ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

ꕥ★ꕥ

BÁO CÁO ĐỀ TÀI
NGHIÊN CỨU KHOA HỌC

XÂY DỰNG HỆ THỐNG NHẬN DẠNG ĐỐI TƯỢNG SỬ DỤNG KHẨU TRANG

Giảng viên hướng dẫn : ThS. LƯƠNG TRẦN NGỌC KHIẾT

Nhóm thực hiện :

+ LÊ TẤN LỘC 45.01.104.135
+ HUỲNH THANH PHONG 45.01.104.172
+ TRẦN THỊ TỨ LINH 45.01.104.127
+ PHẠM DUY MINH 45.01.104.145

TP HỒ CHÍ MINH – 4/2021

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

LỜI CÁM ƠN

Nghiên cứu được thực hiện tại Khoa Công nghệ thông tin – Trường Đại học Sư phạm Thành phố Hồ Chí Minh, dưới sự hướng dẫn khoa học của ThS Lương Trần Ngọc Khiết.

Trước tiên chúng em xin bày tỏ lòng biết ơn sâu sắc tới thầy ThS Lương Trần Ngọc Khiết đã đưa chúng em đến với lĩnh vực nghiên cứu này. Thầy đã tận tình giảng dạy, hướng dẫn chúng em tiếp cận và đạt được những kết quả nhất định trong nghiên cứu của mình. Thầy đã luôn tận tâm động viên, khuyến khích và chỉ dẫn giúp chúng em hoàn thành nghiên cứu này.

Chúng em xin bày tỏ lòng biết ơn tới các Thầy Cô thuộc Khoa Công nghệ thông tin và cán bộ Phòng Khoa học Công nghệ, khoa Công nghệ Thông tin – Trường Đại học Sư Phạm Thành phố Hồ Chí Minh đã tạo mọi điều kiện thuận lợi giúp đỡ chúng em trong quá trình học tập và nghiên cứu.

Sự hướng dẫn của thầy Th. s Lương Trần Ngọc Khiết đã tận tình hướng dẫn, động viên, cổ vũ của gia đình, bạn bè là nguồn động lực quan trọng để chúng em thực hiện đề tài nghiên cứu. Do kiến thức còn hạn chế, nên đề tài nghiên cứu của chúng em không tránh khỏi những thiếu sót, kính mong sự thông cảm, chỉ bảo của quý Thầy Cô.

Chúng em xin chân thành cảm ơn.

Thay mặt nhóm thực hiện. /

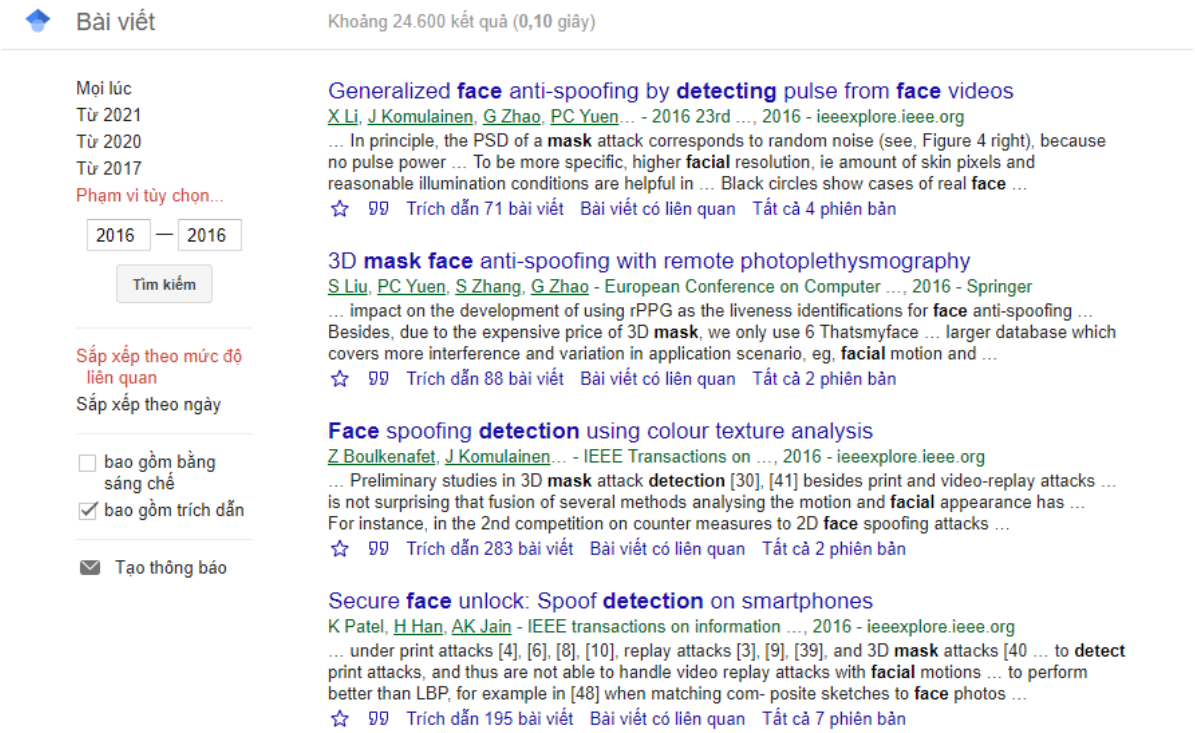
Lê Tấn Lộc

CHƯƠNG 1. TỔNG QUAN

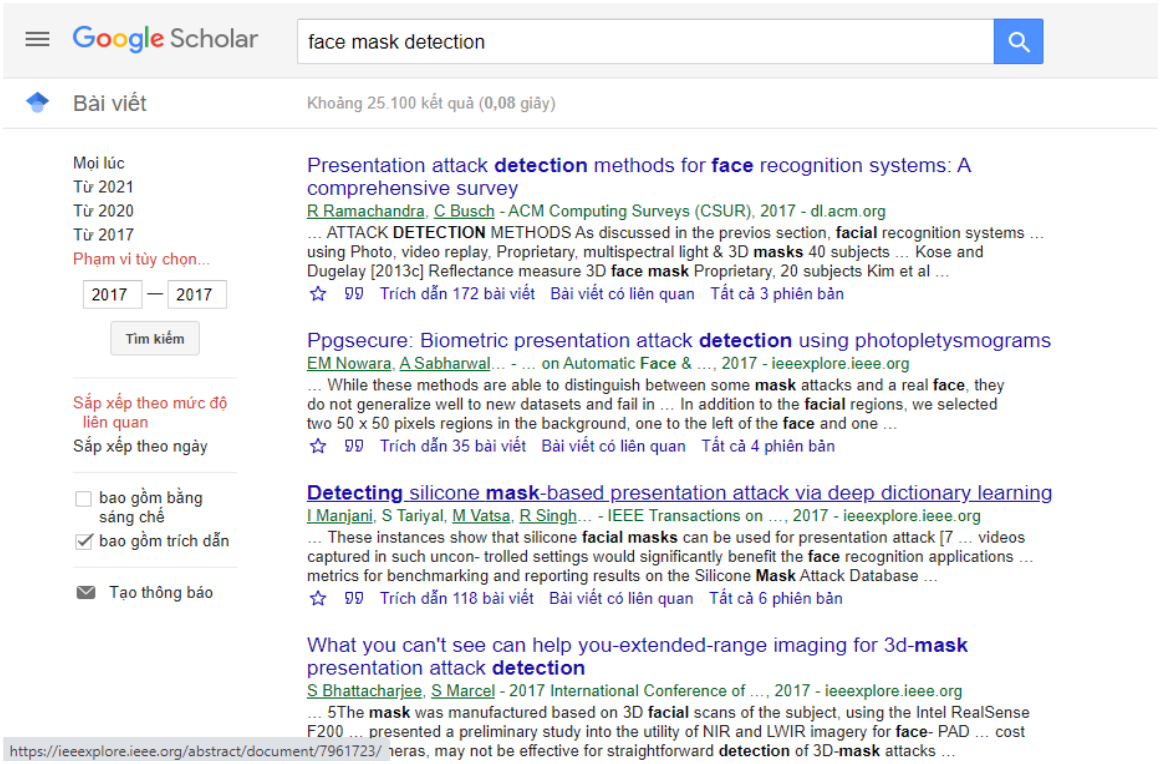
1.1. Giới thiệu bài toán.

1.1.1. Đặt vấn đề.

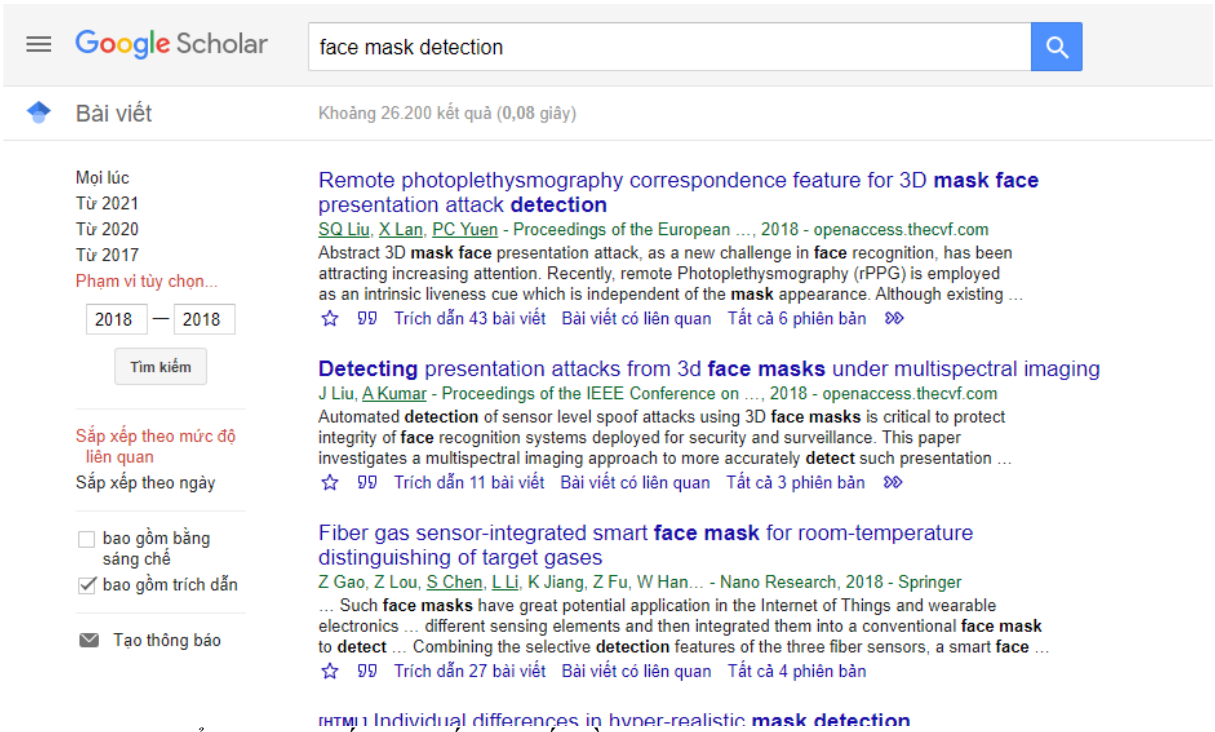
Nhận dạng khẩu trang có tên (Face mask detection: FMD – còn được gọi là nhận dạng đối tượng khi đeo khẩu trang) là một bài toán chính thuộc lĩnh vực thị giác máy tính - Computer Vision. Đây là một bài toán với một bài toán lớn và một bài nhỏ, từ bài toán nhỏ là nhận dạng đối tượng dẫn đến bài toán lớn là nhận diện đối tượng khi đeo khẩu trang. Ngày nay, cùng với sự phát triển công nghệ kĩ thuật số mà trong vòng 5 năm trở lại đây, bài toàn đã được cộng đồng nghiên cứu quan tâm và cũng đạt nhiều thành tựu nổi bật.



