MASK DETECTION WITH DETECTRON2 AND FASTER R-CNN

Hà Nhật Minh Thân Trọng Thành Nguyễn Tiến Dương
Học Viện Kỹ Thuật Quân Sự

ABSTRACT

Đại dịch COVID-19 trên toàn thế giới đã dẫn đến việc đeo khẩu trang rộng rãi để bảo vệ bản thân và những người khác khỏi lây lan dịch bệnh. Điều đặc biệt quan trọng là chúng phải được đeo khi không thể tránh được tiếp xúc gần và khi ở bên trong các tòa nhà. Mục tiêu là tạo ra một mô hình phát hiện đối tượng có thể xác định xem một người có đeo mặt nạ đúng, sai hay không. Ở đây, "chính xác" được định nghĩa là che mũi và miệng. Mô hình này có thể được sử dụng với cả ảnh tĩnh cũng như nguồn cấp dữ liệu video trực tiếp.

Index Terms— Object detection, Detectron2, Mask Detection, Faster R-CNN

1. INTRODUCTION

Sự lây lan của COVID-19 ngày càng đáng lo ngại đối với tất cả mọi người trên thế giới. Virus này có thể bị ảnh hưởng từ con người tới con người qua các giọt nhỏ và trong không khí. Theo hướng dẫn của WHO, để giảm sự lây lan của COVID-19, mọi người cần đeo khẩu trang, tránh xa xã hội, né tránh khu vực đám đông và cũng luôn duy trì khả năng miễn dịch hệ thống. Vì vậy, để bảo vệ nhau, mỗi người hãy đeo khẩu trang đúng cách khi họ ở ngoài trời. Tuy nhiên, hầu hết những người ích kỷ sẽ không đeo mặt nạ đúng với rất nhiều lý do. Để khắc phục tình trạng này, tính năng phát hiện mặt nạ mạnh mẽ cần được phát triển. Để phát hiện mặt nạ, thuật toán phát hiện đối tượng có thể được thực hiện. Tình trạng của nghệ thuật của thuật toán phát hiện đối tượng có một biểu diễn là Detectron2. Detectron2 được xây dựng bởi Facebook AI Research (FAIR) để hỗ trợ triển khai và đánh giá nhanh chóng các nghiên cứu thị giác máy tính mới. Nó bao gồm các triển khai cho các thuật toán phát hiện sau:

- Mask R-CNN
- RetinaNet
- Faster R-CNN
- RPN
- Fast R-CNN
- Tensor Mask
- PointRent
- and more

2. DATASET AND METHOD

Data ở đây lấy từ cuộc thi datacomp.io được tổ chức bởi FPT , bộ dữ liệu cung cấp gồm 976 ảnh có gán nhãn bounding box theo 3 lớp: đeo khẩu trang, không đeo khẩu trang và đeo khẩu trang sai; 1 pre-trained model; 1 tập public test gồm 89 ảnh để có thể đánh giá sơ bộ về mô hình. Đối với thách thức này, phương pháp đánh giá dựa trên Điểm F1 được xác định để cân bằng độ chính xác (p) và độ thu hồi (r).Chúng được định nghĩa là:

$$p = \frac{C_d}{P_d}, r = \frac{C_d}{A_d}, F1 = \frac{2pr}{p+r}$$

trong đó Cd, Pd và Ad là những con số được dự đoán chính xác thiệt hại, thiệt hại dự đoán và tất cả sự thật cơ bản thiệt hại từ tập hợp đánh giá, tương ứng. Hơn nữa, định nghĩa về thiệt hại được dự đoán đúng có hai tiêu chí. Chúng là 1) hộp giới hạn dự đoán phải khớp và 2) nhãn dự đoán là đúng. Điều sau là hiển nhiên, và trước đây được xác định bởi điểm Giao nhau trên Liên minh (IoU), được định nghĩa như sau:

$$IuO = \frac{area(P_b \bigcap G_b)}{area(P_b \bigcup G_b)}$$

trong đó Pb và Gb là hộp dự đoán và hộp chân lý cơ bản, tương ứng. Ngoài ra, diện tích $(Pb \bigcap Gb)$ và diện tích $(Pb \bigcup Gb)$ có nghĩa là khu vực giao nhau và sự kết hợp giữa hai hộp, tương ứng. Trong trường hợp này, nếu $IoU >= 0,5$, thì nó là một trận đấu, và nó không phải là khác.

