Project cuối khóa - Rubik Talent 03

Đề tài: Phân loại thực phẩm từ ảnh sử dụng Mask-RCNN

Ngày 01 tháng 02 năm 2020

—

Danh sách thành viên

1. Nguyễn Văn Thắng
2. Ngô Quốc Đạt

1. Tổng quan

Nhận biết thực phẩm từ hình ảnh là một công cụ hữu ích cho phép mọi người theo dõi lượng thực phẩm tiêu thụ bằng cách chụp ảnh. Theo dõi thực phẩm giúp phân tích được các chất dinh dưỡng một người thu được từ đĩa thức ăn và từ đó đưa ra các gợi ý về thực phẩm có lợi cho sức khỏe. Việc theo dõi lượng thực phẩm mỗi loại tiêu thụ hàng ngày còn giúp con người giảm được tiêu thụ quá nhiều thức ăn và giảm lượng thức ăn thừa.

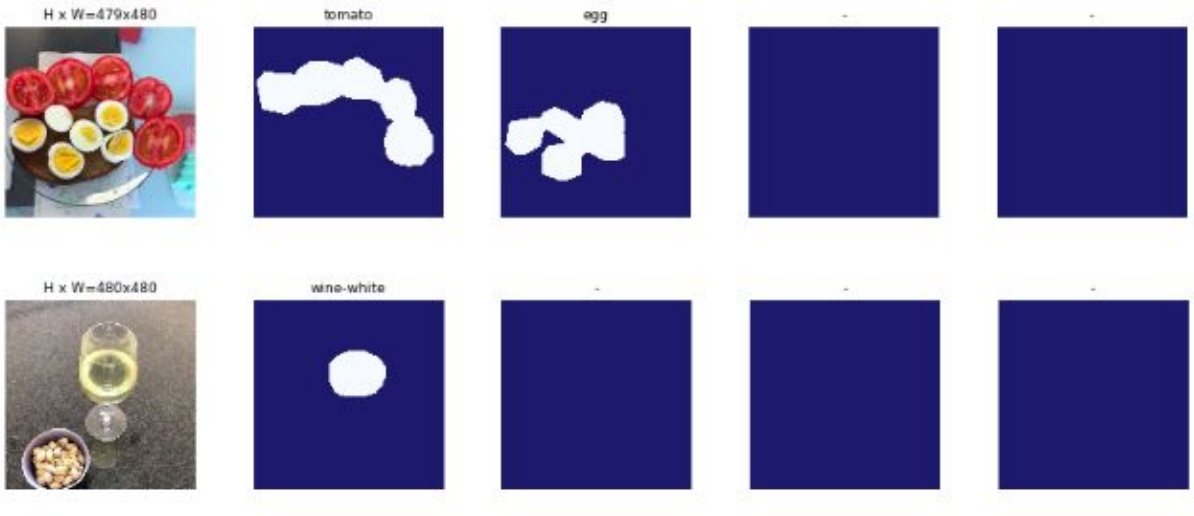
Trong project này, nhóm xây dựng một mô hình để giúp máy tính nhận diện được các loại thức ăn có trên ảnh một đĩa thức ăn. Đề tài project được lấy từ cuộc thi nhận diện thức ăn do AIcrowd tổ chức[1]. Các hình ảnh dùng để huấn luyện và đánh giá mô hình được thu thập từ ứng dụng MyFoodRepo của Thụy Sỹ.

