

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỌC MÁY TÍNH



ĐỒ ÁN CUỐI KỲ
THỊ GIÁC MÁY TÍNH TRONG TƯƠNG TÁC NGƯỜI - MÁY

NHẬN DIỆN MANG KHẨU TRANG THÔNG
QUA CAMERA GIÁM SÁT

GVHD: ThS. Đỗ Văn Tiến

Thành viên:

- Trương Chí Diễm – 19520464
- Trần Hoàn Đức Duy – 19521434
- Nguyễn Anh Dũng – 19521394
- Trịnh Công Danh – 19521326
- Võ Phạm Duy Đức – 19521386

Tp. Hồ Chí Minh, 10 tháng 6 năm 2022

Nội dung

I.	Giới thiệu chung	3
II.	Khó khăn.....	3
III.	Mục tiêu và phạm vi	3
IV.	Phương pháp	4
1.	Mô hình phát hiện việc mang khẩu trang	4
2.	Bộ dữ liệu huấn luyện mô hình	4
3.	Đặt quy cách gắn nhãn dữ liệu	6
4.	Framework để triển khai mô hình lên web.....	7
V.	Các chức năng chính.....	9
1.	Nhận diện đeo khẩu trang thời gian thực	9
2.	Xem danh sách thành viên trong công ty	9
3.	Thêm thành viên vào danh sách	10
4.	Xem lịch sử vi phạm của nhân viên trong công ty.....	11
VI.	Thực nghiệm và đánh giá	12
1.	Phương pháp đánh giá mô hình.....	12
2.	Huấn luyện mô hình phát hiện đeo khẩu trang	12
3.	Đánh giá kết quả triển khai.....	14
VII.	Kết luận	14
VIII.	Tài liệu tham khảo	15
IX.	Bảng phân công	15

I. Giới thiệu chung

Thời kỳ covid gần như đã trôi qua, tuy nhiên xã hội vẫn đang duy trì các thói quen trong thời dịch để ngăn ngừa những hiểm họa có thể xảy ra một lần nữa. Trong đó, việc đeo khẩu trang đi đến nơi đông người và nơi làm việc vẫn đang được các công ty yêu cầu nhân viên chấp hành nghiêm túc. Do vậy, trong đề tài đồ án môn học này chúng tôi muốn xây dựng một ứng dụng web có khả năng phát hiện và ghi nhận các trường hợp không đeo khẩu trang phù hợp sử dụng trong môi trường công ty.

Đầu vào của ứng dụng là hình ảnh từ các camera giám sát tại công ty, ứng dụng sẽ xử lý để ghi nhận các trường hợp là nhân viên trong công ty vi phạm không mang khẩu trang kèm theo ảnh minh chứng.

Input



Output



II. Khó khăn

Một trong những khó khăn của đề tài này là tìm kiếm và xử lý để có được một bộ dữ liệu tốt, giúp cho hiệu năng của mô hình được nâng cao. Các vấn đề gặp phải về dữ liệu bao gồm việc thiếu dữ liệu và mất cân bằng dữ liệu.

Một khó khăn khác là việc ứng dụng mô hình học sâu vào ứng dụng web còn một số hạn chế và bất lợi.

III. Mục tiêu và phạm vi

Mục tiêu chung của bài báo cáo là xây dựng được một ứng dụng web có khả năng phát hiện và ghi nhận các trường hợp không đeo khẩu trang phù hợp sử dụng trong môi trường công ty với độ chính xác cao và tốc độ xử lý nhanh chóng.

Trong nội dung của bài báo cáo này, chúng tôi giới hạn phạm vi của ứng dụng về các nội dung sau: số lượng camera sử dụng trong ứng dụng là 1 camera,

tốc độ xử lý là 20 FPS, độ chính xác đạt trên 95% trong xử dụng thực tế, việc triển khai trong nội dung bài báo cáo vẫn đang nằm trên máy local nên tất cả kết quả đều do 1 người dùng thực hiện request đến server.

IV. Phương pháp

Chúng tôi thực hiện xây dựng ứng dụng trên nền tảng web, huấn luyện một mô hình học sâu có khả năng phát hiện các gương mặt không mang khẩu trang, một mô hình trích xuất đặc trưng để nhận diện cá nhân vi phạm nằm trong danh sách thành viên của công ty. Vì các mô hình được xây dựng dựa trên ngôn ngữ lập trình Python nên để thuận tiện triển khai lên thành ứng dụng web chúng tôi sử dụng framework FastAPI, kết hợp với các template HTML để thiết kế giao diện và xây dựng các hàm chức năng.

