

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

**NGUYỄN HỮU THẮNG
GIẢNG PHÚC VINH**

KHOÁ LUẬN TỐT NGHIỆP
HỆ THỐNG TỰ ĐỘNG KIỂM TRA CHẤP
HÀNH BẢO HỘ PHẦN MẶT

CỬ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

TP.HỒ CHÍ MINH, 2020

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

NGUYỄN HỮU THẮNG – 16521102
GIANG PHÚC VINH – 16521548

KHOÁ LUẬN TỐT NGHIỆP
HỆ THỐNG TỰ ĐỘNG KIỂM TRA CHẤP
HÀNH BẢO HỘ PHẦN MẶT

CỬ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

GIẢNG VIÊN HƯỚNG DẪN
TS. MAI TIỀN DŨNG

TP.HỒ CHÍ MINH, 2020

DANH SÁCH HỘI ĐỘNG BẢO VỆ KHÓA LUẬN

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số
ngàycủa Hiệu trưởng Trường Đại học Công nghệ thông tin.

1. - Chủ tịch.
2. - Thư ký.
3. - Ủy viên.
4. - Ủy viên.

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc Lập - Tự Do - Hạnh Phúc

TP. HCM, ngày.....tháng.....năm.....

**NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP
(CỦA CÁN BỘ HƯỚNG DẪN)**

Tên khóa luận:

HỆ THỐNG TỰ ĐỘNG KIỂM TRA CHẤP HÀNH BẢO HỘ PHẦN MẶT

Nhóm SV thực hiện:

Nguyễn Hữu Thắng 16521102
Giảng Phúc Vinh 16521548

Cán bộ hướng dẫn:

TS. Mai Tiến Dũng

Đánh giá Khóa luận

1. Về cuốn báo cáo:

Số trang	_____	Số chương	_____
Số bảng số liệu	_____	Số hình vẽ	_____
Số tài liệu tham khảo	_____	Sản phẩm	_____

Một số nhận xét về hình thức cuốn báo cáo:

2. Về nội dung nghiên cứu:

3. Về chương trình ứng dụng:

.....
.....
.....

4. Về thái độ làm việc của sinh viên:

.....
.....
.....

Đánh giá chung:

.....
.....
.....

Điểm từng sinh viên:

Nguyễn Hữu Thắng:/10

Giảng Phúc Vinh:/10

Người nhận xét

(Ký tên và ghi rõ họ tên)

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Độc Lập - Tự Do - Hạnh Phúc

TP. HCM, ngày.....tháng.....năm.....

NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP
(CỦA CÁN BỘ PHẢN BIỆN)

Tên khóa luận:

HỆ THỐNG TỰ ĐỘNG KIỂM TRA CHẤP HÀNH BẢO HỘ PHẦN MẶT

Nhóm SV thực hiện:

Nguyễn Hữu Thắng 16521102

Giảng Phúc Vinh 16521548

Cán bộ phản biện:

Số trang _____

Số chương _____

Số bảng số liệu _____

Số hình vẽ _____

Số tài liệu tham khảo _____

Sản phẩm _____

Một số nhận xét về hình thức cuốn báo cáo:

.....
.....
.....

6. Về nội dung nghiên cứu:

.....
.....

.....
.....
.....
.....
.....
.....

7. Về chương trình ứng dụng:

.....
.....
.....
.....
.....

8. Về thái độ làm việc của sinh viên:

Đánh giá chung:

Điểm từng sinh viên:

Nguyễn Hữu Thắng:/10

Giảng Phúc Vinh:/10

Người nhận xét

(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Đầu tiên, chúng em xin chân thành cảm ơn tiến sĩ Mai Tiến Dũng, phó trưởng khoa khoa Khoa học máy tính, đã đồng ý làm giảng viên hướng dẫn khóa luận cho chúng em. Chúng em xin cảm ơn những kinh nghiệm, kiến thức quý báu mà thầy đã truyền đạt cho chúng em để chúng em có thể hoàn thành khóa luận tốt nghiệp. Cảm ơn thầy đã luôn theo sát và hướng dẫn chúng em tận tình suốt quá trình thực hiện khóa luận.

Chúng em cũng xin chân thành gửi lời cảm ơn đến tiến sĩ Ngô Đức Thành, trưởng khoa Khoa học máy tính. Cảm ơn thầy đã đồng hành cùng chúng em từ những bước đầu khi bắt đầu thực hiện khóa luận. Cảm ơn những lời góp ý, hướng dẫn tận tình của thầy để chúng em có thể đạt được kết quả như hôm nay.

Chúng em xin cảm ơn các thầy cô trong khoa Khoa học máy tính, cũng như toàn thể các thầy cô trong trường đã từng giảng dạy cho chúng em từ ngày bước chân vào trường Đại học Công nghệ thông tin. Cảm ơn những kiến thức mà thầy cô đã truyền đạt để chúng em có được những kiến thức làm nền tảng để chúng em thực hiện tốt khóa luận này.

Bên cạnh đó, chúng em cũng xin cảm ơn những anh chị trong MMLab UIT đã hỗ trợ, góp ý cũng như cung cấp cơ sở vật chất trong quá trình thực hiện khóa luận.

Cuối cùng, chúng em xin cảm ơn cha mẹ, bạn bè đã luôn hỗ trợ động viên tinh thần cho chúng em trong bốn năm học tập tại trường Đại học Công nghệ thông tin.

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc Lập - Tự Do - Hạnh Phúc

ĐĂNG KÝ ĐỀ TÀI KHÓA LUẬN TỐT NGHIỆP

TÊN ĐỀ TÀI: Hệ thống Tự động Kiểm tra Chấp hành Bảo hộ phần Mặt.

TÊN ĐỀ TÀI TIẾNG ANH: Automatic Facial Protection Regulation Inspection Application.

Cán bộ hướng dẫn: TS. Mai Tiến Dũng.

Thời gian thực hiện: Từ ngày 10/02/2020 đến ngày 01/07/2020.

Sinh viên thực hiện:

Nguyễn Hữu Thắng - 16521102

Lớp: KHTN2016

Email: 16521102@gm.uit.edu.vn

Điện thoại: 0909512540

Giảng Phúc Vinh - 16521548

Lớp: KHTN2016

Email: 16521548@gm.uit.edu.vn

Điện thoại: 0797156414

Nội dung đề tài: Tự động hóa lao động là một trong những xu hướng mọi doanh nghiệp hướng tới trong thời kì cách mạng công nghiệp 4.0. Lĩnh vực an ninh cũng cần được tự động hóa để mang đến sự công tâm, giảm nhân lực cũng như tăng hiệu năng công việc. Hơn thế nữa, trong hoàn cảnh đại dịch COVID-19 như hiện nay, việc đeo khẩu trang là một biện pháp phòng tránh, giảm yếu tố lây lan quan trọng. Vậy nên, cần có sự giám sát và cảnh báo kịp thời đối với những trường hợp không chấp hành quy định đeo khẩu trang đúng theo yêu cầu. Dựa vào những yếu tố trên, nhóm em sẽ nghiên cứu và đưa ra ứng dụng hỗ trợ giám sát và kiểm tra việc chấp hành đeo khẩu trang.

Với mong muốn có được ứng dụng với độ chính xác và tốc độ đáp ứng được nhu cầu thực tiễn, nhóm sẽ khảo sát và đánh giá một số phương pháp hiện có. Từ đó chọn được phương pháp phù hợp và có gắng cải thiện để đưa vào ứng dụng thực tế.

Kế hoạch thực hiện:**10/02/2020 – 10/03/2020:**

Nguyễn Hữu Thắng + Giảng Phúc Vinh: Khảo sát các bài báo, công nghệ và kĩ thuật hiện tại cho bài toán đang hướng tới.

10/03/2020 – 10/04/2020:

Nguyễn Hữu Thắng + Giảng Phúc Vinh: Chạy lại và đánh giá các phương pháp đã tìm hiểu ở giai đoạn trước.

10/04/2020 – 10/05/2020:

Nguyễn Hữu Thắng: xử lý bộ dữ liệu MAFA để tạo ra bộ dữ liệu riêng.

Giảng Phúc Vinh: xử lý bộ dữ liệu WIDERFACE để tạo ra bộ dữ liệu riêng.

Nguyễn Hữu Thắng + Giảng Phúc Vinh: đánh giá một số mô hình trên bộ dữ liệu riêng đã xây dựng để lựa chọn mô hình phù hợp nhất.

10/05/2020 – 30/06/2020:

Nguyễn Hữu Thắng + Giảng Phúc Vinh: Thử một số phương pháp để cải thiện kết quả, viết báo cáo, làm slide.

Xác nhận của CBHD (Ký tên và ghi rõ họ tên)	TP. HCM, ngày 19 tháng 5 năm 2020 Sinh viên (Ký tên và ghi rõ họ tên)
---	---

Mục lục

TÓM TẮT KHOÁ LUẬN

xvii

1 TỔNG QUAN	1
1.1 Giới thiệu bài toán	1
1.2 Tính ứng dụng	2
1.3 Những thách thức và khó khăn gặp phải	2
1.3.1 Cách thức đánh giá mô hình	2
1.3.2 Tốc độ và độ chính xác của mô hình	3
1.3.3 Dữ liệu	3
1.4 Mục tiêu khóa luận	3
1.5 Cấu trúc Khóa luận tốt nghiệp	4
2 CƠ SỞ LÝ THUYẾT	5
2.1 Các mô hình mạng máy học	5
2.1.1 Mạng nơ-ron nhân tạo (Artificial Neural Network - ANN)	5
2.1.2 Mạng nơ-ron tích chập (Convolutional Neural Network - CNN)	7
Lớp Convolution	7
Lớp Pooling	8
Lớp Fully Connected	10
2.2 Đánh giá mô hình	11
2.2.1 Mức độ tự tin - Confidence Score	11
2.2.2 Intersection over Union - IoU	12
2.2.3 Độ chính xác - precision và độ phủ - recall	13
2.2.4 mean Average Precision - mAP	13
2.3 Các nghiên cứu liên quan	16
2.3.1 Detecting Masked Faces in the Wild with LLE-CNNs - Shiming Ge, Jia Li, Qiting Ye, Zhao Luo[6]	16

2.3.2	Adversarial Occlusion-aware Face Detection - Yujia Chen, Lingxiao Song, Ran He [3]	19
2.3.3	Face Attention Network: An Effective Face Detector for the Occluded Faces-Jianfeng Wang, Ye Yuan, Gang Yu[15]	21
2.3.4	Một vài nghiên cứu gần đây của cộng đồng Pyimagesearch - "COVID-19: Face Mask Detector with OpenCV, Keras/TensorFlow, and Deep Learning" TowardsDataScience - "How I built a Face Mask Detector for COVID-19 using PyTorch Lightning" Github	22 22 23 24
3	NỘI DUNG KHÓA LUẬN	25
3.1	Hướng tiếp cận	25
3.1.1	Giai đoạn một - Trước khi dịch COVID-19 xảy ra:	25
3.1.2	Giai đoạn hai - Khi COVID-19 trở thành đại dịch và lan rộng toàn cầu cho đến nay:	27
3.2	Bộ dữ liệu	28
3.2.1	WIDERMAFA	28
3.2.2	Bộ dữ liệu thu thập ảnh thực tế	42
3.3	Huấn luyện mô hình phát hiện vật thể	48
3.3.1	Một số mô hình CNN	48
	Mobilenetv1 [7]	48
	Mobilenetv2 [14]	49
	YOLOv3[13]	49
	YOLOv5[9]	51
3.3.2	Tensorflow Object Detection API	51
3.3.3	Huấn luyện mô hình phát hiện gương mặt đeo khẩu trang	52
4	THỰC NGHIỆM VÀ ĐÁNH GIÁ	55
4.1	Đánh giá trên bộ dữ liệu WIDERMAFA	55
4.2	Đánh giá trên bộ dữ liệu thực tế tự tạo	57
4.3	Nhận xét kết quả đánh giá và so sánh các mô hình	59
4.4	Thử nghiệm trên hình ảnh thực tế quay ở sân bay Tân Sơn Nhất - TPHCM	60

5 KẾT LUẬN	61
5.1 Kết luận	61
5.2 Hướng phát triển	62
Tài liệu tham khảo	63

Danh sách hình vẽ

1.1	Minh họa cho bài toán	2
2.1	Cấu tạo một tế bào thần kinh	6
2.2	Cấu tạo một Perceptron	6
2.3	Mô tả quá trình tính toán ở lớp Convolution	8
2.4	Mô hình chung của lớp Pooling	9
2.5	Ví dụ về các loại Pooling khác nhau	10
2.6	Mô hình cơ bản của lớp Fully Connected	11
2.7	Công thức tính IoU	12
2.8	Ví dụ các bounding box mà mô hình phát hiện được	14
2.9	Biểu đồ precison-recall	15
2.10	Biểu đồ precison-recall sau khi loại bỏ các đường zig-zag	15
2.11	Mô tả bộ dữ liệu MAFA	17
2.12	Mô hình LLE-CNNs	18
2.13	Kết quả đánh giá mô hình LLE-CNNs	19
2.14	Mô tả đầu ra của mô hình AOFD	20
2.15	Mô hình AOFD	20
2.16	Kết quả đánh giá mô hình AOFD	21
2.17	Cấu trúc mô hình FAN	22
2.18	Kết quả đánh giá mô hình FAN	22
3.1	Mô tả bộ dữ liệu WIDER FACE	29
3.2	Mô tả chi tiết nhãn được chú thích trong tập tin readme.md của WIDER FACE	30
3.3	Hình mẫu cho bounding box (khung viền xanh) và mặt có nhãn "blur" ở mức "clear"	31
3.4	Mặt có nhãn "blur" ở mức "normal blur"	31

3.5	Mặt có nhãn "blur" ở mức "heavy blur"	32
3.6	Mặt có nhãn "expression" ở mức "typical expression"	32
3.7	Mặt có nhãn "expression" ở mức "exaggerate expression"	33
3.8	Mặt có nhãn "illumination" ở mức "normal illumination"	33
3.9	Mặt có nhãn "illumination" ở mức "extreme illumination"	34
3.10	Mặt có nhãn "invalid" là "false"	34
3.11	Mặt có nhãn "invalid" là "true"	35
3.12	Mặt có nhãn "occlusion" ở mức "no occlusion"	35
3.13	Mặt có nhãn "occlusion" ở mức "partial occlusion"	36
3.14	Mặt có nhãn "occlusion" ở mức "heavy occlusion"	36
3.15	Mặt có nhãn "pose" là "typical pose"	37
3.16	Mặt có nhãn "pose" là "atypical pose"	37
3.17	Tập tin ghi nhãn của WIDER FACE	38
3.18	Mô tả bộ dữ liệu MAFA	39
3.19	Số lượng nhãn của WIDERMAFA ở hai lận chọn lọc	41
3.20	Mô hình minh họa việc lọc và xây dựng WIDERMAFA.	42
3.21	Gán nhãn với công cụ labelImg.	43
3.22	Gán nhãn với công cụ labelImg.	43
3.23	Một số hình ảnh trong bộ dữ liệu nhóm tự thu thập.	44
3.24	Ba trong số các trang báo lớn nhóm thu thập hình ảnh.	45
3.25	Các từ khóa được dùng ở các trang báo.	46
3.26	Một số ví dụ về các bài báo mà nhóm thu thập ảnh.	47
3.27	Một số ví dụ về các bài báo mà nhóm thu thập ảnh.	47
3.28	Cấu trúc mô hình mạng Mobilenetv1	48
3.29	Cấu trúc mô hình mạng Mobilenetv2	49
3.30	Mô tả đầu ra của YOLOv3	50
3.31	Kết quả đánh giá của YOLOv5	51
3.32	Mô tả điều chỉnh siêu tham số mô hình YOLOv3	53
3.33	Biểu đồ loss trong quá trình huấn luyện mô hình YOLOv3 qua 56 epochs	54
3.34	Các phiên bản YOLOv5 mà tác giả cung cấp	54
4.1	Thông số bộ test của WIDERMAFA	56
4.2	Thông số bộ test của bộ dữ liệu tự thu thập	58
4.3	Kết quả khi chạy các mô hình trên 1 frame từ clip thực tế	60

Danh sách bảng

4.1	Kết quả precision-recall của các mô hình trên bộ test của WIDERMAFA	56
4.2	Kết quả mAP của các mô hình trên bộ test của WIDERMAFA	57
4.3	Kết quả precision-recall của các mô hình trên bộ test của bộ dữ liệu tự thu thập	58
4.4	Kết quả mAP của các mô hình trên bộ test của bộ dữ liệu tự thu thập	59

TÓM TẮT KHOÁ LUẬN

Bắt nguồn từ một nhánh thách thức trong bài toán phát hiện gương mặt, phát hiện gương mặt bị che khuất dần được cụ thể hóa thành bài toán phát hiện gương mặt đeo khẩu trang. Bài toán này sẽ nhận đầu vào là một ảnh và đầu ra sẽ là bounding box xung quanh những gương mặt đeo khẩu trang và không đeo khẩu trang nếu xuất hiện trong ảnh đầu vào, cùng với nhãn kèm theo tương ứng.

Trước đầu năm 2020, bài toán phát hiện gương mặt bị che khuất nói chung và phát hiện gương mặt đeo khẩu trang nói riêng là bài toán được lượng ít người quan tâm. Khi tìm hiểu và khảo sát các bài báo trong vòng 5 năm gần đây, chúng em đã tìm được số ít bài báo liên quan. Tuy nhiên với hiện trạng dịch COVID-19 bùng phát trên toàn thế giới như hiện nay, bài toán phát hiện gương mặt đeo khẩu trang trở nên đáng chú ý vì tính ứng dụng của nó trong thực tế. Với một hệ thống có khả năng phát hiện gương mặt đeo khẩu trang và không đeo khẩu trang, ta có thể áp dụng hệ thống này vào các camera giám sát để có thể đưa ra nhắc nhở, cảnh báo nhằm giảm thiểu nguy cơ lây lan của dịch bệnh.

Trong khóa luận này, chúng em tập trung nghiên cứu, tìm hiểu về bài toán phát hiện gương mặt đeo khẩu trang. Cùng với đó, chúng em tận dụng các kiến thức đang có cùng với học hỏi thêm kiến thức về máy học, học sâu để áp dụng, xây dựng một mô hình cho bài toán trên. Ngoài ra, chúng em khảo sát, tìm hiểu thêm một số dự án liên quan đến bài toán phát hiện gương mặt đeo khẩu trang trong cộng đồng thị giác máy tính và xây dựng một bộ dữ liệu thực tế tự thu thập. Từ đó, có thể đánh giá, so sánh một cách công bằng các dự án đó với sản phẩm của nhóm.

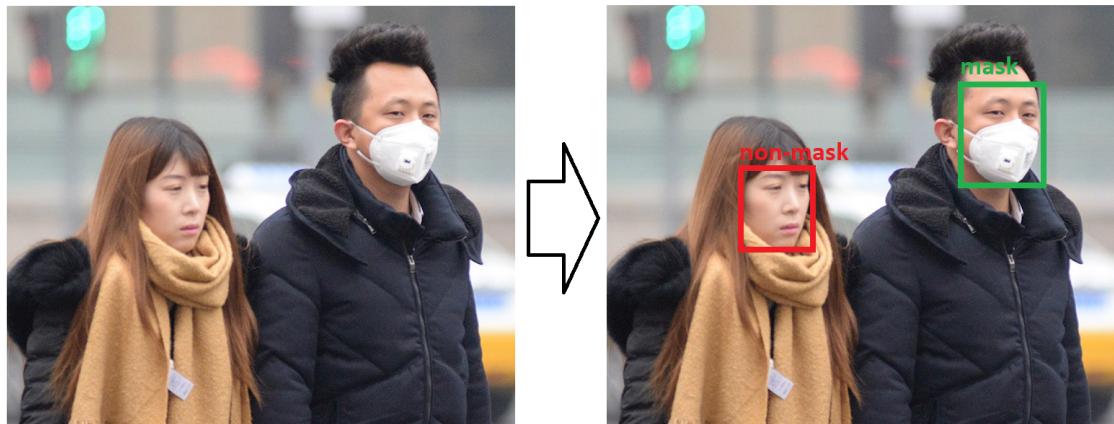
Chương 1

TỔNG QUAN

1.1 Giới thiệu bài toán

Dịch bệnh COVID-19 đã gây ảnh hưởng rất lớn đến đời sống của nhiều quốc gia trên thế giới. Do chưa có thuốc và vaccine điều trị, nên một trong những khuyến cáo được Tổ chức y tế thế giới đưa ra là thực hiện giãn cách và đeo khẩu trang nơi công cộng để ngăn ngừa sự lây lan của dịch bệnh [17],[12]. Bộ ý tế của nước ta cũng đưa ra giải pháp đeo khẩu trang là một trong những việc cần làm nhằm hạn chế dịch bệnh [2]. Vì thế cần thiết phải có các hệ thống có khả năng tự động nhận diện một người có đeo khẩu trang hay không. Điều này sẽ giúp các nhà quản lý đưa ra các chính sách hiệu quả và có những cảnh báo phù hợp hơn.

