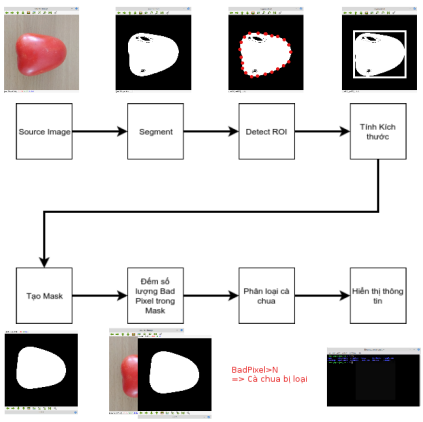
BÁO CÁO DỰ ÁN VER3.0

I. TỔNG QUAN

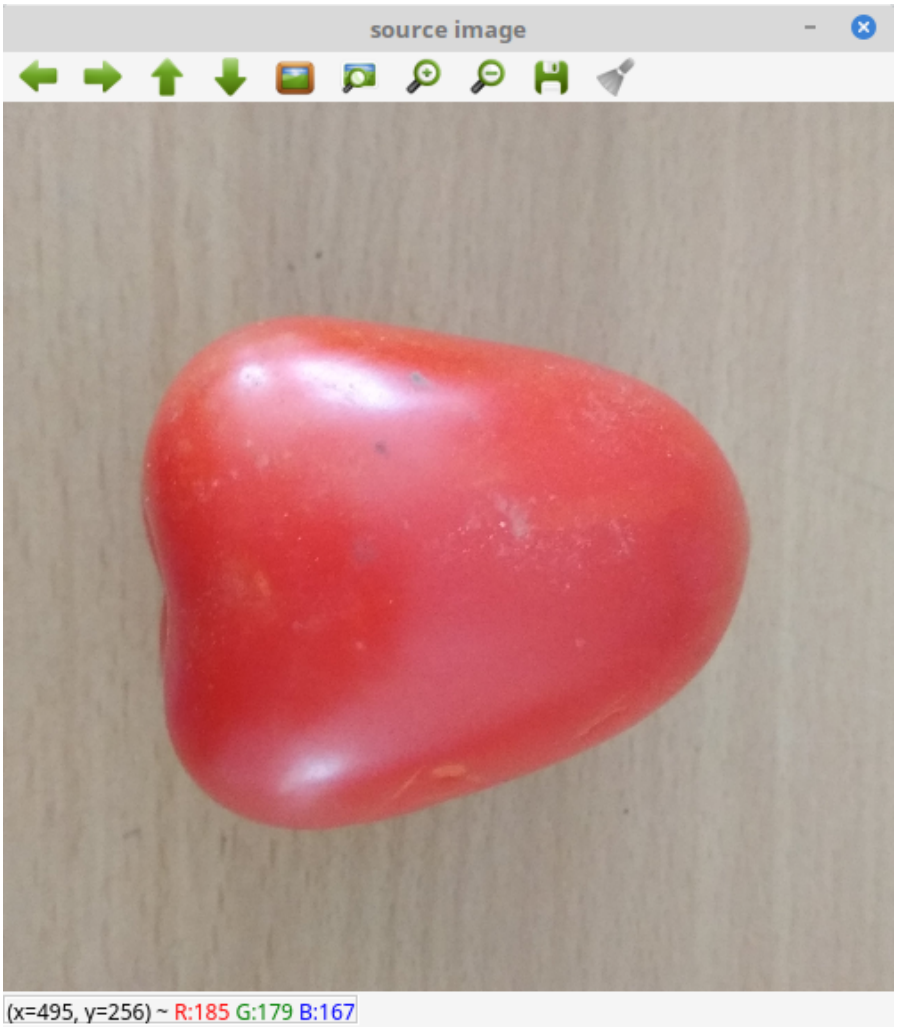


Quá trình xử lý hình ảnh như sau:

Ảnh gốc -> Segment -> Detect ROI -> Tính kích thước -> Tạo mask -> Đếm số lượng Bad Pixel -> Phân loại -> Hiển thị thông tin.