Mặc dù Fast R-CNN cải thiện việc đào tạo và dự đoán thời gian, nó vẫn cần đề xuất khu vực làm đầu vào. Trong khác từ, các đề xuất khu vực cho mỗi hình ảnh vẫn cần được thực hiện riêng biệt (ví dụ: sử dụng kỹ thuật xử lý hình ảnh). Do đó, Ren và cộng sự, đề xuất Faster R-CNN để giải quyết vấn đề này. Cải tiến chính của nó là khả năng kết hợp đề xuất khu vực như một phần của mô hình cuối cùng sử dụng Khu vực Mạng đề xuất (RPN). Nói cách khác, có hai mạng trong kiến trúc này. Đầu tiên là Đề xuất khu vực Mạng (RPN) và mạng thứ hai là Fast R-CNN. Hai mạng con này được đào tạo đồng thời, mặc dù hai nhiệm vụ khác nhau: 1) đề xuất khu vực và 2) hộp giới hạn phân loại và hồi quy. Những chiến lược này giúp cải thiện độ chính xác và thời gian đào tạo và phát hiện đối tượng.

Phương pháp luận chung của chúng tôi là chúng tôi bắt đầu với dữ liệu giai đoạn thăm dò để hiểu tập dữ liệu. Sau đó chúng tôi tiến hành bằng cách tách tập dữ liệu đào tạo thành

quá trình đào tạo và các bộ đánh giá. Việc xác nhận cho phép chúng tôi đánh giá siêu tham số cho các kiến trúc của chúng ta một cách định lượng. Về kiến trúc mô hình học sâu, chúng tôi bắt đầu với các kiến trúc mô hình thường được sử dụng để phát hiện hư hỏng đường và nhiệm vụ phân loại. Sau đó, chúng tôi tiến hành các chiến lược để cải thiện các mô hình cơ sở, chẳng hạn như thay đổi siêu tham số, dữ liệu tàu tăng cường và thử nghiệm tăng thời gian. Điều đáng chú ý quần thể và các mô hình riêng lẻ cho các quốc gia riêng lẻ sẽ có kết quả dự đoán tốt hơn. Tuy nhiên, nó bị hạn chế thách thức rằng chúng ta nên có một thuật toán duy nhất và cách tiếp cận mô hình đơn. Do đó, chúng tôi không cố gắng hướng.

3. EXPERIMENTAL RESULTS AND ANALYSIS

A. Data:

Dựa trên bộ dữ liệu gốc của DatacompFPT 2021, chúng tôi đã sử dụng các công cụ sinh dữ liệu để tạo ra các bộ dữ liệu khác nhau nhằm tăng dữ liệu học cho mô hình. Công cụ chính chúng tôi sử dụng ở đây là Roboflow `https://roboflow.com/` với khác kĩ thuật xoay ảnh, nghiên ảnh, thay đổi độ sáng, độ tương phản, làm mờ và làm xám ảnh. Bộ dữ liệu gồm 3 lớp không đeo khẩu trang (0), đeo khẩu trang (1), đeo khẩu trang sai (2)

Chúng tôi có 5 bộ dữ liệu chính trong thí nghiệm:

- Bộ dữ liệu train1: dữ liệu gốc từ cuộc thi
- Bộ dữ liệu train2: dữ liệu được đánh nhãn lại bằng tay từ train1
- Bộ dữ liệu train3: dữ liệu được sinh từ Roboflow với bộ train2 với các kĩ thuật kể trên
- Bộ dữ liệu train4: dữ liệu được sinh từ Roboflow với bộ train2 tuy nhiên chỉ áp dụng sinh cho class 2 (incorrect mask) và 30
- Bộ dữ liệu train5: dữ liệu được đánh nhãn lại bằng tay và loại bỏ các mẫu lỗi từ bộ train4

Data/nhãn	0	1	2
Train1	308	876	51
Train2	400	978	58
Train3	1327	3536	228
Train4	680	2113	958
Train5	648	1744	741

B. Base Model:

Chúng tôi sử dụng pretrained model cho bài toán này vì các pretrained model được train dựa trên những bộ dữ liệu vô cùng lớn như COCO, ImageNet. Các mô hình pretrained chúng tôi sử dụng gồm R101-FPN và X101-FPN, các mô hình này dựa trên kiến trúc Faster R-CNN và có chỉ số Average Precision (AP) là 42 và 43. Sau một vài thử nghiệm đầu tiên và dựa trên thông số mô hình, cho dù X101-FPN có thông số AP tốn hơn trên bài thử nghiệm ImageNet, tuy nhiên R101-FPN có thời gian dự đoán chỉ bằng một nửa nên chúng tôi quyết định sử dụng mô hình pretrained này làm base model chính cho bài toán của chúng tôi.

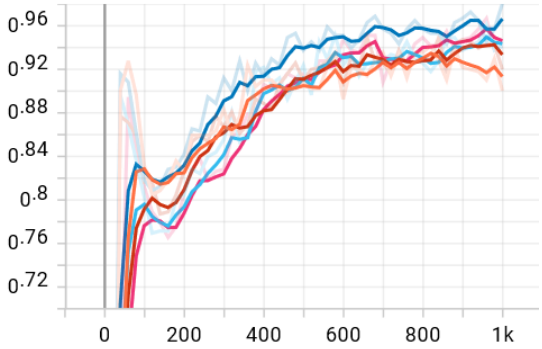
C. Experiment:

Ở những thử nghiệm đầu tiên, chúng tôi đã thay đổi lại một số hyperpaprameters của mô hình để phù hợp với tài nguyên có sẵn. Chúng tôi thiếp lập 'cfg.SOLVER.IMS-PER-BATCH' bằng 4 (images per batch), 'cfg.SOLVER.BASE-LR' bằng 0.001 (learning rate), 'cfg.MODEL.ROI HEADS.NUM CLASSES' bằng 3 (number of classes), 'cfg.SOLVER.MAX-ITER' bằng 1000 (numbers of iteration). Trong bài toán này, chúng tôi lấy AP50 (average precision > 50) làm độ đo chính để đánh giá, kết quả thu được trên tập public test cung cấp bởi DatacompFPT như sau:

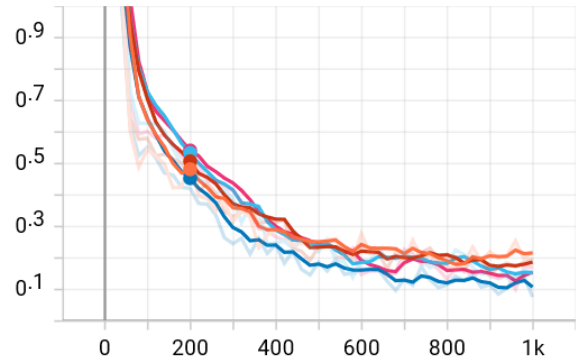
Data	AP	AP50	AP75	APs	APm
Train1	19.438	45.010	9.208	12.086	22.018
Train2	22.067	57.427	6.736	40.400	22.708
Train3	18.080	46.453	6.632	24.835	19.849
Train4	17.761	51.589	5.261	23.758	18.759
Train5	20.411	54.299	5.916	36.647	20.830

