ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

Công Nghệ Phần Mềm mở rộng

Báo cáo

PHÂN LOẠI ICONS THEO TRÒN, VUÔNG HAY MỎNG, DÀY

Giảng viên: Quản Thành Thơ
Sinh viên: Nguyễn Hoàng Việt 1814771

Thành phố Hồ Chí Minh, 6/2021

Mục lục

Mục lục hình ảnh

1 Giới thiệu đề tài

Trí tuệ nhân tạo được áp dụng mạnh mẽ trong hầu hết tất cả các lịnh vực trong thực tiễn hiện nay. Từ nông nghiệp, công nghiệp, tới dịch vụ, giải trí, ... Song song đó, trí tuệ nhân tạo còn có thể dùng như một công cụ để hổ trợ các lập trình viên, các nhà phát triển có thể phát triển ứng dụng của mình một cách tốt hơn.

Ở bài báo cáo này chúng ta sẽ áp dụng trí tuệ nhân tạo vào thiết kế, cụ thể ta sẽ áp dụng vào việc phân loại những nút theo dạng tròn hay vuông, những icons được sử dụng nét dày hay mỏng. Đây là một vấn đề thú vị, bởi khi ta có thể xác định được những icon loại nào sẽ thường được dùng cho mỗi loại thiết kế, phong cách khác nhau, ta có thể mở rộng hơn ở tương lai, một ngày nào đó, ta chỉ cần nói cho máy tính biết được ý tưởng mình muốn thiết kế những gì, máy tính sẽ thay ta làm ở công đoạn thiết kế. Và thậm chí, máy tính còn có thể làm tốt hơn chúng ta làm bởi độ logic, tốc độ tính toán nhanh của máy tính sẽ đưa ra được kết quả tốt hơn.

Ở đề tài này, công việc cụ thể của chúng ta sẽ từ một thiết kế về button hoặc design, qua những công cụ AI có sẵn, cùng với áp dụng một ít công thức toán học vào để có thể tính toán được thiết kế là thiết kế theo hình tròn hay hình vuông, nét dày hay nét mỏng.

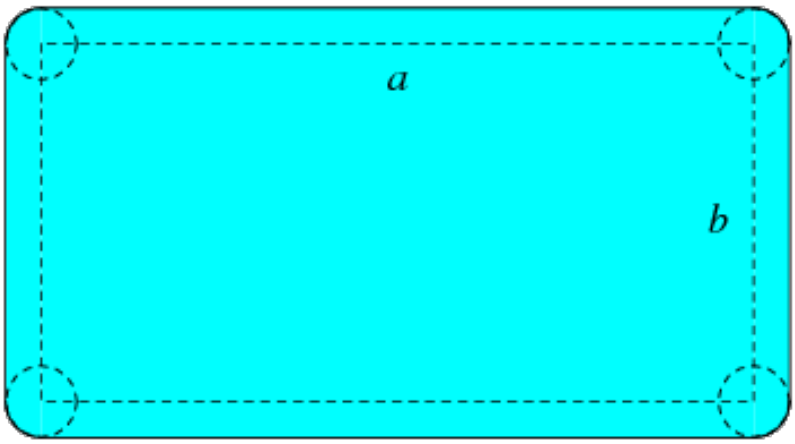
(a) Ví dụ 1 (b) Ví dụ 2

Hình 1.1: Nhận biết button, icon thuộc loại tròn hay vuông, dày hay mỏng

2 Phân loại nút theo tròn hoặc vuông

2.1 Hình chữ nhật bo góc

Bước đầu tiên trong đề tài đó chính là ta sẽ tính toán để phân loại được các nút theo hình tròn hoặc hình vuông. Trong đề tài này, ta sẽ dùng diện tích để phân loại các nút, ở đây ta có hình chữ nhật bo góc [1]:



Hình 2.1: Hình chữ nhật bo góc

Một hình chữ nhật bo góc là một hình dạng được tạo từ một hình chữ nhật, mà ở đó 4 góc có 4 hình tròn bằng nhau về kích thước.