1. Mô hình phát hiện việc mang khẩu trang

Trong phạm vi đề tài này, chúng tôi muốn thu được một ứng dụng có khả năng triển khai thực tế nên điều quan trọng là tốc độ xử lý phải nhanh để có thể thoả mãn yêu cầu tiên quyết là ứng dụng đáp ứng được việc xử lý trong thời gian thực. Việc lựa chọn mô hình phát hiện gương mặt không mang khẩu trang là ưu tiên hàng đầu vì nó là linh hồn của ứng dụng, tất cả mọi chức năng của ứng dụng đều xoay quanh mô hình hoặc kết quả của mô hình đưa ra.

Chúng tôi xem xét các mô hình nổi tiếng và phổ biến trong bài toán Object Detection như RetinaNet, Faster-RCNN và YOLOv5 để nghiên cứu và thực nghiệm nhằm chọn ra mô hình phù hợp nhất với dụng. Dựa trên các tiêu chí như tốc độ và độ chính xác, sau khi thực nghiệm và đánh giá thì chúng tôi chọn YOLOv5 là mô hình đảm nhiệm vai trò này.

2. Bộ dữ liệu huấn luyện mô hình

Dữ liệu cũng là một yếu tố then chốt giúp mô hình có được hiệu năng cao. Một bộ dữ liệu tốt sẽ giảm được chi phí thời gian để thu thập và xử lý, thời gian huấn luyện cũng sẽ được giảm đi nhiều trong khi kết quả lại tăng lên.

Chúng tôi ban đầu sử dụng bộ dữ liệu từ cuộc thi FPT Datacomp chỉ bao gồm 1064 ảnh được trích xuất từ camera giám sát của một công ty. Số ảnh này quá ít để huấn luyện một mô hình học sâu với hàng chục triệu trọng số.

Do đó, chúng tôi quyết định thu thập thêm dữ liệu cho bài toán. Cụ thể, chúng tôi tham khảo các bài toán có liên quan đến nhận diện đeo khẩu trang hiện có trên Google nhưng đa phần, các tập dữ liệu trên đều có ngữ cảnh không phù hợp với bài toán của chúng tôi do có góc camera quá khác biệt.



Hình 1. Data từ một cuộc thi trên Kaggle về nhận diện đeo khẩu trang. Nhưng góc nhìn chính diện trong khi bài toán chúng tôi đang giải quyết là góc nhìn từ camera giám sát.

Sau đó chúng tôi tham khảo các bài báo trên Scholar thì tìm được một số nguồn dữ liệu liên quan đến camera giám sát như [WILDTRACK](#), [BrainWash](#), [Oxford Town Center](#). Và trang web [EarthCam](#) chuyên chia sẻ dữ liệu các camera giám sát được công khai tại khắp nơi trên thế giới. Sau khi duyệt qua các một số camera trên EarthCam, chúng tôi chỉ chọn được 1 camera có góc nhìn tương tự như bài toán và có thể sử dụng được.