Biểu đồ train accuracy và train loss (train1:orange, train2:dark blue, train3:red, train4:blue, train5:pink)

- Train accuracy:



- Train loss:

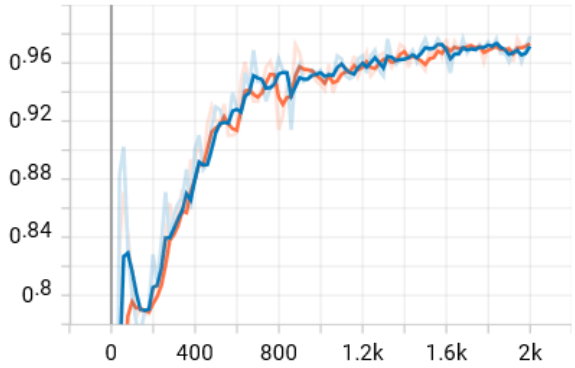


Dựa vào những chỉ số trong những lần thử nghiệm đầu, hầu hết các bộ dữ liệu đã bắt đầu có dấu hiệu hội tụ ở iter 800,

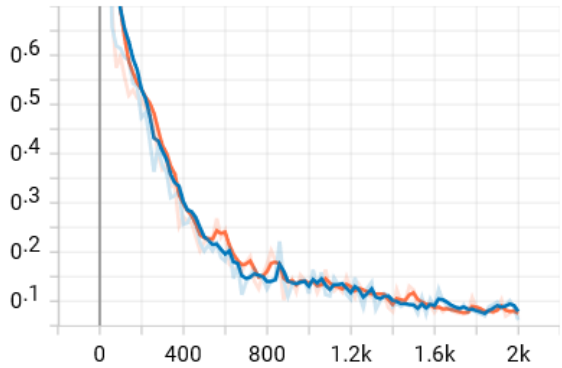
tuy nhiên các chỉ số cho thấy bộ train2 đang thể hiện tốt nhất. Bộ dữ liệu train2 cho AP50 cao nhất với 57, trong khi những bộ dữ liệu được sinh thêm là train3/4/5 lại không đạt được như kì vọng, ngay cả khi đã loại bỏ các mẫu sai và cố gắng tăng cường dữ liệu cho class 0 và 2 ở bộ train5. Tuy nhiên dựa trên lượng dữ liệu lớn hơn, chúng tôi quyết định lấy bộ train2 và train5 (tốt nhất trong các bộ được sinh) để học lại với 'cfg.SOLVER.MAX-ITER' bằng 2000 với dự đoán rằng mô hình chưa hội tụ do học chưa đủ lâu. Kết quả ('cfg.SOLVER.MAX-ITER'=2000):

Data	AP	AP50	AP75	APs	APm
$Train2_1$	25.126	59.467	12.856	36.156	26.128
$Train5_1$	24.078	62.943	11.505	41.270	23.993

-Train accuracy:

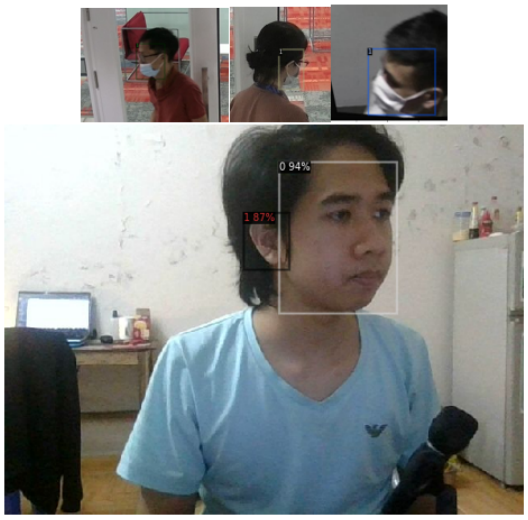


-Train loss:



Dựa theo kết quả trên, khi train tới iter thứ 2000 thì accuracy và loss của 2 tập không còn quá nhiều khác biệt, tuy nhiên AP50 của tập train5 cao hơn hẳn với 62.9, nhưng AP lại thấp hơn với 24. Dựa vào lượng phân bố dữ liệu của tập train5, chúng tôi vẫn chưa giải quyết được vấn đề mất cân bằng dữ liệu, có lẽ đây là nguyên nhân chính dẫn đến việc AP bị giảm đi.

