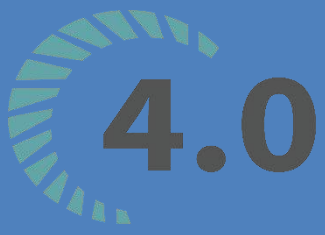
BỘ MÔN THỊ GIÁC MÁY TÍNH – KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐẠI HỌC QUỐC GIA TP HCM

BÁO CÁO MASK-RCNN



Sinh viên thực hiện: Nguyễn Ngọc Khôi Nguyên
Phan Nguyễn Thanh Tùng
Huỳnh Thanh Sang
GV phụ trách: PGS.TS. Lý Quốc Ngọc

ĐỒ ÁN/BÀI TẬP MÔN HỌC - XỬ LÝ ẢNH SỐ VÀ VIDEO SỐ
HỌC KỲ I – NĂM HỌC 2021-2022

Mục lục

YÊU CẦU ĐỒ ÁN - BÀI TẬP

Loại bài tập	☐ Lý thuyết ☐ Thực hành ☑ Đồ án ☐ Bài tập
Ngày bắt đầu	06/10/2021
Ngày kết thúc	23/01/2022

BẢNG THÔNG TIN CÁ NHÂN

Mã sinh viên	**Họ tên sinh viên**
19120106	Nguyễn Ngọc Khôi Nguyên
19120424	Phan Nguyễn Thanh Tùng
1712718	Huỳnh Thanh Sang

A. Yêu cầu Đồ án:

Ứng dụng phân đoạn ảnh dựa vào Mask RCNN.

Nội dung:

- Tầng 1: Môi trường xử lý dữ liệu tiên tiến. Phân đoạn ảnh dựa vào Mask RCNN.
- Tầng 2: Các tác vụ đơn hỗ trợ cho ứng dụng ở tầng 3.
- Tầng 3: Xây dựng ứng dụng phân đoạn ảnh dựa vào Mask RCNN.

B. Báo cáo Đồ án:

I. Giới thiệu

1. *Động lực nghiên cứu*

Phân đoạn hình ảnh là một thuật ngữ thuộc lĩnh vực thị giác máy tính, chỉ tác vụ chia một bức ảnh thành nhiều phần khác nhau dựa trên mục đích phân đoạn.

Phân đoạn cá thể là một nhánh của phân đoạn hình ảnh**,** chỉ tác vụ nhận dạng (bao gồm xác định vị trí và lớp đối tượng) và vẽ một đường bao sát từng cá thể thuộc các lớp đối tượng mà người dùng quan tâm, phần còn lại không thuộc bất kỳ đối tượng nào sẽ được xem là *nền (background)*. Các phương pháp truyền thống sử dụng giải thuật đều chỉ dừng lại ở mức phân đoạn những cá thể có sự đồng nhất về màu sắc**,** hình dạng hay kích thước nhất đinh. Tuy nhiên với sự phát triển của phần cứng máy tính và sự ra đời của phương pháp học máy và học sâu, ngày tính máy tính đã có thể thực hiện tác vụ phân đoạn cá thể bất kể cá thể đó có nhiều màu sắc hay ở bất cứ hình dạng và kích thước nào.

2. *Phát biểu bài toán*

Khi chúng ta có ảnh đầu vào và một tập các lớp đối tượng mong muốn phân đoạn, **Phân đoạn cá thể (Instance Segmentation)** chính là nhận dạng từng cá thể thuộc một trong các lớp đã cho xuất hiện trong ảnh đầu vào bằng cách làm nổi bật các pixel cụ thể của cá thể đó. Ta có input và output của bài toán như sau:

Input: Một ảnh cần phân đoạn với các đối tượng cụ thể (specific object) có kích thước W x H

Output: K kết quả phân đoạn, trong đó mỗi kết quả phân đoạn bao gồm các thông tin:

- Class id (thuộc tập các lớp mong muốn phân đoạn)
- Bounding box: [cx, cy, w, h] (tọa độ tâm + kích thước)
- Binary mask có kích thước W x H (trùng với kích thước ảnh đầu vào)

Để xác định các thông tin trên, ta cần giải quyết từng **tác vụ đơn** dưới đây. Ở đây, chúng tôi sử dụng ảnh đầu vào sau để minh họa cho từng giai đoạn:

Phát sinh ứng viên (Region Proposal): là tác vụ tìm kiếm các khung (bounding box) có khả năng chứa **một** cá thể thuộc lớp bất kỳ trong ảnh đầu vào.

Input: Ảnh đầu vào có kích thước W x H

Output: Các bounding box: [cx, cy, w, h] (tọa độ tâm + kích thước) tại các vị trí nghi ngờ có cá thể → Region of Interest (ROI).

Các bounding box khi vẽ lên lại ảnh đầu vào minh họa ta sẽ được ảnh như sau:

Phân lớp ảnh (Image Classification) là tác vụ giúp nhận dạng ảnh chứa cá thể thuộc lớp trong những lớp đã cho. Tác vụ này thường đi chung với *Phát sinh ứng viên* vì bounding box của *Phát sinh ứng viên* sẽ giúp chúng ta crop được những vị trí nghi ngờ có cá thể từ ảnh đầu vào.

Input: Ảnh cần phân loại có kích thước w x h (có thể được crop ra theo các bounding box của ảnh lớn).