Hình 2a. Tập dữ liệu BrainWash



Hình 2b. Tập dữ liệu WILDTRACK



Hình 2c. Ảnh từ quán [café Miami](#) trên trang EarthCam

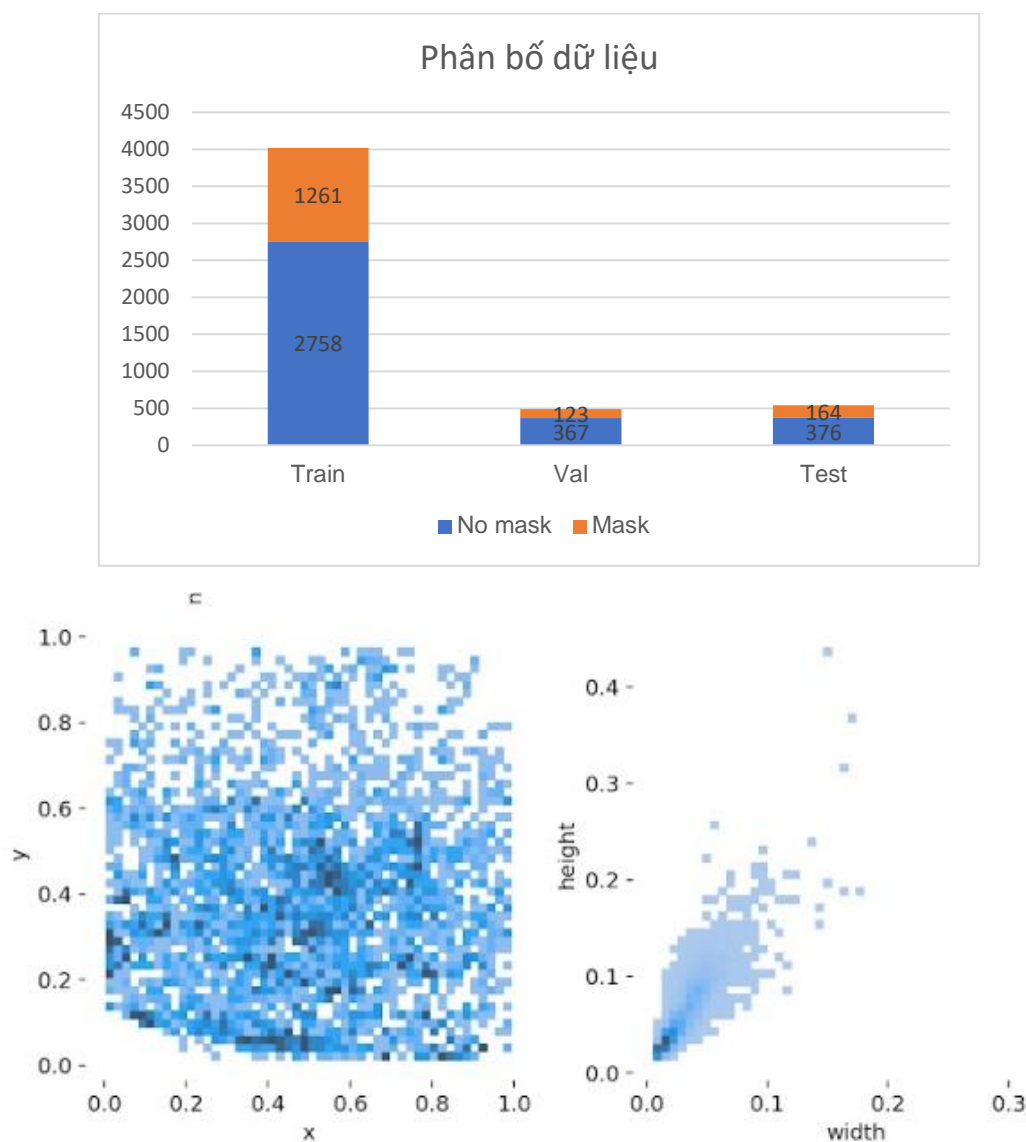


Hình 2d. Ảnh từ bộ dữ liệu Oxford Town Center

Sau khi quá trình tách khung hình từ video và chọn lọc thì chúng tôi thu được thêm 1152 ảnh (WILDTRACK: 572, BrainWash: 219, café Miami: 265, Oxford

Town Center: 96), nâng tổng số ảnh trong tập dữ liệu lên thành 2,216 ảnh. Với số lượng này, chúng tôi tự tin rằng mô hình có thể học được trên nhiều ngữ cảnh hơn so với số lượng dữ liệu ít ỏi ban đầu.

Bộ dữ liệu bao gồm 2.216 ảnh, được chia theo tỉ lệ train:val:test là 8:1:1. Phân bố dữ liệu như biểu đồ bên dưới.



Hình 5. Trái: phân bố của đối tượng trong khung hình. Phải: Phân bố tỉ lệ bounding box