2. Dữ liệu

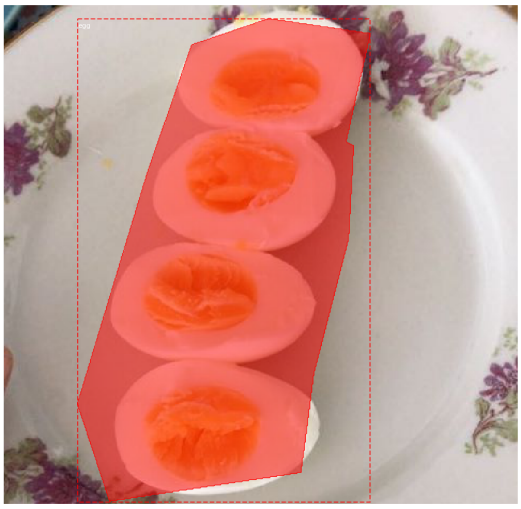
2.1. Khảo sát bộ dữ liệu

- Tập dữ liệu huấn luyện: gồm 5545 hình ảnh dạng RGB thuộc 40 nhóm thực phẩm, cùng với các chú thích tương ứng lưu trong file annotation.json trong cùng thư mục test.
- Tập dữ liệu đánh giá: gồm 291 hình ảnh dạng RGB thuộc 40 nhóm thực phẩm, cùng với các chú thích tương ứng lưu trong file annotation.json trong cùng thư mục val.
- Các file annotation có dạng MS-COCO gồm: Danh sách các nhóm thực phẩm kèm id nhóm, danh sách các ảnh kèm id ảnh, id ảnh và id nhóm tương ứng kèm diện tích vùng nhận diện nhóm thực phẩm, tọa độ các điểm của vùng nhận diện nhóm thực phẩm (mask) và tọa độ của bounding box.
- Bộ dữ liệu được xử lý bằng hàm load_dataset (xem file food_dataset.py), lấy vào địa chỉ file annotation, trả về dữ liệu theo class COCO định nghĩa trong thư viện pycocotools, chứa các thông tin id ảnh, id loại thực phẩm, các thông tin annotation của ảnh.

[1] Đường dẫn cuộc thi: https://www.aicrowd.com/challenges/food-recognition-challenge