Output: Class id tương ứng với ảnh (bao gồm các class muốn phân đoạn và background class).

Với mỗi bounding box từ output của *Phát sinh ứng viên*, ta crop ảnh đầu vào ở vị trí đó và cho qua *Phân lớp ảnh*. Ngoài trừ những ảnh có class background, các ảnh còn lại cùng class id sẽ được ánh xạ ngược lại vào ảnh đầu vào, ta được ảnh minh họa như sau:

Phân đoạn ảnh (Image Segmentation) là tác vụ cuối cùng, khi chúng ta có ảnh và lớp đối tượng cần phân đoạn, *Phân đoạn ảnh* sẽ giúp bật từng pixel của đối tượng thuộc lớp tương ứng lên trong ảnh.

Input: Ảnh cần phân đoạn có kích thước w x h (có thể được crop ra theo các bounding box của ảnh lớn) và class id tương ứng.

Output: Binary mask có kích thước w x h biểu diễn các pixel cụ thể của đối tượng thuộc class id từ input.

Với class id từ *Phân loại ảnh* và bounding box được ánh xạ ngược lại, ta crop ảnh đầu vào ở vị trí đó và cho qua *Phân đoạn ảnh*. Binary mask output sẽ được ánh xạ ngược lại ảnh đầu vào, ta được ảnh minh họa như sau (với mỗi binary mask được tô màu riêng biệt):

Tổng kết lại **Framework** của bài toán sẽ như sau:

3. *Đóng góp*

Trong báo cáo này, chúng tôi sẽ trình bày về giải pháp sử dụng mạng học sâu Mask R-CNN [1] để giải quyết bài toán *Phân đoạn cá thể* và tiến hành chạy thực nghiệm để kiểm tra độ hiệu quả của phương pháp.

Vào thời điểm ra mắt, Mask R-CNN được đánh giá là *State-of-the-art* và đã đánh một dấu mốc quan trọng trong việc sử dụng mạng học sâu, bởi kết quả của nó đạt được hoàn toàn vượt xa những mô hình máy học hay phương pháp truyền thống được đưa ra trước đó.

Với kiến trúc đơn giản và dễ dàng cải tiến, Mask R-CNN đã tạo nền móng cho những *State-of-the-art* sau này phát triển dựa trên nó. Dù vậy, đến ngày hôm nay, Mask R-CNN vẫn được ứng dụng rộng rãi trong nền công nghiệp 4.0 bởi hiệu suất phân đoạn cao và tốc độ xử lý được phần nào cải thiện nhờ sự phát triển của phần cứng máy tính.

II. Nghiên cứu liên quan

1. *R-CNN*:

Mask R-CNN được xây dựng dựa trên hướng tiếp cận Region-based CNN (R-CNN) [2]. Đây là một trong hướng tiếp cận bằng học máy đầu tiên để giải quyết bài toán nhận dạng **nhiều** vật thể trong cùng bức ảnh và được kết quả tương đối tốt. Những cải tiến sau đó như Fast R-CNN [3] hay Faster R-CNN [4] giúp cải thiện tốc độ xử lý nhưng vẫn chỉ dừng ở tác vụ nhận dạng. Đến Mask R-CNN, nhóm tác giả đã thay đổi kiến trúc Faster R-CNN một vài điểm và thêm một nhánh xử lý vào để giúp thực hiện tác vụ phân đoạn.

2. *Instance Segmentation*

Từ khi Mask R-CNN được công bố, tác vụ *phân đoạn cá thể* trong thị giác máy tính đã có nhiều bước chuyển vượt bật, nhưng hầu hết đều dựa trên kiến trúc Mask R-CNN mà cải tiến. PointRend [5] là một cải tiến tuy nhỏ nhưng giúp cải thiện chất lượng phân đoạn của Mask R-

CNN rất nhiều lần, bằng cách thay thế bước nội suy *binary mask* bằng giải thuật truyền thông thành nội suy bằng một mạng neural nhỏ. Một số cải tiến đáng lưu ý khác như YOLACT [6], CenterMask [7] giúp cải thiện tốc độ phân đoạn để đáp ứng nhu cầu thời gian thực bằng cách sử dụng một mô hình nhỏ hơn nhưng bù lại nó cũng làm giảm chất lượng phân đoạn.