3. Đặt quy cách gắn nhãn dữ liệu

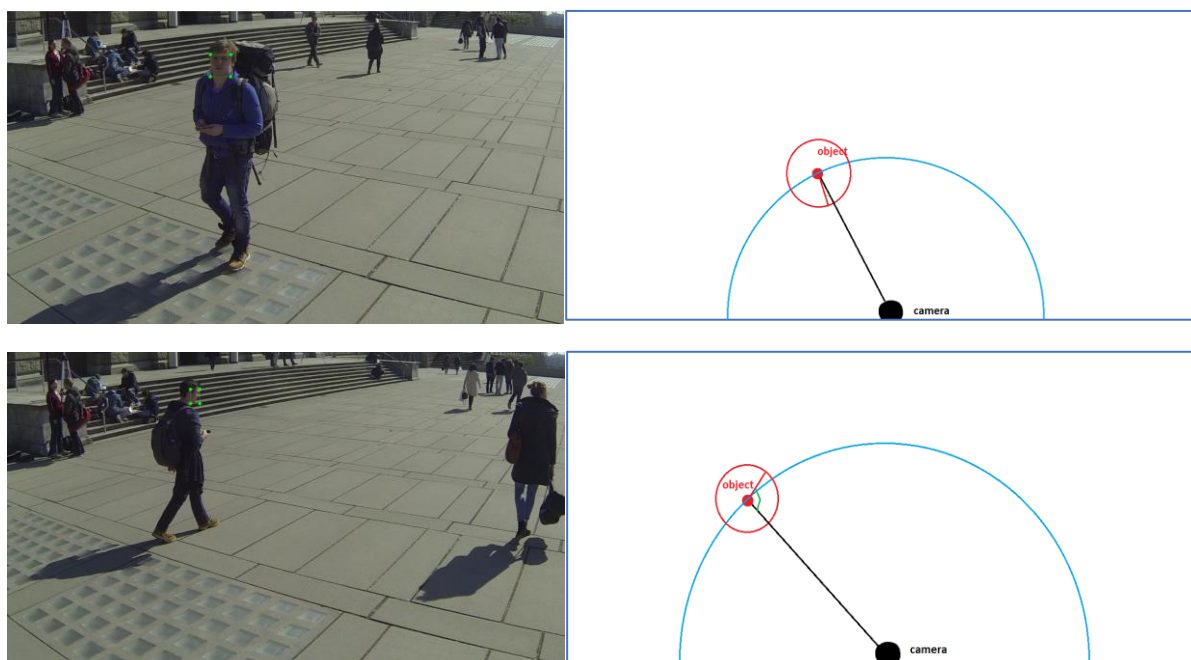
Việc gắn nhãn cho dữ liệu rất quan trọng. Nếu dữ liệu được gắn nhãn tốt thì chỉ với một lượng dữ liệu ít thì mô hình cũng có khả năng đạt được kết quả cao hơn so với một mô hình được huấn luyện trên một bộ dữ liệu lớn nhưng được

gắn nhãn tệp. Do đó chúng tôi đặt ra một quy cách gắn nhãn chung cho toàn bộ các thành viên trong nhóm tham gia gắn nhãn bằng công cụ labelling.

Quy tắc gắn nhãn của chúng tôi tuân theo bộ dữ liệu ban đầu, cụ thể là:

- Đối với các khuôn mặt chính diện hoặc có góc nhìn nghiêng trong khoảng $[-90^\circ; 90^\circ]$, thấy rõ mặt thì sẽ được gắn nhãn từ trán xuống dưới phần cằm, bỏ phần tai.
- Các khuôn mặt có góc nghiêng lớn hơn 90° nhưng vẫn thấy rõ được có đeo khẩu trang hay không thì vẫn được gắn nhãn từ trán xuống dưới cằm và lấy luôn phần tai.
- Đối với các khuôn mặt bị khuất một phần nhưng vẫn nhận diện được bởi mắt người thì sẽ gắn nhãn phần không bị khuất.
- Các khuôn mặt bị khuất quá nhiều, góc quay quá nhiều không thể nhận diện được sẽ bị bỏ qua, không gắn nhãn.
- Những khuôn mặt nhỏ và quá mờ cũng sẽ bị bỏ qua.

Góc nhìn là góc được tạo bởi đoạn thẳng màu đen và đoạn thẳng màu đỏ. Trong đó, đoạn thẳng màu đen chỉ hướng từ người đến camera và đoạn thẳng màu đỏ chỉ hướng mặt người đang nhìn trên hình chiếu bằng.



Hình 3. Mô phỏng cách xác định góc nhìn của mặt

4. Framework để triển khai mô hình lên web

Để triển khai một mô hình học sâu lên web, có nhiều framework hỗ trợ dành cho ngôn ngữ Python như Flask, FastAPI, Django,... Tuy nhiên chúng tôi chọn

FastAPI vì nó dễ sử dụng, nhanh chóng và tiện lợi. Ngoài ra, FastAPI còn cung cấp một Documentation trực tuyến cho phép kiểm tra các API được viết ra có chạy tốt hay không.



Hình 4. Giao diện Documentation của FastAPI

Các API được thiết kế trong FastAPI bao gồm:

- GET('/'): trả về giao diện chính của ứng dụng.
- GET('/listpeople'): trả về giao diện của trang xem danh sách thành viên.
- GET('/history'): trả về giao diện của trang xem danh sách thành viên vi phạm.
- GET('/video_feed'): stream dữ liệu từ output của mô hình lên giao diện chính.
- GET('/get_members'): trả về dữ liệu trong danh sách thành viên.
- GET('/get_history'): trả về dữ liệu trong danh sách vi phạm.
- GET('/add_people'): trả về giao diện của trang thêm thành viên.
- GET('/add_member_feed'): stream dữ liệu từ output của mô hình lên trang thêm thành viên.
- POST('/upload_member_info'): tải lên server thông tin của thành viên được thêm.