Trong khóa luận này, chúng em áp dụng một số phương pháp tiên tiến trong thị giác máy tính nhằm xác định phương pháp hiệu quả cho bài toán. Đầu vào của bài toán là một ảnh có chứa người có khuôn mặt, đầu ra cho biết khuôn mặt có đeo khẩu trang không?



HÌNH 1.1: Minh họa bài toán xử lý

1.2 Tính ứng dụng

Với bài toán này, ta có thể áp dụng vào rất nhiều ứng dụng thực tế như sau:

- Ứng dụng vào camera giám sát ở những nơi công cộng, khi phát hiện có người không mang khẩu trang, sẽ có thông báo phát qua loa để cảnh báo, nhắc nhở.
- Ứng dụng ở các cổng kiểm soát ở sân bay, ga xe lửa, bệnh viện, tòa nhà... nhằm kiểm soát người vào có đeo khẩu trang theo đúng quy định hay không.
- Thông kê số lượng và tỉ lệ người chấp hành quy định người chấp hành đeo khẩu trang ở một địa điểm cụ thể.

1.3 Những thách thức và khó khăn gặp phải

Cũng như một bài toán nhận dạng vật thể đơn thuần, bài toán phát hiện gương mặt đeo khẩu trang cũng gặp phải một số thách thức chung.

1.3.1 Cách thức đánh giá mô hình

Đối với một bài toán phân loại thì đầu vào sẽ là một ảnh và đầu ra sẽ là nhãn của vật trong ảnh đó, vậy nên để đánh giá đầu ra sẽ khá đơn giản. Tuy nhiên với bài toán phát hiện vật thể thì đầu vào sẽ là một ảnh và đầu ra sẽ là bounding box của các vật trong ảnh kèm

với nhãn của vật đó. Ta phải đánh giá cùng một lúc hai công việc của mô hình là xác định vị trí của vật và gắn nhãn cho vật đó. Chính vì vậy mà để đánh giá một mô hình phát hiện vật thể sẽ phức tạp hơn và so với một mô hình phân loại vật thể.

1.3.2 Tốc độ và độ chính xác của mô hình

Ngoài độ chính xác cao, một mô hình phát hiện vật thể cần có thời gian xử lý một ảnh đủ nhanh để có thể áp dụng vào các camera giám sát. Cho dù một mô hình có độ chính xác tuyệt đối nhưng lại xử lý một ảnh với thời gian quá lâu thì sẽ rất khó để áp dụng vào các ứng dụng thực tế. Việc cân bằng giữa tốc độ và độ chính xác của mô hình sẽ được quyết định bởi tính chất của bài toán cần giải.

1.3.3 Dữ liệu

Đối với mọi bài toán phát hiện vật thể, dữ liệu là phần quan trọng nhất trong quá trình giải bài toán. Tính chất dữ liệu cần phải sát với ứng dụng thực tế mình đang hướng tới. Đặc biệt với bài toán phát hiện gương mặt đeo khẩu trang, dữ liệu về mặt đeo khẩu trang vẫn còn hạn chế. Hiện tại chỉ có bộ dữ liệu MAFA[6] là bộ dữ liệu lớn và là benchmark đánh giá các mô hình về bài toán phát hiện gương mặt đeo khẩu trang. Tuy nhiên, bộ dữ liệu MAFA[6] là bộ dữ liệu tổng quát, tổng hợp nhiều trường hợp khác nhau và dữ liệu huấn luyện chỉ gắn nhãn các gương mặt đeo khẩu trang.

Để có bộ dữ liệu phù hợp với bài toán đang làm, chúng em đã phải sử dụng thêm dữ liệu từ bộ dữ liệu WIDER FACE[18] và lọc lại với một số điều kiện để có dữ liệu phù hợp với bài toán đang làm. Cụ thể việc xử lý dữ liệu cho bài toán sẽ được chúng em trình bày tại chương 3.

1.4 Mục tiêu khóa luận

Một số mục tiêu mà chúng em hướng tới khi thực hiện khóa luận:

- Tìm hiểu về bài toán phát hiện vật thể, cách áp dụng các mạng học sâu để giải bài toán đó.

- Hiểu rõ được bài toán phát hiện gương mặt đeo khẩu trang. Khảo sát một số hướng nghiên cứu, ứng dụng hiện tại trong cộng đồng.
- Tự tạo nên một mô hình để giải bài toán phát hiện gương mặt đeo khẩu trang, hướng tới ứng dụng giám sát, cảnh báo tại các nơi công cộng trong hoàn cảnh dịch COVID-19 như hiện nay.

1.5 Cấu trúc Khóa luận tốt nghiệp

Nội dung Khóa luận tốt nghiệp được tổ chức như sau:

- Chương 1: Giới thiệu tổng quan về khóa luận.
- Chương 2: Trình bày về cơ sở lý thuyết, các khảo sát mà nhóm thực hiện được cùng một số kiến thức về máy học và học sâu.
- Chương 3: Trình bày chi tiết nội dung về mô hình và các bước trong quá trình huấn luyện mô hình đó.
- Chương 4: Trình bày phương pháp đánh giá và kết quả thực nghiệm.
- Chương 5: Kết luận và hướng phát triển của khóa luận.

Chương 2

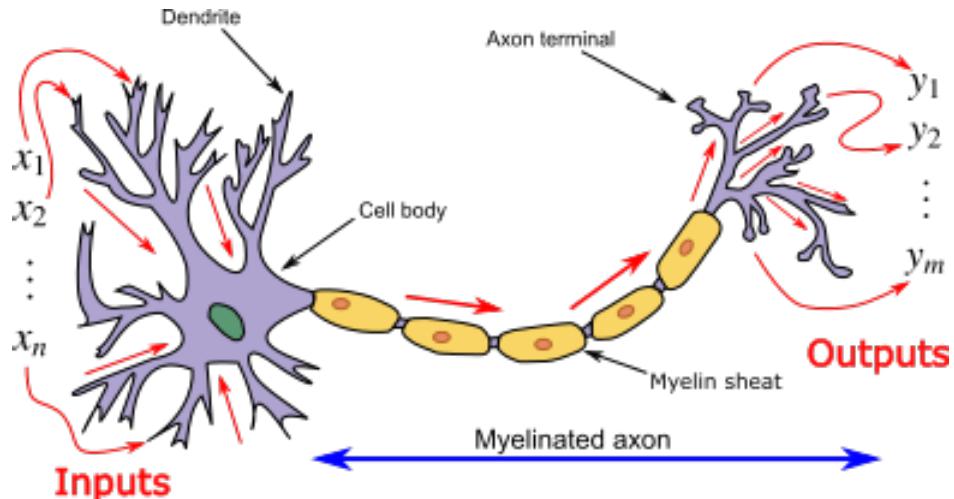
CƠ SỞ LÝ THUYẾT

2.1 Các mô hình mạng máy học

2.1.1 Mạng nơ-ron nhân tạo (Artificial Neural Network - ANN)

ANN được lấy ý tưởng từ cấu trúc của một tế bào thần kinh. Đầu tiên, ta sẽ điểm sơ qua về cấu trúc của một tế bào thần kinh. Một tế bào thần kinh được chia thành 4 phần chính:

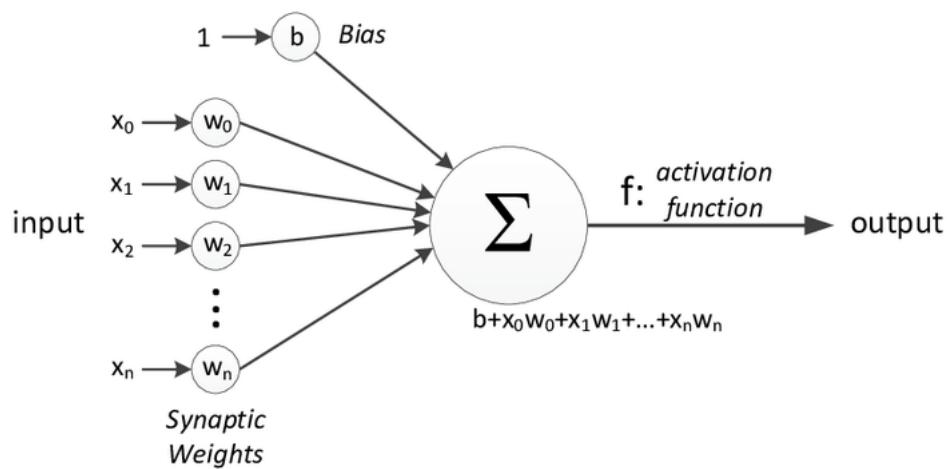
- Thân tế bào (Soma): là chỗ phình to của tế bào thần kinh, là nơi cung cấp dinh dưỡng cho tế bào và có thể phát sinh xung động thần kinh cũng như tiếp nhận xung động thần kinh từ nơi khác truyền đến để xử lý.
- Đuôi gai (Dendrites): là các tua ngắn phát triển từ thân tế bào. Mỗi tế bào thần kinh có nhiều đuôi gai, chia thành nhiều nhánh và có chức năng tiếp nhận các xung thần kinh từ tế bào khác, truyền chúng tới thân tế bào .
- Sợi trục (Axon): là sợi thần kinh đơn dài, có chức năng truyền tín hiệu từ thân tế bào tới tế bào thần kinh khác hoặc tới tế bào cơ .
- Khớp thần kinh (Synapse): là phần cấu trúc phí cuối sợi trục, có chức năng kết nối tế bào thần kinh với các đuôi gai của các tế bào khác hoặc các cơ quan thụ cảm.



HÌNH 2.1: Cấu tạo một tế bào thần kinh

Đuôi gai sẽ nhận tín hiệu từ khớp thần kinh của các tế bào thần kinh khác. Thân tế bào xử lý các tín hiệu nhận được và gửi đến các tế bào thần kinh tiếp theo thông qua sợi trực và khớp thần kinh của tế bào thần kinh đó.

Tương tự như cấu trúc của một tế bào thần kinh, cấu trúc của một lớp trong ANN sẽ có các chức năng nhận tín hiệu, xử lý tín hiệu và truyền tín hiệu đến lớp khác của ANN. Hãy ví dụ một mô hình mạng chỉ với một lớp duy nhất (hay còn gọi là một Perceptron) sẽ có cấu trúc như sau:



HÌNH 2.2: Cấu tạo một Perceptron

Ở ví dụ trên, mô hình sẽ nhận các đầu vào là các số và xử lý tính toán bằng công thức tổng của các tích đầu vào với trọng số tương ứng. Sau đó đưa kết quả qua một hàm số kích hoạt và truyền cho đầu ra. Ở các trường hợp phức tạp hơn, số lượng Perceptron sẽ nhiều hơn tạo thành một lớp và đầu ra này có thể sẽ là đầu vào của lớp tiếp theo của mô hình mang.

2.1.2 Mạng nơ-ron tích chập (Convolutional Neural Network - CNN)

Trong các mạng máy học thì CNN là một trong các mạng được sử dụng rộng rãi dành cho bài toán phân loại hình ảnh, nhận diện, phát hiện vật thể... Tương đồng với ANN, CNN cũng sẽ nhận các đầu vào, xử lý thông tin và cho đầu ra.

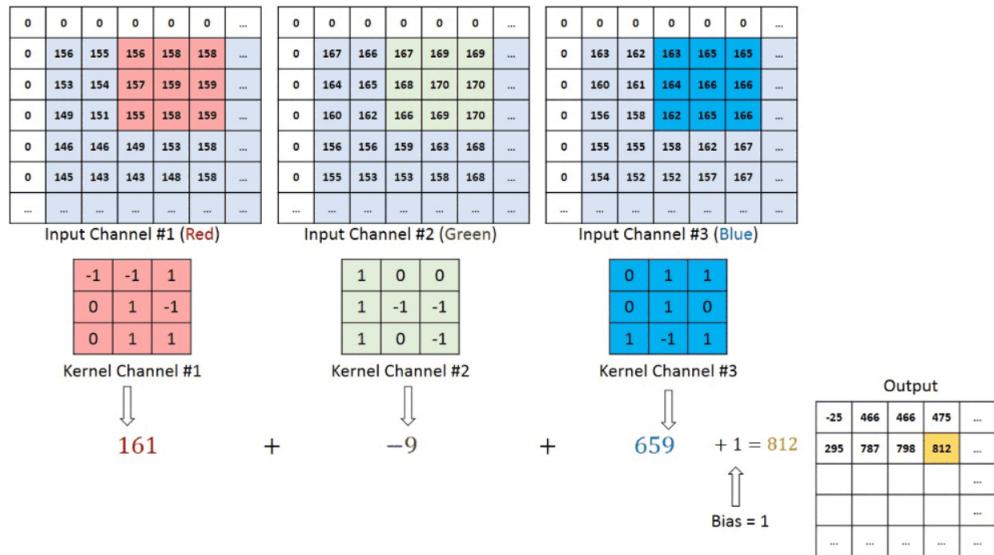
Điều khác biệt ở đây chính là cách xử lý thông tin của CNN và dữ liệu đầu vào. Dữ liệu đầu vào của CNN thường sẽ là hình ảnh, hoặc rộng hơn sẽ là dữ liệu mang tính tuần tự. Về cách xử lý dữ liệu, CNN sẽ có các lớp khác nhau, bao gồm ba lớp chính: Convolution, Pooling, Fully Connected.

Lớp Convolution

Đầu tiên ta sẽ tìm hiểu về phép nhân tích chập (phép tính convolution). Đây là phép tính giữa hai ma trận có cùng kích thước, cho ra kết quả là một số có giá trị bằng tổng các tích của các vị trí tương ứng giữa hai ma trận.

Ta định nghĩa kernel là một ma trận hình vuông với kích thước bất kỳ, thường thì kernel sẽ có kích thước là số lẻ. Ở lớp Convolution, ta ứng dụng kỹ thuật cửa sổ trượt (sliding window) để thực hiện phép tính convolution trên từng vùng của ảnh đầu vào để có được kết quả đầu ra.

Với đầu vào là ảnh với nhiều kênh, có thể là ảnh RGB, SHV... hoặc đầu ra của lớp convolution trước đó có nhiều lớp, ta sẽ dùng kernel có độ sâu tương ứng để thực hiện phép nhân. Hình 2.3 trình bày cụ thể quá trình thực hiện phép nhân convolution trên ảnh với hệ màu RGB.

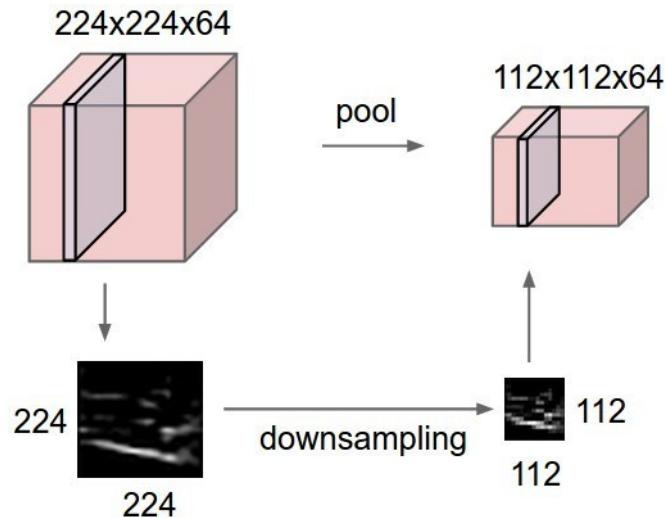


HÌNH 2.3: Mô tả quá trình tính toán ở lớp Convolution

Tại đây, ta sẽ có thêm hai khái niệm mới là “padding” và “stride”: Khi thực hiện phép nhân convolution đơn thuần, ma trận đầu ra sẽ có kích thước nhỏ hơn ma trận đầu vào. Để có thể giữ nguyên kích thước với đầu vào và có thể thực hiện phép nhân với các pixel ở góc và cạnh của ảnh tương tự với cái pixel còn lại, ta sẽ thực hiện kỹ thuật padding, tự tạo thêm các giá trị xung quanh viền của ma trận đầu vào. “Stride” là số pixel mà kernel sẽ dịch chuyển ở mỗi bước khi thực hiện phép tính convolution trên từng vùng của ma trận đầu vào.

Lớp Pooling

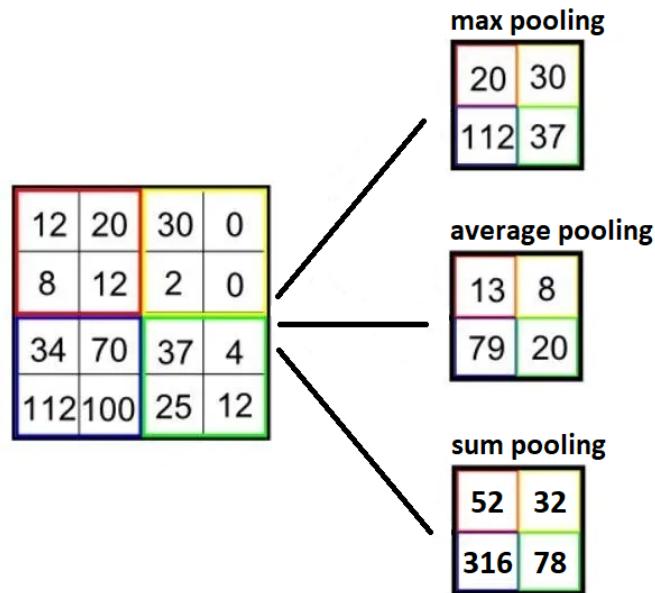
Mục đích chính của lớp Pooling chính là giảm kích thước của ma trận nhưng không làm mất đi quá nhiều thông tin quan trọng trong đó. Từ đó giảm số lượng tham số cần tính toán, thời gian huấn luyện nhưng không ảnh hưởng quá nhiều đến kết quả đầu ra.