III. Phương pháp

1. *Giai đoạn Offline*

a. *Chuẩn bị dữ liệu*

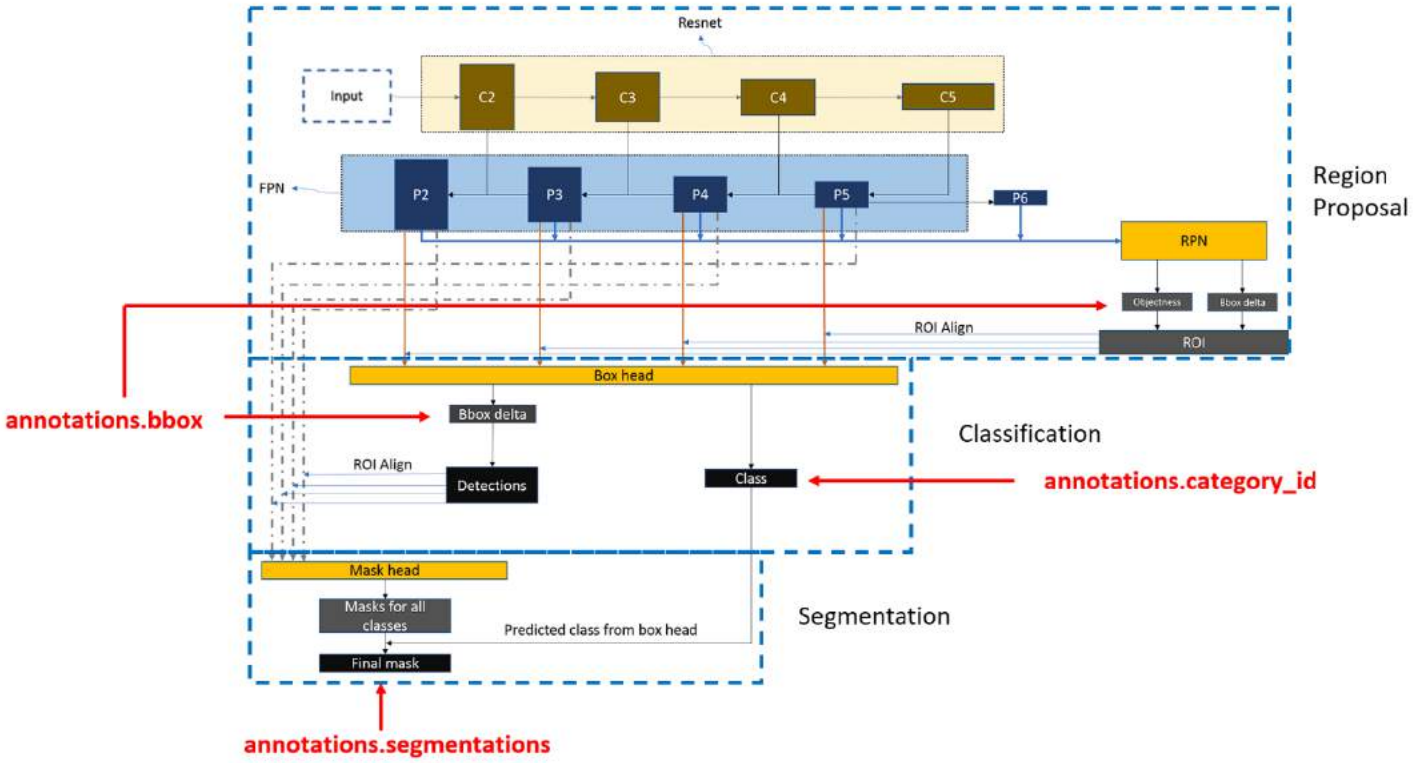
Đối với các giải pháp máy học, việc chuẩn bị dữ liệu là vô cùng quan trọng, dữ liệu huấn luyện là yếu tố quan trọng nhất quyết định sự thành công của mô hình. Mô hình sẽ được tối ưu để cho ra kết quả tốt với dữ liệu huấn luyện. Nếu dữ liệu huấn luyện càng giống với thực tế ứng dụng thì mô hình cũng sẽ chạy càng tốt trong thực tế. Đối với bài toán phân đoạn như Mask R-CNN, dữ liệu huấn luyện phải bao gồm những thông tin sau:

- Danh sách các **ảnh input** được đánh id (có thể là tên ảnh).
- Danh sách các **lớp** đối tượng cần phân đoạn, mỗi phần tử là một map gồm các key sau:
  - class_id: int
  - name: string
- Danh sách các **annotation** cho từng cá thể, dùng cho quá trình đánh giá output của mô hình, mỗi annotation là một map gồm các key sau:
  - image_id: kiểu dữ liệu tùy vào cách đánh id của ảnh input.
  - class_id: int
  - bbox: [cx: float, cy: float, w: float, h: float]: thể hiện vị trí tâm và kích thước của bounding box bao quanh cá thể.
  - segmentations: [x1: float, y1: float, x2: float, y2: float, …, xn: float, yn: float]: thể hiện polygon bao sát vật thể, với điểm (xn, yn) được nối liền với điểm (x1, y1).

Một số tập dữ liệu được cung cấp sẵn như COCO dataset [8] (> 200.000 ảnh, 80 lớp, 1.500.000 cá thể) hoặc có thể dùng những phần mềm gán nhãn thủ công như Labelme [9] nếu muốn sử dụng những bộ dữ liệu đặc thù. Số lượng, độ phân hóa về hình dạng, màu sắc và kích thước của các cá thể trong dữ liệu huấn luyện càng lớn thì chất lượng đầu ra sẽ càng tốt.