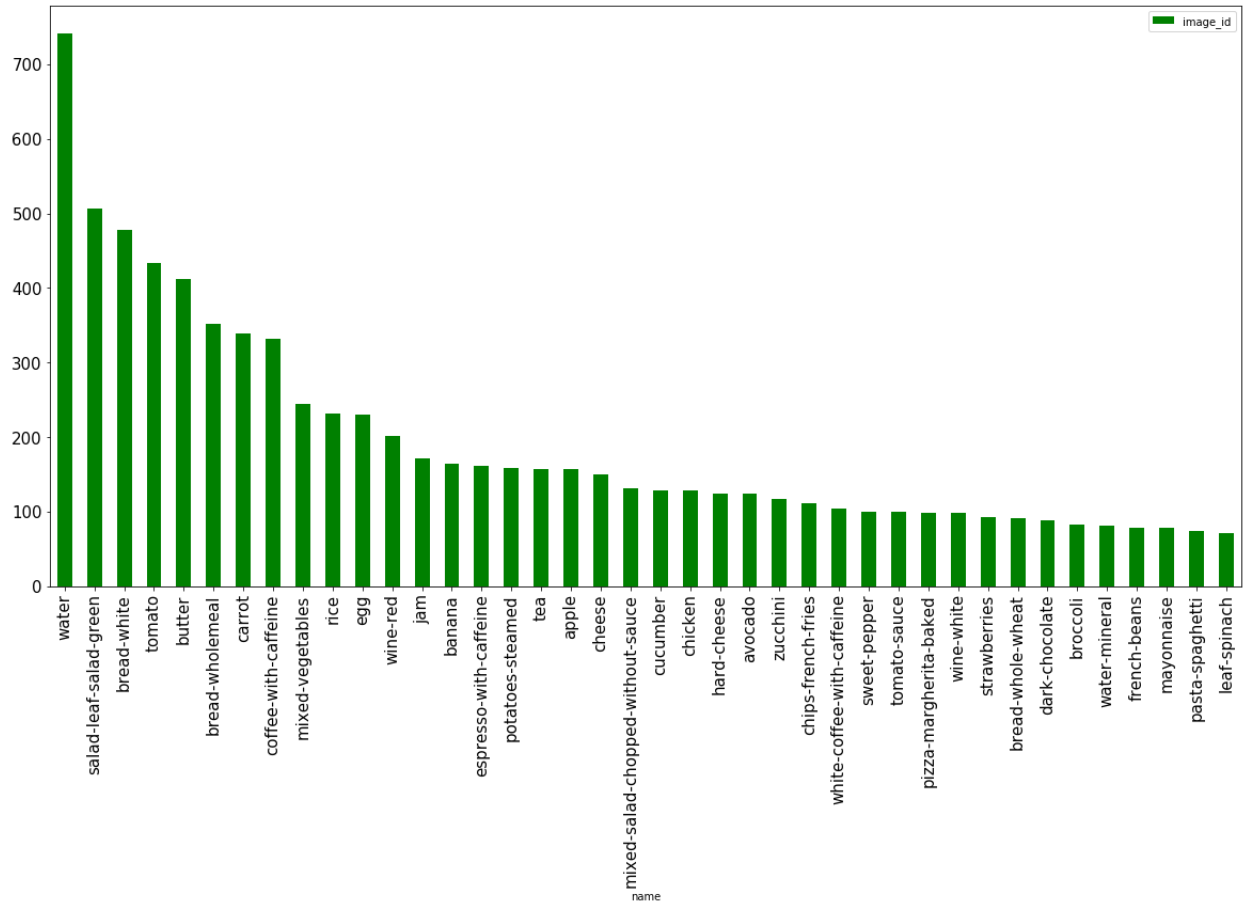


Hình 1: Một số ảnh trong tập training và các mask tương ứng



Hình 2: Ví dụ ảnh training và mask, bounding box

- Khảo sát bộ dữ liệu: Từ các file annotation của tập train, nhóm thống kê số lượng ảnh của từng nhóm. Trong tập test, nhóm có nhiều ảnh nhất (water) có 741 ảnh, nhóm có ít ảnh nhất (leaf-spinach) có 72 ảnh. Để tăng giảm hiệu ứng của việc dữ liệu không cân bằng này và tránh over-fitting, nhóm làm giàu dữ liệu sử dụng thư viện imgaug.



Hình 3: Thống kê số lượng loại thực phẩm xuất hiện trong tập train

2.2. Làm giàu bộ dữ liệu

Nhóm làm giàu bộ dữ liệu sử dụng thư viện imgaug bằng các phương pháp sau:

- Lật ngang các ảnh với xác suất thực hiện 0.5
- Crop ảnh ngẫu nhiên theo tỷ lệ 0.9 đến 1 so với kích thước ban đầu
- Làm mờ ảnh với bộ lọc Guassian, xác suất thực hiện 0.5, sigma ngẫu nhiên từ 0 đến 0.5
- Điều chỉnh độ tương phản của ảnh trong khoảng 0.75 đến 1.25
- Điều chỉnh độ sáng từ 0.8 đến 1.2
- Các phép biến đổi Affine gồm dịch, xoay, cắt ảnh

3. Mô hình

3.1. Kiến trúc mô hình

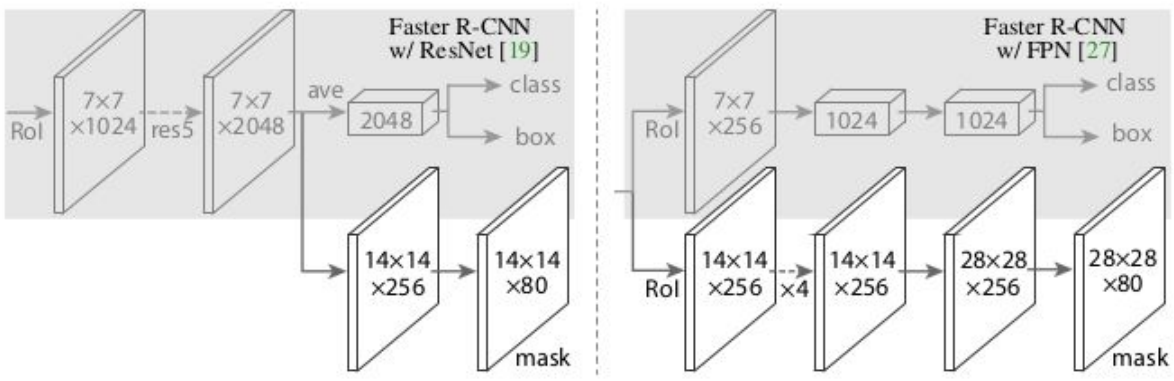
Để ước lượng được khối lượng từng loại thực phẩm trên một đĩa thức ăn, mô hình cần xác định được vùng diện tích của từng loại thực phẩm. Dựa trên yêu cầu đó và các hướng dẫn từ cuộc thi của AICrowd, nhóm lựa chọn mạng Mask-RCNN để xây dựng mô hình cho bài toán.