II. CHI TIẾT

Ảnh gốc để xử lý:

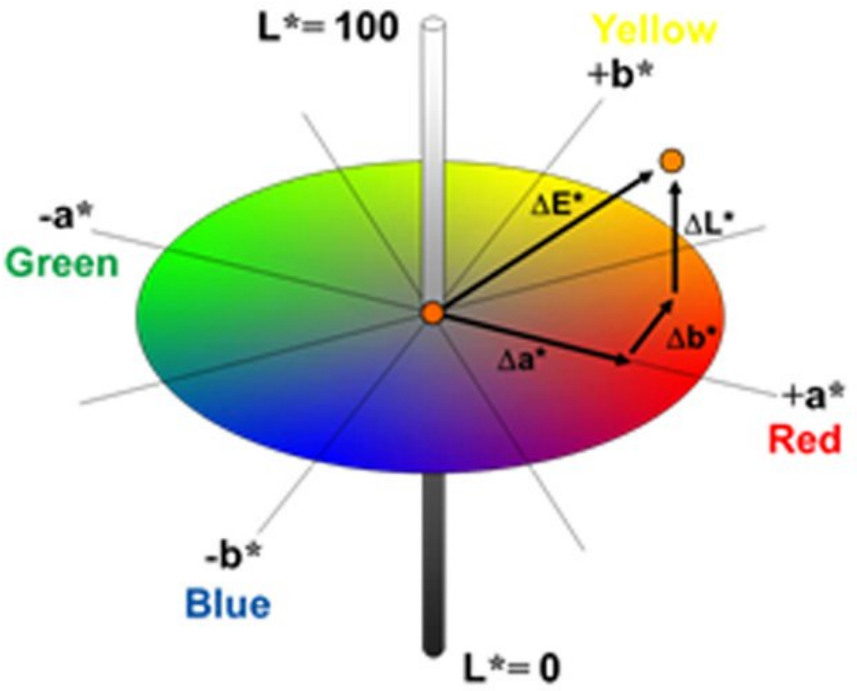


1. Segment

Bước này là bước đầu tiên và là bước quan trọng nhất để có kết quả chính xác nhất. Ở bước này, em phân đoạn ảnh đồng thời threshold hình ảnh luôn, tức những pixel nào là **màu đỏ, màu vàng** hoặc **màu xanh** sẽ được chuyển thành màu trắng, những pixel màu khác sẽ chuyển thành màu đen.

Không gian màu em chọn là `L*a*b*`, có 2 trục xác định màu là trục **a** và trục **b**:

- Trục **a*** chạy từ **(-127) --> 128** với **-127** là màu xanh lá đặc trưng và **128** là màu đỏ đặc trưng.
- Trục **b*** chạy từ **(-127) --> 128** với **-127** là màu xanh dương đặc trưng và **128** là màu vàng đặc trưng.
- Trục **L*** là trục xác định độ sáng của màu sắc, chạy từ **0 -->100** với **0** là tối nhất, tức màu **đen hoàn toàn**, và **100** là sáng nhất, tức màu **trắng hoàn toàn**.



Để xác định màu, em dựa vào công thức:

- Nếu là **màu đỏ**:

$$\begin{cases} 20 < a < 128 \\ 10 < L < 90 \\ 0 < arctan(\frac{|b|}{|a|}) < 45 \end{cases}$$

- Nếu là **màu xanh lá**:

$$\begin{cases} -127 < a < -10 \\ 10 < L < 90 \\ 30 < arctan(\frac{|b|}{|a|}) < 80 \end{cases}$$

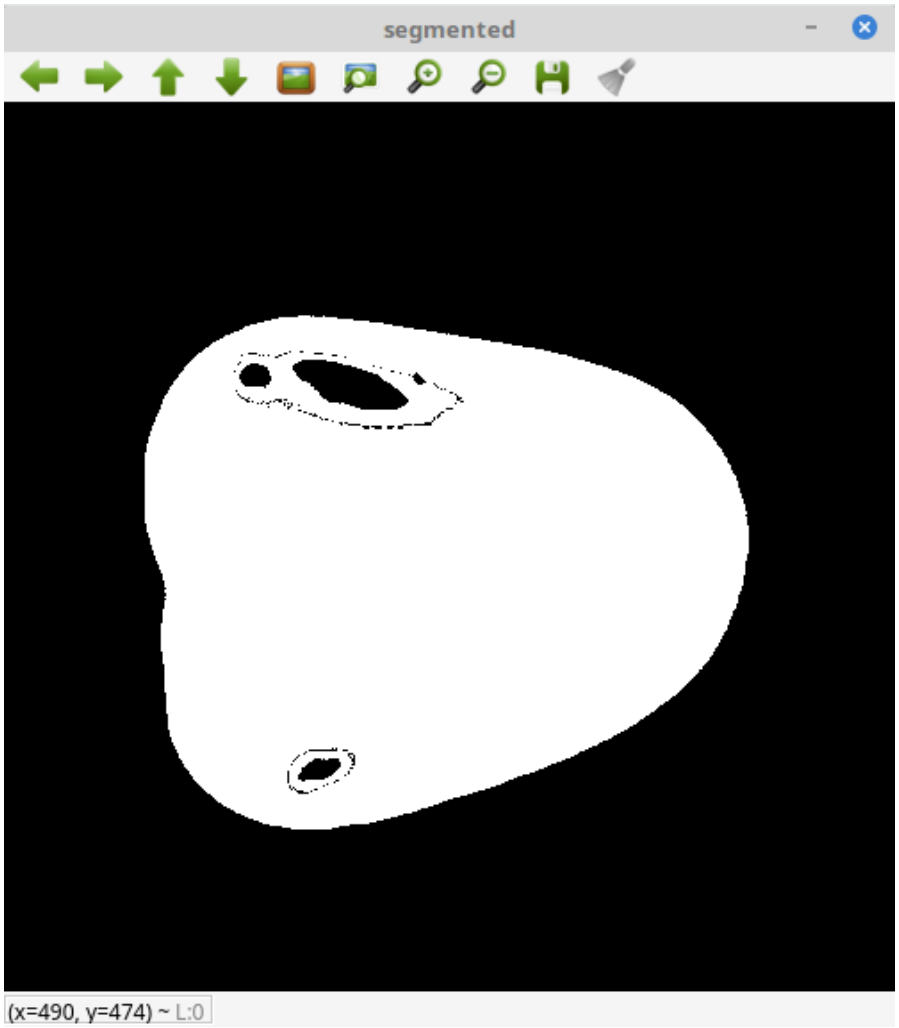
- Nếu là **màu vàng**:

$$\begin{cases} 30 < b < 128 \\ 42 < L < 90 \\ -10 < arctan(\frac{|a|}{|b|}) < 45 \end{cases}$$

Duyệt qua tất cả pixel của hình ảnh và:

1. Kiểm tra pixel đó màu gì, nếu là 1 trong 3 màu **đỏ, xanh lá, vàng** thì **chuyển thành màu trắng,** nếu là **màu khác** thì chuyển thành **màu đen**
2. Đếm tổng số pixel màu đỏ, tổng số pixel màu vàng, tổng số pixel màu xanh lá, so sánh các số này, **số nào lớn nhất thì màu chính của cà chua** sẽ là màu đó.

Kết quả của bước Segment:



2. Detect ROI

Sau khi đã segment hình ảnh gốc, ta thu được một hình ảnh nhị phân (binary image), những pixel cần quan tâm màu trắng và nền màu đen. Ở bước này ta sẽ:

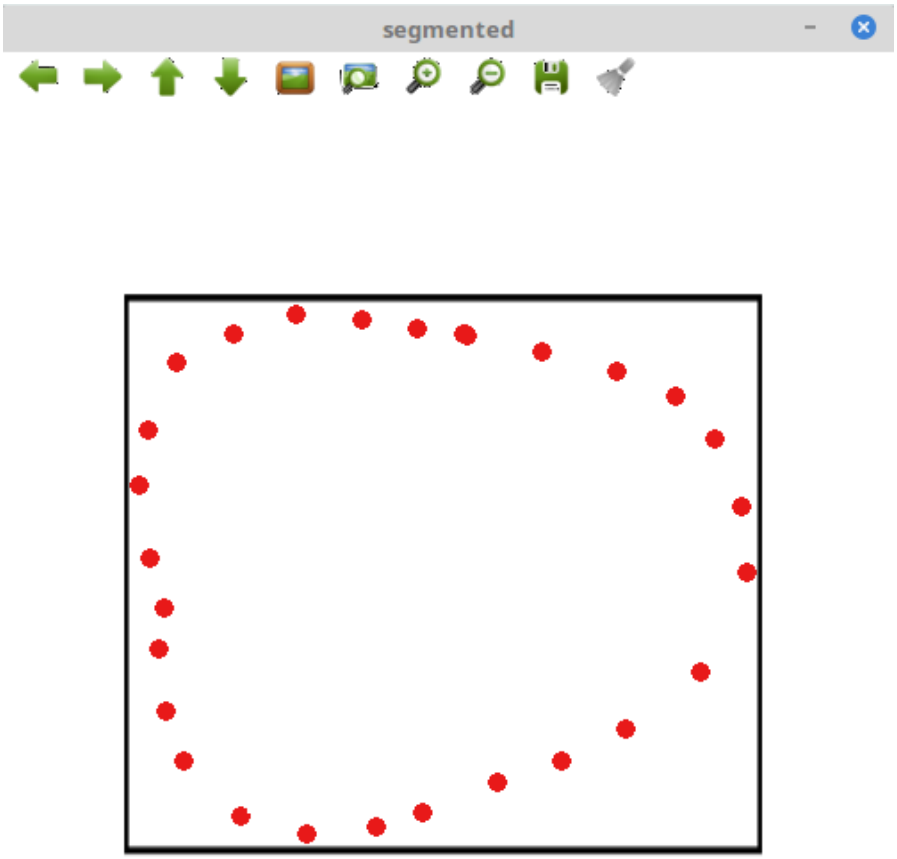
1. Tìm ra các vùng (contour) trong hình, vùng nào có **diện tích lớn nhất** chính là quả cà chua.
2. Convext hull vùng đó