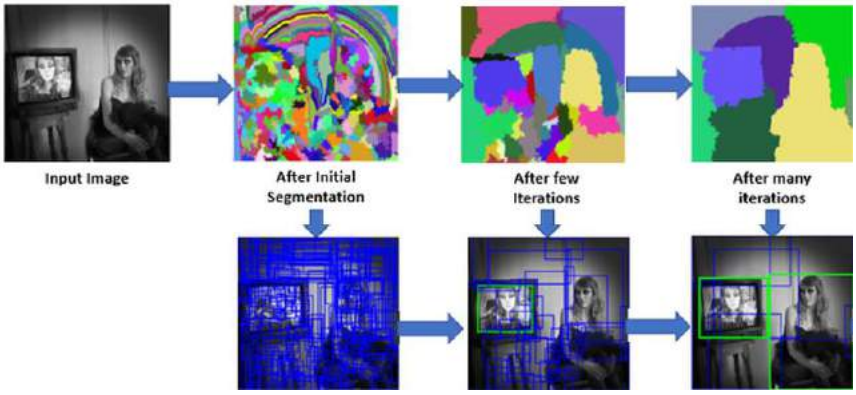
b. Kiến trúc Mask R-CNN

Flow của mô hình Mask R-CNN cho quá trình huấn luyện được minh họa trong hình sau, các thông tin **annotations** của ảnh input tương ứng được đưa vào mô hình để đánh giá và cập nhật trọng số của các lớp mạng thần kinh (source code)



**** Đề xuất ứng viên (Region Proposal):***

Thời kỳ đầu của R-CNN và Fast R-CNN, tác vụ đề xuất ứng viên được thực hiện bằng thuật toán Selective Search, dựa trên ý tưởng Region Growing. Giải thuật bước đầu sẽ sử dụng thuật toán phân đoạn ảnh của Felzenszwalb và Huttenlocher [10] để tìm các vùng phân đoạn cơ sở. Sau mỗi *iteration*, thuật toán sẽ gộp những phân vùng lân cận có độ dị biết thấp hơn một mức ϵ, mức này sẽ tăng dần sau từng *iteration*. Kết thúc nếu kết quả vòng lặp cuối không thay đổi. Cuối cùng, thuật toán sẽ chọn khoảng 2000 phân vùng to nhất từ *iteration* cuối lên đầu để tạo ra các *bounding box* ứng viên. Hình minh họa của giải thuật như sau:



Giải thuật trên vẫn còn nhiều nhược điểm như không thể lấy được các vật nhỏ hay tốc độ xử lý chậm vì phải duyệt ảnh rất nhiều lần. Đến năm 2015, Faster R-CNN sử dụng một mạng nơron nhỏ gọi là *Region Proposal Network (RPN)* để đề xuất ứng viên thay cho *Selective Search*.

Đầu tiên ảnh đầu vào sẽ được rút trích đặc trưng bằng một mạng học sâu như VGG hay ResNet để rút ra *feature map* (đặc trưng về cấu trúc không gian của ảnh). Kích thước của *feature map* nhỏ hơn rất nhiều so với ảnh ban đầu (mạng VGG, ResNet giúp giảm kích thước mỗi chiều đi 32 lần), bù lại số channel tăng lên nên vẫn bảo toàn được thông tin cấu trúc của ảnh.

Sau đó phương pháp sẽ duyệt trên từng ô của *feature map* này một lần duy nhất, tại mỗi ô sẽ sinh ra *k anchor box* có kích thước cố định (được quy định khi xây dựng model).

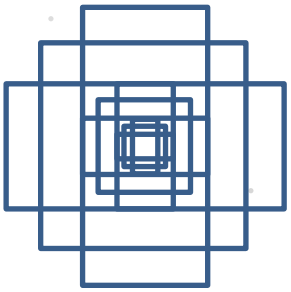
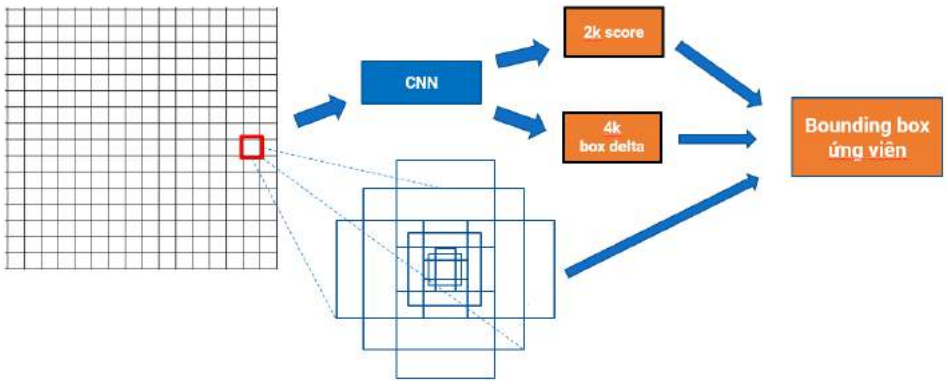


Figure 1: Minh họa cho k = 9

Cách sinh *anchor box* này đảm bảo mô hình có thể nhận dạng được các cá thể thuộc mọi kích thước và hình dạng khác nhau. Ô *feature map* đang xét sau đó sẽ được đưa vô *RPN* để kiểm tra từng *anchor box* có chứa vật thể không và tinh chỉnh *anchor box* đó, sau đó những *anchor box* chứa vật thể sẽ được làm *bounding box* ứng viên, tổng kết quá trình sẽ như sau:



Objectness score và *bbox delta* sẽ được *RPN* học để cực tiểu hóa các hàm *loss* sau trên tập dữ liệu huấn luyện:

$$loss_{rpn\_cls} = loss_{pos} + loss_{neg}$$

$$= -\frac{1}{k}\sum_{i=1}^{k} gt\_scores[i] \cdot \log(scores[2i-1]) + (1 - gt\_scores[i]) \cdot \log(scores[2i])$$

$$loss_{rpn\_delta} = \frac{1}{k}\sum_{i=1}^{k}\sum_{j \in cx,cy,w,h} smooth(gt\_delta[i][j], delta[i][j])$$