Mạng Mask-RCNN được mở rộng từ mạng Faster R-CNN bằng cách thêm một nhánh để dự đoán mặt nạ phân loại (mask) của mỗi vùng quan tâm (Region of Interest – RoI). Một mạng Mask-RCNN bao gồm 3 tầng:

- Tầng thứ nhất: Region Proposal Network (RPN) sử dụng các anchors quét qua các khu vực của feature map. Tầng này đưa ra vị trí các khung đề xuất có thể là bounding box (region proposal) cùng với xác suất chứa vật thể của khung tương ứng.
- Tầng thứ hai: Tầng này trích xuất đặc trưng từ từng khung đề xuất sử dụng ROIPool, từ đó phân loại thực phẩm trong khung và vẽ bounding box
- Tầng thứ 3: Chay song song với tầng thứ 2, tạo ra các mặt nạ phân loại cho từng vùng quan tâm theo từng pixel[2].

Kiến trúc mạng Mask-RCNN là kiến trúc mở rộng dựa trên mạng backbone ResNetC4 hoặc Feature Pyramid Network (FPN)

[2] Mask R-CNN, Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick



Hình 4: Kiến trúc mạng Mask-RCNN

Cụ thể, mạng ResNet được sử dụng trong mô hình có kiến trúc như sau:

Stage 1:

- Thực hiện ZeroPadding kích thước 3x3 với ảnh đầu vào.
- Áp dụng khối convolutiond2D filter 7x7 strides 2x2
- Áp dụng khối BatchNorm
- Áp dụng ReLU activation function
- Áp dụng MaxPooling kích thước 3x3 strides 2, không sử dụng padding.

Stage2:

- Áp dụng khối convolution gồm các filter kích thước 3x3, output [64x64x256]
- Áp dụng 2 lần khối identity gồm các filter kích thước 3x3, output [64x64x256]

Stage3:

- Áp dụng khối convolution gồm các filter kích thước 3x3, output [128x128x512]
- Áp dụng 3 lần khối identity gồm các filter kích thước 3x3 output [128x128x512]

Stage4:

- Áp dụng khối convolution gồm các filter kích thước 3x3, output [256x256x1024]

Tùy thuộc vào việc chọn kiến trúc mạng là ResNet50 hay ResNet101 mà khối identity block sẽ chạy để đủ số lần theo yêu cầu. (Với mạng ResNet101 mà nhóm đã chọn, khối này lặp lại 22 lần)

Nếu training tiếp stage 5, thì sẽ thực hiện các bước sau:

- Áp dụng khối convolution gồm các filter 3x3, output [512x512x2048]
- Áp dụng 2 lần khối identity với cùng kích thước để cho ra kết quả đầu ra ở stage 5.

Hàm mất mát được của mô hình là tổng của các hàm mất mát trong 3 nhiệm vụ phân loại, xác định bounding box và xác định mặt nạ $L = L_{classification} + L_{box} + L_{mask}$

3.2. Huấn luyện mô hình

Trong project này, nhóm sử dụng mô hình mạng Mask-RCNN được có backbone là mạng ResNet101, huấn luyện theo phương pháp transfer learning dựa trên bộ tham số có sẵn từ quá trình huấn luyện mạng Mask-RCNN trên COCO dataset. Phương pháp transfer learning tận dụng kết quả có sẵn giúp tiết kiệm thời gian huấn luyện và cho ra kết quả huấn luyện tốt hơn.