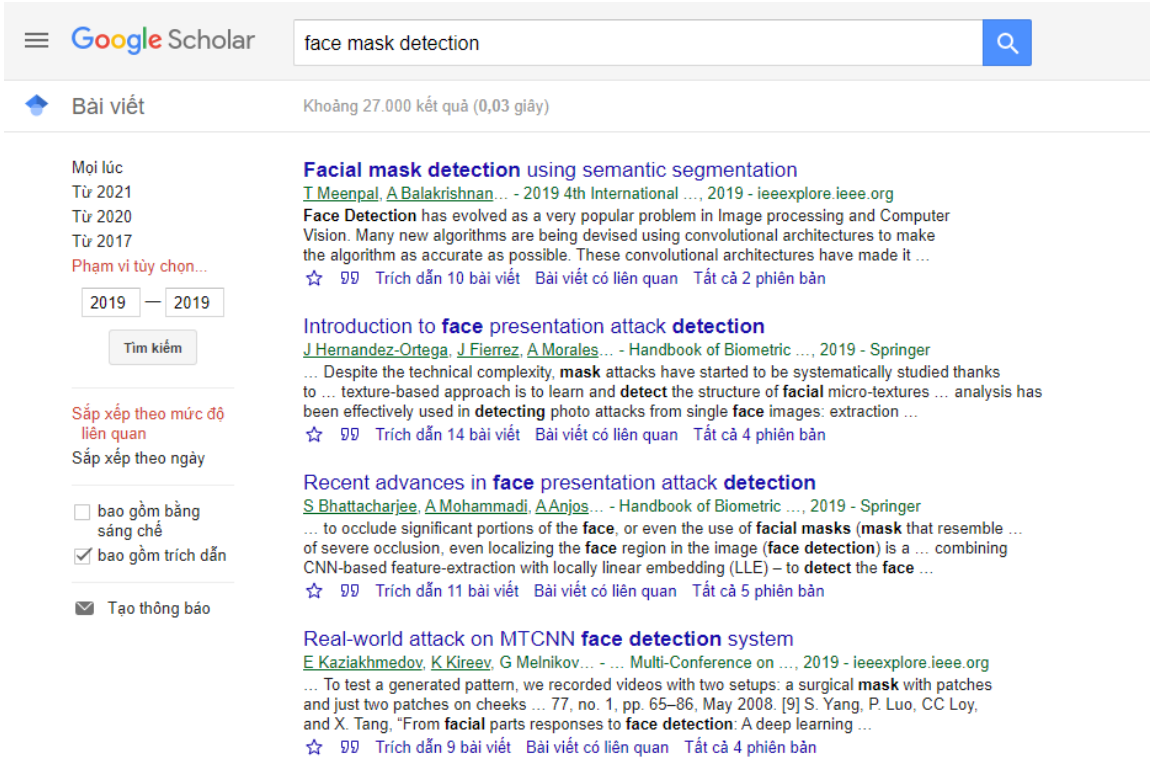
Hình 1-1 Ảnh chụp thống kê số bài viết về cụm từ "Face mask dectection" trên Google schoolar vào năm 2016.



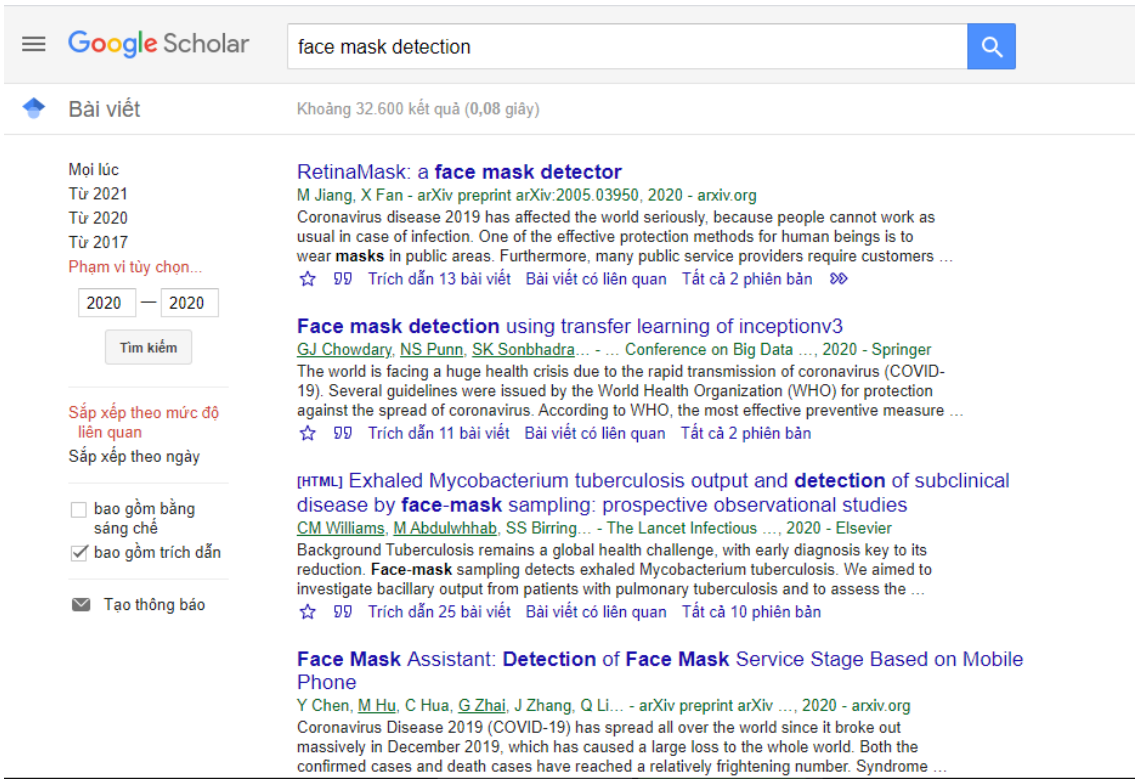
Hình 1-2 Ảnh chụp thống kê số bài viết về cụm từ "Face mask dectection" trên Google schoolar vào năm 2017



Hình 1-3 Ảnh chụp thống kê số bài viết về cụm từ "Face mask dectection" trên Google schoolar vào năm 2018



Hình 1-4 Ảnh chụp thống kê số bài viết về cụm từ "Face mask dectection" trên Google schoolar vào năm 2019.



Hình 1-5 Ảnh chụp thống kê số bài viết về cụm từ "Face mask dectection" trên Google schoolar vào năm 2020.

Hình 1-6 Thống kê các công trình nghiên cứu liên quan đến cụm từ "Face mask detection" 2016 - 2020.

Face mask detection dần trở thành một chủ đề nóng cho cộng đồng nghiên cứu khoa học, thường xuyên được đề xuất tuong các hội nghị như: ICASSP, IFIP, CVPR, … và trên các trang tập chí nổi tiếng: Springer.

Chỉ trong vòng 5 năm trở lại đây, các thành tựu về nhận điện đối tượng đeo khẩu trang đã nhận được môt số lượng đáng kể, tuy nhiên bài toán này vẫn còn nhiều hạn chế và thách thức liên quan dữ liệu cực lớn, bảo mật thông tin, nhận diện sai khi có các ý tố khác tác động (vật thể khác che mặt mà không phải khẩu trang, ánh sáng….).

Nhờ vào sự phát triển vượt bậc về các công nghệ kỹ thuật số, cùng với các công nghệ khác đã tạo cơ hội cho các lĩnh vực về xử lý hình ảnh dần ăn sâu vào đời sống con người về mọi mặt. Việc ứng dụng có hiệu quả các thành tựu này vào đời sống đã góp phần phát triển kinh tế và nâng cao đời sống xã hội, đồng thời nâng cao chất lượng cuộc sống của mọi người. Gần đây dịch covid đã là mối đe dọa đối với toàn nhân loại về vật chất lẫn tinh thần. Vì vậy bài toán nhận diện đối tượng có đeo khẩu trang rất cần thiết trong công cuộc phòng chống sự lây lan của dịch bệnh covid.

1.1.2. Mục tiêu cụ thể.

Ở nghiên cứu này tập trung vào bài toán nhận diện những người có đeo khẩu trang đảm bảo hạn chế sự lây lan của dịch bệnh

Mục tiêu cụ thể của đề tài là phân tích, phát triển vấn đề, đề xuất giải pháp cho việc nhận diện các loại đối tượng có đeo khẩu trang thuộc miền dữ liệu trên.

Xây dựng một chương trình có hiệu quả, đầy đủ và chính xác với mục đích hỗ trợ việc nhận dạng thực thể song xây dựng một hệ thống huấn luyện, thực nghiệm và ứng dụng.

1.2. Xử lý hình ảnh.

Xử lý hình ảnh là một lĩnh vực của thị giác máy tính tập trung vào các ứng dụng trên hình ảnh thực. Trong thị giác máy tính thì xử lý hình ảnh là một trong những phần trọng tâm vì nó liên quan đến việc phải xác định, phân tích hình ảnh.

Xử lý hình ảnh là xác định, phân tích và nghiên cứu cấu trúc các điểm ảnh [1] và tạo ra các hệ thống thông minh có khả năng nhận dạng đối tượng từ hình ảnh và video, giúp giải quyết các vấn đề như phân loại từng đối tượng, xác định những vấn đề tiềm ẩn. Phân tích hình ảnh là một nhiệm vụ thiết yếu cho phép chúng ta nhận dạng ra các đối tượng ở trong hình ảnh hoặc video.

Nhờ vào sự tiến bộ của công nghệ kỹ thuật số đã góp phần tạo điều kiện cho chúng ta có thể thu thập được nguồn dữ liệu phong phú về các hình ảnh thử nghiệm, song việc xác định, phân tích và nghiên cứu cũng trở nên chuẩn xác, truyền cảm hứng cho sự đổi mới, phát triển và có thể dẫn đến công nghê nhận dạng khuôn mặt mạnh mẽ hơn.

1.3. Nhận dạng đối tượng đeo khẩu trang.

Bài toán nhận dạng đối tượng đeo khẩu trang là bài toán xác định (phát hiện) những đối tượng đeo khẩu trang. Bài toán này sẽ dựa trên mô hình mạng nơ-ron thần kinh tích chập (convolutional neural networks - CNN)[2] để phân tích các điểm ảnh khác nhau dựa trên các thuộc tính cả việc nhận dạng các điểm ảnh đặc trưng của hình ảnh bằng phương pháp tính chập mạng nơ-ron (CNNs) [3]. Ví dụ trong trường hợp cung cấp các bức hình có một hay nhiều người đeo khẩu trang cho chương trình phân tích, sau đó lấy một bức ảnh có người đeo khẩu trang nào đó bất kì đưa cho chương trình kiểm tra, kỳ vọng sau khi kiểm tra là chương trình sẽ phản hồi được là có người đeo khẩu trang trong bước hình đó.

Dữ liệu đầu vào cần được nhận diện là một hay nhiều hình ảnh hoặc nhận dạng trực tiếp thông qua các thiết bị ghi hình. Một dữ liệu sau khi được kiểm tra sẽ được xếp vào một trong hai trạng thái là có đeo khẩu trang hoặc không đeo khẩu trang. Như vậy, để có thể nhận diện được một người có đeo khẩu trang hay không, thì hình ảnh là dữ liệu cơ bản nhất, cho dù chương trình nhận diện chạy trên bất kì cấu trúc nào.

Nhận diện người đeo khẩu trang góp phần không nhỏ trong việc đảm bảo an ninh trật tự trong công cộng. Chính vì vậy, bài toán này đã nhận được sự quan tâm sâu sắc của giới khoa học, đặc biệt là trong lĩnh vực thị giác máy tính.

1.4. Các ứng dụng trong thế giới thực của nhận diện đối tượng deo khẩu trang.

1.4.1. Trước mùa dịch COVID.

Năm 2007, một thành phố ở Hà Lan đã cung cấp cho một chủ cửa hàng một hệ thống nhận dạng khuôn mặt để phát hiện những người từng có tiền án trộm cắp ở các cửa hàng. Hệ thống camera này sẽ so sánh các khuôn mặt của khách hàng trong cửa hàng với hình ảnh của các tên trộm trong địa phương từng bị cảnh sát bắt. Các nhân viên trong cửa hàng sẽ chú ý đến những khách hàng đặc biệt này, từ đó có thể giảm được việc mất trộm.

Sản phẩm sẽ tập trung vào những người đeo khẩu trang khi họ đi vào những nơi công cộng vì những người này không muốn người khác nhìn thấy mặt mình nên rất có khả năng họ sẽ làm điều gì đó mờ ám.