Với $smooth(x,y) = \begin{cases} 0.5(x-y)^2, & nếu\ |x-y| < 1 \\ |x-y| - 0.5, & nếu\ ngược\ lại \end{cases}$

Trong đó:

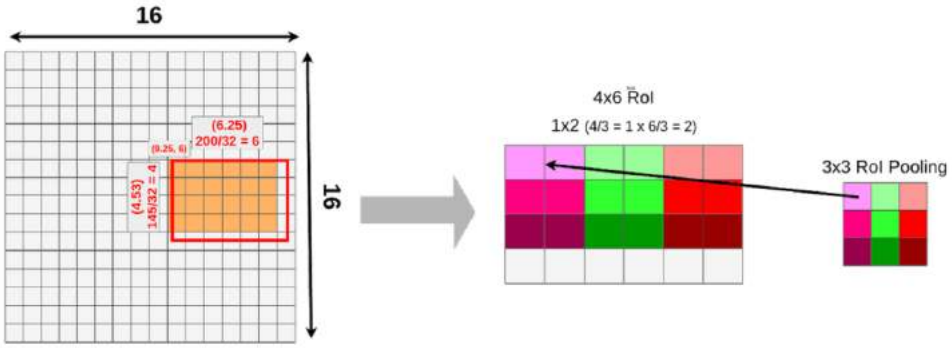
- k: số anchor box
- gt_scores: nếu anchor box thứ i có IoU với bất kỳ **annotations.bbox** > 0.7 thì gt_scores[i] = 1, ngược lại gt_scores[i] = 0.
- scores: kết quả dự đoán (thuộc [0, 1]). (anchor thứ i có xác suất scores[2i-1] chứa cá thể và scores[2i] không chứa cá thể -> chọn kết quả lớn hơn)
- gt_delta: **annotations.bbox** – anchor box.
- delta: kết quả hiệu chỉnh anchor box.

Ý tưởng sử dụng *Region Proposal Network* vẫn được Mask R-CNN kế thừa lại. Mask R-CNN sử dụng mạng rút đặc trưng *Feature Pyramid Network (FPN)* [11] thay vì VGG hay ResNet vì thời gian Faster R-CNN được công bố thì *FPN* vẫn chưa được công bố. Điểm cải tiến của *FPN* là đưa ra nhiều *feature map* biểu diễn các đặc trưng từ đơn giản đến phức tạp, giúp việc nhận dạng các vật thể nhỏ dễ dàng hơn, vì chúng thường bị suy biến khi rút đặc trưng bằng VGG hay ResNet.

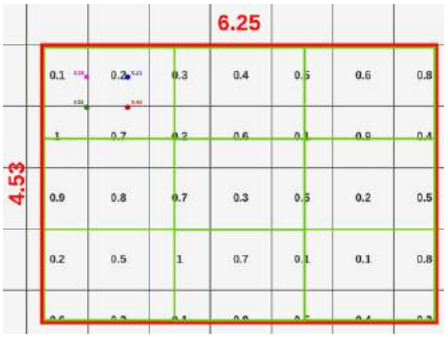
**** Phân lớp (Classification):***

Khi có được các *bounding box* ứng viên, thay vì crop ảnh input, chúng ta sẽ crop trực tiếp trên *feature map* để tránh việc rút trích đặc trưng nhiều lần. Tuy nhiên, mô hình mạng để phân lớp cần input đầu vào có kích thước cố định.

Để giải quyết vấn đề đó, Faster R-CNN đề xuất sử dụng kỹ thuật *ROIPooling*, làm tròn kích thước của *bounding box* sang tọa độ nguyên, tiếp tục làm tròn kích thước *bounding box* về thành bội của 3. Sau đó chia phần *feature map* thuộc *bounding box* thành 9 phần và lấy ô có giá trị lớn nhất trong từng phần (cũng có thể lấy trung bình các ô) làm giá trị cuối.



Kỹ thuật trên có thể đưa ra kết quả tốt với tác vụ nhận dạng, tuy nhiên trong phân đoạn, chúng ta cần sự chính xác nhất có thể, việc làm tròn tới 2 lần sẽ gây mất mát rất nhiều thông tin. Mask R-CNN đã giải quyết vấn đề này bằng *ROIAlign* thay cho *ROIPooling*. *ROIAlign* vẫn giữ nguyên tọa độ thực của *bounding box* và chia thành 9 phần như *ROIPooling*. Trong mỗi phần sẽ chọn 4 điểm mốc cách đều nhau một khoảng bằng 1/3 kích thước từng phần và nội suy song tuyến tính để tìm giá trị 4 điểm mốc đó. Giá trị cuối được lựa chọn bằng cách chọn giá trị lớn nhất trong 4 điểm mốc.