Phần diện tích của hình chữ nhật bo góc này được tính bằng cách:

$$S = ab + 2r(a+b) + \pi r^2 \tag{1}$$

Hoặc có thể tính bằng phép bù:

$$S = S_{BoundingRectangle} - S_{Square-Circle} \tag{2}$$

$$= (a+r) * (b+r) - (4 * r^2 - \pi * r^2) \tag{3}$$

Như vậy, khi ta có được diện tích của phần bên trong của hình chữ nhật bo góc, ta có thể tính được bán kính của 4 hình tròn ở 4 góc của nó. Dựa trên bán kính này, ta sẽ biết được hình mình đang xét sẽ là hình tròn, hình chữ nhật hay hình chữ nhật bo góc.

```
if (w * 0.95 < h < w * 1.05):
    if (curvature < 0.1):
        shape = 'Square'
    elif (curvature < 0.90):
        shape = 'Rounded Square'
    else:
        shape = 'Circle'
else:
    if (curvature < 0.1):
        shape = 'Rectangle'
    else:
        shape = 'Rounded Rectangle'

```

Để có thể tính được phần diện tích bên trong của hình chữ nhật bo góc, hay trong bài toán ta đang xét đó là phần diện tích bên trong của các nút, ta sẽ sử dụng một công cụ của OpenCV, đó là Contours [2]

2.2 OpenCV - Contours

Contour là thuật toán được sử dụng trong xử lý ảnh nhằm tách, trích xuất các đối tượng, tạo điều kiện để các xử lý sau được chính xác. Nên threshold or canny edge detection trước khi sử dụng thuật toán này. Vì vậy contour rất hữu ích trong việc shape analysis và object detection and recognition.

Thư viện OpenCV đã hỗ trợ rất tốt về contour. Ngoài detect những đường nét của nút, thư viện còn hỗ trợ ta tính phần diện tích bounding rectangle, phần diện tích bên trong của đường viền. Áp chụng công thức 3, ta dễ dàng tính được bán kính hình tròn của 4 góc mà ta đang cần.

Từ đó bài toán phân loại nút theo hình tròn, vuông được giải quyết một cách triệt để và chính xác theo lý thuyết, khi hình ảnh đưa vào được thư viên OpenCV đọc ra các đường viền chính xác.

3 Phân loại icon theo nét dày hoặc mỏng

3.1 Phân tích bài toán

Bài toán thứ 2 ta đang cần giải quyết đó chính là từ một hình icon, ta cần phân loại được icon đó sử dụng nét mỏng hoặc nét dày. Dựa trên tình thần mà thực tế ta phân loại bằng cách bình thường, đó chính là xem độ dày của các nét của icons. Nếu như các nét của icon dày hơn mức cho phép, ta sẽ phân loại nó là loại dày, ngược lại là loại mỏng.

Từ tinh thần này, bài toán chúng ta trở thành tính độ dày của các nét trên icon đầu vào. Và để tính được độ dày này, chúng ta sẽ tìm hiểu đến medial axis - trục trung tuyến.

3.2 Medial axis - trục trung tuyến

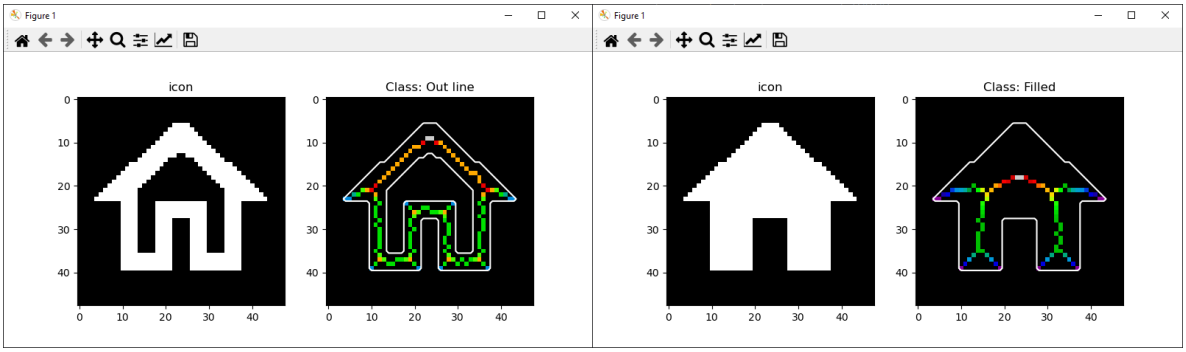
Medial axis - trục trung tuyến của một đối tượng là tập hợp tất cả các điểm có nhiều hơn một điểm gần nhất trên đường biên của đối tượng. Ban đầu được gọi là bộ xương tôpô, nó được Blum giới thiệu như một công cụ để nhận dạng hình dạng sinh học. Trong toán học, việc đóng trục trung gian được gọi là quỹ tích cắt.

Trong 2D, trục trung tuyến của tập con S được giới hạn bởi đường cong phẳng C là quỹ tích của tâm các đường tròn tiếp tuyến với đường cong C tại hai hoặc nhiều điểm, tại đó tất cả các đường tròn đó đều nằm trong S. (Sau đó trục trung tuyến của chính nó nằm trong S.) Trục trung tuyến của một đa giác đơn giản là một cái cây có lá là đỉnh của đa giác và các cạnh của chúng là các đoạn thẳng hoặc cung của parabol.