Ngoài ra, để thiết kế giao diện cho ứng dụng chúng tôi sử dụng ngôn ngữ đánh dấu siêu văn bản HTML. Nó cho phép chúng tôi fetch API từ FastAPI lên để hiển thị trên giao diện web.

V. Các chức năng chính

Trong phạm vi của môn học này, chúng tôi chủ yếu muốn áp dụng mô hình thị giác máy tính vào trong một ứng dụng thực tế nên việc thiếu kinh nghiệm và thời gian khiến ứng dụng này bị hạn chế các chức năng ở một số lượng nhất định. Các chức năng chính của ứng dụng và cách sử dụng được mô tả dưới đây.

1. Nhận diện đeo khẩu trang thời gian thực

Đây là chức năng chính của ứng dụng, là trang mặc định sau khi ứng dụng được mở lên.

Tính năng này cho phép lấy hình ảnh từ camera giám sát, đưa vào mô hình nhận diện, thực hiện kiểm tra và so sánh và lưu lại các trường hợp nhân viên của công ty không mang khẩu trang và stream hình ảnh từ output của mô hình lên màn hình chính.

Trang của chức năng được fetch từ API GET('/') và hình ảnh được stream từ API GET('/video_feed'). Để vào chức năng này khi đang ở các trang chức năng khác thì click vào biểu tượng FaceMask hoặc nút Trang chủ như hình bên dưới.



2. Xem danh sách thành viên trong công ty

Chức năng này cho phép người dùng truy cập vào danh sách thành viên của công ty để xem thông tin về Họ tên, Năm sinh và Giới tính của thành viên. Ngoài ra, tại giao diện của trang còn có nút chức năng thêm thành viên vào danh sách để chuyển hướng đến trang chức năng thêm thành viên.

Trang của chức năng này được fetch từ API GET('/listpeople') và dữ liệu thành viên được fetch từ API GET('/get_members'). Để vào trang chức năng

này khi đang ở các trang chức năng khác thì click vào nút chức năng Danh sách thành viên như hình bên dưới.

FACE MASK

Bấm để vào trang danh sách thành viên

Trang chủ

Danh sách thành viên

Lịch sử vi phạm

DANH SÁCH THÀNH VIÊN		
Họ và tên	Ngày sinh	Giới tính
Truong Chi Dien	1/4/2001	Nam
Võ Phạm Duy Đức	1/4/2001	Nam
Trình Công Danh	15/8/2001	Nam
Do Trọng Khanh	1/5/2001	Nam

Nơi danh sách thành viên được hiển thị

+

3. Thêm thành viên vào danh sách

Chức năng này cho phép người dùng thêm thành viên mới vào danh sách thành viên hiện có. Thành viên mới phải cung cấp thông tin về Họ tên, Năm sinh, Giới tính và thông tin về gương mặt được camera ghi lại để phục vụ cho chức năng ghi thông tin nhân viên vi phạm.

Trang của chức năng này được fetch từ API GET('/add_people) và dữ liệu gương mặt được fetch từ API GET('/add_member_feed). Để vào trang chức năng này khi đang ở các trang chức năng khác thì click vào nút chức năng Danh sách thành viên sau đó click vào hình dấu cộng như hình bên dưới.

FACE MASK

Trang chủ

Danh sách thành viên

Lịch sử vi phạm

DANH SÁCH THÀNH VIÊN		
Họ và tên	Ngày sinh	Giới tính
Truong Chi Dien	1/4/2001	Nam
Võ Phạm Duy Đức	1/4/2001	Nam
Trình Công Danh	15/8/2001	Nam
Do Trọng Khanh	1/5/2001	Nam

Bấm để thêm nhân viên mới

+

THÊM THÀNH VIÊN

Họ và tên

Năm sinh

Giới tính

Nam

Nơi nhập thông tin






Nơi output của mô hình được stream

4. Xem lịch sử vi phạm của nhân viên trong công ty

Chức năng này cho phép người dùng truy cập vào danh sách vi phạm của công ty để xem thông tin về Họ tên người vi phạm, Thời điểm vi phạm và hình ảnh minh chứng tại thời điểm vi phạm. Các vi phạm từ 1 thành viên chỉ được ghi nhận lại ít nhất là 10 phút kể từ lần vi phạm trước đó.