Kết quả thu được sau bước này là **một vector chứa các điểm** bao lấy khu vực có quả cà chua mà ta cần quan tâm (ROI: Region of Interest)

3. Tính kích thước

Tìm một hình chữ nhật xoay (Rotated Rectangle) có *diện tích nhỏ nhất* bao lấy toàn bộ các điểm nằm trong vector trả về từ hàm `DetectROI`, sau đó tính kích thước của hình chữ nhật này, đây chính là **kích thước xấp xỉ của quả cà chua**.

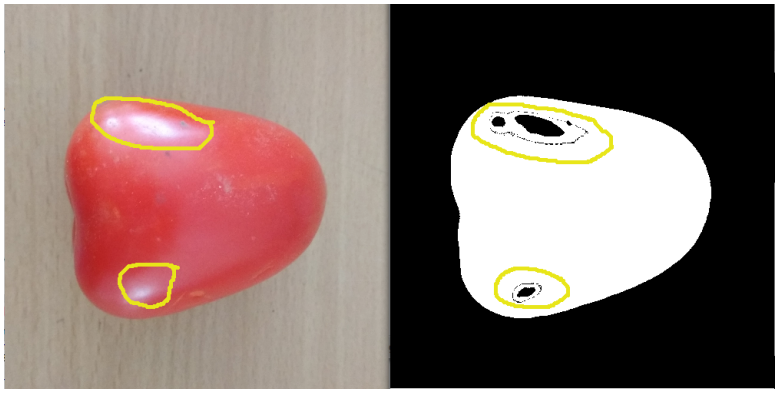
Đến bước này, coi như ta đã xác định được **màu sắc và kích thước của quả cà chua.**



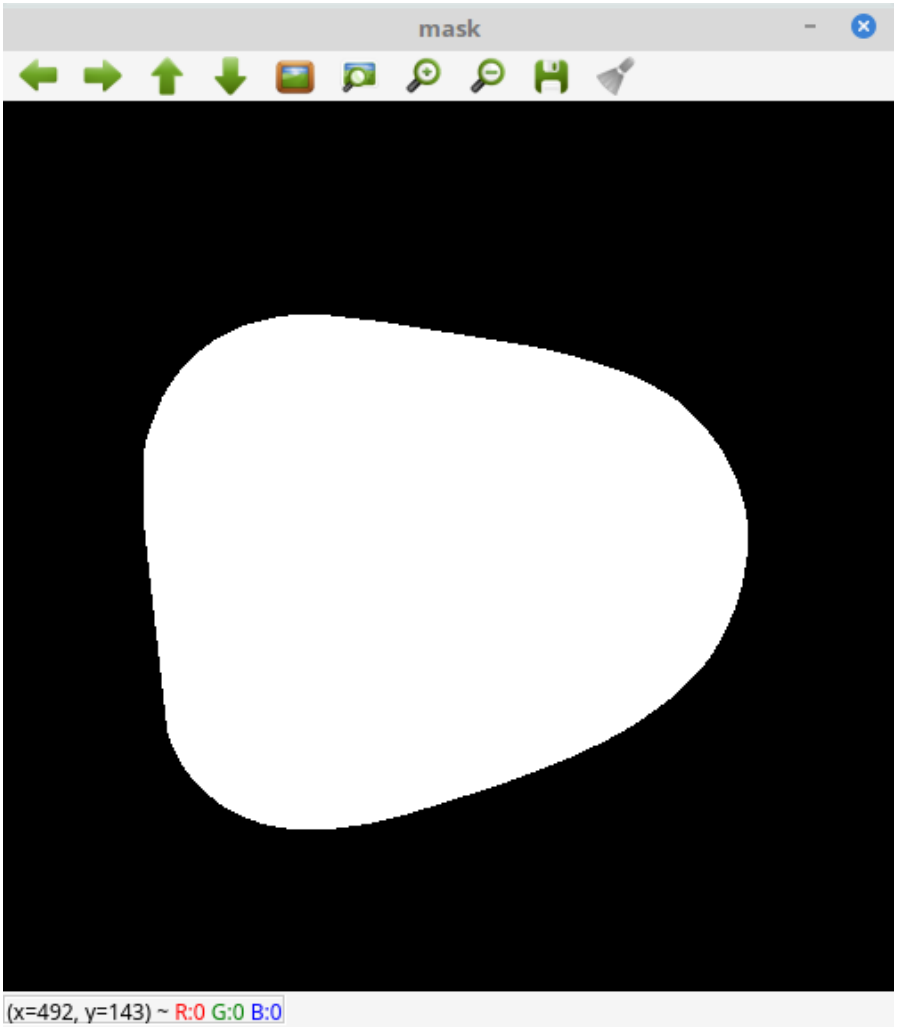
4. Tạo mặt nạ (Mask)