Hình 1-7 Áp dụng chương trình nhận diện khuôn mặt được đặt ở nơi công cộng, các siêu thị.

Khi chưa bùng phát dịch bệnh, mặt nạ da người silicon là một trong những thách thức trong công tác an ninh và phòng chống tội phạm vì lúc này khả năng phân biệt những chiếc mặt nạ này với khuôn mặt thật của mọi người là rất hạn chế. Thử nghiệm cho thấy độ chính xác kém (40%) và sự khác biệt lớn (5–100%) đối với mặt nạ có độ chân thực cao giữa mặt nạ độ chân thực thấp và khuôn mặt thật [4].

Hình 1-8 Mặt nạ da người silicon nam *(trái) được làm bởi RJ (phải)*

Giả mạo khuôn mặt, kẻ giả mạo có thể sử dụng một số mặt nạ giả bắt chước khuôn mặt người dùng thật. Các biện pháp đối phó hiện tại chống lại sự giả mạo thông qua phân tích kết cấu khuôn mặt, phát hiện chuyển động và phân tích phản xạ bề mặt. Da bao gồm cấu trúc nhiều lớp tạo ra nhiều phản xạ: phản xạ bề mặt và phản xạ dưới bề mặt, đề xuất một biện pháp để phân biệt giữa mặt thật và ảnh giấy in dựa trên các đặc tính vật lý của vật liệu góp phần tạo nên các giá trị phản chiếu đặc biệt của nó [5]. Các vụ phạm tội liên quan đến máy ATM bằng cách sử dụng xử lý hình ảnh giám sát video, để phát hiện khuôn mặt bị che khuất, phân tích hành vi, cử chỉ bất thường của con người và phát hiện đối tượng bất hợp pháp có thể không hoạt động đối với ATM [6].

Mặc khác, hệ thống nhận diện học khuôn mặt dễ bị tấn công giả mạo. Các cuộc tấn công như vậy có thể được thực hiện theo nhiều cách, bao gồm hiển thị hình ảnh, video hoặc mặt nạ 3D giả mạo của một người dùng hợp lệ. Do đó, trong thời gian này, đã có một số bài viết đề ra các phương án như phát hiện xung từ video khuôn mặt, dựa trên thực tế là tín hiệu xung tồn tại trong khuôn mặt sống thực nhưng không có trong bất kỳ mặt nạ hoặc vật liệu in nào, phương pháp này có thể là một giải pháp tổng quát để phát hiện khuôn mặt sống động. Phương pháp đề xuất được đánh giá đầu tiên trên cơ sở dữ liệu giả mạo mặt nạ 3D 3DMAD để chứng minh tính hiệu quả của nó trong việc phát hiện các cuộc tấn công mặt nạ 3D [7], chụp ảnh quang tuyến (PPGSecure) một phương pháp luận mới dựa trên các phép đo sinh lý học dựa trên camera để phát hiện và ngăn chặn các cuộc tấn công trình bày sinh trắc học như vậy. PPGSecure sử dụng hình ảnh quang tuyến (PPG), là một ước tính các dấu hiệu quan trọng từ những thay đổi nhỏ về màu sắc trong video được quan sát do những biến động nhỏ trong lượng máu chảy đến

mặt, phổ tần số thời gian của tín hiệu PPG ước tính cho các cá thể sống thực sự khác biệt rõ ràng so với phổ tần số của các cuộc tấn công trình chiếu và khai thác những khác biệt này để phát hiện các cuộc tấn công trình chiếu, đạt được hiệu suất tốt hơn đáng kể so với các phương pháp phát hiện tấn công trình bày hiện đại [8], học kết cấu động phức hợp sâu - học thông tin kết cấu động mạnh mẽ từ các tính năng phức hợp sâu chi tiết. Hơn nữa, ràng buộc phân biệt kênh được kết hợp một cách thích ứng để cân bằng khả năng phân biệt của các kênh đặc trưng trong quá trình học tập [9],...

1.4.2. Trong mùa dịch COVID.

COVID-19 lây lan chủ yếu từ người sang người qua các giọt bắn từ đường hô hấp. Các giọt bắn từ đường hô hấp bay vào không khí khi quý vị ho, hắt hơi, trò chuyện, la hét hoặc ca hát. Sau đó, những giọt bắn này có thể rơi vào miệng hoặc mũi của những người ở gần quý vị hoặc họ có thể hít phải những giọt bắn này. Do vậy để đảm bảo sức khỏe của mọi người và tránh tình trạng lây lan dịch bệnh thì việc đeo khẩu trang đến các nơi công cộng là điều bắt buộc bởi, khẩu trang là một rào chắn đơn giản giúp ngăn các giọt bắn từ đường hô hấp khi tiếp xúc với người khác. Các nghiên cứu chỉ ra rằng đeo khẩu trang che mũi và miệng giúp làm giảm các giọt bắn ra ngoài. Vì vậy những ai không đeo khẩu trang khi đến những nơi công cộng là người có những hành vi bất thường [10].

Phát hiện mặt nạ bằng MobileNetV2 trong Kỷ nguyên Đại dịch COVID-19 [11]

Đại dịch Corona Virus Disease (COVID-19) đang gây ra môt cuộc khủng hoảng sức khỏe. Một trong những phương pháp hiệu quả để chống lại virus là đeo khẩu trang. Tính năng phát hiện mặt nạ có thể được các cơ quan chức năng sử dụng để lập kế hoạch giảm thiểu, đánh giá, phòng ngừa và hành động chống lại COVID-19. Nhận dạng mặt nạ trong nghiên cứu này được phát triển bằng thuật toán máy học thông qua phương pháp phân loại hình ảnh: MobileNetV2. Các bước để xây dựng mô hình là thu thập dữ liệu, xử lý trước, chia nhỏ dữ liệu, thử nghiệm mô hình và thực hiện mô hình. Mô hình đươc xây dựng có thể phát hiện những người đang đeo mặt nạ và không đeo mặt nạ và không đeo với độ chính xác 96.85%.

Xác định danh tính cá nhân của người đeo mặt nạ [12].

Một công ty Nhật Bản NEC, đã phát triển một hệ thống nhận diện khuôn mặt để danh tính đối tượng đeo khẩu trang. Cảnh sát đã sử dụng hệ thống nhận diện khẩu trang trong thời gian thực của công ty để dò tìm một số đối tượng trong danh sách.

Nhận diện khẩu trang ở thời gian thực trong thời kỳ COVID-19 dựa trên học sâu

Trong thời kỳ COVID-19 đang hoành hành, nó mang đến một mối hiểm họa cho thế giới, song song đó cũng nhắc nhở cho chúng ta cần phải kiểm soát sự hoàng hành của virus. Dẫn đến sự bùng nổ kết nổ không gian mạng giữa các thiết bị giúp cho việc thu nhập dữ liệu. Chẳng hạn như: hệ thống chăm sóc và thông báo cho mọi người chăm sóc sức khỏe. Đặc biệt, trong thời kỳ COVID-19 hiện nay, khẩu trang là một vật dụng không thể thiếu. Vì vậy, chúng ta cần một hệ thống nhận diện khẩu trang trong thời gian thực ngăn chặn sự lây lan. Trong bài viết này, chúng ta sẽ nói tới thư viện nhận dạng khẩu trang dựa trên máy tính (EC Mask) để giúp sức khỏe cộng đồng có thể chạy trên những camera ít năng lượng. Hệ thống EC Mask bao gồm 3 giai đoạn chính: chạy trên camera video, xác định khuôn mặt, nhận dạng danh tính.

1.5. Phạm vi đề tài.

Phạm vi nghiên cứu của đề tài là nhận diện được đối tượng có đeo khẩu trang hay không trong video thời gian thực (ghi hình trực tiếp) trên các thiết bị ghi hình.

Phạm vi áp dụng có thể áp dụng trong các siêu thị, cửa hàng tiện lợi, tiệm tạp hóa, trường học, trung tâm thương mai…

Đối tượng được xử lý phải đảm bảo có 2/3 khuôn mặt được xác định qua khung hình camera thì mới có thể nhận diện được.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Nhận diện khuôn mặt.

Nhận diện khuôn mặt (Face Detection) là một nhánh của Trí tuệ nhân tạo, tập trung vào việc nghiên cứu các bài toán để máy tính có thể tìm và nhận diện khuôn mặt trên những hình ảnh kỹ thuật số, được phát triển dựa trên những nghiên cứu của Thị giác máy tính (Computer Vision) cùng với sự tiến bộ các lĩnh vực như: máy học (Machine Learning), Mạng nơ-ron nhân tạo (Neural Networks), và nhiều công nghệ khác. Mục tiêu của lĩnh vực này là giúp máy tính tìm và nhận diện có hiệu quả với những đối tượng là khuôn mặt, ngoài việc xử lý để nhận diện khuôn mặt thì lĩnh vực này còn tập trung việc xác định các thông tin thông qua khuôn mặt như: tuổi, giới tính và cả cảm xúc của con người.

Những người tiên phong về nhận dạng khuôn mặt là Woody Bledsoe, Helen Chan Wolf và Charles Bisson. Năm 1964 và 1965, Bledsoe cùng với Wolf và Bisson bắt đầu sử dụng máy tính để nhận dạng khuôn mặt người.

Thừa kế những thành tựu của Bledsoe, bước nhảy tiếp theo của Goldstein, Harmon và Lesk vào những năm 1970 đã được cải tiến thêm việc tự động hóa nhận diện 21 điểm dấu đó bao gồm cả màu tóc, độ dày của miệng.

Cuối những năm 1980, chúng ta mới thấy sự tiến bộ hơn nữa với sự phát triển của nhận dạng khuôn được áp dụng ở các doanh nghiệp. Năm 1988, Sirovich và Kirby bắt đầu áp dụng đại số tuyến tính vào bài toán nhận dạng khuôn mặt.

Năm 1991, Turk và Pentland tiếp bước Sirovich và Kirby bằng cách phát hiện khuôn mặt trong một bức ảnh, dẫn đường cho việc nhận dạng khuôn mặt tự động sớm nhất. Tuy bước đột phá đáng kể này đã bị cản trở bởi các yếu tố công nghệ và môi trường, nhưng nó cũng đã góp phần cho công nghệ Nhận diện khuôn mặt được phát triển hơn trong tương lai.

Vào đầu những năm 1990, để khuyến khích thị trường nhận dạng khuôn mặt thương mại, Cơ quan Chỉ đạo các Dự án Nghiên cứu Quốc phòng Tiên tiến (DARPA) và Viện Tiêu chuẩn và Kỹ thuật quốc gia (NIST) đã triển khai chương trình Công nghệ Nhận dạng Khuôn mặt (FERET).

Vào đầu những năm 2000, Viện Tiêu chuẩn và Kỹ thuật quốc gia (NIST) đã bắt đầu Thử nghiệm nhà cung cấp nhận dạng khuôn mặt (FRVT). Dựa trên FERET và FRVTs được thiết kế để cung cấp các đánh giá độc lập của chính phủ về các hệ thống nhận dạng khuôn mặt có sẵn trên thị trường cũng như các công nghệ nguyên mẫu.

Face Recognition Grand Challenge (FRGC) được ra mắt vào năm 2006 với mục tiêu chính là thúc đẩy và nâng cao công nghệ nhận dạng khuôn mặt, được thiết kế nhằm hỗ trợ các nỗ lực nhận dạng khuôn mặt hiện có của Chính phủ Hoa Kỳ.

Năm 2010, Facebook bắt đầu triên khai tính năng nhận dạng khuôn mặt giúp xác định những người có khuôn mặt có thể xuất hiện trong ảnh mà người dùng Facebook cập nhật hằng ngày. Tuy nhiên, tính năng này ngay lập tức gây tranh cái bởi các phương tiện truyền thông tin tức, làm dấy lên một loạt các bài báo liên quan đến quyền riêng tư. Dù vậy, nhưng người dùng Facebook dường như không bận tâm.