HÌNH 2.4: Mô hình chung của lớp Pooling

Với lớp pooling, ta vẫn sử dụng các kernel, nhưng các kernel này không có trọng số. Tùy vào loại pooling, đầu ra của lớp này sẽ có cách tính riêng. Có nhiều loại pooling, trong đó có ba loại phổ biến và được sử dụng rộng rãi:

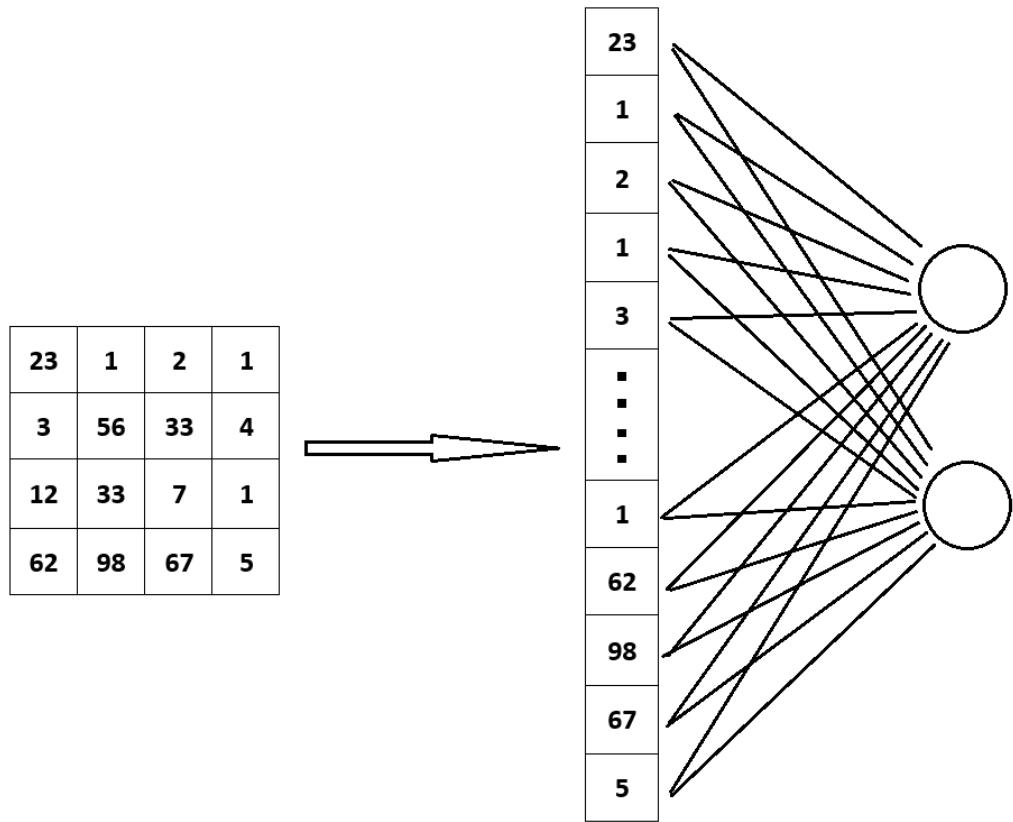
- Max pooling sẽ chọn ra giá trị lớn nhất trong kernel để đưa ra kết quả.
- Average pooling sẽ tính trung bình các giá trị trong kernel để đưa ra kết quả.
- Sum pooling sẽ tính tổng các giá trị trong kernel để đưa ra kết quả.



HÌNH 2.5: Ví dụ về các loại Pooling khác nhau

Lớp Fully Connected

Các lớp Fully Connected thường sẽ được đặt ở phần cuối cùng của mô hình. Kết quả đầu ra của các lớp này sẽ quyết định đến kết quả cuối cùng. Dữ liệu ảnh sau khi đi qua các lớp convolution và pooling trước đó sẽ có là cấu trúc là một ma trận có kích thước $W \times H \times D$, các giá trị của ma trận này sẽ kết hợp tạo thành một vector có độ dài $W \times H \times D$ phần tử. Từ đây, cấu trúc lớp Fully Connected sẽ tương tự như một ANN, mỗi đầu ra của lớp này sẽ được kết nối tới tất cả đầu vào của lớp kế tiếp để kết nối các đặc điểm của ảnh và cho ra được đầu ra của cả model.



HÌNH 2.6: Mô hình cơ bản của lớp Fully Connected

2.2 Đánh giá mô hình

Như đã nói ở phần thách thức ở chương 1, mô hình phát hiện vật thể thực hiện hai tác vụ: xác định vị trí của vật thể và phân loại vật thể ở vị trí đó. Khi đánh giá mô hình phát hiện vật thể, ta cần đánh giá cho cả hai tác vụ nói trên.

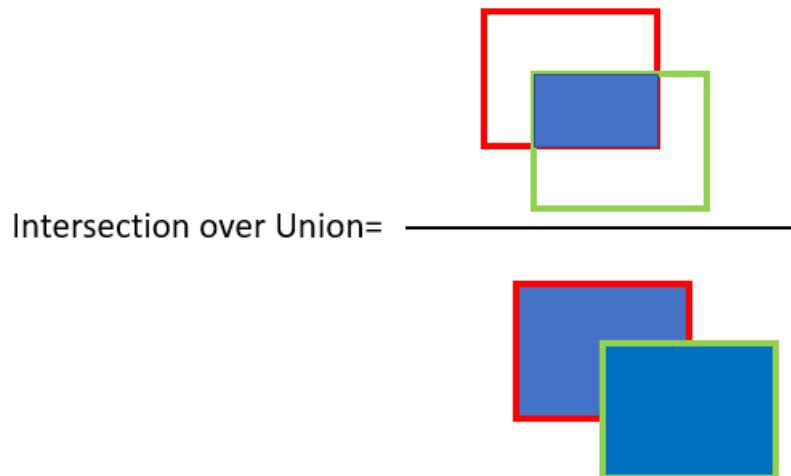
2.2.1 Mức độ tự tin - Confidence Score

Mỗi bounding box mà mô hình phát hiện được sẽ kèm theo một chỉ số gọi là confidence score . Đây là chỉ số đánh giá mức độ tự tin của mô hình đó khi phân loại nhãn cho vật thể.

Có thể lấy ví dụ đơn giản như sau, với bài toán phát hiện gương mặt đeo khẩu trang, sau khi phát hiện được một bounding box, mô hình có nhiệm vụ phân loại bounding box ấy là gương mặt đeo khẩu trang hoặc gương mặt không đeo khẩu trang. Cụ thể, nếu một mô hình cho đầu ra là thông tin của một bounding box có nhãn là "mask face" với confidence score là 0.9. Điều này có nghĩa là mô hình phát hiện tại vị trí của bounding box đó có một gương mặt đeo khẩu trang với mức độ tự tin 90%.

2.2.2 Intersection over Union - IoU

Trước tiên, ta cần hiểu được khái niệm Intersection over Union (IoU). IoU là phương thức để xác định bounding box được dự đoán có chính xác hay không. Giả sử A là bounding box mà mô hình dự đoán và B là bounding box đúng, cần được phát hiện. IoU là tỉ lệ của diện tích phần giao A và B trên diện tích phần hợp A và B.



HÌNH 2.7: Công thức tính IoU

Ta có thể xác định một ngưỡng thích hợp cho IoU để xác định bounding box dự đoán có chính xác hoặc không. Cụ thể trong trường hợp ngưỡng IoU được đặt là 0.5 thì :

- $\text{IoU} \geq 0.5$: bounding box được dự đoán chính xác (True Positive - TP).
- $\text{IoU} < 0.5$: bounding box được dự đoán sai (False Positive - FP).

- Khi trong ảnh có vật thể nhưng mô hình không phát hiện được, tình huống này được xem là False Negative - FN
- True Negative - TN là tất cả những phần ảnh không có vật thể và mô hình không phát hiện vật thể. Đối với bài toán phát hiện vật thể, TN không hữu dụng, vậy nên ta sẽ không quan tâm đến TN

Với bốn con số trên, ta hình thành được ma trận lỗi (Confusion matrix). Ứng với mỗi vật thể mà mô hình được huấn luyện để phát hiện, ta sẽ hình thành được một ma trận lỗi khi đánh giá mô hình. Nếu ngưỡng IoU được đặt với giá trị khác nhau, kết quả ma trận lỗi cũng sẽ khác nhau. Ngưỡng này có thể đặt tùy ý. Tuy nhiên, thông thường, các nhà nghiên cứu sẽ đặt ở 0.5. Ở một số trường hợp, ngưỡng này có thể đặt ở 0.75, 0.9, 0.95 để đánh giá chi tiết mô hình hơn.

2.2.3 Độ chính xác - precision và độ phủ - recall

Độ chính xác xác định tỷ lệ phân loại chính xác những bounding box mà mô hình phát hiện. Mô hình có precision=1 có nghĩa rằng mô hình phân loại chính xác tất cả bounding box mà mô hình phát hiện được. Độ bao phủ xác định tỷ lệ phát hiện vật thể có trong bộ dữ liệu đánh giá. Mô hình với recall=1 có nghĩa rằng mô hình phát hiện hết tất cả các bounding box có trong bộ dữ liệu đánh giá. Độ chính xác và độ bao phủ được xác định riêng biệt cho từng loại vật thể mà mô hình được huấn luyện để phát hiện theo công thức:

$$\text{Precision} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalsePositives}}$$

$$\text{Recall} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalseNegatives}}$$

2.2.4 mean Average Precision - mAP

Đầu tiên, chúng ta cần làm rõ khái niệm Average Precision (AP). AP được xem là độ chính xác trung bình dành cho một vật thể mà mô hình có khả năng phát hiện. Để tính được AP của một loại vật thể, ta cần sắp xếp tất cả các bounding box của vật thể đó mà mô hình phát hiện được theo confidence score của bounding box đó. Ta có thể tính được độ chính xác và độ phủ ứng với từng mức trên bảng xếp hạng đó. Bounding box được xem

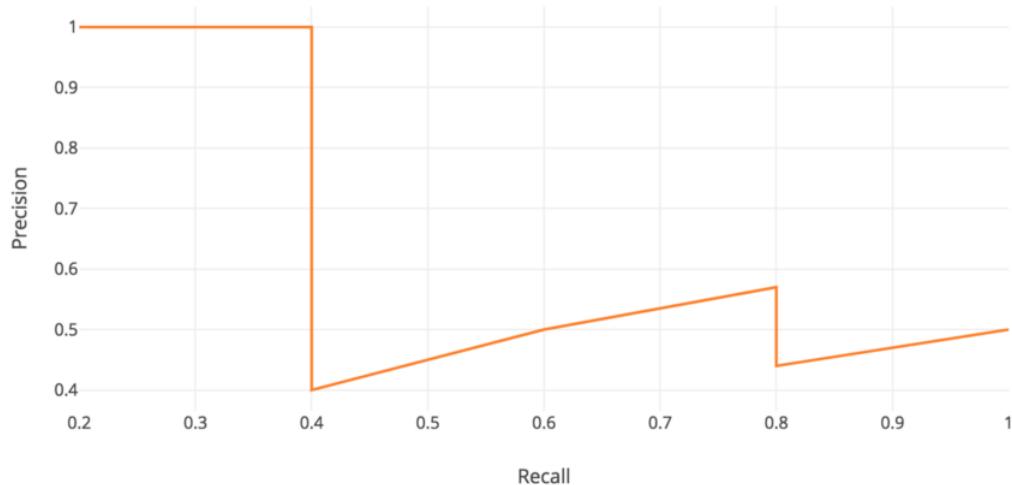
là chính xác khi mà IoU của nó với bounding box đúng lớn hơn ngưỡng nào đó, thông thường ngưỡng này sẽ được đặt ở mức 0.5.

Lấy một ví dụ như sau, giả sử ta có 5 vật thể trong ground-truth và đây là bảng xếp hạng các bounding box mà mô hình đã phát hiện được cùng với thông tin về tính chính xác, độ chính xác cũng như độ phủ của mô hình khi chỉ tính các bounding box tại mỗi mức trở lên.

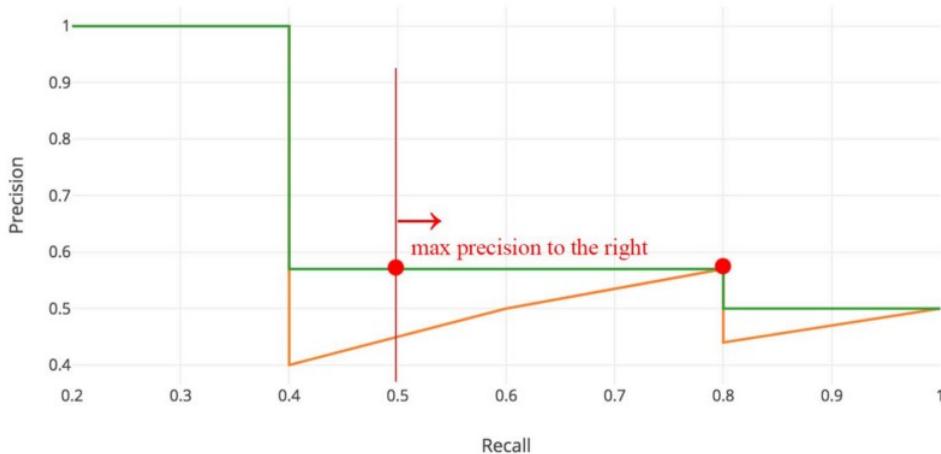
Rank	Correct?	Precision	Recall
1	True	1.0	0.2
2	True	1.0	0.4
3	False	0.67	0.4
4	False	0.5	0.4
5	False	0.4	0.4
6	True	0.5	0.6
7	True	0.57	0.8
8	False	0.5	0.8
9	False	0.44	0.8
10	True	0.5	1.0

HÌNH 2.8: Ví dụ về bảng xếp hạng các bounding box mà mô hình phát hiện được

Từ bảng xếp hạng trên, ta có thể biểu diễn thành một biểu đồ như hình 2.9 Từ đó, ta có thể thấy được rằng biểu đồ có dạng zig-zag. Để loại bỏ điều này, ứng với mỗi mức recall, ta sẽ lấy giá trị precision lớn nhất về phía bên phải mức đó. Bằng cách này, từ đường zig-zag, ta có thể chuyển thành các đoạn thẳng biểu diễn mức độ giảm dần của precision khi recall tăng như đường thẳng mới ở hình 2.10.



HÌNH 2.9: Biểu đồ precison-recall



HÌNH 2.10: Biểu đồ precison-recall sau khi loại bỏ các đường zig-zag

Từ đường thẳng mới này, AP sẽ có giá trị bằng giá trị trung bình precision ở 11 mức recall 0, 0.1, 0.2, 0.3, ..., 0.9, 1. Ở một số trường hợp, nếu mô hình không có khả năng detect tất cả vật thể có trong ground-truth, đồng nghĩa với việc recall chỉ đạt lớn nhất ở một mức bé hơn 1, thì precision ở các mức sau đó sẽ có giá trị bằng 0. mAP sẽ là giá trị trung bình của các AP ứng với mỗi vật thể mà mô hình được huấn luyện để phát hiện.

2.3 Các nghiên cứu liên quan

Ngoài các lý thuyết về phát hiện vật thể và cách đánh giá mô hình, chúng em có khảo sát thêm các bài báo, nghiên cứu của cộng đồng, từ những hội nghị đỉnh cao như CVPR-Conference on Computer Vision and Pattern Recognition, ACCV-Asian Conference on Computer Vision,... đến các trang web, blog uy tín trong ngành như PyImageSearch, Towards Data Science, ... Mục đích của việc khảo sát này là để chúng em có cái nhìn tổng quan hơn cũng như hiểu được về các hướng giải quyết, cách làm mà cộng đồng hiện tại đang hướng đến.

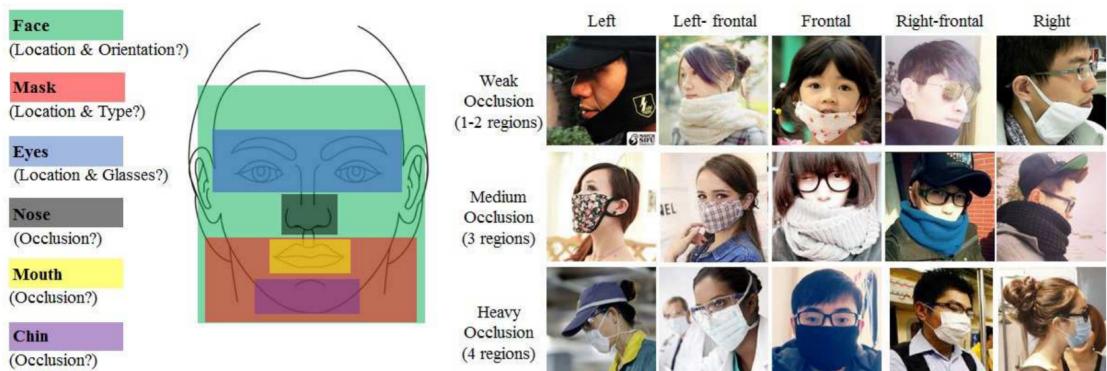
2.3.1 Detecting Masked Faces in the Wild with LLE-CNNs - Shiming Ge, Jia Li, Qiting Ye, Zhao Luo[6]

Đây là một bài báo đã được công bố tại hội nghị CVPR 2017. Theo như nhóm tác giả của bài báo, bài toán phát hiện gương mặt đang được rất nhiều nhà nghiên cứu theo làm và phát triển. Đến thời điểm hiện tại thì đã có rất nhiều mô hình khác nhau đạt được độ chính xác và tốc độ đáng ngạc nhiên khi được thử trên các bộ dữ liệu phổ biến.

Tuy nhiên trong ngữ cảnh thực tế, thì việc phát hiện gương mặt gặp rất thách thức khác nhau. Cụ thể một thách thức rõ ràng đó là việc gương mặt người đeo khẩu trang, bị che khuất. Việc phát hiện được những gương mặt này là điều thực sự cần thiết với ứng dụng giám sát kiểm tra. Nhưng vào thời điểm hiện tại, không có bộ dữ liệu về gương mặt đeo khẩu trang nào đủ lớn. Từ đó, nhóm tác giả đã đề xuất một bộ dữ liệu mới dành riêng cho bài toán phát hiện gương mặt đeo khẩu trang gọi là MAFA và đồng thời đề xuất một mô hình mới cho bài toán vừa nêu với tên LLE-CNNs.

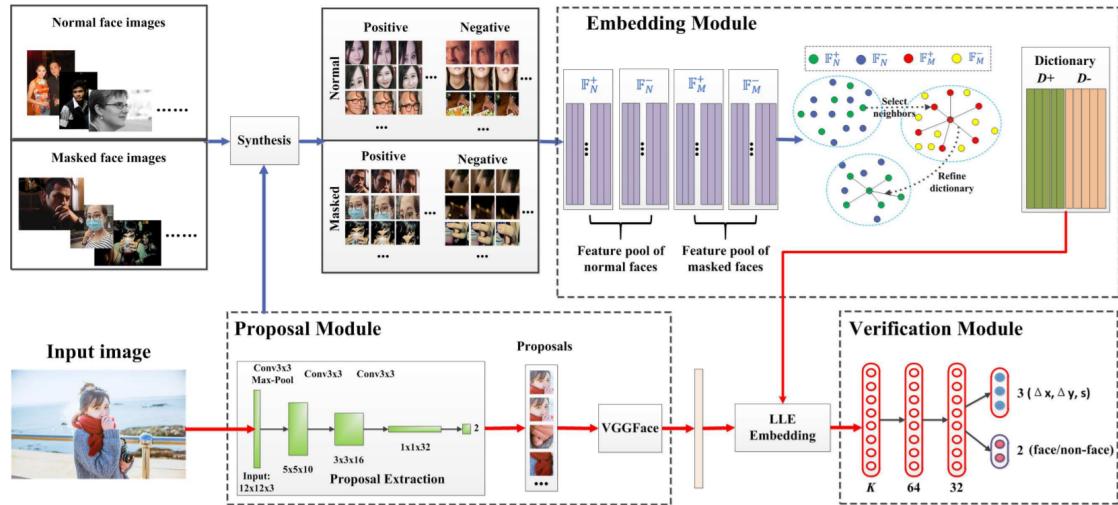
Về phần bộ dữ liệu mới - MAFA, nhóm tác giả đã thu thập các hình ảnh về mặt từ Internet, các trang tìm kiếm thông tin như Google, Bing... và các mạng xã hội như Flickr... Trong tất cả các hình ảnh mà nhóm tác giả thu thập, họ đã loại bỏ đi những hình ảnh mà chỉ chứa gương mặt không bị che khuất và họ chỉ giữ lại những hình ảnh có kích thước cạnh nhỏ nhất là 80 pixels. Với những bức ảnh còn lại, một nhóm gồm 9 người sẽ thực hiện việc gắn nhãn cho phần dữ liệu này. Mỗi bức ảnh sẽ được gắn nhãn bởi 2 người khác nhau và sẽ được kiểm tra lại bởi 1 người khác. Nhãn của mỗi gương mặt có trong ảnh sẽ được xác định các yếu tố như sau:

- Vị trí của gương mặt nằm trong bức ảnh, vị trí này sẽ được đánh dấu bằng một bounding box hình vuông. Một số gương mặt quá mờ, hình ảnh bị biến dạng hoặc độ dài khung ảnh có kích thước nhỏ hơn 32 điểm ảnh sẽ được gắn nhãn "Ignore". Những gương mặt có nhãn này sẽ không được tính vào kết quả bài toán.
- Vị trí tâm hai mắt của gương mặt.
- Vị trí của khẩu trang hoặc kính (nếu có) của gương mặt
- Hướng của gương mặt. Hướng sẽ được chia thành 5 nhóm khác nhau: thẳng, trái, phải, trái - thẳng, phải - thẳng.
- Mức độ bị che khuất. Nhóm tác giả chia gương mặt thành 4 vùng chính, bao gồm: cầm, miệng, mũi, mắt. Số lượng vùng bị che khuất sẽ là mức độ bị che khuất của gương mặt đó.
- Loại khẩu trang. Nhóm tác giả định nghĩa bốn loại khẩu trang như sau:
 - “Simple mask”: Loại khẩu trang chỉ có một màu đơn giản.
 - “Complex mask”: Loại khẩu trang bao gồm nhiều màu kết hợp hoặc có gắn thêm các hình ảnh khác.
 - “Human body”: Gương mặt bị che khuất bởi các bộ phận con người như tay, tóc...
 - “Hybrid mask”: Gương mặt bị che khuất bởi cả hai trong các nhóm nói trên hoặc một nhóm trên và có đeo kính.