Nhóm sử dụng framework Keras để huấn luyện mô hình. Quá trình huấn luyện gồm 3 giai đoạn:

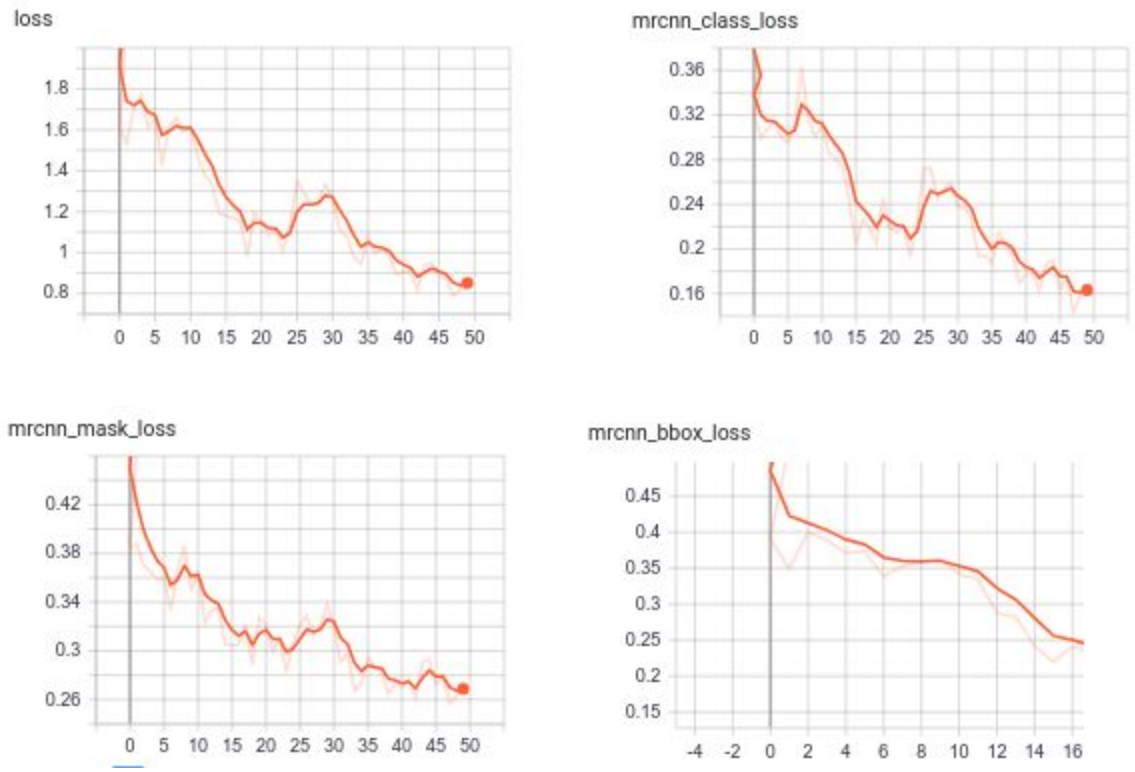
- Giai đoạn 1: Sử dụng bộ tham số pretrained từ mô hình Mask-RCNN training trên COCO dataset, loại trừ tham số của các lớp cuối cùng (mrcnn_class_logits, mrcnn_class, mrcnn_bbox, mrcnn_bbox_fc) để khớp số lượng classification trong bộ dữ liệu với số lượng classification sau khi training giai đoạn này. Nhóm huấn luyện giai đoạn 1 trong 12 epochs.

- Giai đoạn 2: Tinh chỉnh mô hình trên các lớp thuộc stage từ thứ 1 đến 4 của mạng ResNet. Nhóm huấn luyện giai đoạn 2 trong 24 epochs.