Trong quá trình thử nghiệm ở môi trường thực tế, chúng tôi cũng nhận thấy những vấn đề khi mô hình đánh nhãn không tốt, cụ thể là gần như tất cả các bộ có thể sẽ nhận tai người làm một đối tượng đeo mặt nạ hoặc không, điều này có thể là do ở các bộ này, chúng tôi đánh nhãn người đeo khẩu trang với những bouding box bao trùm cả phần mang tai. Hoặc có trường hợp người đeo tai nghe cũng bị nhận diện là có đeo khẩu trang, có thể do mô hình nhầm lẫn dây tai nghe màu trắng với dây khẩu trang, và trong dữ liệu có rất nhiều mẫu chỉ chụp cạnh mặt và chỉ thấy phần dây.



D. Kết quả đánh giá trên DatacompFPT

$$score = wAP@50 = 0.2*AP50_w + 0.3*AP50_n w + 0.5*AP50_w i$$

Kết quả submit với bộ dữ liệu train2, trong khi bộ dữ liệu gốc của cuộc thi là 0.32

#	Tên nhóm	Tài khoản đội trưởng	Score	Tên file	Thời gian gửi
604	cs17	trongthanht3	0.477	dataset_r.zip	15/11/2021
647	cs17	trongthanht3	0.471	dataset_relabel.zip	15/11/2021

- Kết quả evaluate trên tập train5 với wAP.5 là 49.2, tốt hơn so với bộ dữ liệu train2

Class	Images	Labels	Boxes	P	R	wAP@.5	mAP@.5	mAP@.5:.95: 100% 2
all	88	147	1645	0.536	0.644	0.492	0.558	0.202
no_mask	42	42	644	0.544	0.796	0.648	0.648	0.212
mask	91	91	572	0.732	0.78	0.718	0.718	0.226
incorrect_mask	14	14	429	0.332	0.357	0.308	0.308	0.169

4. CONCLUSION

Công việc này khám phá khả năng phát hiện đối tượng hiện đại khác nhau các phương pháp và khả năng ứng dụng của chúng để phát hiện khẩu trang và phân loại cách đeo đúng hay sai. Cụ thể, chúng tôi thử nghiệm triển khai Detectron2's Faster R-CNN với các cơ sở khác nhau mô hình Tập dữ liệu DataComp.io.Bộ dữ liệu cần label lại và cải thiện để cân bằng với model của Detectron2 và Faster R-CNN.Mã nguồn của các thử nghiệm có sẵn tại trang Github của dự án này:

Trong những bài toán xử lý ảnh thì mô hình tốt là chưa đủ, bước quan trọng nhất trong giải quyết một bài toán AI đó là xử lí dữ liệu, chúng ta cần nguồn dữ liệu "sạch" và được đánh nhãn tốt, bên cạnh đó, việc sinh dữ liệu và các kĩ thuật sinh cũng là một bước cần thiết nhằm tăng dữ liệu cho mô hình, và việc sinh dữ liệu đó cũng nên được xác nhận một cách cẩn

thận. Qua đó chúng ta mới có thể có một mô hình đủ tốt để giải quyết bài toán.

5. REFERENCES

(1) Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun ' Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks ' https://arxiv.org/abs/1506.01497
(2) Detectron2 'Facebook Ai': https://ai.facebook.com/tools/detectron2/
(3) Van Vung Pham' road damage detection with Detectron2': https://arxiv.org/abs/2010.15021
(3) Hirotos ' Dig into Detectron2' : https://medium.com/@hirotoschwert