HÌNH 2.11: Mô tả bộ dữ liệu MAFA

Ngoài ra, nhóm tác giả công bố thêm một mô hình mới và đặt tên là LLE-CNNs. Mô hình được kết hợp bởi ba module khác nhau: “Proposal Module”, “Embedded Module”, “Verification Module”.



HÌNH 2.12: Mô hình LLE-CNNs

- **Proposal Module:** Trích xuất các gương mặt đề xuất.
- **Embedding Module:** Xác định lại các gương mặt được đề xuất từ proposal module bằng lượng lớn hình ảnh gương mặt và không phải gương mặt được thu thập trước đó.
- **Verification Module:** Thực hiện việc phân loại gương mặt đeo khẩu trang và gương mặt không đeo khẩu trang.

Kết quả so sánh khi thử nghiệm các mô hình khác và LLE-CNNs trên bộ dữ liệu MAFA cho thấy LLE-CNNs vượt trội so với các mô hình khác.

Attributes	SURF [18]	NPD [20]	ZR [37]	HH [22]	HPM [7]	MT [35]	OUR	Min ↑
Left	0.02	1.01	5.02	7.91	1.29	6.89	17.2	9.29
Left-Fr.	2.17	4.37	29.3	28.5	26.6	31.9	61.7	29.8
Front	19.7	16.9	45.5	51.6	64.4	62.2	79.6	15.2
Right-Fr.	1.93	2.34	13.8	20.4	18.9	20.2	54.5	34.1
Right	0.02	0.23	1.34	5.43	0.93	1.94	14.3	8.87
Weak	18.1	5.87	37.1	47.7	58.5	56.2	75.8	17.3
Medium	12.7	17.0	13.9	46.4	34.8	45.6	67.9	22.3
Heavy	0.05	0.52	7.12	5.59	5.31	5.24	22.5	15.4
Simple	10.7	12.8	39.3	45.3	54.7	51.6	74.3	19.6
Complex	11.8	8.52	33.3	42.1	46.1	48.2	71.6	23.4
Body	12.3	4.12	21.4	34.7	23.4	30.4	62.0	27.3
Hybrid	0.17	0.63	7.64	7.58	6.00	6.48	24.2	16.6
All	16.1	19.6	41.6	50.9	60.0	60.8	76.4	15.6

HÌNH 2.13: Kết quả so sánh mô hình LLE-CNNs với các mô hình khác được công bố trong bài báo

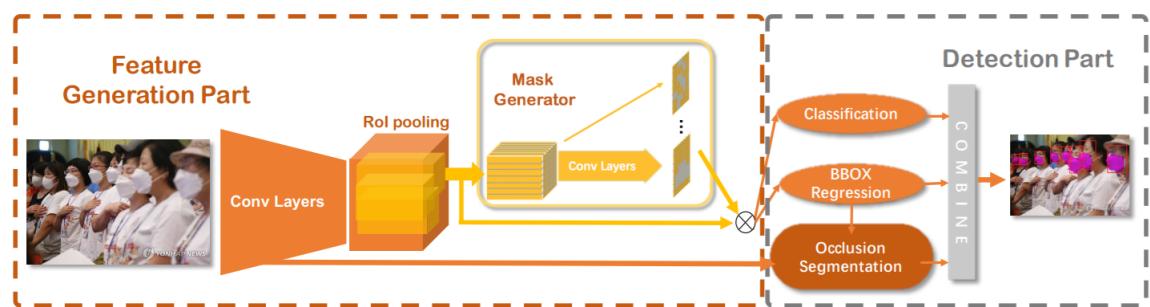
2.3.2 Adversarial Occlusion-aware Face Detection - Yujia Chen, Lingxiao Song, Ran He [3]

Bài báo này được công bố lại hội nghị quốc tế về lý thuyết sinh trắc học, ứng dụng và hệ thống lần thứ 9 của IEEE năm 2018 (BTAS 2018). Nhóm tác giả của bài báo công bố một mô hình mới với tên gọi AOFD. Mô hình AOFD đồng thời phát hiện gương mặt đeo khẩu trang và phân loại vùng có khẩu trang.



HÌNH 2.14: Mô tả đầu ra của mô hình AOFD

Nhóm tác giả áp dụng phương pháp huấn luyện nghịch cảnh, tức là tạo ra các đặc trưng thay thế cho vùng gương mặt bị che khuất hoặc đeo khẩu trang. Từ đó, AOFD có khả năng phát hiện gương mặt bị che khuất nặng, ít phần gương mặt lộ với mức độ tự tin cao.



HÌNH 2.15: Mô hình AOFD

Methods	All	'masked' only	w/o 'Ignored'
AOFD	81.3%	83.5%	91.9%
FAN	-	76.5%	88.3%
LLE-CNNs	-	-	76.4%
MTCNN	-	-	60.8%

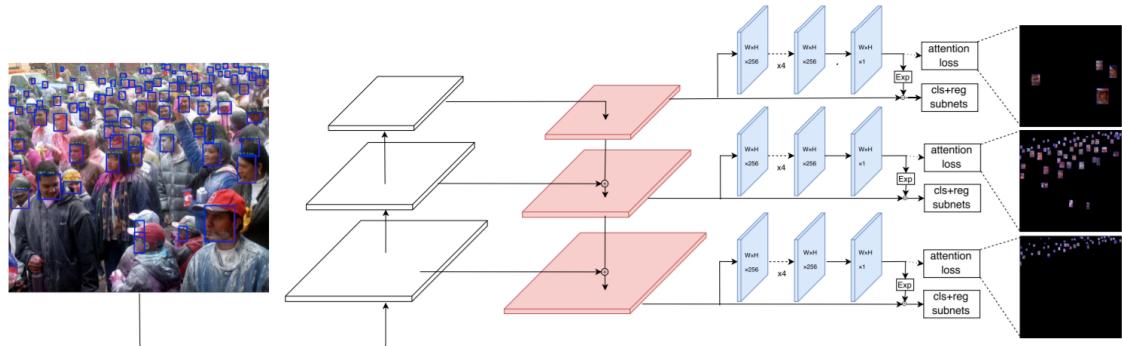
HÌNH 2.16: Kết quả so sánh độ chính xác trung bình của AOFD với các mô hình khác trên bộ test của bộ dữ liệu MAFA

Hình 2.16 là kết quả so sánh của mô hình AOFD với các mô hình khác được công bố trong bài báo. Như đã nói ở phần trước, trong bộ dữ liệu MAFA, có những gương mặt được gắn nhãn "Ignore" và sẽ bị bỏ qua trong quá trình tính kết quả. Ở đây, độ chính xác trung bình của mô hình AOFD đạt đến 91.9%, cao hơn tất cả các mô hình khác.

2.3.3 Face Attention Network: An Effective Face Detector for the Occluded Faces-Jianfeng Wang, Ye Yuan, Gang Yu[15]

Đây là bài báo được công bố tại trang “arXiv.org” của Đại học Cornell. “arXiv.org” là nơi các bài báo khoa học trong lĩnh vực toán học, vật lý, khoa học máy tính, sinh học, thống kê... được công bố và lưu trữ, hiện đang thuộc quyền sở hữu của Đại học Cornell.

Nhóm tác giả của bài báo công bố một mô hình mới với tên gọi FAN. Theo bài báo, các mô hình phát hiện gương mặt thường cho kết quả độ phủ thấp khi gặp các trường hợp mặt bị che khuất bởi khẩu trang hoặc mắt kính. FAN là mô hình được phát triển dựa trên mạng RetinaNet, kết hợp với một số phương pháp khác như, mô hình kim tự tháp giúp phát hiện các gương mặt với kích thước khác nhau, tăng cường dữ liệu huấn luyện, phương pháp phát hiện "one-shot"...



HÌNH 2.17: Cấu trúc mô hình FAN

FAN được nhóm tác giả công bố rằng có khả năng tăng đáng kể độ phủ với bài toán phát hiện các gương mặt đeo khẩu trang nhưng không ảnh hưởng tới tốc độ của mô hình. Kết quả so sánh FAN với các mô hình khác được tác giả công bố như sau:

Method	mAP
LLE-CNNs [6]	76.4
AOFD [2]	77.3
FAN	88.3

HÌNH 2.18: Kết quả so sánh của mô hình FAN

2.3.4 Một vài nghiên cứu gần đây của cộng đồng

Thời gian gần đây, trước tình hình dịch COVID-19 đang lây lan rộng và diễn biến phức tạp trên thế giới, có rất nhiều diễn đàn, blog uy tín, tiêu biểu trong giới thị giác máy tính công bố một số dự án về bài toán phát hiện gương mặt đeo khẩu trang với mục đích đóng góp sức nghiên cứu của riêng mình cho cộng đồng. Nhóm chúng em đã khảo sát cách làm của các dự án đó và so sánh với các mô hình mà nhóm sử dụng cho bài toán này.

Pyimagesearch - "COVID-19: Face Mask Detector with OpenCV, Keras/TensorFlow, and Deep Learning"

Pyimagesearch là một trang web chuyên về trí tuệ nhân tạo, đặc biệt là thị giác máy tính nổi tiếng trong ngành. Có rất nhiều bài viết hướng dẫn, bàn luận về các bài toán, chủ đề nổi bật trong lĩnh vực thị giác máy tính. Bài viết “COVID-19: Face Mask Detector with OpenCV, Keras/TensorFlow, and Deep Learning” được chủ của trang web, anh

Adrian Rosebrock, viết để giới thiệu một hướng làm về bài toán ứng dụng phát hiện gương mặt đeo khẩu trang.

Mô hình mà Adrian sử dụng được chia thành hai phần, phát hiện gương mặt và phân loại gương mặt đó có đeo khẩu trang hoặc không đeo khẩu trang. Anh ấy đã sử dụng mô hình “res10_300x300_ssd_iter_140000” đã được train sẵn trước đó cho phần phát hiện gương mặt. Đối với phần phân loại gương mặt, Adrian train mô hình Mobilenet_v2 với bộ dữ liệu tự tạo của mình.

Bộ dữ liệu của Adrian được tạo ra bằng cách thêm các khẩu trang giả vào những gương mặt có trong một tấm hình. Đầu tiên, Adrian thu thập các tấm ảnh có gương mặt bình thường, không bị che khuất hoặc đeo khẩu trang. Sau đó, anh ấy trích xuất các vùng có gương mặt và dùng thư viện “dlib” để tìm các vùng cụ thể trên gương mặt như mắt, mũi, miệng, cằm, khuôn mặt... để gắn vào đó một chiếc khẩu trang.

Tuy nhiên với hướng tiếp cận của Adrian, vì phần phát hiện gương mặt sử dụng mô hình có sẵn và mô hình này không được train với những gương mặt đeo khẩu trang nên nếu gương mặt bị che khuất quá nhiều bởi khẩu trang thì sẽ không phát hiện được. Trên blog của mình, Adrian cũng có nói về điều này.

TowardsDataScience - "How I built a Face Mask Detector for COVID-19 using PyTorch Lightning"

Towards Data Science là một trang blog nổi tiếng trong ngành. Các bài viết tại đây dành cho mọi đối tượng, từ những bài viết dành cho người mới bắt đầu tìm hiểu về khoa học máy tính, đến những bài viết chuyên sâu về kĩ thuật, lí thuyết phức tạp cho người nhiều kinh nghiệm.

Tại trang blog này, chúng em đã tìm hiểu về cách thiết kế một hệ thống nhận diện gương mặt đeo khẩu trang từ bài viết "How I built a Face Mask Detector for COVID-19 using PyTorch Lightning" của tác giả Jay Haddad. Hướng tiếp cận của Jay tương tự với bài viết ở trang Pyimagesearch, hệ thống gồm hai phần phát hiện và phân loại.

Về phần phát hiện, tác giả sử dụng mô hình “res10_300x300_ssd_iter_140000” được huấn luyện sẵn từ framework caffe. Về phần phân loại, tác giả đã sử dụng framework Pytorch để tự xây dựng một mô hình mạng CNN và huấn luyện mô hình này với bộ dữ liệu "Real World Masked Face Dataset" - RMFD [16]. RMFD là bộ dữ liệu dành cho bài toán nhận diện gương mặt đeo khẩu trang. Trong bộ dữ liệu này, sẽ có hình ảnh các gương mặt đeo khẩu trang cùng với các gương mặt không đeo khẩu trang và được gắn nhãn theo bộ những gương mặt của cùng một người. Tác giả không sử dụng đến thông tin về gương mặt của ai mà chỉ sử dụng thông tin về gương mặt có đeo khẩu trang hay không và dùng chúng để huấn luyện cho mô hình phân loại gương mặt đeo khẩu trang của mình.

Github

GitHub là một dịch vụ lưu trữ trên web dành cho các dự án có sử dụng hệ thống kiểm soát Git revision. Đây là một cộng đồng rất lớn của những nhà phát triển. Khi tìm kiếm chủ đề phát hiện gương mặt đeo khẩu trang tại đây, có rất nhiều kết quả trả về từ những mã nguồn mở của các công ty lớn đến những dự án nhỏ của các cá nhân. Sau khi tìm kiếm với một số từ khoá như "mask face detection", "face mask detection", "masked face detection"... Chúng em lựa chọn 2 dự án để chạy thử nghiệm và đánh giá so với mô hình của nhóm.

- [deepinsight/insightface/RetinaFaceAntiCov](#) - Insightface là một thư viện lớn, cung cấp các triển khai thực tế của những thuật toán phân tích gương mặt state-of-the-art, phát triển bởi công ty dữ liệu khoa học deepinsight. Sau khi tìm hiểu mã nguồn mở mà insightface cung cấp, chúng em nhận thấy tác giả sử dụng mô hình có khả năng phát hiện đa vật thể được phát triển dựa trên cấu trúc của mạng RetinaFace[4].
- [chandrikadeb7/Face-Mask-Detection](#) - Đây là dự án cá nhân nhưng được sự ủng hộ của nhiều người từ cộng động (280 sao). Tác giả xây dựng hệ thống gồm 2 phần, phát hiện và phân loại. Để phát hiện gương mặt, tác giả sử dụng mô hình “res10_300x300_ssd_iter_140000” được huấn luyện sẵn để phát hiện gương mặt của framework caffe. Về phân loại gương mặt, tác giả sử dụng dữ liệu riêng gồm 1916 gương mặt đeo khẩu trang và 1919 gương mặt không đeo khẩu trang để huấn luyện mô hình Mobilenetv2 thông qua thư viện Keras.

Chương 3

NỘI DUNG KHÓA LUẬN

Ở chương này chúng em xin trình bày về phần thực hiện của nhóm trong bài toán "Phát hiện mặt người đeo khẩu trang". Nội dung chương này được trình bày gồm các phần:

- Hướng tiếp cận: Trình bày hai giai đoạn nhóm đã tìm hiểu và nghiên cứu hai hướng tiếp cận cho bài toán.
- Bộ dữ liệu: Trình bày về ba giai đoạn, tìm kiếm, xây dựng, thu thập bộ dữ liệu cho quá trình nghiên cứu của nhóm.

3.1 Hướng tiếp cận

Như đã giới thiệu, đối với hướng tiếp cận bài toán, nhóm chúng em xin được chia thành hai giai đoạn như sau:

3.1.1 Giai đoạn một - Trước khi dịch COVID-19 xảy ra:

Ở giai đoạn này, nhóm xin phép được gọi là giai đoạn tìm hiểu. Để chuẩn bị cho khóa luận tốt nghiệp, nhóm chúng em đã bắt đầu tìm hiểu sớm hơn trước thời gian chính thức. Và đã chọn cho mình bài toán "Phát hiện mặt người đeo khẩu trang", từ đó, chúng em đã đặt những "viên gạch" đầu tiên trong việc nghiên cứu và tìm hiểu một bài toán khoa học mà cụ thể ở đây là bài toán ứng dụng thực tế trong lĩnh vực Thị giác Máy tính.

Bõ ngõ là những điều không thể tránh khỏi, nhóm chúng em đã gặp khá nhiều khó khăn trong giai đoạn khảo sát các bài báo và phương pháp về bài toán mà nhóm đã chọn. So với giai đoạn đại dịch COVID-19 bùng nổ, các dự án về bài toán "Phát hiện mặt người

"đeo khẩu trang" được nhiều nhóm và cá nhân thực hiện mà nhóm em có trình bày ở chương II, thì quay lại thời điểm giai đoạn một này, gần như không có một tài liệu, bài báo nào giải quyết cụ thể cho bài toán nhóm tìm hiểu. Và nhóm đã chỉ tìm được ba bài báo "*Detecting Masked Faces in the Wild with LLE-CNNs - Shiming Ge, Jia Li2, Qiting Ye, Zhao Luo*"[6], "*Adversarial Occlusion-aware Face Detection - Yujia Chen, Lingxiao Song, Ran He*"[3], "*Face Attention Network: An Effective Face Detector for the Occluded Faces*"[15] có liên quan về bài toán tổng quát hơn - Occluded Face Detection.

Từ kinh nghiệm dạy dặn thầy Mai Tiến Dũng - giảng viên hướng dẫn, cũng như thầy Ngô Đức Thành đã giúp nhóm chúng em tìm được những hướng đi đúng đắn hơn và cụ thể là hai hướng tiếp cận thử cho bài toán mà nhóm đã chọn. Hai hướng tiếp cận mà các thầy đề xuất cho nhóm em ở thời điểm khó khăn đó là mô hình có khả năng phát hiện đa vật thể - multi-object detection và hướng chia bài toán thành hai phần nhỏ hơn, dùng hai mô hình phát hiện và phân loại riêng biệt – object detection + classification. Đây cũng chính là hai hướng mà ở giai đoạn hai nhóm chúng em đã khảo sát được.

Sau khi tiếp thu đề xuất của các thầy, nhóm em còn được thầy Đỗ Văn Tiên cho tham khảo thêm một bài khóa luận "*Hệ thống phát hiện khuôn mặt bị che thông qua camera giám sát*" có liên quan đến bài toán nhóm đang nghiên cứu của anh Lý Trung Dũng mà thầy Tiên hướng dẫn trong khóa trước. Đọc và hiểu được hướng tiếp cận của anh Dũng là hướng chia nhỏ bài toán thành hai phần sử dụng hai mô hình riêng biệt cho từng phần phát hiện và phân loại - object detection + classification. Đó cũng chính là một hướng tiếp cận mà thầy Dũng và thầy Thành đã đề xuất với nhóm. Tuy nhiên vào thời điểm này, nhóm em không tìm được bài báo nào có thể chứng minh hướng tiếp cận nào phù hợp hơn với bài toán của nhóm em cũng như giới hạn về thời gian của khóa luận tốt nghiệp nên nhóm đã thống nhất tập trung theo hướng tiếp cận sử dụng mô hình phát hiện đa vật thể - multi-object detection để giải quyết bài toán của mình, đồng thời so sánh với hướng tiếp cận còn lại.