Phần *feature map* sau khi thực hiện *ROIAlign* sẽ được đưa vô một mạng *fully-connected* gồm 2 nhánh để tìm ra *class id* tương ứng với *bounding box* đầu vào và tinh chỉnh lại *bounding box* đầu vào để tăng độ chính xác. Mạng sẽ tìm *class_id* và *bbox delta* sao cho những hàm *loss* sau đạt cực tiểu trên tập dữ liệu huấn luyện:

$$loss_{cls} = -\frac{1}{num\_class} \sum_{i=1}^{num\_class} gt\_class[i] \cdot \log\left(pred\_class[i]\right)$$

$$loss_{delta} = \sum_{i \in cx,cy,w,h} smooth(gt\_delta[i], delta[i])$$

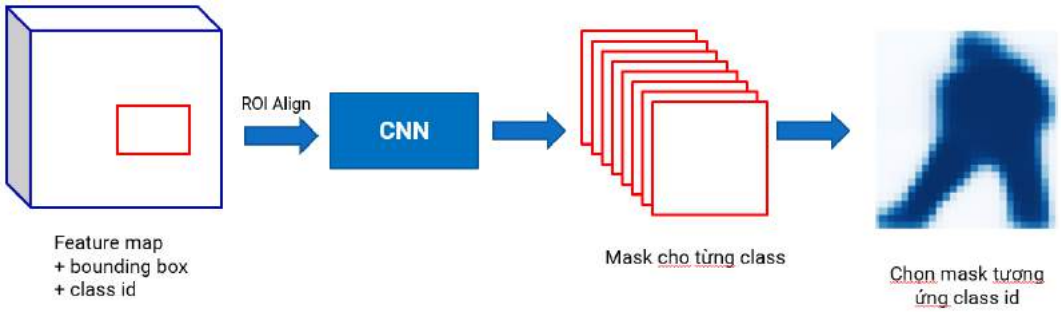
Trong đó:

- num_class: số lượng class được định nghĩa trước.
- gt_class: gt_class[**annotations.class_id**] = 1, gt_class[j] = 0 với mọi j khác.
- pred_class: kết quả dự đoán (thuộc [0,1]) từng class cho bounding box đang xét.
- gt_delta: **annotations.bbox** – bounding box.
- delta: kết quả hiệu chỉnh bounding box.

Nếu chúng ta dừng tại đây, những phần phía trên chính là kiến trúc của Faster R-CNN. Nhưng trong Mask R-CNN, nhóm tác giả đã gắn thêm một kiến trúc mạng nhỏ phía sau để phục vụ tác vụ phân đoạn.

**** Phân đoạn (Segmentation):***

Với *bounding box* được tinh chỉnh ở phần phân lớp, chúng ta lại crop *feature map* từ bước rút trích đặc trưng và cho qua *ROIAlign*. Phần *feature map* kết quả sẽ được cho qua một 1 mạng *Convolution* với output có kích thước **tổng số class** x 28 x 28. Cuối cùng, lấy output thứ *class_id* (kết quả phân lớp) và *resize* lại cho kích thước ban đầu của *bounding box* ta được *binary mask* cần tìm.



Chỉ có *binary mask* của lớp *class_id* mới tham gia quá trình huấn luyện. Hàm *loss* cho mạng CNN phân đoạn như sau:

$$loss_{mask} = -\frac{1}{mask\_size}\sum_{i=1}^{mask\_size} gt\_mask[i] \cdot \log\left(pred\_mask[class\_id][i]\right) + (1 - gt\_mask[i]) \cdot \log\left(1 - pred\_mask[class\_id][i]\right)$$

- mask_size: tổng số pixel của một mask (28 x 28)
- gt_mask: gt_mask[i] = 1 nếu pixel thứ i thuộc **polygon annotations.segmentations**, ngược lại gt_mask[i] = 0.
- pred_class: kết quả dự đoán (thuộc [0,1]) từng pixel cho từng class.

Binary mask sẽ được *padding* dựa trên vị trí của *bounding box* trên ảnh đầu vào để tạo ra *binary mask* có kích thước giống với ảnh ban đầu.

2. *Giai đoạn Online*

Với mô hình Mask R-CNN đã được huấn luyện ở giai đoạn offline, chúng ta có thể đưa mô hình lên giai đoạn online. Khi cho ảnh input cần phân đoạn vào, mô hình sẽ xuất cho chúng ta danh sách các kết quả phân đoạn, mỗi kết quả bao gồm:

- Class id: int hoặc string (thuộc tập các lớp trong tập huấn luyện)
- Bounding box: [cx: float, cy: float, w: float, h: float] (tọa độ tâm + kích thước)
- Binary mask có kích thước W x H (trùng với kích thước ảnh input)

IV. Thực nghiệm và báo cáo kết quả

1. *Mô tả thí nghiệm*

Tập ảnh train (lấy từ COCO 2017 train dataset) bao gồm 2500 ảnh, 80 lớp, 23533 cá thể được gán nhãn: Ảnh, Nhãn

Tập ảnh test bao gồm 300 ảnh, 80 lớp, 2594 cá thể được gán nhãn (lấy từ COCO 2017 validation dataset): Ảnh, Nhãn