Công nghệ Nhận dạng khuôn mặt đã phát triển nhanh chóng từ năm 2010 trở lại đây và ngày 12 tháng 9 năm 2017 đánh dấu một bước đột phá quan trọng khác để tích hợp tính năng nhận dạng khuôn mặt vào cuộc sống hằng ngày của chúng ta. Song cũng là ngày Apple ra mắt iPhone X, qua đó người dùng iPhone đầu tiên có thể mở khóa điện thoại bằng FaceID – thuật ngữ tiếp thị của Apple về nhận dạng khuôn mặt. [13][14]

2.1.1. Một số thuật toán nhận diện khuôn mặt kinh điển.

Năm 1960, Bled đã phát triển một hệ thống có thể phân loại khuôn mặt bằng tay trên thiết bị là RAND tablet. Là một thiết bị có thể nhập được tọa độ ngang dọc bằng cách xử dụng một cây bút cảm ứng để có thể gửi mã hóa thông tin tọa độ cho máy hiểu. Bằng cách này ông đã có thể ghi lại những đặc điểm quan trọng trên khuôn mặt và những thông tin đó sẽ được lưu trữ trong cơ sở dữ liệu. Và khi đưa một hình ảnh ai đó vào hệ thống, nó sẽ có đủ dữ liệu cũng như khả năng để so sánh giữa thông tin trong dữ liệu và thông tin lấy được từ tấm ảnh mới được đưa vào

Eigenfaces được Sirovich và Kirby nghĩ ra khi áp dụng tuyến tính vào trong vấn đề nhận diện khuôn mặt. Khi mà đưa vào một bức ảnh vào trong máy cần rất nhiều thời gian để có thể nhận dạng được do phải cần thêm thời gian để xác định các đặc điểm nào quan trọng trên khuôn mặt. Và Eigenface đã giúp giảm đi thời gian xác định đó bằng cách làm giảm đi hoặc bớt đi những đặc điểm không quan trọng và giữ những lại những đặc điểm để đi so sánh. Và không những giảm thiểu thời gian cho máy xử lý mà còn làm giảm đi những dự đoán sai của máy đưa ra [15].

Năm 1980, Fukushima đề xuất mô hình mạng nơ-ron có cấp bậc gọi là neocognitron. Mô hình này dựa trên khái niệm về S cell và C cell. Mạng neocognitron có thể nhận diện mẫu dựa trên việc học hình dáng của đối tượng.

Mạng Nơ-ron Tích Chập lấy cảm hứng từ não người. Nghiên cứu trong những thập niên 1950 và 1960 của D.H Hubel và T.N Wiesel trên não của động vật đã đề xuất một mô hình mới cho việc cách mà động vật nhìn nhận thế giới. Trong báo cáo, hai ông

đã diễn tả 2 loại tế bào nơ-ron trong não và cách hoạt động khác nhau: tế bào đơn giản (simple cell – S cell) và tế bào phức tạp (complex cell – C cell).

Sau đó vào năm 1998, mạng nơ-ron tích chập được giới thiệu bởi Bengio, Le Cun, Bottou và Haffner. Mô hình đầu tiên của họ được gọi tên là LeNet-5. Mô hình này có thể nhận diện chữ số viết tay. Và mạng nơ-ron tích chập của LeCun là bước đạp cho ngành thị giác máy tính thời nay. Các tế bào đơn giản được kích hoạt khi nhận diện các hình dáng đơn dản như đường nằm trong một khu vực cố định và một góc cạnh của nó. Các tế bào phức tạp có vùng tiếp nhận lơn hơn và đầu ra của nó không nhạy cảm với những vị trí cố định trong vùng.

Trong thị giác, vùng tiếp nhận của một nơ-ron tương ứng với một vùng trên võng mạc nơi mà sẽ kích hoạt nơ-ron tương ứng.

2.1.2. Các thuât ngữ cơ bản trong nhận diện khuôn mặt.

Face detection: Phát hiện khuôn mặt trong ảnh Các máy ảnh camera hiện tại điều có chức năng này.

Thuật toán được sử dụng phổ biến nhất hiện nay là Viola-Jones (Thư viện OpenCV hỗ trợ nhận diện khuôn mặt theo thuật toán này). Ngoài nó nó còn có thể nhận diện được đồ vật, hình dạng.

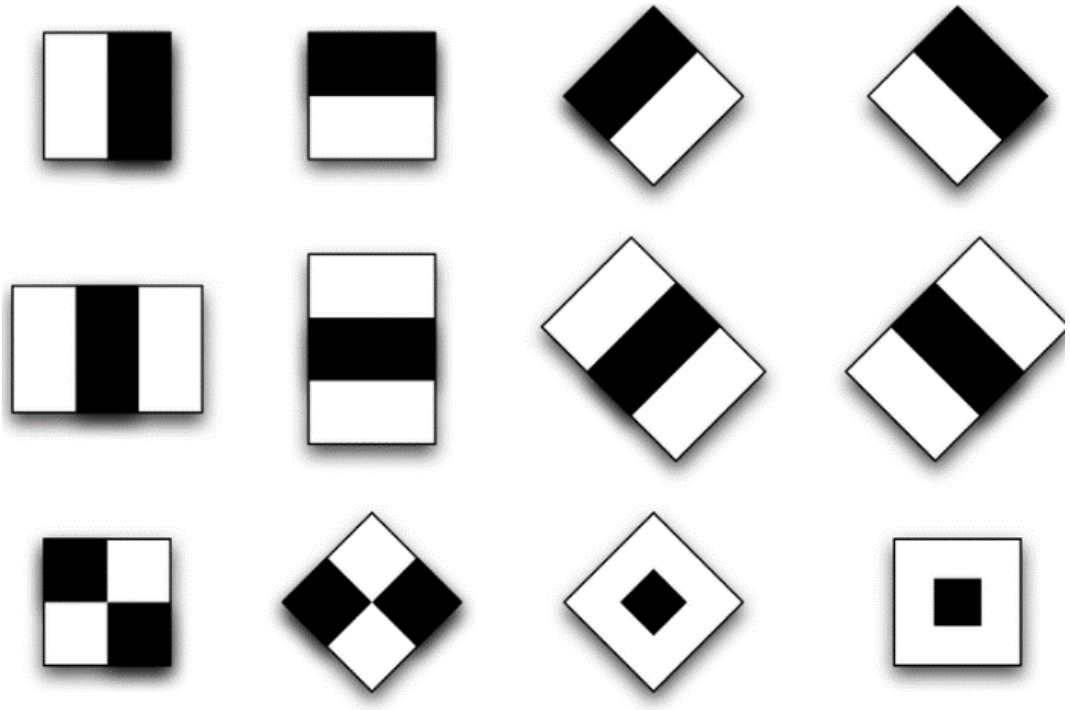
Cơ chế hoạt động:

- Tạo ra một frame hình vuông, lần lượt di chuyển frame này khắp tấm ảnh gốc
- Mỗi khi frame di chuyển, check xem khu vực trong frame có phải là khuôn mặt hay đồ vật không (Bằng cách check các vùng sáng tối trong frame)
- Sau khi di chuyển hết ảnh, tăng kích cở frame lên là scan lại từ đầu
- Dừng thuật toán khi frame đủ lớn

Face Recognition: quá trình nhận diện khuôn mặt từ những gương mặt đã quét được trong ảnh. Các khuôn mặt này sẽ được so với các khuôn mặt có trong cơ sở dữ liệu để trả về kết quả

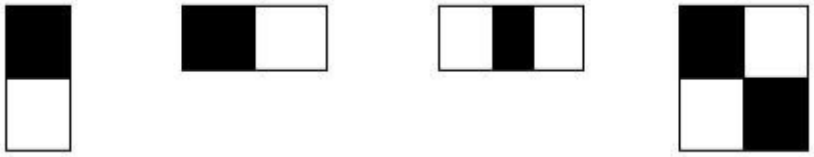
2.1.2.1. Haar-Like.

Haar-Like [16] là những hình chữ nhật được phân thành các vùng khác nhau.



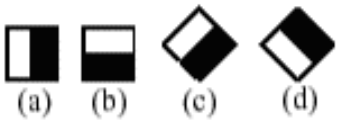
Hình 2-1 Một số ví dụ Haar-Like

Đặc trưng do Viola và Jones công bố gồm 4 đặc trưng cơ bản để xác định khuôn mặt người. Mỗi đặc trưng Haar-Like là sự kết hợp của hai hay ba hình chữ nhật trắng hay đen.

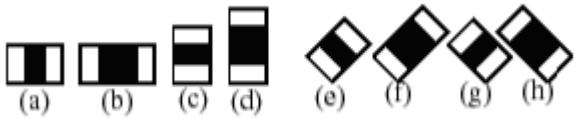


Hình 2-2 Bốn đặc trưng cơ bản của Haar-Like

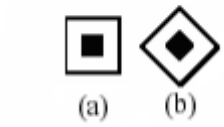
Để sử dụng các đặc trưng này vào việc xác định khuôn mặt người, 4 đặc trưng Haar-Like cơ bản được mở rộng ra và được chia làm 3 tập đặc trưng như sau



Hình 2-3 Đặc trưng cạnh (edge feature).



Hình 2-4 Đặc trưng đường (line feature).

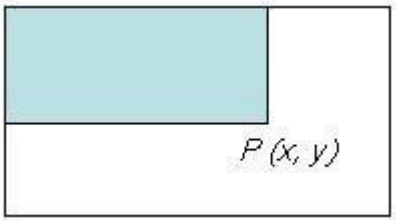


Hình 2-5 Đặc trưng xung quanh tâm (center-surround features).

Dùng các đặc trưng trên, ta có thể tính được các giá trị của đặc trưng Haar-Like là sự chênh lệch giữa tổng của các pixel của vùng đen và vùng trắng theo công thức.

f(x) = $\text{Tổng}_{\text{vùng đen}}$ (các mức xám của pixel) – $\text{Tổng}_{\text{vùng trắng}}$ (các mức xám của pixel).

Viola và Joines đưa ra một khái niệm gọi là Integral Image, là một mảng 2 chiều với kích thước bằng với kích thước của ảnh cần tính đặc trưng Haar-Like, với mỗi phần tử của mảng này được tính bằng cách tính tổng của điểm ảnh phía trên (dòng-1) và bên trái (cột-1) của nó theo công thức $P(x,y) = \sum_{x' \le x, y' \le y} i(x', y')$.



Hình 2-6 Minh họa về Integral Image.

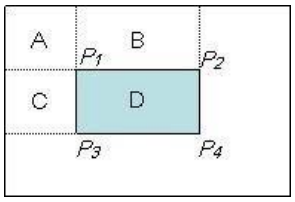
Sau khi tính được Integral Image, việc tính tổng các giá trị mức xám của một vùng bất kỳ nào đó trên ảnh thực hiện rất đơn giản theo cách sau:

Giả sử ta cần tính tổng giá trị mức xám của vùng D như hình dưới, ta có thể tính được như sau:

D = A + B + C + D – (A+B) – (A+C) + A

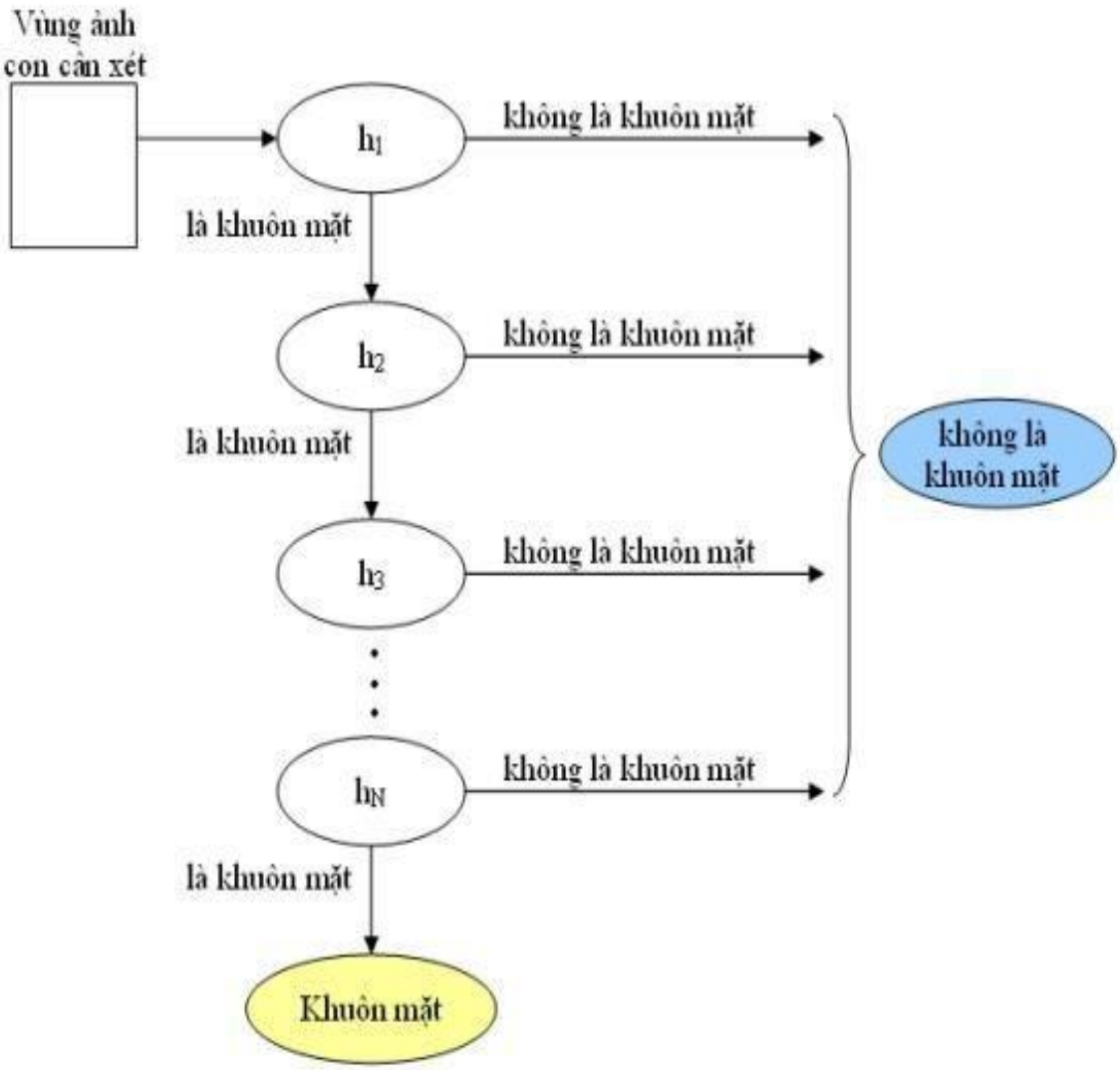
Với A + B + C + D chính là giá trị tại điểm P4 trên Integral Image, tương tự như vậy A+B là giá trị tại điểm P2, A+C là giá trị tại điểm P3, và A là giá trị tại điểm P1. Vậy ta có thể viết lại biểu thức tính D ở trên như sau:

$$D = \underbrace{(x4,y4)}_{A+B+C+D} - \underbrace{(x2,y2)}_{(A+B)} - \underbrace{(x3,y3)}_{(A+C)} + \underbrace{(x1,y1)}_{A}$$



2.1.2.2. AdaBoost.

AdaBoost là một bộ phân loại mạnh phi tuyến phức dựa trên hướng tiếp cận boosting được Freund và Schapire đưa ra vào năm 1995. Adaboost cũng hoạt động trên nguyên tắc kết hợp tuyến tính các weak classifiers để hình thành một trong các classifiers.



Hình 2-7 Viola và Jones dùng AdaBoost kết hợp các bộ phân loại yếu sử dụng các đặc trưng Haar-like theo mô hình phân tầng (cascade).

2.2. Nhận diện khẩu trang trên khuôn mặt.

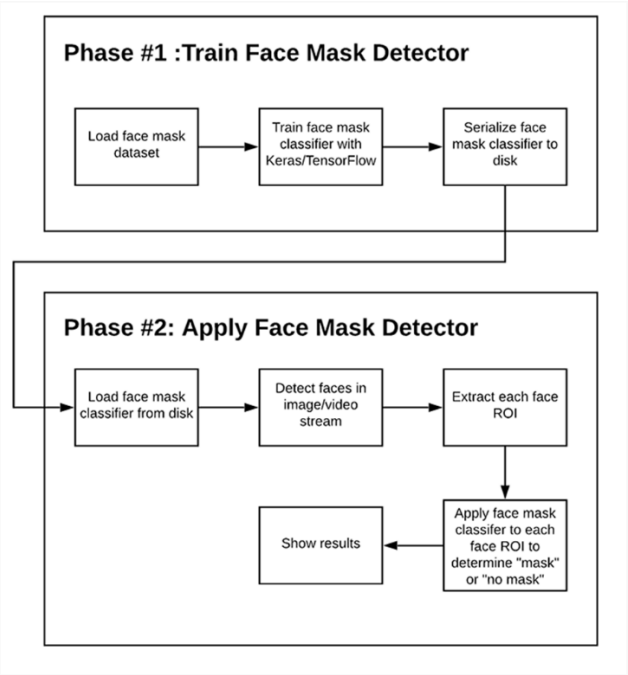
2.2.1. Bài toán.

Để máy có thể xác định khuôn mặt đeo khẩu trang qua hai giai đoạn, từ đó triển khai quy trình Học sâu (Deep Learning)/ Thị giác máy tính (Computer Vision) thì từ đó sẽ xem xét tập dữ liệu sẽ đươc sử dụng để đào tạo trình phát hiện khẩu trang tùy chỉnh. Sau đó triển khai tập lệnh Python để đào tạo trình phát hiện khẩu trang trên tập dữ liệu bằng Keras và TensorFlow [17].

Chúng ta sẽ sử dụng tập lệnh Python này để đào tạo trình phát hiện khẩu trang và xem xét kết quả. Với trình phát hiện khuôn mặt đeo khẩu trang đươc đào tạo, chúng ta sẽ tiến thành triển khai thêm hai tập lệnh Python bổ sung được sử dụng để:

- Phát hiện khuôn mặt đeo khẩu trang qua hình ảnh.
- Phát hiện khuôn mặt đeo khẩu trang qua luồng video trong thời gian thực.

Hai giai đoạn xác định khuôn mặt đeo khẩu trang:



Hình 2-8 Các giai đoạn và các bước riêng lẻ để xây dựng máy nhận diện khuôn mặt đeo khẩu trang với Thị giác máy tính (Computer Vision – CV) và Học sâu (Deep Learning – DL) sử dụng Python, OpenCV và TensorFlow/Keras.

Để đào tạo một máy nhận diện mặt na tùy chỉnh, chúng ta cần chia dự án của mình thành hai giai đoạn riêng biệt, mỗi giai đoan có các bước phụ tương ứng riêng (được thể hiện như trong Hình 2-8).

Đào tạo: Tập trung vào việc tải tập dữ liệu phát hiện khuôn mặt đeo khẩu trang từ đĩa, đào tạo mô hình (sử dụng Keras/TensorFlow) trên tập dữ liệu, sau đó là tuần tự hóa trình phát hiện khuôn mặt đeo khẩu trang vào đĩa.

Triển khai: Tải trình phát hiện khuôn mặt đeo khẩu trang, thực hiện phát hiện khuôn mặt và sau đó phân loại từng khuôn mặt vào hai thư mục with_mask hoặc without_mask.

2.2.2. Tình hình nghiên cứu về bài toán nhân diện khẩu trang trên khuôn mặt trong và ngoài nước.

2.2.2.1. Ngoài nước.

Thế giới đang đối mặt với sự lây lan chóng mặt của virus corona (COVID-19). Theo như World Health Organization (WHO) đã thông báo cách hiệu quả nhất để chống lại COVID-19 chỉ đơn giản là hãy mang một cái khẩu trang khi ra ngoài được hay những nơi đông người. Và sẽ là một điều không thể cho nhưng người quan sát những khu với số lượng người quá đông. Thế là chúng ta nảy ra ý tưởng dùng máy học để có thể tạo ra mô hình với kỹ thuật học tên là transfer learning để có thể phát hiện những người không đeo khẩu trang và xác định danh tính của họ thông qua camera giám sát [18].

Với một model tên là SSDMNV2 sử dụng những thư viện như OpenCV Deep Neural Network (DNN), TensorFlow, Keras và MobileNetV2.Model SSDMNV2 đã được so sánh với những model trước đo như LeNet-5, AlexNet, VGG-16 và ResNet-50 thông sử dụng cùng một bộ dataset và đã cho ra độ chính xác cao hơn nhưng model trước đây. Không chỉ hơn về kết quả SSDMNV2 model còn rất dễ cài đặt trên nhiều thiết bị mà không yêu cầu quá nặng về mặt cấu hình [19].

Để có thể ngăn chặn sự lây lan của COVID 19, chúng tôi có ý tưởng tạo nên một hệ thống nhận diện khẩu trang cài đặt trên thiết bị di động sử dụng thuật toán K Nearest Neighbor (KNN). Và độ chính xác đưa ra từ hệ thống của chúng tôi chạy trên một bộ test dataset là 82.87%. Ý tưởng tạo một hệ thống nhận diện khẩu trang trên thiết bị di động là một ý tưởng có chiều sâu khi mà nó thật sự tiện lợi khi ta một camera nhỏ gọn nhận diện khẩu trang ngay trong túi của mình và rất có thể đóng một vai trò quan trọng trong nhiệm vụ phòng chống COVID-19 [20].

2.2.2.2. Trong nước.

Hệ thống cảnh báo đeo khẩu trang phòng chống covid 19 đơn giản với OpenCV và Keras. Hệ thống này sẽ nhận diện một người có mang khẩu trang hay ko từ đó đưa ra những cảnh báo phù hợp. Ưu điểm hệ thống hoạt động mượt mà chạy được realtime trên các thiết bị cấu hình thấp như Raspberry Pi, Jetson Nano, MobilePhone với độ chính sát tương đối cao. Nhược điểm chỉ có thể nhận diện được khẩu trang y tế thông dụng màu xanh hoặc xám [21].

Chung tay chống COVID-19, làm thử hệ thống phát hiện khẩu trang và nhắc nhở nếu không đeo bằng OpenCV (Mì AI) [22]. Hệ thống thông qua camera giám sát kiểm tra xem người dùng có đeo khẩu trang hay ko để thông báo lên màn hình có thể kết hợp với thông báo qua loa bên ngoài để cảnh báo, hoặc thông báo cho lực lượng bảo vệ tòa

nhà và người dùng. Ưu điểm đơn giản dễ thực hiện. Nhược điểm chỉ nhận diện được 1 số loại khẩu trang.

Việt Nam nghiên cứu thành công công nghệ nhận diện khi đeo khẩu trang [23].

NDĐT - Ngày 20-4, Viện Nghiên cứu Trí tuệ nhân tạo VinAI Research (thuộc Tập đoàn Vingroup) công bố đã nghiên cứu thành công công nghệ nhận diện khuôn mặt chính xác và ổn định cả khi sử dụng khẩu trang, trở thành một trong những đơn vị đầu tiên trên thế giới nghiên cứu thành công công nghệ này. Công nghệ này có thể nhận diện chính sát khuôn mặt của người dùng dù người đó có đeo khẩu trang. Ngoài ra nó còn được tích hợp tính năng giám sát đeo khẩu trang trên camera hỗ trợ tích.

2.3. Môi trường lập trình.

2.3.1. Ngôn ngữ Python.

Hình 2-9 Logo cảu ngôn ngữ Python.