3.1.2 Giai đoạn hai - Khi COVID-19 trở thành đại dịch và lan rộng toàn cầu cho đến nay:

Bước sang giai đoạn hai, khi đại dịch COVID-19 đã bùng nổ, nhiều dự án về bài toán mà nhóm em đang làm đã được công bố. Trong đó, một số bài đã được nhóm chúng em khảo sát và nêu ra ở chương 2. Từ các dự án đó, nhóm chúng em nhận thấy rằng quả thật như đã được các thầy đề xuất, có hai hướng tiếp cận chính cho bài toán phát hiện gương mặt đeo khẩu trang.

- Hướng thứ nhất là mô hình có khả năng phát hiện đa vật thể. Với hướng tiếp cận này, ta có thể xem gương mặt đeo khẩu trang và gương mặt không đeo khẩu trang là hai vật thể tách biệt và tiến hành phát hiện hai vật thể này bằng mô hình của ta.
- Hướng thứ hai là ta sẽ chia bài toán của ta thành hai phần nhỏ, gồm phần phát hiện gương mặt và phần xác định gương mặt ấy là gương mặt đeo khẩu trang hoặc gương mặt không đeo khẩu trang. Với hướng tiếp cận này, ta sẽ dùng hai mô hình riêng biệt cho hai phần như trên.

Mỗi hướng tiếp cận sẽ có ưu điểm và nhược điểm riêng. Với hướng phát hiện đa vật thể, đây là một mô hình toàn vẹn, với đầu vào là một tấm ảnh, mô hình có thể cho ra ngay kết quả mà ta mong muốn. Ngoài ra, hướng tiếp cận này còn cho ta sự tiện lợi trong việc huấn luyện mô hình. Ta chỉ cần một bộ dữ liệu đã được gắn nhãn kết quả mong muốn thực tế và sử dụng bộ dữ liệu ấy để huấn luyện mô hình.

Tuy nhiên, việc đánh giá mô hình với hướng tiếp cận này còn chưa thực sự rõ ràng, tường minh. Mô hình phát hiện đa vật thể sẽ bao gồm hai tác vụ chính là phát hiện vật thể và phân loại vật thể. Với hướng tiếp cận như trên, ta sẽ không thể đánh giá chi tiết cụ thể cho từng tác vụ. Từ đó, ta sẽ gặp nhiều khó khăn hơn trong việc cải thiện mô hình của mình.

Ngược lại, với hướng tiếp cận thứ hai, sử dụng hai mô hình riêng cho từng tác vụ, việc đánh giá mô hình sẽ rõ ràng, tường minh hơn. Ta sẽ biết được lý do thiếu sót của mô hình ở đâu. Từ đó, ta sẽ dễ dàng điều chỉnh, cải thiện mô hình của mình hơn.

Vì theo hướng tiếp cận gồm hai mô hình cho hai tác vụ cụ thể, ta cần phải có bộ dữ liệu riêng, phù hợp cho hai mô hình khác nhau này. Với phần phát hiện gương mặt, ta cần dữ liệu các hình ảnh với nhãn là vị trí các gương mặt đeo khẩu trang và không đeo khẩu trang trong hình. Với phần phân loại gương mặt, dữ liệu cần có dạng là các hình ảnh của từng gương mặt đeo, nhãn của các hình ảnh này sẽ là gương mặt đeo khẩu trang hoặc gương mặt không đeo khẩu trang tương ứng. Cuối cùng, sau khi đã khảo sát một số dự án hiện có, nhóm em nhận thấy hướng tiếp cận mà nhóm đã chọn là phù hợp cho bài toán này.

3.2 Bộ dữ liệu

Để tiến hành thực nghiệm trong bài toán này, nhóm chúng em sử dụng một bộ dữ liệu do nhóm tự xây dựng và đặt tên là WIDERMAFA cùng một bộ dữ liệu thu thập ảnh thực tế.

3.2.1 WIDERMAFA

Bộ dữ liệu WIDERMAFA được chúng em xây dựng từ hai bộ dữ liệu lớn, WIDER FACE và MAFA. Vào thời điểm sau khi tìm ra hướng tiếp cận để thử nghiệm, một bộ dữ liệu phù hợp với hướng tiếp cận sử dụng mô hình "Phát hiện đa vật thể" mà nhóm đã chọn là rất cần thiết cho việc tiếp tục. Tuy nhiên, ở giai đoạn trước khi dịch COVID-19 bùng nổ, nhóm em đã không thể tìm được một bộ dữ liệu đặc thù nào phù hợp cho bài toán của nhóm. Song, trong quá trình tìm kiếm, nhóm đã tìm thấy bộ dữ liệu MAFA[6], một bộ dữ liệu về mặt bị che khuất, đặc biệt là che bởi khẩu trang, rất phù hợp cho bài toán của mình. Cuối cùng nhóm đã quyết định xây dựng một bộ dữ liệu từ MAFA cùng bộ dữ liệu WIDER FACE[18] để tiến hành thực nghiệm cho bài toán "Kiểm tra Chấp hành Bảo hộ Phần Mặt" nhóm đang làm. Và từ đây WIDERMAFA đã bắt đầu được xây dựng.

Chi tiết về hai bộ dữ liệu WIDER FACE và MAFA:

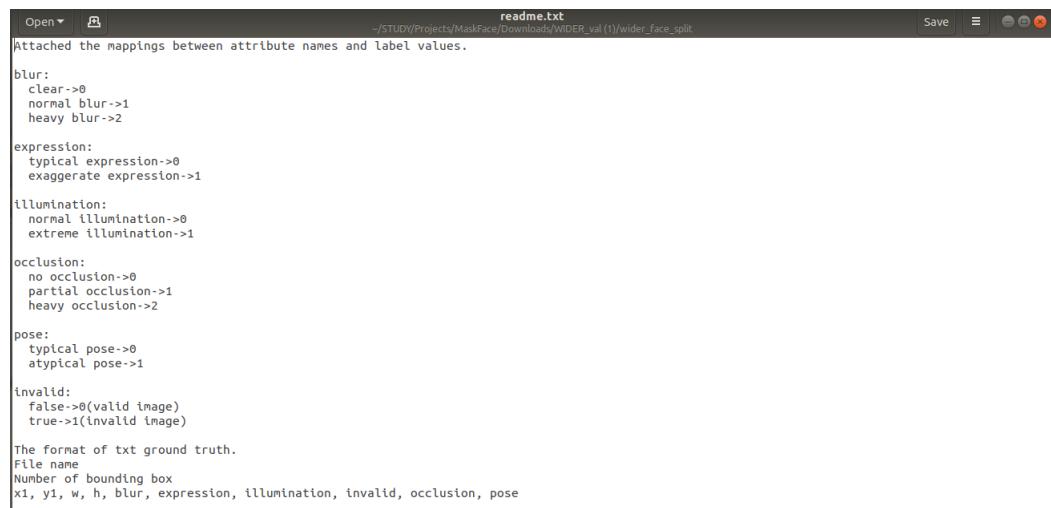
- WIDER FACE:



HÌNH 3.1: Mô tả bộ dữ liệu WIDER FACE

Đây là một bộ dữ liệu chuẩn dành cho bài toán phát hiện khuôn mặt, được biết đến rộng rãi trong lĩnh vực nghiên cứu khoa học máy tính nói chung và thị giác máy tính nói riêng.

Bộ dữ liệu có khối lượng dữ liệu khá lớn, gồm 32,203 ảnh và 393,703 nhãn cho khuôn mặt với độ phong phú cao về kích thước, tư thế, độ che khuất cũng các điều kiện khác như biểu cảm, ánh sáng,... Được tổ chức dựa trên 61 lớp sự kiện. Đối với từng lớp sự kiện, được chọn ngẫu nhiên theo tỷ lệ 40/10/50 tương ứng cho các tập huấn luyện, tập kiểm tra và tập đánh giá. Cụ thể với nhãn của mỗi gương mặt có trong ảnh sẽ được xác định bởi các yếu tố như sau:



```

Open ▾ Save readme.txt
Attached the mappings between attribute names and label values.

blur:
  clear->0
  normal blur->1
  heavy blur->2

expression:
  typical expression->0
  exaggerate expression->1

illumination:
  normal illumination->0
  extreme illumination->1

occlusion:
  no occlusion->0
  partial occlusion->1
  heavy occlusion->2

pose:
  typical pose->0
  atypical pose->1

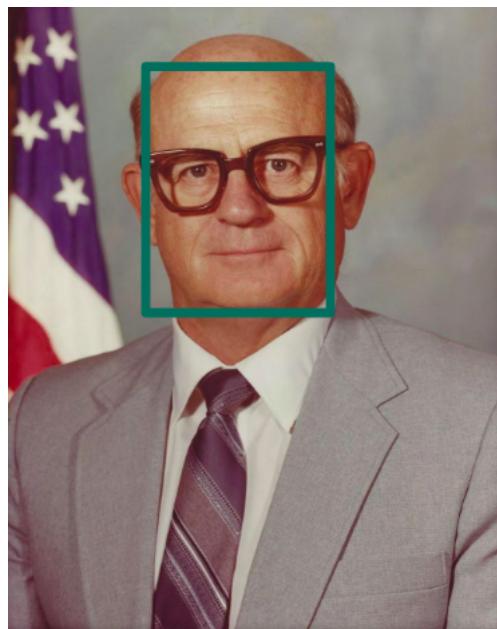
invalid:
  false->0(valid image)
  true->1(invalid image)

The format of txt ground truth.
File name
Number of bounding box
x1, y1, w, h, blur, expression, illumination, invalid, occlusion, pose

```

HÌNH 3.2: Mô tả chi tiết nhãn được chú thích trong tập tin readme.md của WIDER FACE

- "x1,y1,w,h": Đây là bốn con số nguyên, biểu diễn vị trí của gương mặt được xác định bởi tọa độ một bounding box trong bức ảnh, với x1,y1 là tọa độ của góc trái trên của bounding box và có chiều rộng là "w" điểm ảnh, chiều cao là "h" điểm ảnh.
- "blur": Mức độ gương mặt bị làm mờ, được chia thành ba mức tương ứng ba số nguyên trong tập tin ghi nhãn:
 - * "clear": ứng với "0", thể hiện gương mặt rõ ràng, không bị làm mờ.



HÌNH 3.3: Hình mẫu cho bounding box (khung viền xanh) và mặt có nhãn "blur" ở mức "clear"

- * "normal blur": ứng với "1", thể hiện gương mặt bị làm mờ ở mức thấp, hơi mờ.



HÌNH 3.4: Mặt có nhãn "blur" ở mức "normal blur"

- * "heavy blur": ứng với "2", thể hiện gương mặt bị làm mờ ở mức cao, rất mờ.



HÌNH 3.5: Mặt có nhãn "blur" ở mức "heavy blur"

- "expression": Mức độ biểu cảm của gương mặt, được chia thành 2 mức tương ứng "0" và "1" trong tập tin ghi nhãn:
 - * "typical expression": ứng với "0", thể hiện gương mặt có biểu cảm ở mức thông thường hoặc không có.



HÌNH 3.6: Mặt có nhãn "expression" ở mức "typical expression"

- * "exaggerate expression": ứng với "1", thể hiện gương mặt có cảm xúc một cách thái hóa, một cách rõ ràng.



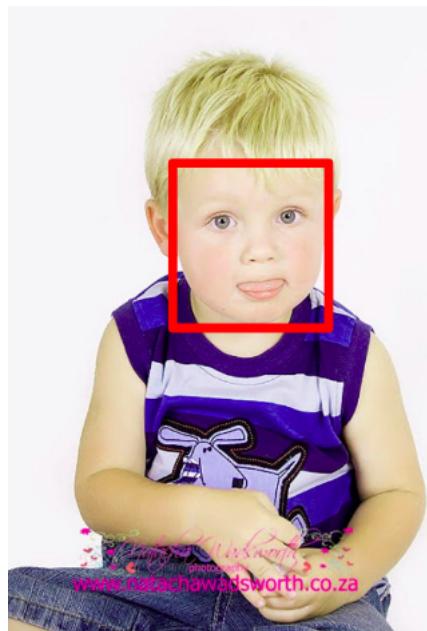
HÌNH 3.7: Mặt có nhãn "expression" ở mức "exaggerate expression"

- "illumination": Mức độ sáng hoặc bị chiếu sáng của gương mặt, được chia thành hai mức tương ứng "0" và "1" trong tập tin ghi nhãn:
 - * "normal illumination": ứng với "0", thể hiện gương mặt được chiếu sáng một cách bình thường.



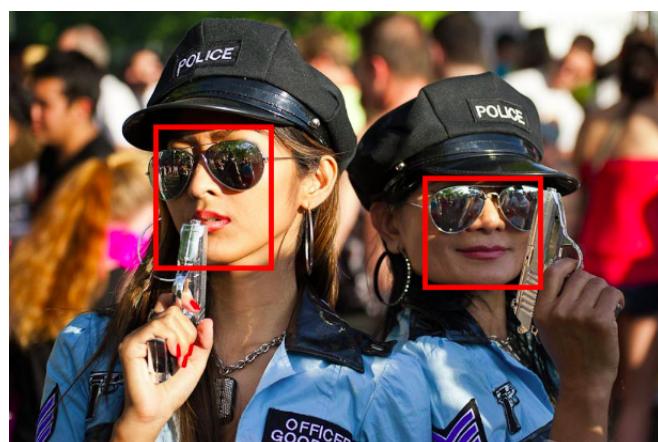
HÌNH 3.8: Mặt có nhãn "illumination" ở mức "normal illumination"

- * "extreme illumination": ứng với "1" thể hiện gương mặt bị chiếu sáng quá sáng hoặc quá tối.



HÌNH 3.9: Mặt có nhãn "illumination" ở mức "extreme illumination"

- "invalid": Đây là yếu tố để xét xem gương mặt đang được gán nhãn là có giá trị hay không tương ứng với "0" và "1".
 - * "false": ứng với "0", thể hiện gương mặt đang được gán nhãn là có giá trị, có thể nhận ra là mặt bởi con người.



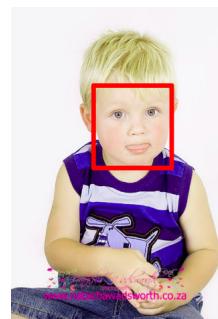
HÌNH 3.10: Mặt có nhãn "invalid" là "false"

- * "true": ứng với "1", thể hiện gương mặt đang được gán nhãn là không có giá trị, không thể nhận ra là mặt bởi chính con người, thông thường là các mặt quá nhỏ, con người chỉ có thể vô thức đoán là mặt chứ không thể nhận ra bất cứ chi tiết nào trên mặt.



HÌNH 3.11: Mặt có nhãn "invalid" là "true"

- "occlusion": Thể hiện mức độ gương mặt bị che khuất, được chia thành ba mức tương ứng với "0", "1" và "2" trong tập tin ghi nhãn:
 - * "no occlusion": ứng với "0", thể hiện gương mặt rõ ràng, không bị che khuất.



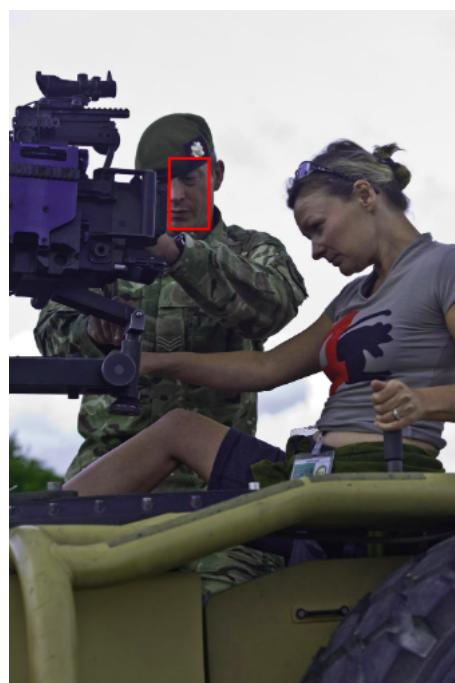
HÌNH 3.12: Mặt có nhãn "occlusion" ở mức "no occlusion"

- * "partial occlusion": ứng với "1", thể hiện gưỡng mặt bị che khuất một phần.



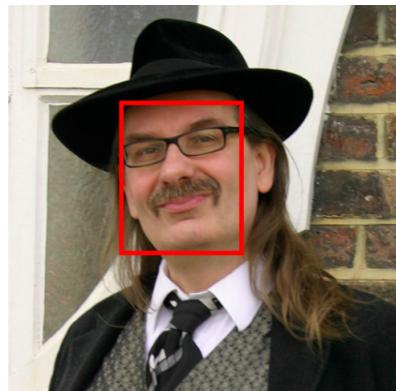
HÌNH 3.13: Mặt có nhãn "occlusion" ở mức "partial occlusion"

- * "heavy occlusion": ứng với "2", thể hiện gưỡng mặt bị che khuất rất nhiều, gần như bị che khuất hoàn toàn.



HÌNH 3.14: Mặt có nhãn "occlusion" ở mức "heavy occlusion"

- "pose": cuối cùng yếu tố này thể hiện dáng người, góc độ của gương mặt và cũng được chia thành hai loại:
 - * "typical pose": ứng với "0", thể hiện gương mặt có góc độ bình thường, trực diện.



HÌNH 3.15: Mặt có nhãn "pose" là "typical pose"

- * "atypical pose": ứng với "1", thể hiện gương mặt có một góc độ khác thường, có thể sẽ không còn thấy rõ ràng gương mặt.



HÌNH 3.16: Mặt có nhãn "pose" là "atypical pose"

- Và cuối cùng định dạng của tập tin ghi nhãn của bộ dữ liệu WIDER FACE sẽ có cấu trúc:
 - dòng 1: là một chuỗi để biểu diễn đường dẫn của bức ảnh được gán nhãn.
 - dòng 2: là một số nguyên n, thể hiện số lượng nhãn của mặt được gán trong bức ảnh "dòng 1".
 - n dòng tiếp theo: ứng với mỗi dòng trong n dòng là một chuỗi 10 chữ số nguyên dương để thể hiện vị trí và các yếu tố của từng gương mặt được gán nhãn trong ảnh theo thứ tự "x1, y1, w, h, blur, expression, illumination, invalid, occlusion, pose".

```
wider_face_train_bbx_gt.txt
~/STUDY/Projects/MaFace/Datasets/WIDER_val(1)/wider_face_split
Save | 三 | ☰ | ○
0--Parade/0_Parade_marchingband_1_849.jpg
1
449 330 122 149 0 0 0 0 0 0
0--Parade/0_Parade_marchingband_1_904.jpg
1
361 98 263 339 0 0 0 0 0 0
0--Parade/0_Parade_marchingband_1_799.jpg
21
78 221 7 8 2 0 0 0 0 0 0
78 238 14 17 2 0 0 0 0 0 0
113 212 11 15 2 0 0 0 0 0 0
134 266 15 15 2 0 0 0 0 0 0
163 250 14 17 2 0 0 0 0 0 0
201 218 10 12 2 0 0 0 0 0 0
182 266 15 17 2 0 0 0 0 0 0
245 279 18 15 2 0 0 0 0 0 0
304 265 16 17 2 0 0 0 0 2 1
328 295 16 20 2 0 0 0 0 0 0
389 281 17 19 2 0 0 0 0 2 0
406 293 21 21 2 0 1 0 0 0 0
436 290 22 17 2 0 0 0 0 0 0
522 328 21 18 2 0 1 0 0 0 0
643 320 23 22 2 0 0 0 0 0 0
653 224 17 25 2 0 0 0 0 0 0
793 337 23 30 2 0 0 0 0 0 0
535 311 16 17 2 0 0 0 0 1 0
29 220 11 15 2 0 0 0 0 0 0
3 232 11 15 2 0 0 0 0 2 0
20 215 12 16 2 0 0 0 0 2 0 |
0--Parade/0_Parade_marchingband_1_117.jpg
9
69 359 50 36 1 0 0 0 0 1
227 382 56 43 1 0 1 0 0 0 1
296 305 44 26 1 0 0 0 0 0 1
353 280 46 36 2 0 0 0 0 2 1
885 377 63 41 1 0 0 0 0 0 1
819 391 34 43 2 0 0 0 0 1 0
777 299 37 31 2 0 0 0 0 0 1 0
```

HÌNH 3.17: Tập tin ghi nhãn của WIDER FACE

• MAFA:

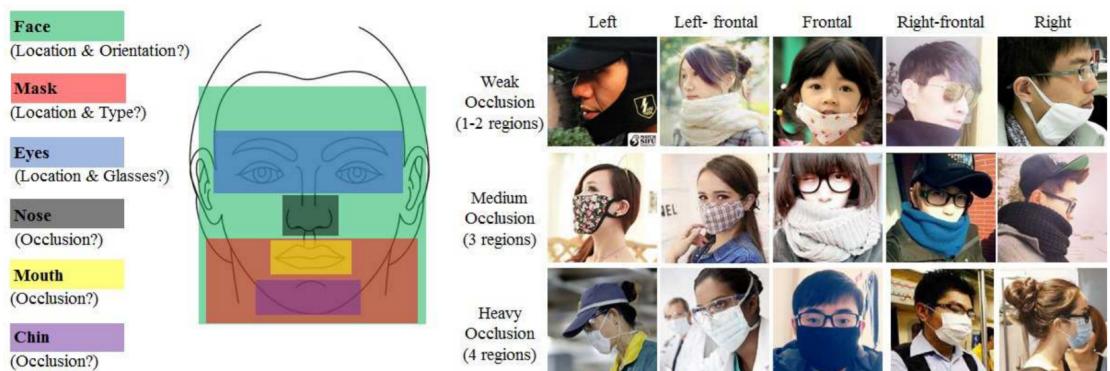
Về bộ dữ liệu MAFA[6], như đã được giới thiệu trong bài báo "*Detecting Masked Faces in the Wild with LLE-CNNs - Shim-ing Ge, Jia Li, Qiting Ye, Zhao Luo*"[6] mà nhóm đã trình bày trong chương 2. Nhóm xin trình bày tóm tắt lại về các nhãn của bộ dữ liệu này.