3.3 Sử dụng scikit-image để tìm trục trung tuyến

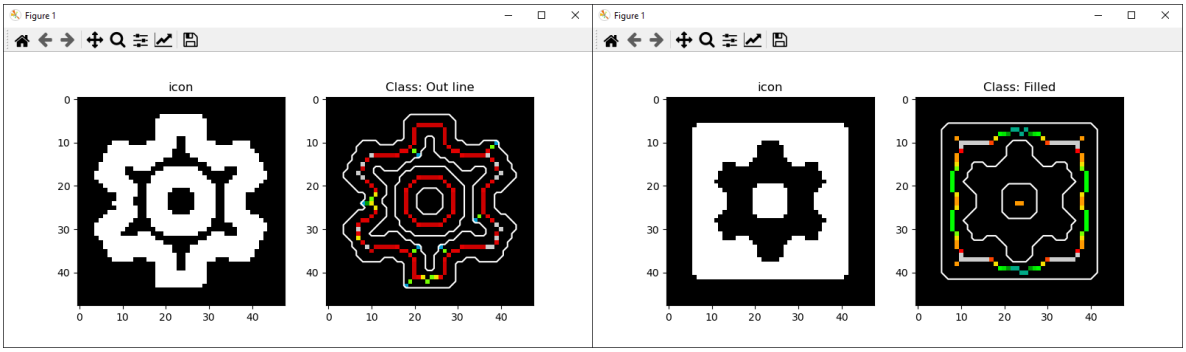
Thư viện scikit-image đã hỗ trợ ta tìm được trục trung tuyến của hình ảnh. [4] "Vì hàm medial_axis (skimage.morphology.medial_axis) trả về biến đổi khoảng cách ngoài trục trung gian (với đối số từ khóa return_distance = True), có thể tính toán khoảng cách tới nền cho tất cả các điểm của trục trung gian với chức năng. Điều này cung cấp một ước tính về chiều rộng cục bộ của các đối tượng. Đối với một bộ xương có ít nhánh hơn, tồn tại một thuật toán tạo bộ xương khác trong skimage: skimage.morphology.skeletonize, tính toán bộ xương bằng cách làm mỏng hình thái lặp đi lặp lại."

Như vậy, từ kết quả mà thư viện scikit-image trả về, ta có thể dễ dàng tìm được độ dày của các đường nét của icon. Từ đó ta có thể tạo một ngưỡng giá trị để phân loại mỏng và dày các icon.