Trang của chức năng này được fetch từ API GET('/hisroty') và dữ liệu thành viên được fetch từ API GET('/get_history'). Để vào trang chức năng này khi đang ở các trang chức năng khác thì click vào nút chức năng Lịch sử vi phạm như hình bên dưới.

LỊCH SỬ VI PHẠM

Họ và tên	Thời điểm vi phạm	Ảnh minh chứng
Truong Chi Dien	2022/06/19 11:31	
Truong Chi Dien	2022/06/16 14:07	
Võ Phạm Duy Đức	2022/06/16 14:07	

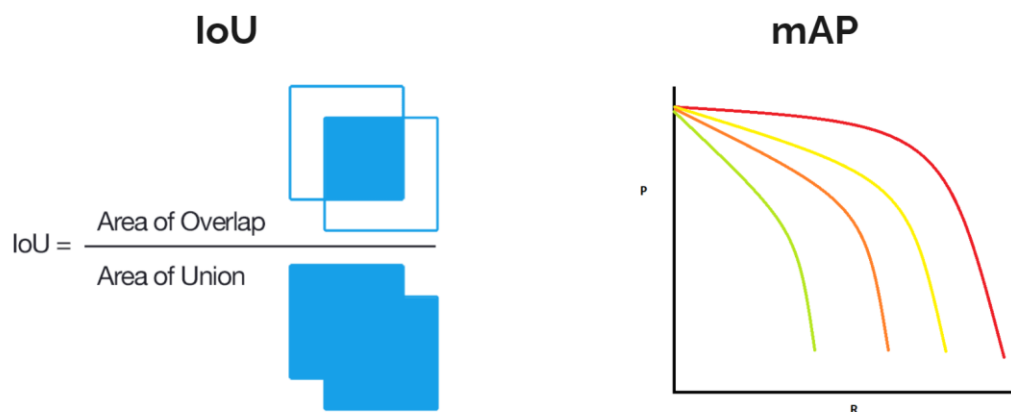
VI. Thực nghiệm và đánh giá

1. Phương pháp đánh giá mô hình

IoU là độ đo được tính bằng diện tích phần giao giữa bounding box dự đoán và bounding box thực tế chia cho phần hợp giữa chúng (được mô tả như hình bên dưới).

Precision curve là đường cong thể hiện độ đánh đổi giữa Precision và Recall khi thay đổi ngưỡng Confidence score.

Chúng tôi sử dụng điểm mAP để đánh giá bài toán này. Điểm mAP được tính toán bằng trung bình phần diện tích bên dưới Precision Curve. Tuy nhiên, với các ngưỡng IoU khác nhau sẽ có 1 Precision curve khác nhau tương ứng với điểm mAP khác nhau. Chúng tôi sử dụng điểm [mAP@0.5](#) và [mAP@0.5:0.95](#) và cả tốc độ nhận diện mỗi ảnh (FPS) để đánh giá mô hình.