Nhãn của mỗi gương mặt có trong ảnh sẽ được xác định các yếu tố như sau:

- Vị trí của gương mặt nằm trong bức ảnh, vị trí này sẽ được đánh dấu bằng một bounding box hình vuông. Một số gương mặt quá mờ, hình ảnh bị biến dạng hoặc độ dài khung ảnh có kích thước nhỏ hơn 32 điểm ảnh sẽ được gán nhãn

"Ignore". Những gương mặt có nhãn này sẽ không được tính vào kết quả bài toán.

- Vị trí tâm hai mắt của gương mặt.
- Vị trí của khẩu trang hoặc kính (nếu có) của gương mặt
- Hướng của gương mặt. Hướng sẽ được chia thành 5 nhóm khác nhau: thẳng, trái, phải, trái - thẳng, phải - thẳng.
- Mức độ bị che khuất. Nhóm tác giả chia gương mặt thành 4 vùng chính, bao gồm: cầm, miệng, mũi, mắt. Số lượng vùng bị che khuất sẽ là mức độ bị che khuất của gương mặt đó.
- Loại khẩu trang. Nhóm tác giả định nghĩa bốn loại khẩu trang như sau:
 - * “Simple mask”: Loại khẩu trang chỉ có một màu đơn giản.
 - * “Complex mask”: Loại khẩu trang bao gồm nhiều màu kết hợp hoặc có gắn thêm các hình ảnh khác.
 - * “Human body”: Gương mặt bị che khuất bởi các bộ phận con người như tay, tóc...
 - * “Hybrid mask”: Gương mặt bị che khuất bởi cả hai trong các nhóm nói trên hoặc một nhóm trên và có đeo kính.



HÌNH 3.18: Mô tả bộ dữ liệu MAFA

Thông qua các chi tiết trên, nhóm em đã nhận thấy được sự đa dạng của từng bộ dữ liệu cũng như sự phong phú về số lượng của chúng rất phù hợp cho bộ dữ liệu dùng trong bài toán mà nhóm đang nghiên cứu. Và đó là lý do nhóm em đã quyết định chọn WIDER

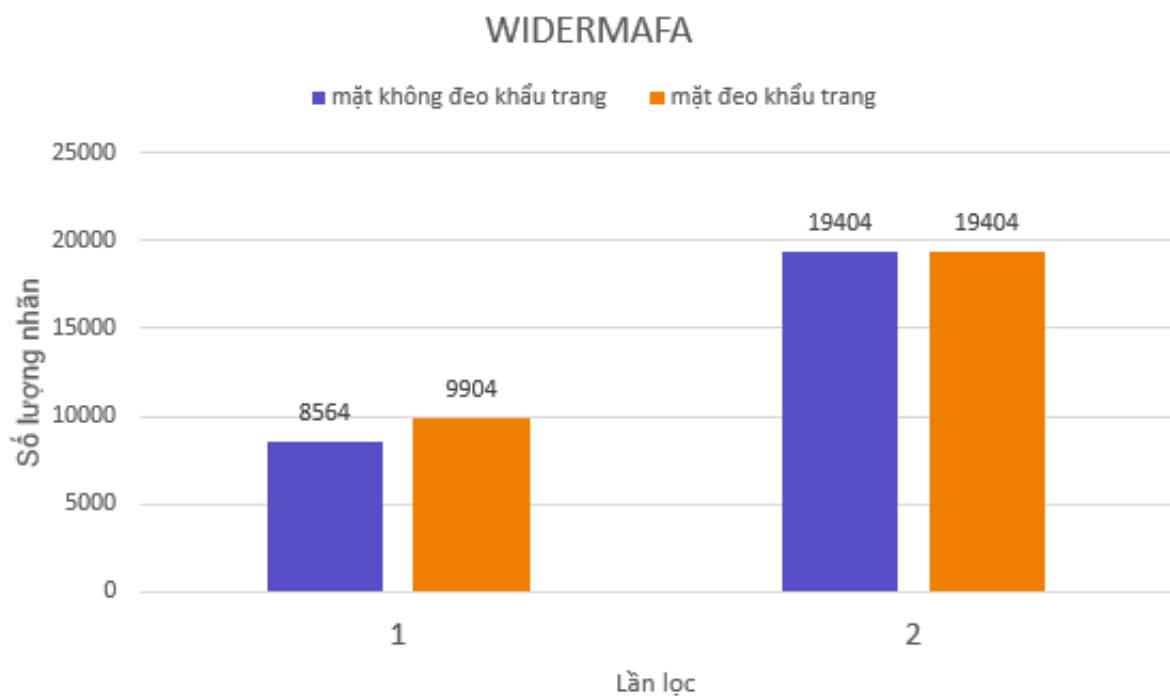
FACE và MAFA để xây dựng bộ dữ liệu WIDERMAFA.

Hướng tiếp cận mà nhóm đã chọn là sử dụng mô hình "phát hiện đa vật thể" nên nhóm cần xây dựng bộ dữ liệu cho hai lớp, lớp "mặt không đeo khẩu trang" và lớp "mặt đeo khẩu trang". Bộ dữ liệu WIDER FACE và MAFA thực sự phù hợp tương ứng cho hai lớp đó. Tiến hành kiểm tra hình ảnh trong các bộ dữ liệu dựa trên các yếu tố đã được trình bày, nhóm chúng em cảm thấy mình cần lọc lại để phù hợp cụ thể cho bài toán của nhóm. Tổng cộng bộ dữ liệu WIDERMAFA mà nhóm em xây dựng đã trải qua hai lần chọn lọc.

- Ban đầu, nhóm đã chọn lọc dữ liệu đơn giản với một vài điều kiện:
 - WIDER FACE: số lượng nhãn trong một ảnh (từ 5 nhãn trở xuống), blur (clear), expression (typical expression), illumination (normal illumination), invalid (false), occlusion (no occlusion), pose (typical pose).
 - MAFA: occ_type (simple), occ_degree(3), orientation (frontal) và không quan tâm những điều kiện còn lại.
- Sau khi lọc với những điều kiện trên, nhóm em thu được 5914 ảnh với 8546 nhãn từ bộ dữ liệu WIDER FACE (tập huấn luyện và tập kiểm tra) cho lớp "mặt không đeo khẩu trang" và ... ảnh với ... nhãn từ bộ dữ liệu MAFA (tập huấn luyện) cho lớp "mặt đeo khẩu trang".
- Tuy nhiên sau khi kiểm tra lại các điều kiện khác của các bộ dữ liệu, chúng em cảm thấy có thể tăng số lượng hình ảnh lên với một vài điều kiện đã bỏ qua nhưng vẫn phù hợp cho bài toán. Nhóm tiến hành chọn lọc lại dữ liệu một cách kĩ càng với một vài điều kiện bổ sung:
 - WIDER FACE: giữ nguyên các điều kiện cũ, tăng số lượng nhãn trong một ảnh (từ 13 nhãn trở xuống) với diện tích mỗi nhãn phải chiếm 0.3% diện tích tấm ảnh và bổ sung cá điều kiện sau, blur (normal blur), expression (exaggerate expression), illumination (extreme illumination), occlusion (partial occlusion).
 - MAFA: giữ nguyên các điều kiện cũ, bổ sung một vài điều kiện sau occ_type (complex), orientation (left-frontal và right-frontal).
- Kết thúc việc chọn lọc lần thứ hai, số lượng ảnh thu được tăng đáng kể, từ 5914 ảnh với 8564 nhãn tăng thành 7946 ảnh với 19404 nhãn cho lớp "mặt không đeo khẩu

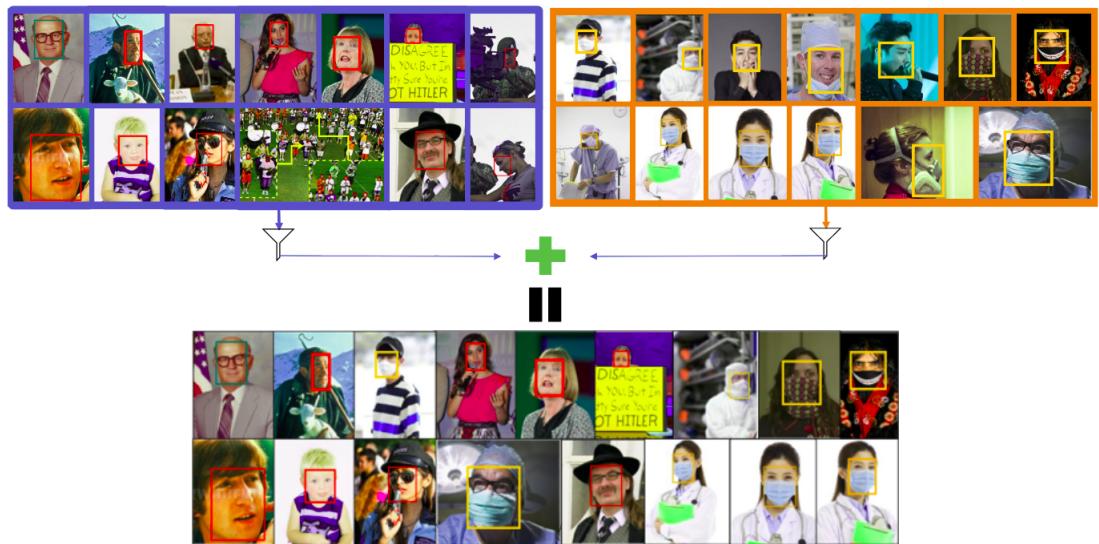
trang" và từ 9139 ảnh với 9940 tăng lên thành 17,169 ảnh với 19404 nhãn cho lớp "mặt có đeo khẩu trang".

- Sau cùng, nhóm em đã hoàn thiện được bộ dữ liệu WIDERMAFA mong muốn, để sử dụng cho việc thực nghiệm trong bài toán. Bộ dữ liệu có tổng cộng 26904 ảnh với 19404 nhãn cho mỗi lớp.



HÌNH 3.19: Số lượng nhãn của WIDERMAFA ở hai lận chọn lọc

- Những tiêu chí mà chúng em đã dùng để chọn lọc, xây dựng bộ dữ liệu MAFA được đưa ra trước khi thực nghiệm nên vẫn đảm bảo được tính khách quan. Mô hình minh họa cho việc lọc và xây dựng WIDERMAFA như sau.



HÌNH 3.20: Mô hình minh họa việc lọc và xây dựng WIDERMAFA.

3.2.2 Bộ dữ liệu thu thập ảnh thực tế

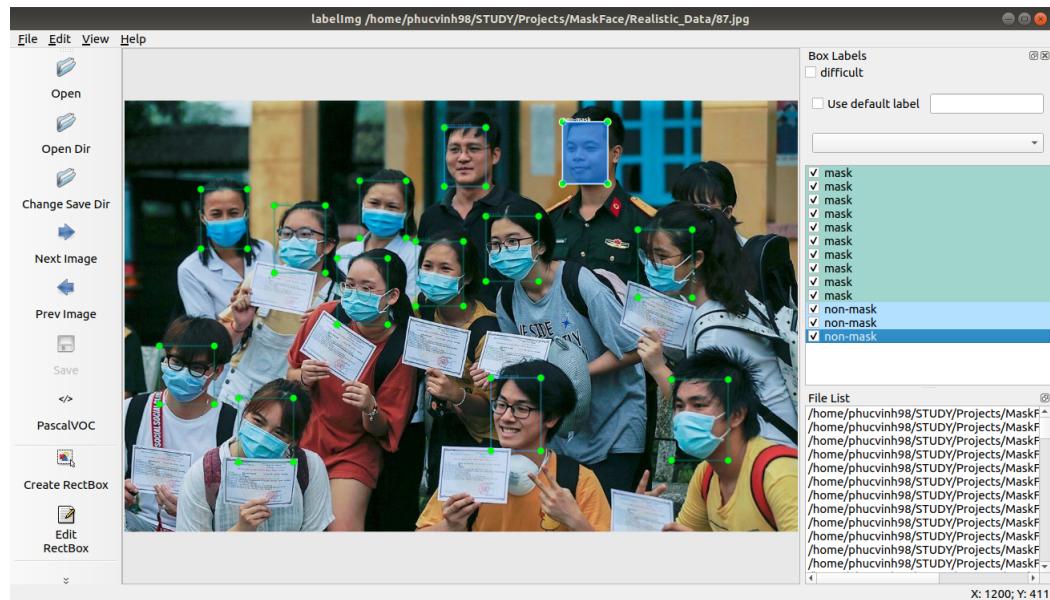
Ở phần này, nhóm chúng em muốn giới thiệu về bộ dữ liệu thực tế mà chúng em đã tự thu thập thông qua các kênh tin tức trên mạng như các trang báo Tuổi Trẻ¹, báo Thanh Niên², báo VNEXPRESS³,... và các nguồn khác được ghi rõ trong file excel online (shorturl.at/dklo7). Bộ dữ liệu này bao gồm 930 ảnh với 2914 nhãn "mặt đeo khẩu trang" và 608 nhãn "mặt không đeo khẩu trang", nhãn này cũng được chính nhóm chúng em gán thủ công bằng công cụ labelImg⁴. Dưới đây là một số hình ảnh về công cụ labelImg và hình ảnh nhóm thu thập cho bộ dữ liệu này.

¹<https://tuoitre.vn/>

²<https://thanhnien.vn/>

³<https://vnexpress.net/>

⁴<https://github.com/tzutalin/labelImg>



HÌNH 3.21: Gán nhãn với công cụ labelImg.



HÌNH 3.22: Gán nhãn với công cụ labelImg.



HÌNH 3.23: Một số hình ảnh trong bộ dữ liệu nhóm tự thu thập.

Trong quá trình so sánh và đánh giá với các bài đã được thực hiện bởi cộng đồng hiện nay, nhóm chúng em đã gặp khó khăn ở phần dữ liệu đánh giá được lấy ra từ bộ WIDER-MAFA mà nhóm đã xây dựng. Việc không thể kiểm soát được sự trùng lặp giữa dữ liệu đánh giá của nhóm với dữ liệu huấn luyện mà các bài hiện có dùng, đã thúc đẩy nhóm em xây dựng một bộ dữ liệu khách quan được giới thiệu ở trên. Hơn nữa đây cũng là bộ dữ liệu giúp chúng em kiểm chứng được tính thực tế sát ngữ cảnh nhóm hướng tới hơn so với bộ dữ liệu đánh giá trích ra từ WIDERMAFA.

Bộ dữ liệu thực tế này đã được tổng hợp từ các hình ảnh mà nhóm thu thập trên các trang báo mạng và thông qua các công cụ tìm kiếm như Google, Bing, Facebook Search,... Cụ thể quá trình nhóm em xây dựng bộ dữ liệu này như sau:

- Đầu tiên về thời gian, vì khó khăn này xảy ra vào tháng cuối trước khi báo cáo nên nhóm đã tiến hành thu thập các dữ liệu thực tế và cả việc gán nhãn chỉ trong vòng một tuần sau khi bàn bạc cùng với thầy Mai Tiến Dũng. Việc thu thập diễn ra trong vòng năm ngày đầu và hai ngày còn lại nhóm em dành trọn cho việc gán nhãn.
- Tiến hành phần thu thập dữ liệu, nhóm em đã bắt đầu bằng việc tìm kiếm trên các công cụ tìm kiếm với từ khóa "tin tức ở Việt Nam" với mong muốn tìm ra những bài

báo có chứa hình ảnh phù hợp. Với từ khóa trên nhóm đã tìm ra được khá nhiều trang báo từ những trang lớn gồm có báo Thanh Niên, báo Tuổi trẻ, báo VNEXPRESS, Zing News⁵ và không thể thiếu trang tin tức của Bộ Y Tế về tình hình dịch bệnh COVID-19⁶, ... cùng nhiều trang báo khác như Pháp luật Tuổi trẻ Thủ Đô⁷, trang tin tức Kênh 14⁸, trang tin tức 24h⁹, ...

- Tiếp nối việc tìm kiếm nguồn tin báo, nhóm em truy cập sâu hơn vào chuyên mục "dịch COVID-19" ở vài trang báo lớn.



HÌNH 3.24: Ba trong số các trang báo lớn nhóm thu thập hình ảnh.

Ở chuyên mục này chúng em tổng hợp và chọn lọc các từ khóa mà những bài báo thường dùng cho chuyên mục COVID-19. Một vài từ khóa mà nhóm đã chọn để tìm kiếm ở những trang báo khác như #Dịch COVID-19, #Deo Khẩu trang, #Virus Corona. Sau đó, lướt tìm ở vài trang báo trong số các trang đã được tổng hợp trước

⁵<https://zingnews.vn/>

⁶<https://ncov.moh.gov.vn/>

⁷<https://phapluat.tuoitrethudo.com.vn/>

⁸<https://kenh14.vn/>

⁹<https://24h.com.vn/>

đó, nhóm chúng em tìm ra thêm vài trang báo ngoài nước như The Wall Street Journal¹⁰, Yahoo! News¹¹, The Star¹².



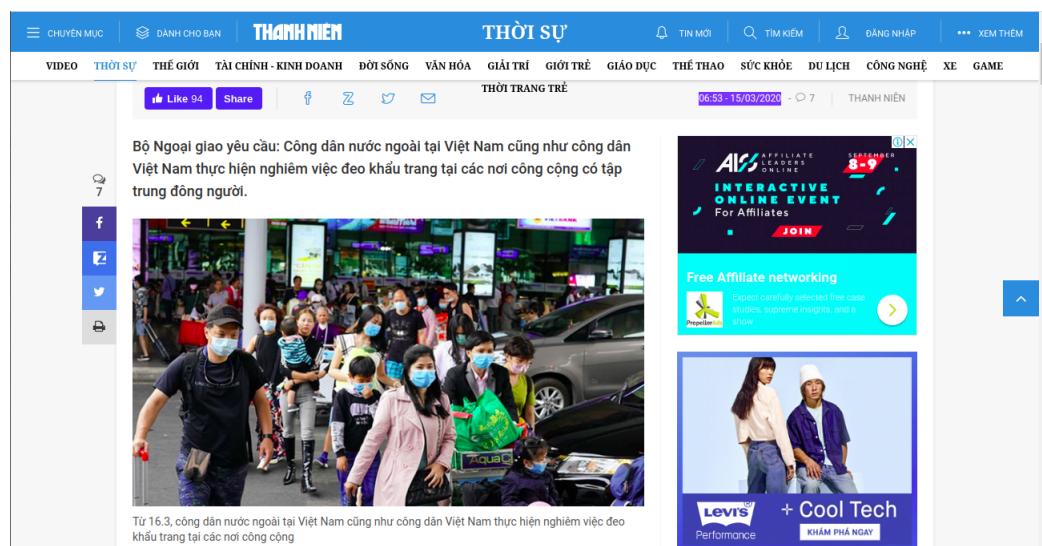
HÌNH 3.25: Các từ khóa được dùng ở các trang báo.