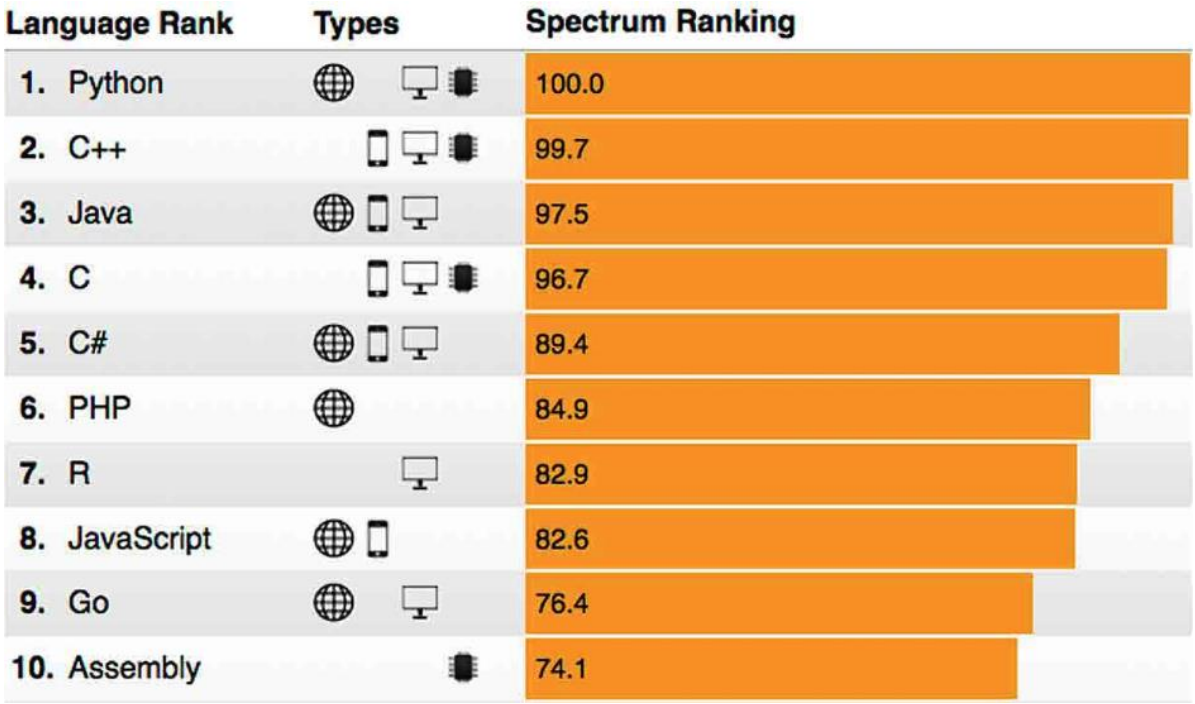
Python là một ngôn ngữ lập trình thông dịch, hướng đối tượng, cấp cao với ngữ nghĩa động được thiết kế bởi Guido van Rossum. Python thiết kế vào cuối những năm 1980 và được phát hành lần đầu tiên vào tháng 2 năm 1991, với quy tắc thiết kế của nó rất thuận tiện cho việc đọc hiểu code, đơn giản và rõ ràng.

Cấu trúc dữ liệu tích hợp sẵn ở cấp độ cao, kết hợp với tính năng gõ động và liên kết động, làm cho nó trở nên hấp dẫn đối với Phát triển ứng dụng nhanh, cũng như được sử dụng như một ngôn ngữ kịch bản hoặc ngôn ngữ keo để kết nối các thành phần hiện có với nhau. Cú pháp đơn giản, dễ học của Python nhấn mạnh tính dễ đọc và do đó giảm chi phí bảo trì chương trình. Python hỗ trợ các mô-đun và các gói, khuyến khích mô-đun chương trình và tái sử dụng mã. Trình thông dịch Python và thư viện chuẩn mở rộng có sẵn ở dagnj nguồn hoặc nhị phân miễn phí cho tất cả các nền tảng chính và có thể được phân phối miễn phí.

Thông thường, các lập trình viên yêu thích Python vì sự gia tăng năng suất công việc mà nó mang lại. Vì không có bước biên dịch, chu trình chỉnh sửa - kiểm tra – gỡ lỗi diễn ra cực kỳ nhanh chóng. Gỡ lỗi các chương trình Python rất dễ dàng: một lỗi hoặc đầu vào không tốt sẽ không bao giờ gây ra lỗi phân đoạn. Thay vào đó, khi trình thông dịch phát hiện ra lỗi, nó sẽ tạo ra một ngoại lệ. Khi chương trình không bắt được

ngoại lệ trình thông dịch sẽ in một dấu vết ngăn xếp. Trình gỡ lỗi cấp nguồn cho phép kiểm tra các biến cục bộ và toàn cục, nhằm đánh giá các biểu thức tùy ý, thiết lập các điểm ngắt, lướt qua mã một dòng tại một thời điểm, v…v… Trình gỡ lỗi được viết bằng chính Python, minh chứng cho sự ảnh hưởng bên trong chính Python. Mặt khác, cách nhanh nhất để gỡ lỗi một chương trình là thêm một vài câu lệnh in vào nguồn: chu trình sửa - kiểm tra – gỡ lỗi nhanh làm cho cách tiếp cận đơn giản này rất quả. Cộng đồng người sử dụng ngôn ngữ này rất đông, nếu so sánh từ bảng xếp hạng các ngôn ngữ năm 2018 thì Python đã leo lên vị trí số 1 trên bảng xếp hạng những ngôn ngữ lập trình phổ biến.

Hiện nay, Python cũng là một ngôn ngữ rất phát triển trong lĩnh vực Data Science và Machine Learning. Python cũng cung cấp những hàm và thư viện xử lý hình ảnh. Chính vì vậy, Python là một lựa chọn hợp lý khi thực hiện xử lý hình ảnh và nhận diện khuôn mặt.



Hình 2-10 IEEE Spectrum The Top Programming languages 2018.

2.3.2. Thư viện Tensorflow.

Hình 2-11 Logo của thư viện Tensorflow.

Được tạo bởi nhóm Google Brain, TensorFlow là một thư viện mã nguồn mở để tính toán số và học máy quy mô lớn. TensorFlow kết hợp một loạt các mô hình và thuật toán học máy và học sâu (hay còn gọi là mạng Neural) và làm cho chúng trở nên hữu ích bằng một phép ẩn dụ thông thường. Nó sử dụng Python để cung cấp một API front-end thuận tiện cho việc xây dựng các ứng dụng với framework, đồng thời thực thi các ứng dụng đó bằng C++ hiệu suất cao.

Với sự giúp đỡ của thư viện Tensorflow mà Máy học – bộ môn phức tạp đã trở nên thuận tiện hơn qua việc triển khai các mô hình máy học nhờ các khuôn khổ máy học, chẳng hạn như Google’s TensorFlow, đơn giản hóa quá trình thu thập sữ liệu, mô hình đào tạo, cung cấp các dự đoán và tinh chỉnh các kết quả trong tương lai

TensorFlow có thể đào tạo và chạy các mạng neural sâu để phân loại chữ số viết tay, nhận hình dạng, nhúng từ, mạng neural lặp lại, mô hình tuần tự để dịch máy, xử lý ngôn ngữ tự nhiên và mô phỏng dựa trên PDE (phương trình vi phân từng phần). Hơn hết, TensorFlow hỗ trợ dự đoán sự sản xuất trên quy mô lớn, các mô hình tương tự được sử dụng để đào tạo.

Tesorflow kết hợp các mô hình và thuật toán Machine Learning và Deep Learning lại với nhau và chạy trên Python, giúp việc tiếp cận các bài toán trở nên đơn giản, nhanh chóng và tiện lợi hơn nhiều.

Cho phép tạo ra một đồ thị tính toán để thực hiện. Mỗi một đỉnh trong biểu đồ đại diện cho một phép toán và mỗi kết nối đại diện cho dữ liệu, giúp có cái nhìn trực quan hơn về lối logic của bài toán.

Các hàm được dựng sẵn trong thư viện cho từng bài toán cho phép TensorFlow xây dựng được nhiều neural network. Nó còn cho phép tính toán song song trên nhiều máy tính khác nhau, thậm chí trên nhiều CPU, GPU trong cùng một máy hau tạo ra các dataflow graph – đồ thị luồng dữ liệu để dựng nên các model.

Để cài đặt Tensorflow trên môi trường Python, cài đặt trực tiếp thông qua pip.

pip install tensorflow

2.3.3. Thư viện Numpy.

Hình 2-12 Logo của thư viện Numpy.

NumPy là một gói (package) cơ bản cho tính toán khoa học bằng Python. Nó là một thư viện Python cũng cấp một đối tượng mảng đa chiều, các đối tượng dẫn xuất khác nhau (chẳng hạn như các mảng và ma trận như mặt nạ) và một loạt các quy trình cho các hoạt động nhanh trên mảng, bao gồm toán học, logic, thao tác hình dạng, sắp xếp, lựa chọn, I/O, các phép biến đổi Fourier rời rạc, đại số tuyến tính cơ bản, các phép toán thống kê cơ bản, mô phỏng ngẫy nhiên và hơn thế nữa.

Cốt lõi của gói (package) NumPy là đối tượng ma trận. Nó đóng gói các mảng n – chiều của các kiểu dữ liệu đồng nhất, với nhiều hoạt động được thực hiện trong mã đã được biên dịch để thực hiện. Sau đây là một số sự khác nhau quan trọng giữa NumPy và trình tự Python tiêu chuẩn:

+ Mảng NumPy có kích thước cố định khi tạo, không giống nhưng các danh sách Python (có thể phát động). Thay đổi kích thước của một ndarray sẽ tạo ra một mảng mới và xóa bản gốc.

+ Tất cả các phần thử trong mảng NumPy đều được yêu cầu phải có cùng một kiểu dữ liệu và do đó sẽ có cũng kích thước bộ nhớ. Ngoại trừ người ta có thể có các mảng đối tượng (Python, bao gồm NumPy), do đó cho pháp các mảng có các phần tử có kích thước khác nhau.

+ NumPy hỗ trợ toán học nâng cao và các loại phép toán khác trên lượng dữ liệu lớn. Thông thường, các mã hoạt động như vậy được thực thi hiệu quả hơn và ít mã hơn có thể sử dụng bằng các trình tự tích hợp của Python.

+ Ngày càng có nhiều gói (package) phát triển dựa trên Python về khoa học và toán học đang sử dụng mảng NumPy; mặc dù chúng thường hỗ trợ đầu vào theo trình tự Python, chúng chuyển đổi đầu vào như vậy thành mảng NumPy trước khi xử lý và chúng thường xuất ra mảng NumPy. Nói cách khác, để sử dụng hiệu quả (thậm chí có thể là hầu hết) phần mềm khoa học/ toán học ngày nay dựa trên Python, chỉ cần biết cách sử dụng các loại trình tự tích hợp của Python là không đủ – người ta cũng cần biết cách sử dụng mảng NumPy.

Để cài đặt Numpy trên môi trường Python, có thể cài đặt trực tiếp thông qua pip.

pip install numpy

Nếu sử dụng conda:

Sử dụng environment thay vì cài đặt trong base env

conda create -n my-env

conda activate my-env

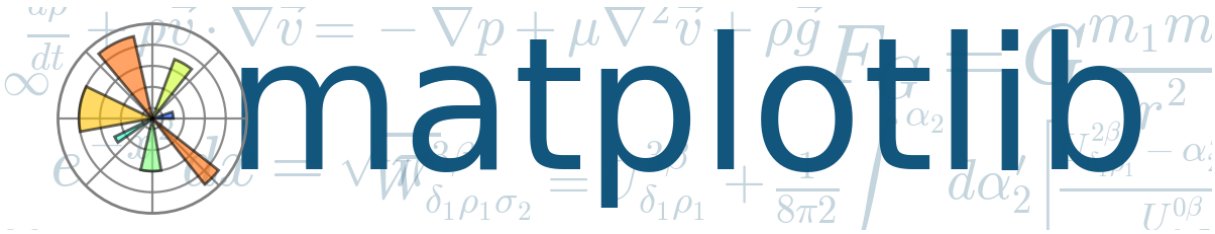
Muốn cài đặt từ conda-forge

conda config --env --add channels conda-forge

Sau đó nhập lệnh cài đặt trên terminal

conda install numpy

2.3.4. Thư viện Matplotlib.



Hình 2-13 Logo của thư viện matplotlib.

Matplotlib là một thư viện vẽ đồ thị cấp thấp trong Python đóng vai trò như một tiện ích trực quan hóa được tạo ra bởi John D. Hunter vào năm 2002.

Matplotlib là mã nguồn mở và chúng ta có thể sử dụng nó một cách tự do. Nó cũng là ngôn ngữ hướng đối tượng và có thể đươc sử dụng với bộ công cụ GUI mục đích chung như wxPython, Qt và GTK+. Matplotlib chủ yếu được viết bằng Python, một vài phân đoạn được viết bằng C, Objective-C và Javascript để tương thích với nền tảng.

Matplotlib - thư viện trực quan hóa tuyệt vời cho các mảng 2D trong Python. Matplotlib là một thư viện trực quan hóa dữ liệu nên tảng được xây dựng trên mảng NumPy và được thiết kế để hoạt động với ngăn xếp SciPy rộng lớn hơn.