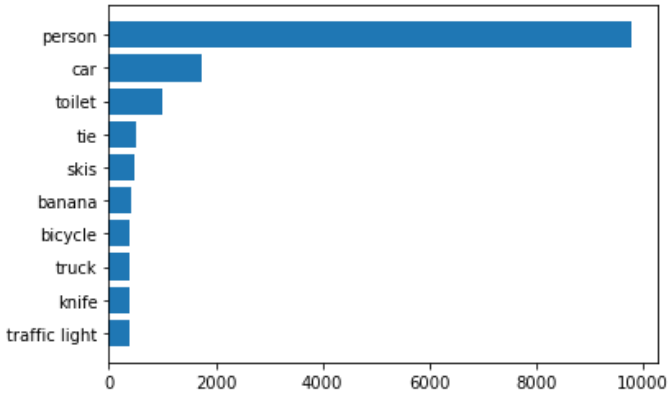


Figure 2: 10 lớp có nhiều cá thể được gán nhãn nhất trong tập train

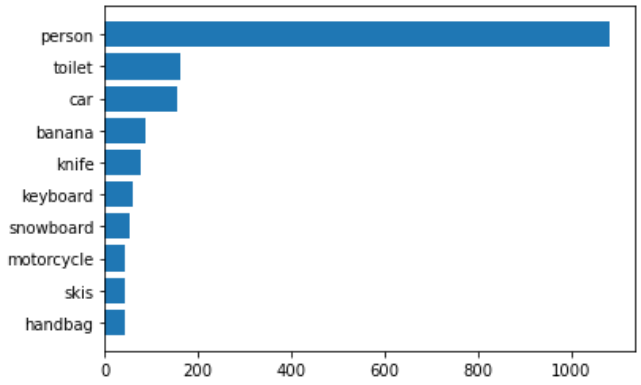


Figure 3: 10 lớp có nhiều cá thể được gán nhãn nhất trong tập test

Code sử dung: Notebook, backbone rút trích đặc trưng là ResNet-101-FPN

Môi trường thử nghiệm: Google Colaboratory

Thời gian train: 8 tiếng trên GPU Nvidia K80 Tesla

Transfer learning với model pretrained: COCO. Chúng tôi chỉ giữ lại lớp rút trích đặc trưng và đề xuất ứng viên, các lớp còn lại được train lại từ đầu.

2. *Kết quả và phân tích đánh giá*

Kết quả được công bố trong bài báo, trong đó Average Precision (AP) là một độ đo phổ biến dùng trong các tác vụ nhận dạng và phân đoạn:

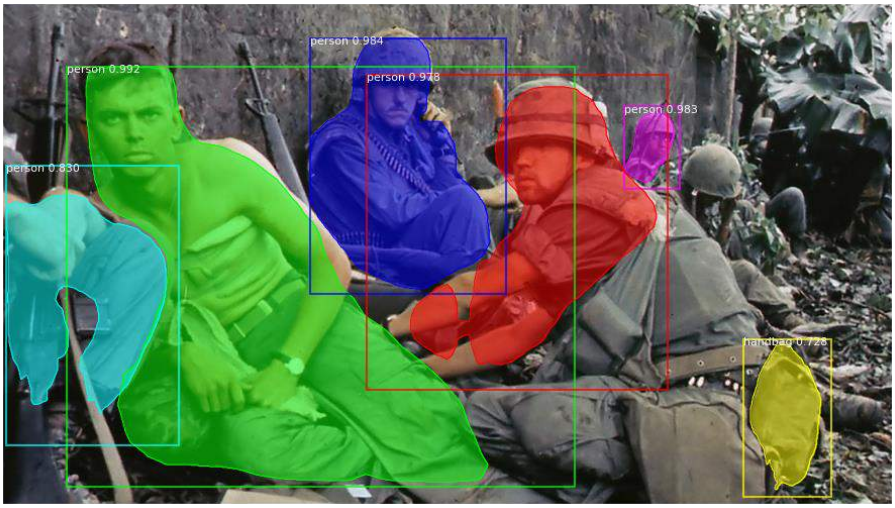
net-depth-features	AP	AP_{50}	AP_{75}
ResNet-50-C4	30.3	51.2	31.5
ResNet-101-C4	32.7	54.2	34.3
ResNet-50-FPN	33.6	55.2	35.3
ResNet-101-FPN	35.4	57.3	37.5
ResNeXt-101-FPN	**36.7**	**59.5**	**38.9**

Chúng tôi thực hiện đánh giá kết quả từ model đã được train. Chúng tôi có bảng so sánh sau:

	Kích thước tập train - test	Số lớp	AP @ IoU=0.5
Chúng tôi	2500 - 300	80	60.4%
Tác giả	80000 - 5000	80	57.3%

Dựa vào kết quả trên, chúng tôi có thể kết luận mô hình hoạt động hiệu quả và có thể dễ dàng áp dụng trên tập dữ liệu mới. Các lớp rút đặc trưng có thể sử dụng trong hầu hết mọi loại đối tượng nên có thể áp dụng transfer learning để rút ngắn thời gian huấn luyện.

Demo phân đoạn của ảnh input:



Ngoài ra chúng tôi còn kiểm tra thời gian truy xuất kết quả của mô hình:

	Chúng tôi	Công bố
FPS	1.03	5

GPU chúng tôi sử dụng (K80 Tesla) chỉ có *compute capability* 3.7, nằm ở mức thấp trong số các GPU của NVIDIA nên tốc độ như trên là hoàn toàn hợp lý. Qua đây chúng tôi kết luận rằng Mask R-CNN vẫn chưa thể áp dụng vào các ứng dụng thời gian thực một cách rộng rãi được.