- Truy cập và tìm kiếm ở hơn 400 bài báo từ các trang báo khác nhau đã được tổng hợp trước đó, chúng em thu thập những hình ảnh có chứa mặt người. Để đảm bảo tính khách quan của dữ liệu, nhóm em lấy cả mặt có đeo khẩu trang và mặt không đeo khẩu trang, tất nhiên là cũng không có trong dữ liệu WIDERMAFA. Dữ liệu được thu thập thì đa dạng về loại mặt, kích thước ảnh, kích thước gương mặt, số lượng mặt, điều kiện sáng,... và yếu tố quốc tịch theo một thứ tự ưu tiên giảm dần từ Việt Nam, Châu Á, Châu Âu, Châu Mỹ và các dân tộc khác. Mọi hình ảnh chúng em thu thập cũng đều được lưu trữ nguồn của bài báo và thời gian đăng của bài trong một tập tin excel online [Hình ??]. Sau tròn năm ngày thu thập, nhóm đã có cho mình một bộ 930 hình ảnh (chưa có nhãn). Và tiếp tục hoàn tất việc gán 3522 nhãn trong hai ngày cuối cùng.

¹⁰<https://wsj.com/>

¹¹<https://news.yahoo.com/>

¹²<https://www.thestar.com.my/>



HÌNH 3.26: Một số ví dụ về các bài báo mà nhóm thu thập ảnh.



HÌNH 3.27: Một số ví dụ về các bài báo mà nhóm thu thập ảnh.

Như vậy, khó khăn do việc không thể kiểm soát được sự trùng lặp giữa dữ liệu đánh giá của nhóm với dữ liệu huấn luyện mà các bài hiện nay dùng đã có thể được giải quyết, cũng như mong muốn kiểm chứng tính ứng dụng của mô hình mà nhóm chọn đã có thể thực hiện nhờ bộ dữ liệu thực tế này.

3.3 Huấn luyện mô hình phát hiện vật thể

3.3.1 Một số mô hình CNN

Mobilenetv1 [7]

Mô hình Mobilenet lần đầu tiên được nhóm tác giả tại Google công bố tại arXiv vào năm 2017. Lúc bấy giờ, xu hướng nghiên cứu về các mạng nơ-ron nhân tạo là mô hình sâu và nhiều biến số, từ đó có thể đạt được độ chính xác cho mô hình cao hơn. Tuy nhiên, mô hình học sâu sẽ dẫn đến hệ quả tốc độ kém, điều này rất ảnh hưởng đến tính ứng dụng thực tế. Đi ngược lại với số đông, nhóm tác giả đã cho ra mắt mô hình Mobilenet với kiến trúc nhỏ gọn, ít biến số tuy nhiên vẫn đạt độ chính xác chấp nhận được.

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

HÌNH 3.28: Cấu trúc mô hình mạng Mobilenetv1

Với kỹ thuật xử lí ở các lớp Convolution, thay vì sử dụng các kernel có kích thước $M \times M \times D$, Mobilenet sử dụng các kernel có kích thước $M \times M \times 1$ và $1 \times 1 \times D$ để giảm số

lượng thực hiện phép nhân, chi phí tính toán cho máy tính, từ đó có thể làm tăng tốc độ của mô hình. Kỹ thuật này được gọi là "depthwise separable convolutions".

Mobilenetv2 [14]

Một năm sau kể từ khi công bố Mobilenet, nhóm tác giả từ Google tiếp tục công bố bài báo về Mobilenetv2 tại hội nghị CVPR2018. Cùng với kỹ thuật "depthwise separable convolutions", nhóm tác giả đã thêm hai tính chất mới vào kiến trúc mạng của mình, đó là "linear bottlenecks between the layers" và "shortcut connections between the bottlenecks". Bằng các kỹ thuật trên, nhóm tác giả công bố rằng số lượng biến số và tốc độ tính toán ở Mobilenetv2 ít hơn và nhanh hơn cả Mobilenet nhưng không ảnh hưởng quá nhiều đến độ chính xác.

Input	Operator	<i>t</i>	<i>c</i>	<i>n</i>	<i>s</i>
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	<i>k</i>	-	

HÌNH 3.29: Cấu trúc mô hình mạng Mobilenetv2

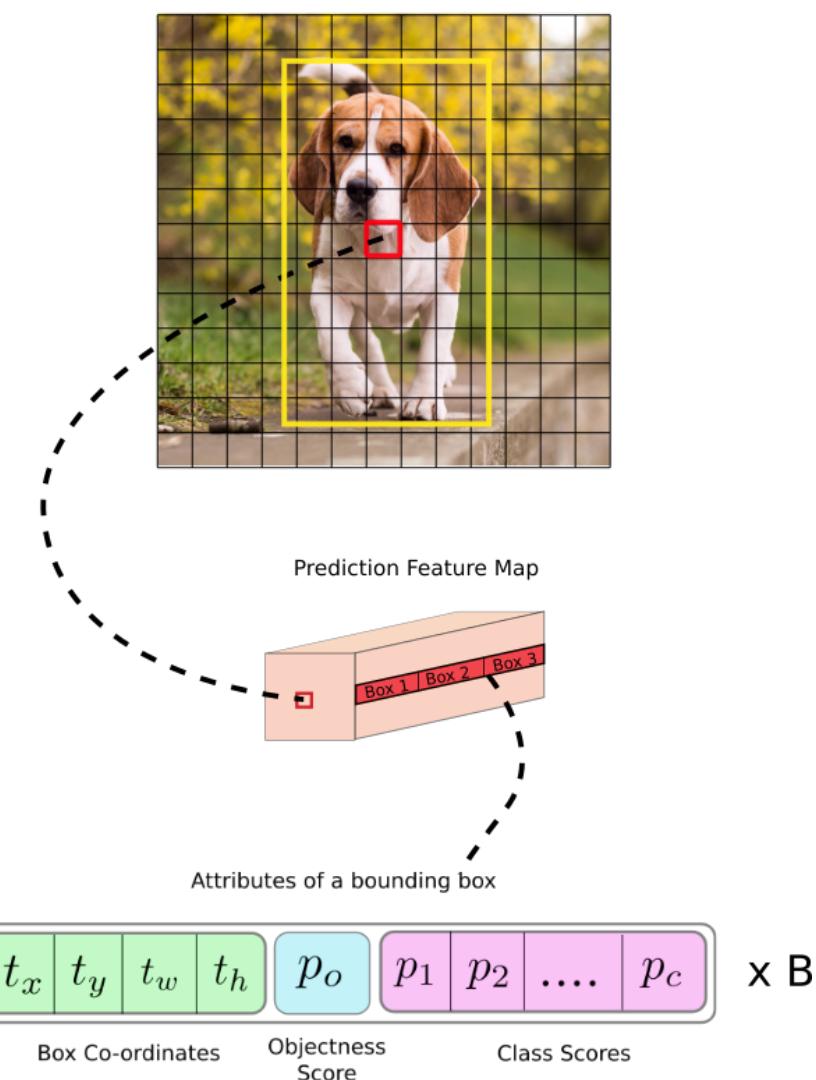
YOLOv3[13]

YOLOv3 được nhóm tác giả đến từ đại học Washington công bố tại arXiv năm 2018. YOLOv3 nổi tiếng với độ chính xác cao khi thực hiện trên các benchmark nổi tiếng như VOC 2007[5] và COCO test-dev[11]. YOLOv3 sẽ chia ảnh đầu vào thành các grid-cell, mỗi grid-cell sẽ cho đầu ra là một vector có các thông tin về việc phát hiện vật thể ở grid-cell đó. Độ dài của vector này sẽ bằng $B(5+C)$. B là số lượng bounding box mà mô hình có khả năng phát hiện trong một grid-cell. 5 là số lượng thông tin về bounding box, trong đó sẽ có 4 thông tin về vị trí bounding box và confidence score cho bounding box đó. C là số lượng vật thể mà mô hình có khả năng phát hiện, tương đương với C chỉ số khả

năng tồn tại của vật thể tương ứng đối với grid-cell.

Hình 3.30 thể hiện cách chia grid-cell 13x13 với $B=3$. Điều này tương đương với đầu ra của mô hình sẽ có kích thước $13 \times 13 \times 24$ nếu mô hình có khả năng phát hiện 3 vật thể ứng với một grid-cell ($C=3$).

Image Grid. The Red Grid is responsible for detecting the dog



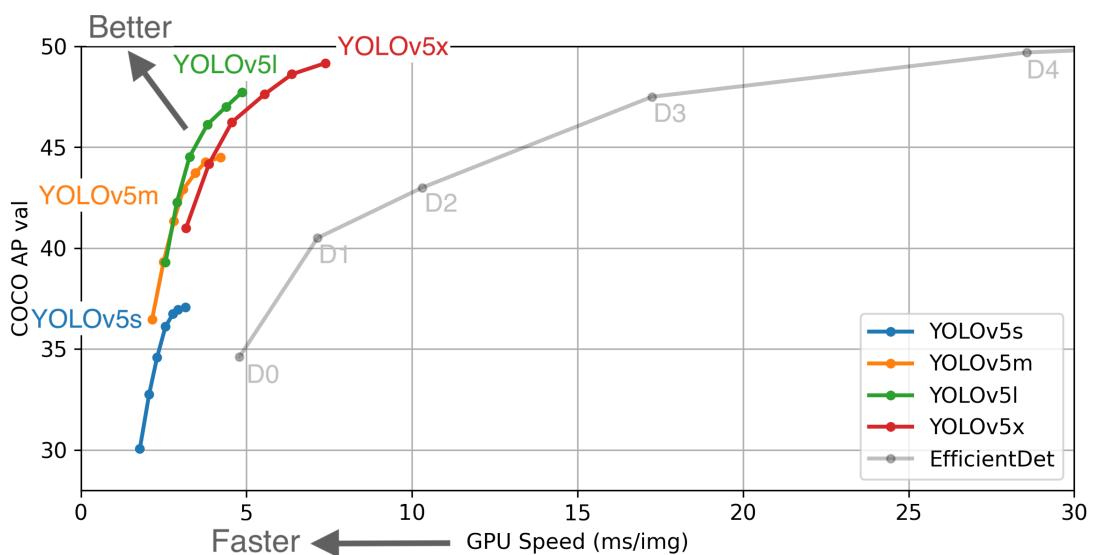
HÌNH 3.30: Mô tả đầu ra của YOLOv3

Ngoài ra, YOLOv3 còn định nghĩa các anchor box với những tỉ lệ cho trước. Cụ

thể, nếu huấn luyện mô hình detect xe hơi, các anchor box sẽ được định nghĩa với những tỉ lệ như hình chữ nhật nằm ngang. YOLOv3 sẽ phát hiện những vật thể với tỉ lệ này tốt hơn.

YOLOv5[9]

Tháng 6 năm 2020, tác giả Glenn Jocher công bố một mô hình mới với tên gọi YOLOv5. Lựa chọn tên gọi này gây nên nhiều tranh cãi trong cộng đồng bởi Glenn Jocher không phải là tác giả của các phiên bản mô hình YOLO trước đó. Mặc dù chưa có bài báo chính thức nào về mô hình YOLOv5[9] và có những tranh cãi về tên gọi nhưng độ chính xác của của mô hình này là không thể phủ định. Tác giả cung cấp nhiều phiên bản của mô hình này với số lượng lớp trong mạng và số lượng trọng số khác nhau trên framework Pytorch tại repo github: <https://github.com/ultralytics/yolov5>.



HÌNH 3.31: Kết quả đánh giá các phiên bản của YOLOv5 (công bố tại link
github của tác giả)

3.3.2 Tensorflow Object Detection API

Như đã nói ở phần hướng tiếp cận, chúng em chọn theo hướng phát hiện đa vật thể. Để làm được điều này, chúng em sử dụng framework do Tensorflow cung cấp. Framework này hỗ trợ xây dựng và huấn luyện các mô hình mạng học sâu nhằm giải quyết bài toán phát

hiện vật thể. Tensorflow Object Detection API cung cấp một số mô hình đã được huấn luyện trên các bộ dữ liệu như COCO[11], KITTI [1], Open Images[10]... hỗ trợ người dùng huấn luyện dựa trên các mô hình đã được huấn luyện trước đó hoặc người dùng cũng có thể tự huấn luyện một mô hình mới mà không sử dụng đến các mô hình này.

3.3.3 Huấn luyện mô hình phát hiện gương mặt đeo khẩu trang

Sau khi tham khảo một số kiến trúc mô hình học sâu cùng với sự tìm hiểu về Tensorflow, chúng em quyết định chọn mô hình Mobilenetv1[7] và Mobilenetv2[14] để huấn luyện cho bài toán chúng em đang làm. Ngoài ra, sau khi nghiên cứu chúng em quyết định huấn luyện thêm mô hình YOLOv3 [13] để so sánh vì độ chính xác nổi bật mà mô hình này đã thể hiện ở các benchmark lớn.

- Với Mobilenetv1[7] và Mobilenetv2 [14], chúng em áp dụng kĩ thuật transfer learning, sử dụng mô hình tương ứng đã được huấn luyện trên bộ dữ liệu COCO[11] mà framework Tensorflow Object Detection API cung cấp để tiếp tục huấn luyện trên hai bộ dữ liệu mà chúng em đã thực hiện nhằm tận dụng các đặc trưng đã được học từ bộ dữ liệu COCO[11].

Cụ thể hơn, với cả hai mô hình này, chúng em sẽ đặt learning rate ban đầu ở mức 0.0005 và sẽ đặt decay factor ở mức 0.95 sau khi thực hiện được 2000 bước. Điều này có ý nghĩa rằng, cứ sau mỗi 2000 bước thì learning rate sẽ giảm còn bằng với 0.95 mức trước đó. Khi bắt đầu huấn luyện, giá trị learning rate sẽ là 0.0005, sau 2000 bước đầu learning rate sẽ có giá trị 0.000475, sau 2000 bước tiếp theo, learning rate sẽ có giá trị 0.00045125. Và chúng em đã huấn luyện mô hình Mobilenetv1[7] với tổng cộng là 241400 bước, mô hình Mobilenetv2[14] với tổng cộng là 168020 bước.

- Với YOLOv3[13], nhóm chúng em sử dụng code từ "ultralytics", một nguồn trên github¹³ cùng với tài nguyên của Google Colaboratory¹⁴ để huấn luyện mô hình YOLOv3. Tương tự hai mô hình trước, nhóm em tiếp tục áp dụng kĩ thuật transfer

¹³<https://github.com/ultralytics/yolov3>

¹⁴<https://colab.research.google.com/>

learning, sử dụng mô hình YOLOv3 đã được huấn luyện trên bộ dữ liệu COCO[11] để train mô hình với tập dữ liệu WIDERMAFA mà nhóm xây dựng.

Ở nguồn code này, tác giả cho phép điều chỉnh khá nhiều siêu tham số như "GIoU loss", "Classification loss", "ngưỡng IoU trong huấn luyện", "chỉ số học ban đầu và cuối cùng" và một vài tham số khác cùng các tham số điều chỉnh cho quá trình tăng dữ liệu trong huấn luyện.

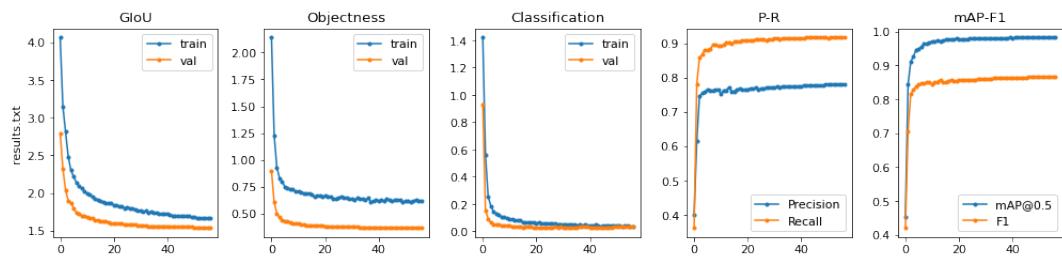
```
# Hyperparameters
hyp = {
    'giou': 3.54, # giou loss gain
    'cls': 37.4, # cls loss gain
    'cls_pw': 1.0, # cls BCELoss positive weight
    'obj': 64.3, # obj loss gain (*=img_size/320 if img_size != 320)
    'obj_pw': 1.0, # obj BCELoss positive weight
    'iou_t': 0.20, # iou training threshold
    'lr0': 0.01, # initial learning rate (SGD=5E-3, Adam=5E-4)
    'lrf': 0.0005, # final learning rate (with cos scheduler)
    'momentum': 0.937, # SGD momentum
    'weight_decay': 0.0005, # optimizer weight decay
    'fl_gamma': 0.0, # focal loss gamma (efficientDet default is gamma=1.5)
    'hsv_h': 0.0138, # image HSV-Hue augmentation (fraction)
    'hsv_s': 0.678, # image HSV-Saturation augmentation (fraction)
    'hsv_v': 0.36, # image HSV-Value augmentation (fraction)
    'degrees': 1.98 * 0, # image rotation (+/- deg)
    'translate': 0.05 * 0, # image translation (+/- fraction)
    'scale': 0.05 * 0, # image scale (+/- gain)
    'shear': 0.641 * 0} # image shear (+/- deg)
```

HÌNH 3.32: Các siêu tham số có thể điều chỉnh trong nguồn code trên

Để tránh làm mất mát những đặc trưng mà mô hình được huấn luyện trên bộ COCO[11], nhóm em vẫn chọn một learning rate ban đầu ở mức nhỏ là 0.0005 và vì đặc thù của mô hình YOLOv3 cùng nghiên cứu của tác giả về chiến lược hiệu chỉnh learning rate nên nhóm ở mô hình này nhóm chỉ số học của mô hình này được thay đổi theo một hàm cosine[8] mà tác giả đã áp dụng thử nghiệm.

$$\eta_t = \frac{1}{2}(1 + \cos(\frac{t\pi}{T}))\eta$$

Và nhóm đã tiến hành huấn luyện mô hình này trong 56 epochs, mất gần 21 tiếng. Ngoài ra, nhóm cũng đã điều chỉnh và thử nghiệm với một số tham số, siêu tham số khác như "batch size", "augmentation", "optimizer",...



HÌNH 3.33: Biểu đồ loss trong quá trình huấn luyện mô hình YOLOv3 qua 56 epochs

- Đối với mô hình YOLOv5[9], chúng em lấy source code từ repository¹⁵ mà tác giả công bố. Trong repository trên, tác giả cung cấp 5 phiên bản của mô hình YOLOv5 tương đương với số lớp và tham số khác nhau.

Model	AP ^{val}	AP ^{test}	AP ₅₀	Speed _{GPU}	FPS _{GPU}	params	FLOPS
YOLOv5s	37.0	37.0	56.2	2.4ms	416	7.5M	13.2B
YOLOv5m	44.3	44.3	63.2	3.4ms	294	21.8M	39.4B
YOLOv5l	47.7	47.7	66.5	4.4ms	227	47.8M	88.1B
YOLOv5x	49.2	49.2	67.7	6.9ms	145	89.0M	166.4B
YOLOv5x + TTA	50.8	50.8	68.9	25.5ms	39	89.0M	354.3B

HÌNH 3.34: Các phiên bản YOLOv5 mà tác giả cung cấp

Nhóm chúng em quyết định lựa chọn phiên bản YOLOv5m vì có số lượng tham số phù hợp với phần cứng chúng em sử dụng để huấn luyện. Sau khi đọc hướng dẫn của tác giả, chúng em đã huấn luyện thành công mô hình YOLOv5m với phần dữ liệu huấn luyện của bộ dữ liệu WIDERMAFA trong 25 epoch với Optimizer Adam.