- Giai đoạn 3: Tinh chỉnh toàn bộ các lớp của mô hình. Nhóm huấn luyện giai đoạn 3 trong 14 epochs.

Trong quá trình huấn luyện, nhóm thay đổi các siêu tham số sau để thu được thu được kết quả tốt hơn. Với kết quả training tốt nhất thu được, các siêu tham số được lựa chọn là:

- Learning rate trong giai đoạn 1: 0.001
- Learning rate trong giai đoạn 2 và 3: 0.0001
- Steps per epoch: 600
- Batch size: 8
- Validation steps: 5
- Detection min confidence: 0.3



Hình 5: Kết quả các hàm mất mát qua quá trình training

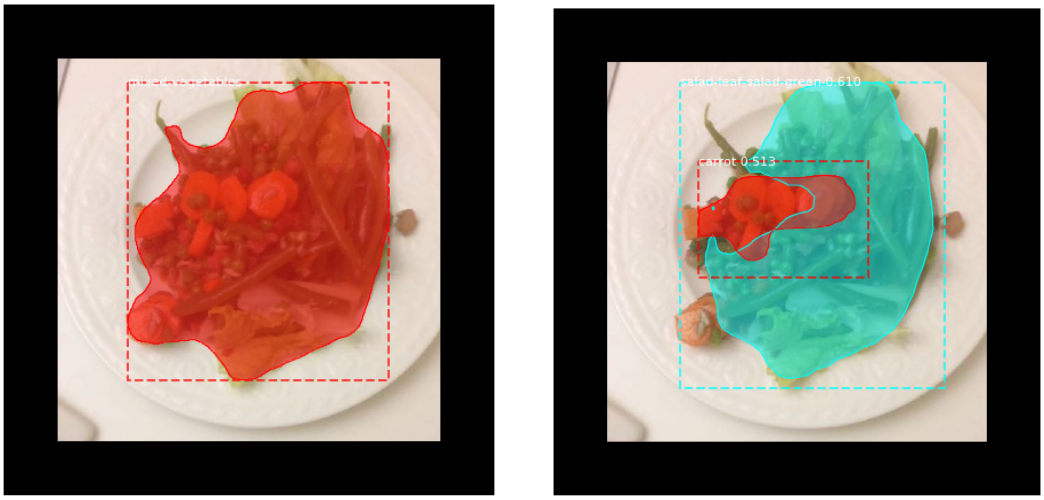
4. Đánh giá kết quả

Mô hình được đánh giá thông qua các bước:

Bước 1: Dự đoán ảnh

- Các đặc trưng của ảnh của tập validation đi qua mạng RPN (Region Proposal Network) của mô hình và đưa ra các khu vực tiềm năng chứa đối tượng. Mạng RPN gán nhãn cho các anchors tìm được và chọn ra top các anchor tích cực với nhiều kích thước khác nhau mà RPN nghi ngờ chứa đối tượng.
- Mạng RPN sử dụng Non-maximum suppression để tinh chỉnh chọn ra các bounding box,
- Mô hình phân lớp để phân loại các box đã được nhận diện phía trên, chỉ giữ lại những box thuộc các loại đối tượng có trong nhóm thực phẩm cần nhận diện.

- Sau khi tạo ra được bounding box bao bọc đối tượng, và phân nhóm cho đối tượng, mô hình tạo ra Mask để bao bọc đối tượng đã nhận diện.



Hình 6: So sánh kết quả của một mẫu ảnh bất kỳ trong tập validation

(bên trái: ảnh và mask gốc, bên phải: ảnh và mask nhận diện từ mô hình)

Bước 2: Tính trên hệ số mAP (mean Average Precision):

Hệ số mAP của 10 ảnh ngẫu nhiên trong tập validation sau khi dự đoán từ mô hình là 0.39. Hệ số này có thể được cải thiện khi tăng cường training. Bên cạnh đó, một số ảnh trong tập validation bị đánh thiếu mask cũng ảnh hưởng đến kết quả đánh giá mô hình