V. Ứng dụng

Sự ra đời của Mask R-CNN đã góp phần thúc đẩy quá trình sản xuất tự động lên một tầm cao mới. Việc máy tính có thể biết được vị trí và hình dáng chính xác của các vật thể sẽ giúp máy tính đưa ra quyết định thay cho con người trong những công việc có tính chất lặp lại. Hơn nữa, máy tính có thể hoạt động liên tục không ngừng nghỉ, đó là lý do mọi người luôn muốn tự động hóa những tác vụ có tính chất lặp lại để tiết kiệm chi phí và dành nhân công cho những tác vụ khác. Vì tốc độ xử lý khá chậm, các ứng dụng của Mask R-CNN thường không chú trọng vào tốc dộ và ưu tiên độ chính xác của kết quả. Một số ứng dụng của Mask R-CNN có thể kể đến như:

- Phân loại rác [12]: Phân loại rác là một công việc có môi trường đặc biệt độc hại. Do đó Mask R-CNN được sử dụng để máy tính có thể phân đoạn các loại rác khác nhau, sau đó các cánh tay robot để dựa vào kết quả đó để sắp xếp các khối rác về đúng loại của nó. Việc tự động này còn giúp làm giảm số bệnh nhân mắc những chứng bệnh liên quan đến đường hô hấp do làm việc trong môi trường độc hại.
- Robot thu hoạch trái cây [13]: Việc kiểm tra cả vườn trái cây, đặc biệt ở những vùng nông thôn, tốn rất nhiều thời gian và công sức. Đó là lý do robot thu hoạch trái cây ra đời. Mask R-CNN giúp robot nhận dạng riêng biệt những trái cây nằm chồng lên nhau, từ đó robot có thể thu hoạch chính xác những trái còn chính hay loại bỏ những trái bị sâu bọ ăn.
- Quản lý tàu thuyền ra vào cảng [14]: Để có một cái nhìn tổng quan, các cảng thuyền sử dụng các flycam để liên chụp ảnh cảng từ trên khung trung và gửi cho trung tâm. Tại đây, Mask R-CNN sẽ được sử dụng để chỉ ra chính xác vị trị, hình dáng và góc độ của từng con thuyền, bởi không phải con thuyền nào cũng có thể nhìn rõ bằng mắt thường nếu được chụp từ trên cao. Từ đó, các nhân viên có thể đưa ra quyết định chính xác để điều hướng, giúp các tàu thuyền không va chạm vào nhau.
- Quy hoạch đất đai [15]: Từ những bức ảnh vệ tinh hoặc từ trên không trung, các tổ chức, nhà thầu muốn xác định rõ những căn nhà nào nằm ở trong vùng quy hoạch. Mask R-CNN sẽ giúp họ phân đoạn từng ngôi nhà riêng biệt, từ đó họ có thể tính diện tích để đưa ra kế hoạch đền bù một cách hợp lý.

C. Tài liệu tham khảo

[1] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

[2] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[3] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[4] Jiang, H., & Learned-Miller, E. (2017, May). Face detection with the faster R-CNN. In 2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017) (pp. 650-657). IEEE.

[5] Kirillov, A., Wu, Y., He, K., & Girshick, R. (2020). Pointrend: Image segmentation as rendering. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9799-9808).

[6] Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). Yolact: Real-time instance segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 9157-9166).

[7] Lee, Y., & Park, J. (2020). Centermask: Real-time anchor-free instance segmentation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 13906-13915).

[8] Tsung-Yi Lin and Michael Maire and Serge J. Belongie and Lubomir D. Bourdev and Ross B. Girshick and James Hays and Pietro Perona and Deva Ramanan and Piotr Dollár and C. Lawrence Zitnick (2014). Microsoft COCO: Common Objects in Context. CoRR, abs/1405.0312.

[9] Wada, K. Labelme: Image Polygonal Annotation with Python [Computer software]. https://doi.org/10.5281/zenodo.5711226.

[10] P. F. Felzenswalb, D. P. Huttenlocher. Efficient graph-based image segmentation. International Journal of Computer Vision 59 (2) (2004) 167–181.

[11] Tsung-Yi Lin and Piotr Dollár and Ross B. Girshick and Kaiming He and Bharath Hariharan and Serge J. Belongie (2016). Feature Pyramid Networks for Object Detection. CoRR, abs/1612.03144.

[12] Instance Segmentation for Waste Management AI (2021). https://keymakr.com/blog/instance-segmentation-for-waste-management-ais

[13] Weikuan Jia, Yuyu Tian, Rong Luo, Zhonghua Zhang, Jian Lian, and Yuanjie Zheng. Detection and segmentation of overlapped fruits based on optimized mask r-cnn application in apple harvesting robot.Computers and Electronics in Agriculture, 172:105380, 2020.

[14] Shanlan Nie, Zhiguo Jiang, Haopeng Zhang, Bowen Cai, and Yuan Yao. Inshore ship detection based on mask r-cnn. InIGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium,pages 693–696, 2018.

[15] Sharada Prasanna Mohanty, Jakub Czakon, Kamil A. Kaczmarek, Andrzej Pyskir, Piotr Tarasiewicz, Saket Kunwar, Janick Rohrbach, Dave Luo, Manjunath Prasad, Sascha Fleer, Jan Philip G¨opfert,Akshat Tandon, Guillaume Mollard, Nikhil Rayaprolu, Marcel Salathe, and Malte Schilling. Deep learning for understanding satellite imagery: An experimental survey. Frontiers in Artificial Intelligence,3:85, 2020.