¹⁵<https://github.com/ultralytics/yolov5>

Chương 4

THỰC NGHIỆM VÀ ĐÁNH GIÁ

Để có thể đánh giá kết quả các mô hình mà nhóm huấn luyện cũng như để hiểu rõ sản phẩm của nhóm hơn, chúng em đã tiến so sánh với các dự án của cộng đồng trong thời gian gần đây. Cụ thể, chúng em so sánh với các mô hình mà đã được giới thiệu ở phần 2.3.4:

- SCI: <https://github.com/chandrikadeb7/Face-Mask-Detection>
- Insightface: <https://github.com/deepinsight/insightface/tree/master/RetinaFaceAntiCov>
- Pyimagesearch: <https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/>
- Cov: <https://github.com/JadHADDAD92/covid-mask-detector>

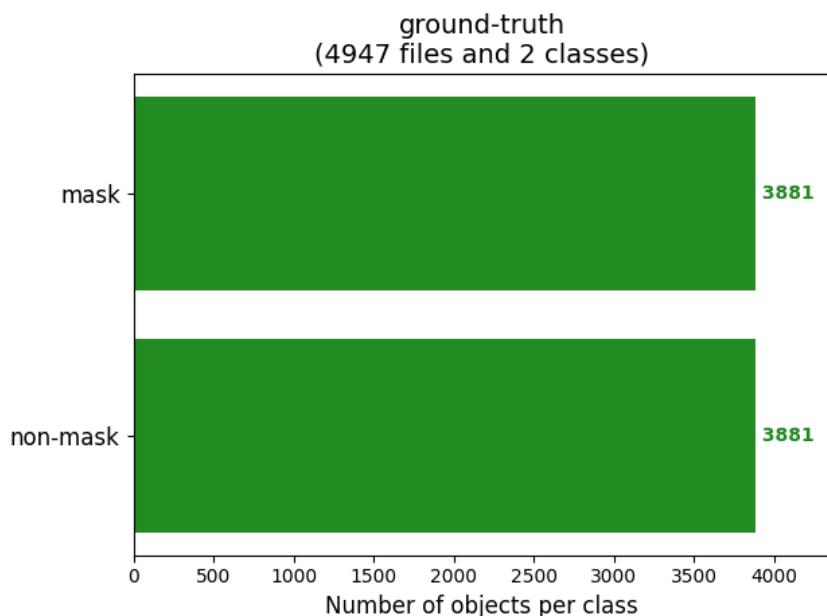
4.1 Đánh giá trên bộ dữ liệu WIDERMAFA

Sau khi đã tự tạo nên được mô hình phát hiện gương mặt đeo khẩu trang với mạng Mobilenetv1, Mobilenetv2 và YOLOv3, YOLOv5 chúng em tiến hành kiểm tra, đánh giá những mô hình này với các dự án, hướng làm khác mà chúng em đã khảo sát được. Cụ thể, chúng em đánh giá về một số độ đo như precision, recall, mAP. Các độ đo này được tính với mức ngưỡng Confidence Score là 0.5.

WIDERMAFA là tên gọi tạm chúng em đặt cho bộ dữ liệu mà chúng em đã tiến hành chọn lọc từ hai bộ dữ liệu lớn là WIDERFACE và MAFA như đã nói ở phần 3.2.1. Một số thông tin cơ bản về phần dữ liệu test trong bộ dữ liệu này:

- Tổng cộng có 4947 ảnh.

- Số lượng bounding box gương mặt đeo khẩu trang là 3881.
- Số lượng bounding box gương mặt không đeo khẩu trang là 3881.



HÌNH 4.1: Thông số bộ test của WIDERMAFA

Model	Precision		Recall	
	Mask	Non-mask	Mask	Non-mask
Mobilenetv1	0.95	0.85	0.71	0.38
Mobilenetv2	0.92	0.54	0.4	0.07
YOLOv3	0.97	0.86	0.63	0.74
YOLOv5	0.96	0.83	0.9	0.88
SCI	0.71	0.64	0.54	0.62
Insightface	0.76	0.53	0.85	0.88
Pyimagesearch	0.74	0.68	0.56	0.66
Cov	0.75	0.48	0.3	0.39

BẢNG 4.1: Kết quả precision-recall của các mô hình trên bộ test của WIDERMAFA

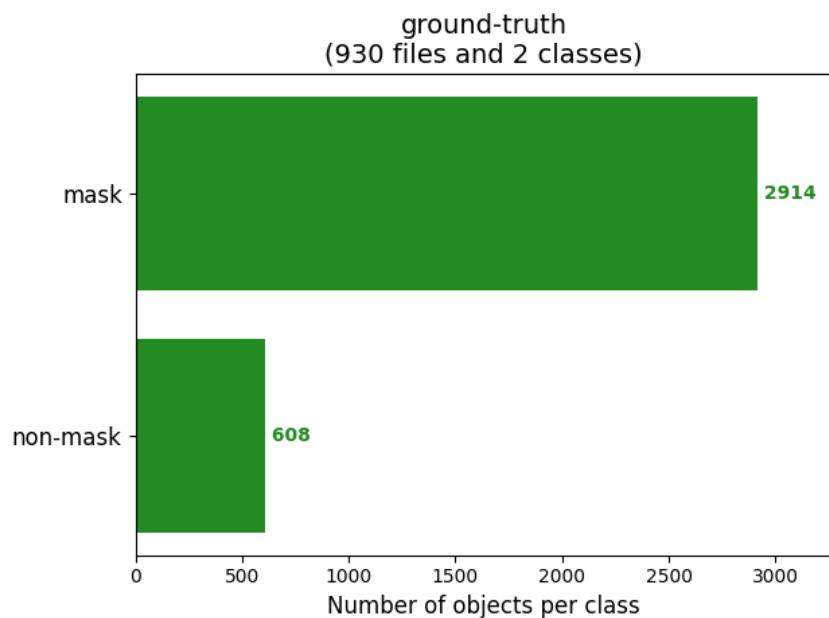
Model	mAP (%)
Mobilenetv1	52.58
Mobilenetv2	22.47
YOLOv3	65.85
YOLOv5	84.8
SCI	44.76
Insightface	72.76
Pyimagesearch	49.04
Cov	24.34

BẢNG 4.2: Kết quả mAP của các mô hình trên bộ test của WIDERMAFA

4.2 Đánh giá trên bộ dữ liệu thực tế tự tạo

Ngoài bộ dữ liệu WIDERMAFA, chúng em có tự thu thập thêm một số hình ảnh gần đây từ internet. Vì nhóm không thể biết rõ dữ liệu huấn luyện của các mô hình khác, nên có thể xảy ra trường hợp dữ liệu test trong bộ dữ liệu WIDERMAFA có thể nằm trong dữ liệu huấn luyện của các mô hình đó, điều này có thể dẫn đến sự đánh giá không công bằng giữa các mô hình. Để tránh trường hợp trên, chúng em đã tiến hành thu thập thêm một số hình ảnh và tạo nên một bộ dữ liệu kiểm tra mới và tiến hành đánh giá trên bộ dữ liệu này. Thông tin về bộ dữ liệu thực tế tự tạo như sau:

- Tổng cộng có 930 ảnh.
- Số lượng bounding box gương mặt đeo khẩu trang là 2914.
- Số lượng bounding box gương mặt không đeo khẩu trang là 608.



HÌNH 4.2: Thông số bộ test của bộ dữ liệu tự thu thập

Kết quả đánh giá của mỗi mô hình nhìn chung sẽ thấp hơn với khi đánh giá trên bộ dữ liệu WIDERMAFA.

Model	Precision		Recall	
	Mask	Non-mask	Mask	Non-mask
Mobilenetv1	0.92	0.85	0.13	0.17
Mobilenetv2	0.85	0.56	0.01	0.03
YOLOv3	0.96	0.87	0.26	0.33
YOLOv5	0.96	0.86	0.47	0.38
SCI	0.89	0.7	0.15	0.33
Insightface	0.85	0.28	0.45	0.83
Pyimagesearch	0.9	0.7	0.16	0.33
Cov	0.88	0.68	0.06	0.2

BẢNG 4.3: Kết quả precision-recall của các mô hình trên bộ test của bộ dữ liệu tự thu thập

Model	mAP (%)
Mobilenetv1	14.14
Mobilenetv2	1.75
YOLOv3	28.22
YOLOv5	41.23
SCI	21.28
Insightface	45.61
Pyimagesearch	21.69
Cov	11.36

BẢNG 4.4: Kết quả mAP của các mô hình trên bộ test của bộ dữ liệu tự thu thập

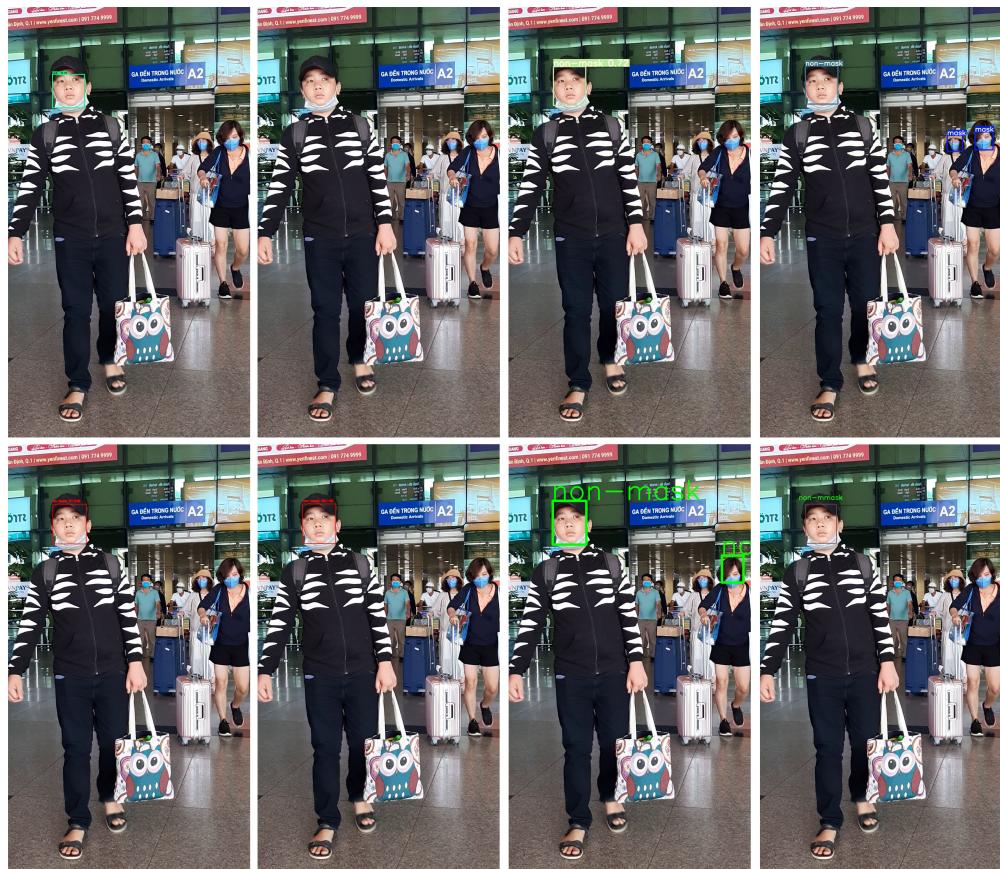
4.3 Nhận xét kết quả đánh giá và so sánh các mô hình

Dựa vào kết quả so sánh như trên, chúng em có một số nhận xét như sau:

- Kết quả trên bộ dữ liệu WIDERMAFA nhìn chung sẽ cao hơn kết quả trên bộ dữ liệu tự thu thập ở mọi chỉ số.
- Trong bốn mô hình mà nhóm huấn luyện (Mobilenetv1, Mobilenetv2, YOLOv3 và YOLOv5):
 - Vì Mobilenetv1 và Mobilenetv2 là các mô hình có kích thước nhỏ, ít biến số nên kết quả của hai mô hình này sẽ thấp hơn YOLOv3 và YOLOv5.
 - Với YOLOv3 và YOLOv5, chỉ số precision của hai mô hình này tương đồng nhau nhưng vì YOLOv5 có kết quả recall cao hơn nên sẽ có kết quả mAP cao hơn YOLOv3.
- Khi so sánh với các dự án khác của cộng đồng, mô hình YOLOv5 của nhóm cho kết quả khá tốt khi có chỉ số mAP cao nhất trong bộ dữ liệu WIDERMAFA và cao thứ hai khi đánh giá trên bộ dữ liệu tự thu thập.

4.4 Thủ nghiệm trên hình ảnh thực tế quay ở sân bay Tân Sơn Nhất - TPHCM

Để chứng minh tính ứng dụng cao của bài toán, đặc biệt trong hoàn cảnh dịch COVID-19, chúng em đã tiến hành quay video ngay tại cổng ra quốc nội tại sân bay Tân Sơn Nhất. Hình 4.3 thể hiện kết quả của các mô hình trên cùng một khung ảnh mà chúng em đã cắt từ video.



HÌNH 4.3: Kết quả khi chạy các mô hình trên 1 frame từ clip thực tế (thứ tự từ trên xuống dưới, từ trái qua phải: Mobilenetv1, Mobilenetv2, YOLOv3, YOLOv5, Pyimagesearch, SCI, Insignface, Cov)

Chương 5

KẾT LUẬN

5.1 Kết luận

Thông qua khóa luận này, chúng em đã hiểu rõ được bài toán phát hiện gương mặt đeo khẩu trang. Ngoài ra, chúng em đã khảo sát từ nhiều nguồn thông tin đáng tin cậy để biết được những hướng tiếp cận để giải quyết bài toán trên cũng như biết được tình hình nghiên cứu của cộng đồng. Đặc biệt là trong tình hình dịch bệnh COVID-19 trên thế giới đang diễn biến rất phức tạp thì việc ứng dụng bài toán trên vào thực tiễn là thật sự cần thiết. Với mục tiêu đó và sự tìm hiểu của nhóm, nhóm đã chọn được hướng đi phát hiện đa vật thể và mô hình mạng phù hợp là YOLOv3 để thực hiện bài toán. Cùng với đó, chúng em có tìm hiểu cách đánh giá mô hình phát hiện vật thể và tiến hành so sánh mô hình của nhóm với các mô hình được những cộng đồng uy tín trong ngành phát triển. Mặc dù không đạt được độ chính xác cao nhất những chúng em vẫn rất hài lòng với sản phẩm của mình khi cho chạy thực tế với những video clip mà nhóm thu thập. Kết quả từ những video này cho thấy tính khả thi khi tiếp tục phát triển sản phẩm của chúng em.

Ngoài ra, sau khi kết thúc khóa luận, chúng em đã tìm hiểu được sâu hơn về thị giác máy tính và trí tuệ nhân tạo. Chúng em cảm nhận được đây sẽ là tương lai trong các ứng dụng thực tế hỗ trợ đời sống con người. Mặc dù để ứng dụng vào thực tế còn nhiều khó khăn, việc huấn luyện yêu cầu cao về phần cứng và thách thức nhất là dữ liệu cho bài toán phải sát với thực tế và đủ đa dạng để máy tính có thể học được nhiều nhất có thể. Nhưng chúng em tin rằng với sự phát triển khoa học công nghệ như hiện nay thì những trở ngại trại trên sẽ không còn quá khó khăn.

5.2 Hướng phát triển

Kết thúc khóa luận, chúng em cảm thấy sản phẩm của nhóm còn nhiều thiếu sót. Nhưng qua những thiếu sót này, chúng em thấy được nhiều hướng phát triển cho sản phẩm của chúng em cũng như bài toán phát hiện gương mặt đeo khẩu trang. Cụ thể các hướng phát triển tiếp theo tại em có thể làm như sau:

- Thủ nghiệm hướng tiếp cận phát hiện gương mặt và phân loại gương mặt với nhiều mô hình mạng khác nhau.
- Thu thập thêm dữ liệu thực tế để có bộ dữ liệu sát với ứng dụng hơn.
- Điều chỉnh các siêu tham số trong mô hình mạng YOLOv5 cũng như thử nghiệm các hàm mất mát khác để chọn được bộ siêu tham số phù hợp nhất.
- Nghiên cứu thêm các mô hình, phương pháp để có thể cải thiện tốc độ hệ thống nhằm dễ dàng áp dụng vào các ứng dụng thực tiễn khác.

Tài liệu tham khảo

- [1] Jens Behley et al. *SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences*. 2019. arXiv: 1904.01416 [cs.CV].
- [2] *Bộ ý té: 5 việc cần làm tốt để phòng chống dịch COVID-19*. <https://ncov.moh.gov.vn/-/5-viec-can-lam-tot-e-phong-chong-dich-covid-19>. Accessed: 2020-07-24.
- [3] Y. Chen et al. “Adversarial Occlusion-aware Face Detection”. In: *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. 2018, pp. 1–9.
- [4] Jiankang Deng et al. “RetinaFace: Single-stage Dense Face Localisation in the Wild”. In: *CoRR* abs/1905.00641 (2019). arXiv: 1905 . 00641. URL: <http://arxiv.org/abs/1905.00641>.
- [5] M. Everingham et al. *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*. <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>.
- [6] S. Ge et al. “Detecting Masked Faces in the Wild with LLE-CNNs”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 426–434.
- [7] Andrew G. Howard et al. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”. In: *CoRR* abs/1704.04861 (2017). arXiv: 1704.04861. URL: <http://arxiv.org/abs/1704.04861>.
- [8] Frank Hutter Ilya Loshchilov. “SGDR: STOCHASTIC GRADIENT DESCENT WITH WARM RESTARTS”. In: (2017).
- [9] Glenn Jocher et al. *ultralytics/yolov5: v3.0*. Version v3.0. Aug. 2020. DOI: 10 . 5281/zenodo . 3983579. URL: <https://doi.org/10.5281/zenodo.3983579>.

- [10] Alina Kuznetsova et al. “The Open Images Dataset V4”. In: *International Journal of Computer Vision* 128.7 (Mar. 2020), pp. 1956–1981. ISSN: 1573-1405. DOI: 10.1007/s11263-020-01316-z. URL: <http://dx.doi.org/10.1007/s11263-020-01316-z>.
- [11] Tsung-Yi Lin et al. “Microsoft COCO: Common Objects in Context”. In: *CoRR* abs/1405.0312 (2014). arXiv: 1405.0312. URL: <http://arxiv.org/abs/1405.0312>.
- [12] World Health Organization. *Advice on the use of masks in the context of COVID-19: interim guidance, 6 April 2020*. Technical documents. 2020, 5 p.
- [13] Joseph Redmon and Ali Farhadi. “YOLOv3: An Incremental Improvement”. In: *CoRR* abs/1804.02767 (2018). arXiv: 1804.02767. URL: <http://arxiv.org/abs/1804.02767>.
- [14] Mark Sandler et al. “Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation”. In: *CoRR* abs/1801.04381 (2018). arXiv: 1801.04381. URL: <http://arxiv.org/abs/1801.04381>.
- [15] Jianfeng Wang, Ye Yuan, and Gang Yu. “Face Attention Network: An effective Face Detector for the Occluded Faces”. In: (Nov. 2017).
- [16] Zhongyuan Wang et al. *Masked Face Recognition Dataset and Application*. 2020. arXiv: 2003.09093 [cs.CV].
- [17] WHO:Coronavirus disease (COVID-19) advice for the public: When and how to use masks. <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public/when-and-how-to-use-masks>. Accessed: 2020-07-24.
- [18] Shuo Yang et al. “WIDER FACE: A Face Detection Benchmark”. In: *CoRR* abs/1511.06523 (2015). arXiv: 1511.06523. URL: <http://arxiv.org/abs/1511.06523>.