

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



CÔNG NGHỆ PHẦN MỀM MỞ RỘNG

Báo cáo bài tập lớn
Shadow Detection

Giảng viên hướng dẫn: PGS. TS Quản Thành Thơ
Sinh viên: Giang Văn Lực - 1813021

Tp. Hồ Chí Minh, Tháng 08/2021



Mục lục

1	Giới thiệu	2
2	Component detection	3
3	Shadow detection	6

1 Giới thiệu

box-shadow là 1 thuộc tính CSS được sử dụng nhiều trong giao diện trang web nhằm giúp giao diện thân thiện với người dùng hơn. Shadow offset được định nghĩa là khoảng cách từ phần viền của shadow đến viền của phần tử đó. Để xác định được



Hình 1: Shadow offset của 1 phần tử web

phần shadow offset của 1 phần tử chúng ta có 2 bước thực hiện chính: xác định các thành phần trên 1 trang web và xác định shadow offset trên các thành phần có khả năng chứa shadow, ở đây là các button. Sau đó, với input là các hình button được cắt từ bước trước đó, ta xác định shadow offset trên các hình đó.

2 Component detection

Để xác định các thành phần trên 1 hình ảnh trang web, ta cần train 1 model với 1 tập dataset được gán nhãn với các loại thành phần như input, heading, image,... Framework được sử dụng cho quá trình training là [detectron2](#), 1 framework về Deep Learning của Facebook.



Hình 2: Detectron2

Model baseline sử dụng cho quá trình training là Faster-RCNN với backbone là Resnet và FPN. Faster-RCNN là model hiệu quả trong lĩnh vực object detection, còn Resnet+FPN đạt được sự cân bằng giữa tốc độ và độ chính xác. Tham khảo thêm về các model sử dụng trong detectron2 tại [đây](#).

Dataset sử dụng là tập dữ liệu về ảnh chụp màn hình các trang web, gồm các thành phần được gán nhãn sẵn như: button, heading, link, text,... Dataset hỗ trợ nhiều format cho các thư viện khác nhau, với [COCO](#) json format là format được detectron2 sử dụng.

Sau quá trình training, chúng ta có 1 file model weight, sử dụng để detect các thành phần trên 1 trang web.



Hình 3: Output của 1 hình ảnh trang web

Bởi vì chúng ta chỉ giới hạn detect shadow trên 1 button nên các component được gán nhãn khác sẽ được loại bỏ. Các predicted boxes được gán nhãn của detectron2 có điểm số trong khoảng từ 0 đến 1, tương ứng cho xác suất dự đoán chính xác của box đó, các box này có thể lồng vào nhau trong quá trình prediction. Vậy nên chúng ta cần qua thêm 1 bước xử lý các box lồng nhau, sẽ chỉ có box có xác suất cao nhất được lựa chọn.

はてなブログタグ

はてなブログタグとはよくある質問

言葉からつながる ネットの話題

「同じ興味の人とつながりたい!」「ネットの話題について知りたい!」そう思ったら、はてなブログタグの出番。言葉の解説だけでなく、ブロガーの意見も、ネットで話題の記事も、これひとつでチェックできます。



はてなブログ タグ とは？

はてなブログ タグでは、ある言葉についてネット上の意見や感想をさまざまな角度から知ることができます。はてなブログ・はてなブックマーク等の情報を活用しており、はてなブログをご利用の方も、そうでない方も、自分の興味のある記事を探せます。

たとえばこんなタグをチェック！

仮面ライダー

VTuber

萩尾望都

ノーベル賞

プリキュア

YAPC

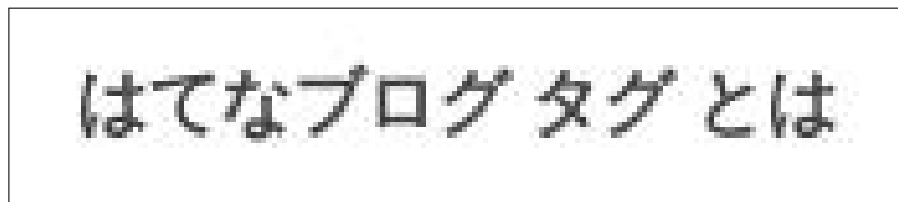
十二国記

Hình 4: Các box với nhãn là button được lựa chọn

3 Shadow detection

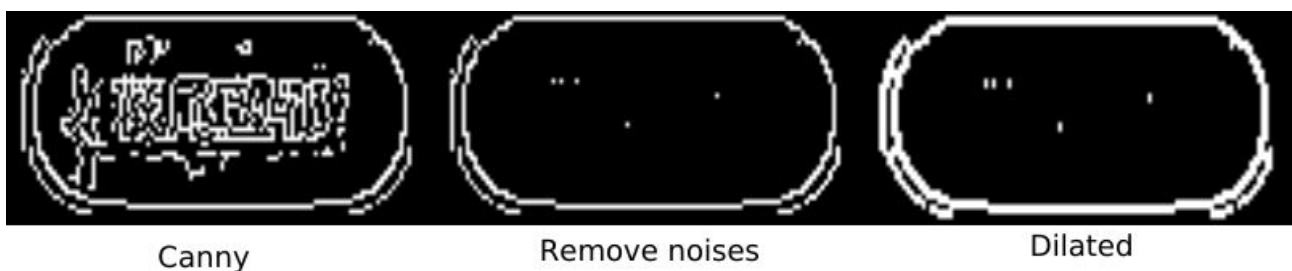
Từ output của phần trước sẽ tạo ra 1 tập ảnh các button là input cho bước tiếp theo là Shadow Detection. Để detect shadow trên 1 hình ảnh button, có 2 phương pháp chính được sử dụng là Canny Edge Detection và Contours. Canny Edge Detection để tìm các cạnh trên ảnh, lọc nhiễu và đưa ảnh về dạng binary(chỉ gồm pixel đen và trắng) sử dụng cho việc tìm các contour(đường biên) trên hình.

Với mỗi ảnh, ta sử dụng Canny trên ảnh, sau đó sử dụng `cv2.findContours()` để lấy là cấu trúc của các contours. Contour chứa viền của button được xác định là contour có diện tích lớn nhất. Để lọc ra các input không hợp lệ, tỷ lệ giữa bounding box của button và của hình ảnh nằm trong khoảng từ 0.7 đến 0.9. Những hình ảnh không phải 1 button sẽ được loại bỏ.



Hình 5: Input không hợp lệ

Khi xác định được contour là đường viền của button, tất cả các contour là con của contour đó sẽ được loại bỏ(xem [contour hierachy](#) của OpenCV). Cuối cùng sử dụng `cv2.dilate()` để tăng độ dày các đường viền để các đường viền rời rạc nối lại với nhau.



Hình 6: Các bước xử lý

Sau đó, sử dụng `cv2.findContours()` với hình ảnh đã được tăng độ dày đường viền, phần shadow dễ dàng được xác định. Sau khi có được bounding box cho cả button và shadow thì offset sẽ được tính là khoảng cách giữa các cạnh tương ứng. Nếu như offset của cả 4 cạnh quá nhỏ (< 3) thì có thể coi như input ban đầu không chứa shadow.



Hình 7: Input và output của quá trình Shadow Detection