(a) nét mỏng (b) nét dày

Hình 3.1: Ví dụ về phân loại icon home mỏng và dày



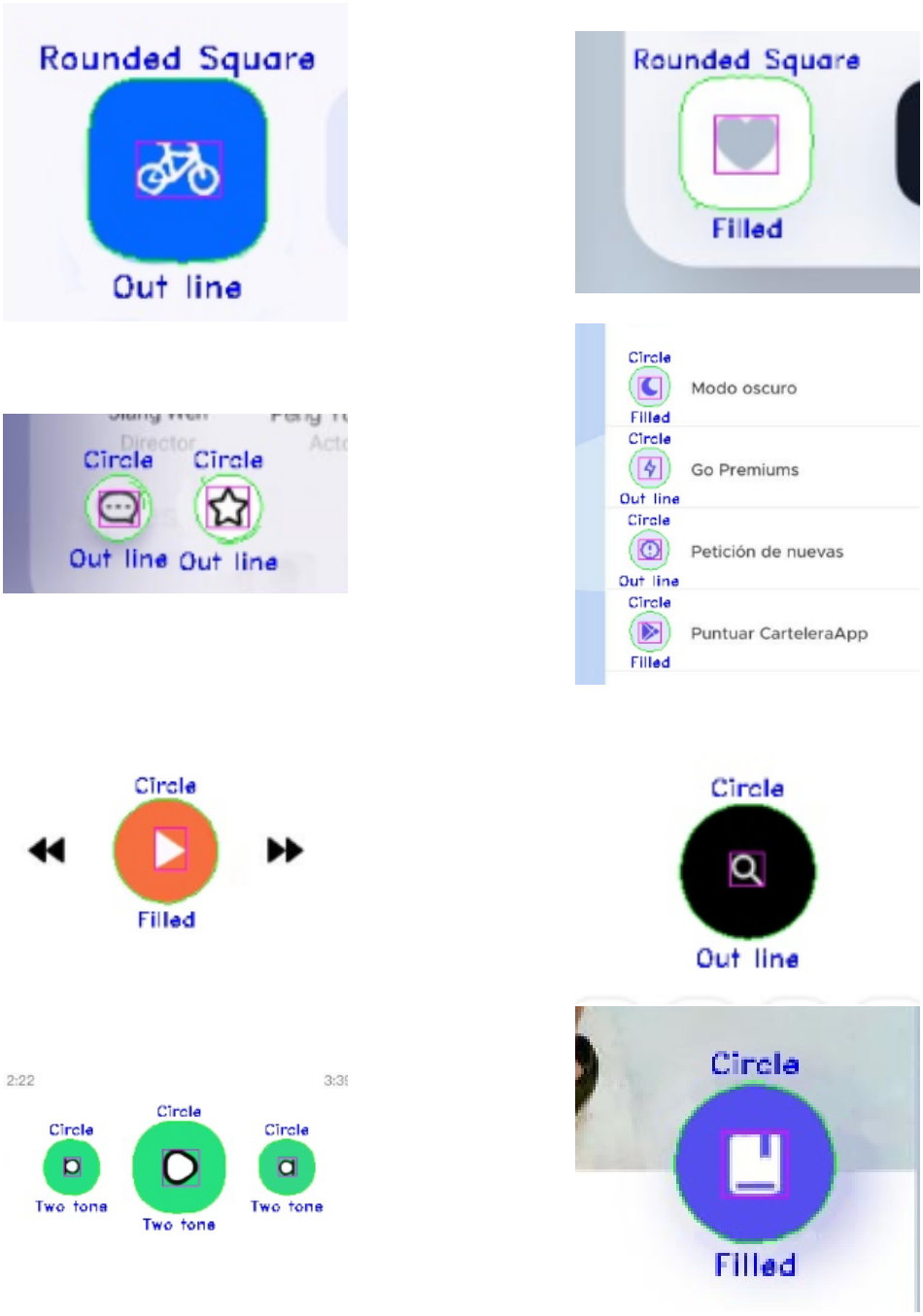
(a) nét mỏng (b) nét dày

Hình 3.2: Ví dụ về phân loại icon setting mỏng và dày

Hiện tại, ngưỡng em dùng để phân loại được em chạy bằng tay, và tinh chỉnh cho thích hợp. Ta có thể chạy kết quả này cho nhiều một lượng lớn vừa đủ các icons, từ đóápp dụng thêm những kỹ thuật phân loại bằng AI để có thể tìm được một ngưỡng hợp lý và chính xác nhất, từ đó kết quả trả về sẽ vượt trội hơn hẳn.

4 Kết quả chạy được

Đây là kết quả chạy thử với hình ảnh rõ nét, không bị nhòe và được định sẵn vị trí của các nút và các icons.



Hình 4.1: Kết quả chạy được

Các hình ảnh này đã được qua xử lý, làm giảm noise, nhòe, để đạt được điều kiện lý tưởng. Các kết quả chạy khác ở: Repository

5 Tổng kết

Qua bài tập lớn của học phần mở rộng này em hiểu hơn về ứng dụng của AI trong cuộc sống đời thường, và cụ thể là áp dụng AI vào chính design của các ứng dụng mà mình có thể sẽ xây dựng lên. Và qua đó có thể thấy được tầm quan trọng và sức ảnh hưởng của AI to lớn đến thế nào.

Kết quả chạy được vẫn chưa được hoàn hảo. Kết quả ảnh hưởng nhiều đến độ nhiễu, độ nhòe của hình ảnh. Nếu hình ảnh là định dạng jpg, khó có thể nhận dạng chính xác được. Thay vào đó, nếu ta có thêm 1 bước tiền xử lý hình ảnh, làm giảm nhiễu của hình, thì kết quả sẽ đưa ra tốt hơn rất nhiều. Các ngưỡng giá trị để phân loại, nhận biết icon vẫn chưa được chính xác. Ta có thể mở rộng áp dụng kết quả này cho một tập dữ liệu lớn hơn, từ đó có thể đưa ra một ngưỡng giá trị có độ tin cậy tốt hơn. Nhưng trên cơ bản, các yêu cầu của bài tập lớn đều đã giải quyết được ở mức chấp nhận.

Tài liệu tham khảo

[1] Weisstein, Eric W. "Rounded Rectangle." From MathWorld–A Wolfram Web Resource.

[2] OpenCV - Contours : Getting Started

[3] Wikipedia, 1 April 2021. - Medial axis

[4] Scikit-image - Medial axis skeletonization

Link repository: ClassifyIcon