là vị trí tốt nhất xuất hiện vật thể.



1. Sau khi đã có được ví trị của vật thể trong ảnh lớn tiến hành vẽ một hình vuông bao quanh điểm đó với kích thước bằng với ảnh template.



1. Cuối cùng mã hóa lại ảnh và hiện thị lên phía người dùng.