Từ bước này, ta sẽ kiểm tra quả cà chua có bị sâu bệnh đục khoét hư hại hay không. Ý tưởng là **kiểm tra trong vùng nằm bên trong ROI**, tức nằm trên bề mặt quả cà chua có bao nhiêu điểm mà không thuộc 1 trong 3 màu: **đỏ, xanh lá, vàng.** Ta gọi những pixel như vậy là **Bad Pixel.** Và nếu số lượng pixel loại đó lớn hơn một giá trị cho trước là `MAX_NUMBER_OF_BAD_PIXELS` thì ta kết luận là trái cà chua bị hư hỏng, nên loại bỏ. Cách này đơn giản nhưng có những nhược điểm:

- Những điểm được đánh dấu trên hình cũng bị đếm vào Bad Pixel, mặc dù những điểm đó là do ánh sáng gây ra.
- Khó phát hiện những trái bị sâu đục lỗ không quá to, khi mà số lượng pixel này không vượt quá `MAX_NUMBER_OF_BAD_PIXELS` thì cà chua vẫn được tính là bình thường.



Để bắt đầu dùng cách này, ta phải tạo một mặt nạ (**Mask Image**) để chỉ xét những pixel nằm trong ROI, tức nằm trên bề mặt quả cà chua chứ không xét cả những điểm là background. Kết quả tạo mặt nạ như hình dưới.

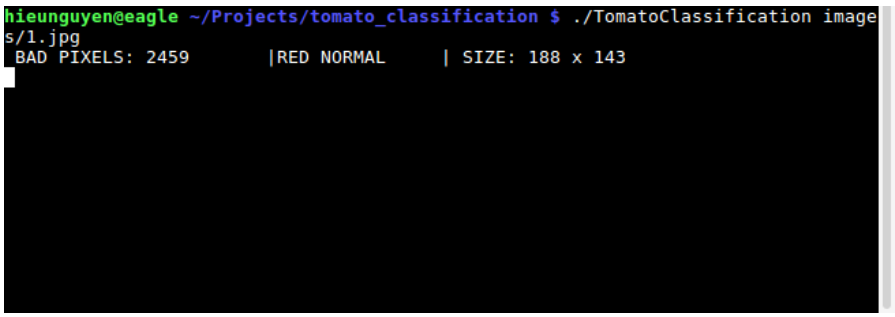


5. Đếm số lượng Bad Pixel

Ta sẽ chồng mặt nạ (**Mask Image**) lên trên hình ảnh gốc (**Original Image**) để bắt đầu đếm bằng cách xét, nếu một pixel đó ở Mask Image là màu trắng thì pixel tương ứng đúng vị trí đó ở ảnh gốc mới được xét, nếu không thì bỏ qua vì là background. Kết quả trả về là **số lượng Bad Pixel đếm được.** Ta dùng số này để phân loại cà chua.

6. Phân loại cà chua và hiển thị thông tin

Dựa vào các thông tin: **màu chính của cà chua, kích thước, số lượng BadPixel** trên cà chua, ta sẽ phân loại cà chua thành các loại khác nhau tùy yêu cầu.



III. KẾT QUẢ HIỆN TẠI

Chương trình đã chạy được **real-time** trên camera.

IV. HƯỚNG ĐI TIẾP THEO

1. Cải thiện kết quả bằng cách chọn màu băng chuyền (tức màu nền background) là màu khác xa các màu mà chúng ta quan tâm, từ đó kết quả đỡ sai lệch hơn.
2. Xem xét điều kiện chiếu sáng để hạn chế tối đa vấn đề ảnh hưởng của ánh sáng lên việc tính BadPixel
3. Nghiên cứu cách phát hiện sâu bệnh chính xác hơn.

V. SOURCE CODE DỰ ÁN

Link **Github**: https://github.com/hieunvce/tomato_classification