Một trong những lợi ích lớn nhất của trực quan hóa là nó cho phép chúng ta truy cập trực quan vào lượng dữ liệu khổng lồ dưới dạng hình ảnh dễ đồng hóa. Matplotlib bao gồm một số biểu đồ như đường thẳng, thanh, phân tán, biểu đồ, v…v…

Matplotlib có thể được sử dụng để tạo ra những số liệu (figures) đủ chất lượng cho một loạt các định dạng bản cứng (hardcopy) và môi trường tương tác trên nền tảng. Không những thế hỗ trợ rất mạnh mẽ hữu ích trong việc vẽ đồ thị cho những người làm việc với Python và Numpy

Để cài đặt Numpy trên môi trường Python

python -m pip install -U pip

python -m pip install -U matplotlib

Để nhập thư viện Matplotlib (Importing matplotlib):

from matplotlib import pyplot as plt

hoặc

import matplotlib.pyplot as plt

2.3.5. Thư viện Keras.

Hình 2-14 Logo của thư viện Keras.

Keras là một thư viện cấp cao được xây dựng dựa trên Theano hoặc TensorFlowđược viết bằng Python và phát triển vào năm 2005 bởi Francios Chollet - kỹ sư nghiên cứu Deep Learning . Chạy dễ dàng trên cả CPU và GPU, nó cung cấp một API loại scikit-learning (được viết bằng Python) để xây dựng Mạng thần kinh (Neural Networks).

Hơn nữa, các mô hình này có thể được kết hợp để xây dựng các mô hình phức tạp hơn. Keras, có bản chất mô-đun, cực kỳ nhanh chóng, linh hoạt và thích nghiên cứu đổi mới. Song còn là một khung hoàn toàn dựa trên Python giúp bạn dễ dàng gỡ lỗi và khám phá

Các nhà phát triển có thể sử dụng Keras để nhanh chóng xây dựng mạng nơ-ron mà không cần lo lắng về các khía cạnh toán học của đại số tensor, kỹ thuật số và phương pháp tối ưu hóa.

Ý tưởng quan trọng đằng sau sự phát triển của Keras là tạo điều kiện thuận lợi cho các thí nghiệm bằng cách tạo mẫu nhanh. Khả năng đi từ một ý tưởng đến kết quả với thời gian trì hoãn ít nhất có thể là phương án để nghiên cứu tốt.

Điều này mang lại một lợi thế to lớn cho các nhà khoa học cũng như các nhà phát triển mới bắt đầu vì họ có thể tham gia ngay vào Học sâu (Deep Learning) mà không phải làm việc với các phép tính cấp thấp. Sự gia tăng nhu cầu về Học sâu (Deep Learning) đã dẫn đến sự gia tăng nhu cầu về những người có kỹ năng về Học sâu (Deep Learning).

Mọi cách cấu tạo đều đang cố gắng kết hợp Học sâu theo cách này hay cách khác và Keras cung cấp một API rất dễ sử dụng cũng như đủ trực quan để hiểu về cơ bản giúp bạn kiểm tra và xây dựng các ứng dụng Học sâu ít khó khan nhất. Điều này là tốt vì nghiên cứu Học sâu (Deep Learning) là một chủ đề nóng hiện nay và các nhà khoa học cần một công cụ để thử các ý tưởng của họ mà không mất thời gian vào việc đưa ra một mô hình Mạng thần kinh (Neural Network).

Cài đặt

Bước 1: Cài thư viện tensorflow

pip install tensorflow

Bước 2: import thư viện tensorflow trong code

import tensorflow as tf

from tensorflow import keras

2.3.6. Thư viện OpenCV-Python.

Hình 2-15 Logo của thư viện OpenCV-Python.

OpenCV được bắt đầu tại Intel vào năm 1999 bởi Gary Bradsky, và bản phát hành đầu tiên ra mắt vào năm 2000. Vadim Pisarevsky cùng với Gary Bradsky quản lý nhóm OpenCV phần mềm của Intel. Năm 2005, OpenCV được sử dụng trên Stanley và đã giành chiến thắng trong cuộc thi DARPA Grand Challenge năm 2005. Sau đó, sự phát triển tích cực của nó tiếp tục dưới sự hỗ trợ của Willow Garage với Gary Bradsky và Vadim Pisarevsky dẫn đầu dự án. OpenCV hiện hỗ trợ vô số thuật toán liên quan đến Thị giác máy tính và Học máy và đang mở rộng từng ngày.

OpenCV hỗ trợ nhiều ngôn ngữ lập trình như C ++, Python, Java, v.v. và có sẵn trên các nền tảng khác nhau bao gồm Windows, Linux, OS X, Android và iOS. Các giao

diện cho hoạt động GPU tốc độ cao dựa trên CUDA và OpenCL cũng đang được phát triển tích cực.

OpenCV-Python là API Python của OpenCV, kết hợp những tiêu chuẩn tốt nhất của OpenCV C ++ API và ngôn ngữ Python.

OpenCV-Python là một thư viện liên kết Python được thiết kế để giải quyết các vấn đề về thị giác máy tính. hỗ trợ rất hiệu quả trong xử hình ảnh, quay video, phân tích và có cả các tính năng như nhận diện khuôn mặt và phát hiện đối tượng.

Có thể sử dung nhiều ngôn ngữ lập trình khác nhau để làm việc với OpenCV như C++, Java, Python, C#, …

OpenCV-Python sử dụng Numpy, là một thư viện được tối ưu hóa cao cho các phép toán số với cú pháp kiểu MATLAB. Tất cả các cấu trúc mảng OpenCV được chuyển đổi sang và từ mảng Numpy. Điều này cũng giúp dễ dàng tích hợp với các thư viện khác sử dụng Numpy như SciPy và Matplotlib.

Để cài đặt OpenCV-Python trên môi trường Python, có thể cài đặt trực tiếp thông qua pip.

```
pip install opencv-python
```

2.3.7. Thư viện Imutils.

Imutils là một thư viện cung cấp một loạt các hàm có sẵn nhằm cung cấp các chức năng sử lý hình ảnh cơ bản như: dịch hình ảnh, xoay ảnh, điều chỉnh kích thước ảnh, xây dựng cấu trúc liên kết của đối tượng trong hình ảnh, hỗ trợ hiện thị hình ảnh Matplotlib, phân loại đường viền, phát hiện các cạnh và còn nhiều tính năng khác, sử dụng dễ dàng với OpenCV và cả Python.

Để cài đặt Numpy trên môi trường Python, có thể cài đặt trực tiếp thông qua pip.

```
pip install imutils
```

2.3.8. Thư viện SciPy.

Hình 2-16 Logo của thư viện SciPy.

Scipy (Scientific Python) là một thư viện tính toán khoa học sử dụng Numpy, đươc tạo ra bởi Travis Olliphant (người tạo ra NumPy). Kể từ lần phát hành đầu tiên

vào năm 2001, SciPy đã trở thành tiêu chuẩn thực tế để tận dụng các thuật toán khoa học trong Python, với hơn 600 người đóng góp mã duy nhất, hàng nghìn gói phụ thuộc, hơn 100.000 kho phụ thuộc và hàng triệu lượt tải xuống mỗi năm.

SciPy cung cấp nhiều chức năng tiện ích hơn để tối ưu hóa, thống kê và xử lý tín hiệu. Nó cho phép người dùng thao tác dữ liệu và trực quan hóa dữ liệu bằng một loạt các lệnh Python cấp cao. SciPy được xây dựng dựa trên mở rộng Python NumPy. SciPy cũng được phát âm là "Sigh Pi."

Thư viện SciPy là một trong những nền tảng cơ bản để tạo nên ngăn xếp SciPy (SciPy stack). Nó sẽ cung cấp nhiều tính năng nhằm hỗ trợ cho việc phân tích và tính toán các vấn đề liên quan với số học như: tích phân, nội suy, tối ưu hóa, đại số tuyến tính, số liệu thống kê, ...

Giống như NumPy, SciPy là mã nguồn mở và được cấp phép BSD, nên chúng ta có thể thoải mái sử dụng. Thư viện SciPy phụ thuộc vào thư viện NumPy, do đó việc học các kiến thức cơ bản về NumPy sẽ giúp việc hiểu dễ dàng.

Cả NumPy và SciPy đều là các thư viện Python được sử dụng để phân tích toán học và số được sử dụng. NumPy chứa dữ liệu mảng và các hoạt động cơ bản như sắp xếp, lập chỉ mục, v.v. trong khi, SciPy bao gồm tất cả các mã số. Mặc dù NumPy cung cấp một số hàm có thể giúp giải quyết đại số tuyến tính, biến đổi Fourier, v.v., SciPy là thư viện thực sự chứa các phiên bản đầy đủ tính năng của các hàm này cùng với nhiều hàm khác. Tuy nhiên, nếu bạn đang thực hiện phân tích khoa học bằng Python, bạn sẽ cần cài đặt cả NumPy và SciPy vì SciPy được xây dựng dựa trên NumPy.

Để cài đặt Numpy trên môi trường Python, có thể cài đặt trực tiếp thông qua lệnh cài đạt python: pip.

python -m pip install --user numpy scipy matplotlib ipython jupyter pandas sympy nose.

2.3.9. Thư viện Scikit-Learn (Sklearn).

Hình 2-17 Logo của thư viện Scikit_Learn.

Scikit-learning là một thư viện phân tích dữ liệu mã nguồn mở và là tiêu chuẩn vàng cho Học máy (ML) trong hệ sinh thái Python.

Ban đầu nó được gọi là scikits-learn, được phát triển bởi David Cournapeau như một dự án mã mùa hè của Google vào năm 2007. Sau đó, vào năm 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort và Vincent Michel, từ FIRCA (Viện Nghiên cứu Pháp tại Khoa học máy tính và Tự động hóa), đã đưa dự án này ở một cấp độ khác và phát hành bản phát hành công khai đầu tiên (v0.1 beta) vào ngày 1 tháng 2 năm 2010.

Scikit-learning có lẽ là thư viện hữu ích nhất cho máy học (Machine Learning) trong Python. Thư viện sklearn chứa rất nhiều công cụ hiệu quả cho máy học (Machine Learning) và lập mô hình thống kê bao gồm phân loại, hồi quy, phân cụm và giảm kích thước. Thư viện này, phần lớn được viết bằng Python, được xây dựng dựa trên NumPy, SciPy và Matplotlib.

Scikit-learning cung cấp một loạt các thuật toán học có giám sát và không giám sát thông qua một giao diện nhất quán trong Python.

Nó được cấp phép theo giấy phép BSD, đơn giản hóa cho phép và được phân phối dưới nhiều bản phân phối Linux, sử dụng trong học thuật và thương mại.

Scikit-learning đều được ghi chép đầy đủ và dễ học và dễ sử dụng. Là một thư viện cấp cao, nó cho phép bạn xác định mô hình dữ liệu dự đoán chỉ trong một vài dòng mã, sau đó sử dụng mô hình đó để phù hợp với dữ liệu của bạn. Nó linh hoạt và tích hợp tốt với các thư viện Python khác, chẳng hạn như matplotlib để vẽ biểu đồ, numpy để vector hóa mảng và pandas cho khung dữ liệu.

Nếu cài đặt NumPy và Scipy, sau đây là hai cách dễ nhất để cài đặt scikit-learning:

Cài đặt scikit-learn thông qua pip –

```
pip install -U scikit-learn
```

Cài đặt scikit-learn thông qua conda –

```
conda install scikit-learn
```

CHƯƠNG 3. THỰC NGHIỆM CHƯƠNG TRÌNH

3.1. Cài đặt.

Cấu hình máy thực hiện

Bảng 3-1 Cấu hình máy để huấn luyện.

	Máy 1: Acer Aspire 7	**Máy 2: Asus Vivobook x510u**
CPU	Intel core I5 8300H	Intel Core i5 8250U
GPU	NVDIA GTX 1050	NVIDIA GeForce 940MX
RAM	DDR4 8GB	DDR4 8GB
Ổ cứng	Ổ cứng: SSD 256GB	HDD 1TB

Phiên bản Python 3.8.2

IDE: Pycharm community edition 2020.2.2

3.2. Dữ liệu

3.2.1. Yêu cầu dữ liệu

Dữ liệu để kiểm tra là phải cho máy tính nhìn thấy được trên 2/3 khuôn mặt có đeo hoặc không đeo khẩu trang mà không phải là một đối tượng nào khác và trong điều kiện không thiếu sáng, hoặc ngược sáng khi quay video trực tiếp trên không gian thực, thiết bị ghi hình phải bảo có chất lượng hình ảnh trên 360p.