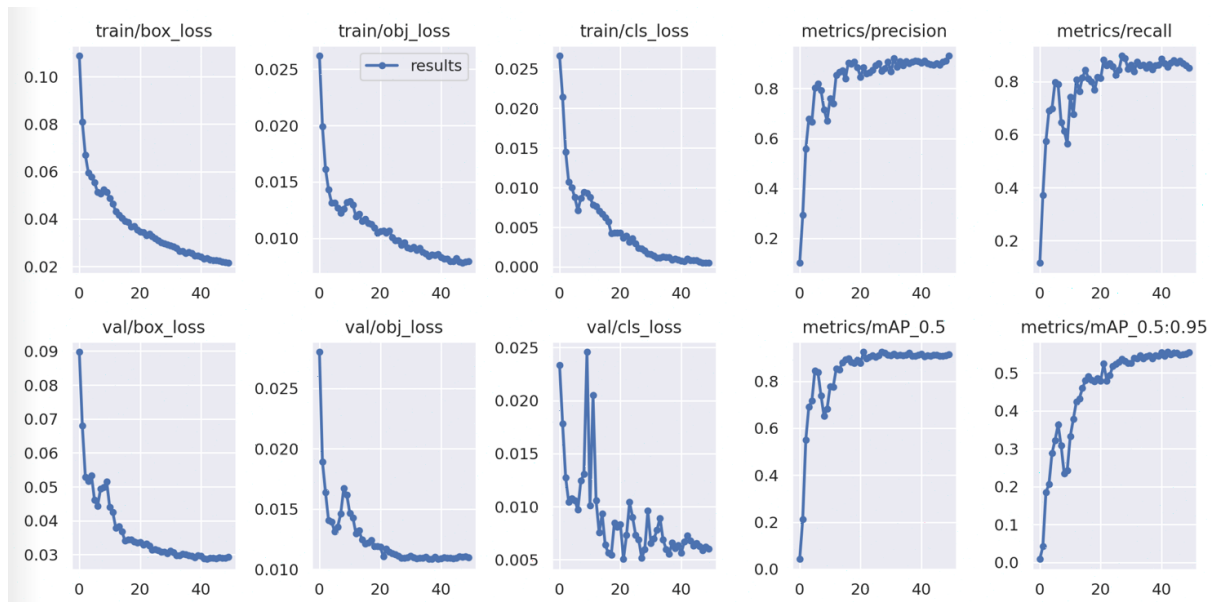


Hình 6. Mô tả độ đo IoU và mAP

2. Huấn luyện mô hình phát hiện đeo khẩu trang

Trong thực nghiệm này, các mô hình đã được pretrained trên bộ dữ liệu [COCO](#), sau đó được fine-tuning với bộ dữ liệu khẩu trang.

Train trên GPU: Tesla P100-PCIE-16GB.

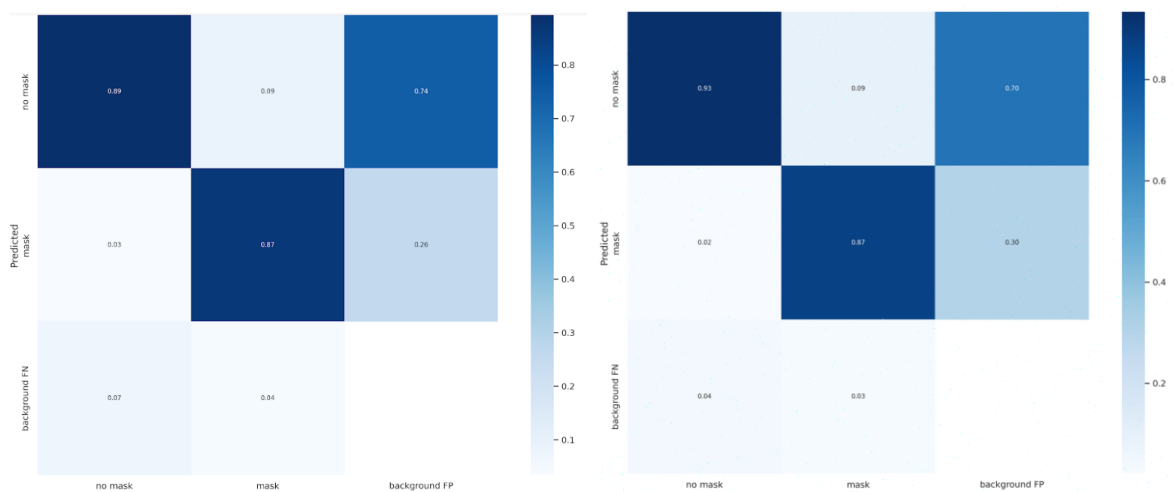


Hình 7: Các thông số giám sát quá trình huấn luyện.

Trong quá trình huấn luyện losses của tập train vẫn giảm nhưng losses và mAP trên bộ val không còn giảm nữa, nếu tiếp tục train sẽ bị overfit.

Phương pháp	mAP@0.5	mAP@0.5:0.95	Inference time (FPS)
RetinaNet	0.865	0.496	14.7
Faster R-CNN	0.888	0.509	13.7
YOLOv5	0.899	0.535	48.1
YOLOv5 (bỏ Oxford Town Center)	0.918	0.547	

Bảng 1 Kết quả trên các phương pháp đã thực hiện



Hình 8: Confusion matrix của YOLO trên bộ test. Trái: Train trên toàn tập dữ liệu đã thu thập. Phải: Loại bỏ bộ Oxford Town Center.

Từ kết quả thực nghiệm, ta thấy YOLOv5 hoàn toàn tốt hơn các mô hình còn lại trên tất cả các độ đo. Với điểm [mAP@0.5](#) $\approx 90\%$, đây là con số có thể chấp nhận được cho bài toán này nếu không quá khắt khe. Vì bộ dữ liệu Oxford Town Center có bối cảnh khác biệt khá lớn với các bộ dữ liệu còn lại, nên khi bỏ đi thì kết quả bài toán tăng nhẹ.

Phiên bản YOLOv5 được sử dụng trong huấn luyện và so sánh là YOLOv5x với hơn 80M tham số, khiến cho việc triển khai trong thực tế không được nhanh như kỳ vọng mặc dù độ chính xác rất tốt. Do đó, chúng tôi huấn luyện trên các phiên bản thấp hơn của YOLOv5 và quyết định chọn YOLOv5s với hơn 20M tham số để thực hiện triển khai. Việc hạ độ phức tạp của mô hình có làm giảm độ chính xác mô hình không đáng kể nhưng tăng được tốc độ xử lý lên nhiều lần. Để đảm bảo mô hình tối thiểu hoá những kết quả sai, chúng tôi tăng threshold của confident score thành 0.8 và mô hình cho ra kết quả chính xác mặc dù recall có phần bị giảm đi.

3. Đánh giá kết quả triển khai

Vì thời gian làm xong ứng dụng cũng là lúc kết thúc các lớp học lý thuyết trên trường, thêm nữa là việc di chuyển các thiết bị liên quan để thực hiện đánh giá thực tế là khó khăn nên chúng tôi vẫn chưa có được một đánh giá khách quan nhất về ứng dụng do chưa triển khai được trên thực tế với số lượng người tham gia lớn.

Tuy nhiên, thực nghiệm của chúng tôi trên 8 người tham gia cho thấy ứng dụng hoạt động tốt với FPS trung bình là 22, độ chính xác trong việc phân biệt các đối tượng là 93%, recall của các trường hợp cố tình gian lận hệ thống là 13%, recall của các trường hợp bình thường là 89%.

VII. Kết luận

Qua các đánh giá thực nghiệm cho thấy ứng dụng hoạt động tốt về mặt chức năng phát hiện và nhận diện các cá nhân vi phạm. Nhưng nhìn chung ứng dụng chưa đủ tốt để triển khai sử dụng thực tế do số lượng chức năng vẫn còn hạn chế và chưa áp dụng các nguyên tắc tiến hoá, bảo mật, an toàn,... trong phát triển phần mềm.

Các công việc trong tương lai chúng tôi có thể làm để cải thiện ứng dụng này bao gồm việc nâng cấp bộ dữ liệu, huấn luyện mô hình để phát hiện được các hành vi cố tình qua mặt hệ thống. Xây dựng lại toàn bộ ứng dụng theo một kiến trúc phần mềm để đảm bảo chất lượng của ứng dụng có thể sử dụng trong thực tế.

VIII. Tài liệu tham khảo

1. REDMON, Joseph, et al. You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. p. 779-788.
2. REDMON, Joseph; FARHADI, Ali. YOLO9000: better, faster, stronger. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017. p. 7263-7271.
3. REDMON, Joseph; FARHADI, Ali. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
4. BOCHKOVSKIY, Alexey; WANG, Chien-Yao; LIAO, Hong-Yuan Mark. YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
5. [YOLOv4. While object detection matures in the... | by Jonathan Hui | Medium](#)
6. [YOLO v4 or YOLO v5 or PP-YOLO? Which should I use? | Towards Data Science](#)
7. [Object Detection in 2022: The Definitive Guide - viso.ai](#)
8. [PASCAL VOC 2007 Benchmark \(Object Detection\) | Papers With Code](#)
9. [FastAPI \(tiangolo.com\)](#)
10. [HTML Tutorial \(w3schools.com\)](#)

IX. Bảng phân công

Tên	MSSV	Công việc	Mức độ hoàn thành
Trương Chí Diễm ©	19520464	Phân công, giám sát tiến độ, đề xuất ý tưởng, thu thập dữ liệu, thiết kế chức năng, viết báo cáo, thuyết trình.	100%
Trịnh Công Danh	19521326	Thiết kế giao diện, đề xuất ý tưởng, gắn nhãn dữ liệu, tìm hiểu nội dung liên quan, viết báo cáo.	100%
Võ Phạm Duy Đức	19521383	Thiết kế giao diện, đề xuất ý tưởng, gắn nhãn dữ liệu, tìm hiểu nội dung liên quan, viết báo cáo.	100%
Trần Hoàn Đức Duy	19521434	Huấn luyện mô hình, so sánh kết quả, thu thập dữ liệu, phân tích dữ liệu, xử lý dữ liệu, viết báo cáo.	100%
Nguyễn Anh Dũng	19521394	Huấn luyện mô hình, so sánh kết quả, gắn nhãn dữ liệu, thiết kế chức năng, viết báo cáo.	100%