Hình 3-1 Ví dụ một số dữ liệu đã đặt yêu cầu.

Hình 3-2 Ví dụ một số dữ liệu không đặt yêu cầu.

Do yêu cầu đầu ra là kết quả "Có" hoặc "Không" nên gắn nhãn dữ liệu thủ công bằng cách phân chia 2 thư mục lưu trữ là *with_mask; without_mask* được lưu trong thư mục *dataset*.

Với quy ước gắn nhãn cho ảnh theo cấu trúc: Attribute_Status (nếu có)_Index

Trong đó:

Attribute: nhận giá trị mask nếu có khẩu trang, hoặc nhận giá trị notmak nếu không khẩu trang.

Status: nhận giá trị multi nếu trong ảnh có nhiều người.

Index: là số thứ tự ảnh với 4 số. Ví dụ: 0001

3.2.2. Thống kê dữ liệu

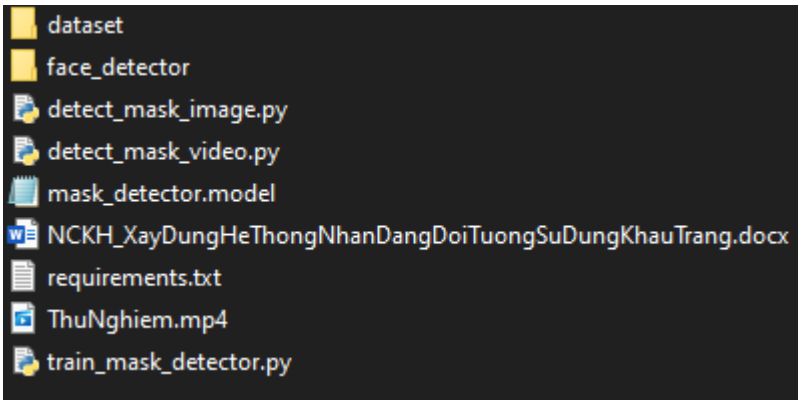
Bộ dữ liệu: Sriman_Mitra và sưu tập từ cá nhân của các thành viên trong nhóm.

Ảnh chụp tổ chức lưu trữ: một thư mục tên *dataset* lưu trữ hai thư mục là *with_mask* (chứa các hình ảnh có khẩu trang, số lượng: 2059 hình ảnh); *without_mask* (chứa các hình ảnh không có khẩu trang, số lượng: 1997 hình ảnh).

Bảng 3-2 Thống kê số lượng hình ảnh training và testing.

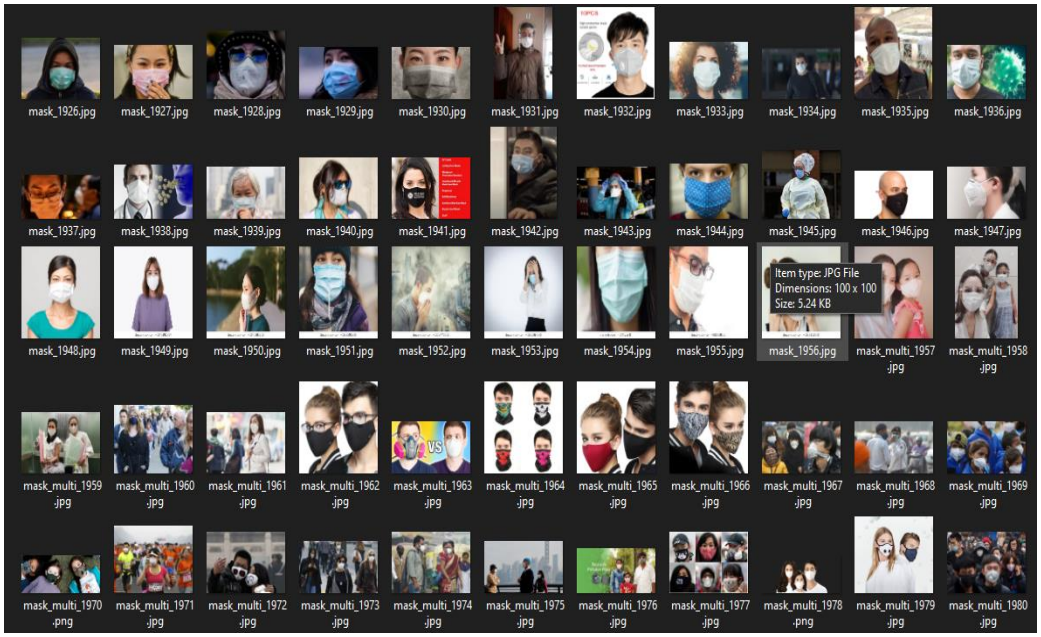
	Training (80%)	Testing (20%)
Có khẩu trang	1647	412
Không khẩu trang	1598	399

Cấu trúc thư mục.

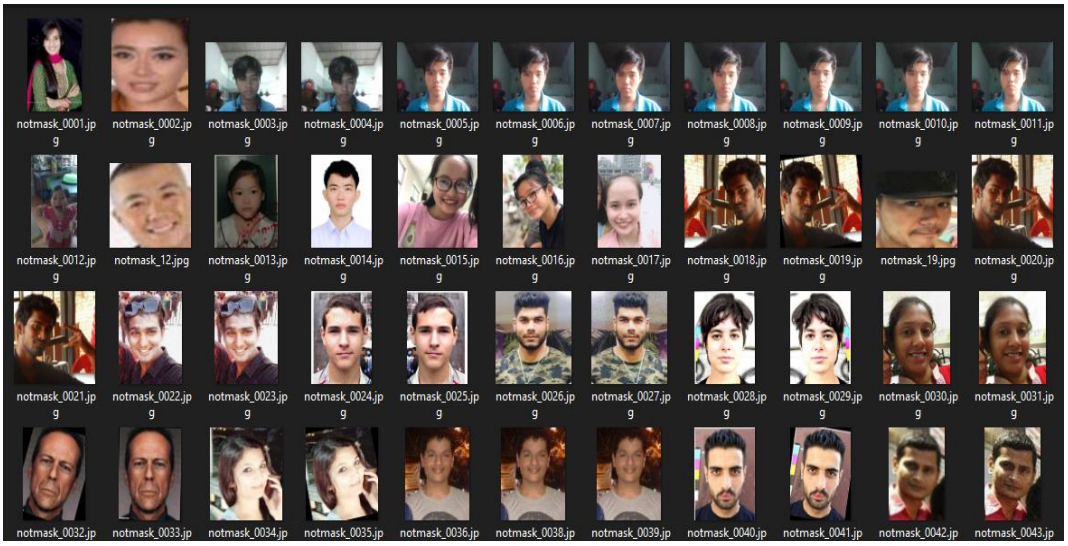


Hình 3-3 Cấu trúc thư mục của đề tài.

dataset là một thư mục lưu trữ hai thư mục là with_mask (chứa các hình ảnh có khẩu trang, số lượng: 2059 hình ảnh); without_mask (chứa các hình ảnh không có khẩu trang, số lượng: 1997 hình ảnh).



Hình 3-4 Một số hình ảnh trong thư mục with_mask.



Hình 3-5 Một số hình ảnh trong thư mục without_mask.

ThuNghiem.mp4 là đoạn video ghi lại kết quả thực thi đề tài ngoài thực nghiệm.

Train_mask_detector.py là file chính của đề tài dùng để phân tích, xây dựng nên một cấu trúc nền tảng cho hệ thống, dựa trên dữ liệu được lấy từ file dataset, kết quả thu được là file model *mask_detector. model*.

NCKH_XayDungHeThongNhanDangDoiTuongSuDungKhauTrang.docx là file word của đề tài.

Requirements.txt là một file chứa các thư viện với phiên bản cần thiết.

detect_mask_video.py là chương trình để hỗ trợ cho cho việc nhận diện khuôn mặt đeo khẩu trang trên không gian thực.

detect_mask_image.py là chương trình để hỗ trợ cho cho việc nhận diện khuôn mặt đeo khẩu trang trong một bức ảnh cho trước.

3.2.3. Phân tích dữ liệu.

3.2.3.1. Thiết lập mô hình

Đề tài sử dụng mô hình Mobilenetv - một kiến trúc mạng nơ-ron phức hợp nhằm tìm cách hoạt động tốt trên các thiết bị di động. Dung lượng và tốc độ xử lý trên các thiết bị động thường hạn chế, nên nhóm chúng em đã áp dụng mô hình vào đề tài.

Bảng 3-3 Các tham số trong mô hình

Tham số	Giá trị
INIT_LR	10^{-4}
EPOCHS	20
BS	32

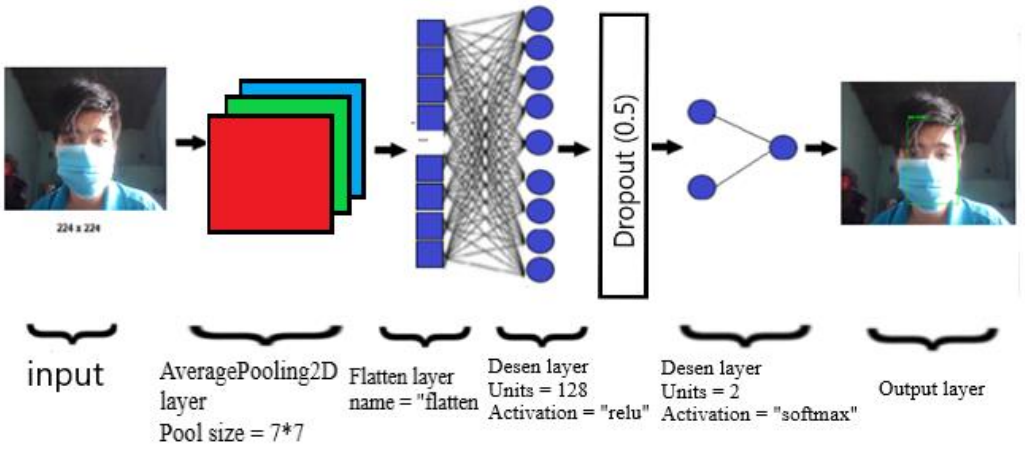
Trong đó:

Learning rate (INIT_LR) – Tốc độ học là một siêu tham số sử dụng trong việc huấn luyện các mạng nơ ron. Giá trị của nó là một số dương, thường nằm trong khoảng giữa 0 và 1. Tốc độ học kiểm soát tốc độ mô hình thay đổi các trọng số để phù hợp với bài toán. Tốc độ học lớn giúp mạng nơ ron được huấn luyện nhanh hơn nhưng cũng có thể làm giảm độ chính xác

Một *Epoch* được tính là khi chúng ta đưa tất cả dữ liệu vào mạng neural network 1 lần. Khi dữ liệu quá lớn, chúng ta không thể đưa hết mỗi lần tất cả tập dữ liệu vào để huấn luyện được. Buộc lòng chúng ta phải chia nhỏ tập dữ liệu ra thành các batch (size nhỏ hơn).

Batch size (BS) là số lượng mẫu dữ liệu trong một batch. Ở đây, khái niệm batch size và số lượng batch (number of batch) là hoàn toàn khác nhau. Như đã nói ở trên, chúng ta không thể đưa hết toàn bộ dữ liệu vào huấn luyện trong 1 epoch, vì vậy chúng ta cần phải chia tập dữ liệu thành các phần (number of batch), mỗi phần có kích thước là batch size.

Mô hình được thiết lập bao gồm 4 lớp:



Hình 3-6 Mô hình training của đề tài.

AveragePooling2D layer: là lớp sẽ gộp những chi tiết của quan trọng thông qua bài toán tính chập hai chiều [24] từ ảnh có kích thước 224 x 224 x 3 pixel xuống còn 32 x 32 x 3 pixel (Pool size = 7 * 7 tức là gộp những chi tiết quan trọng của hình ảnh sao cho kích thước chiều rộng và chiều cao giảm gấp 7 lần).

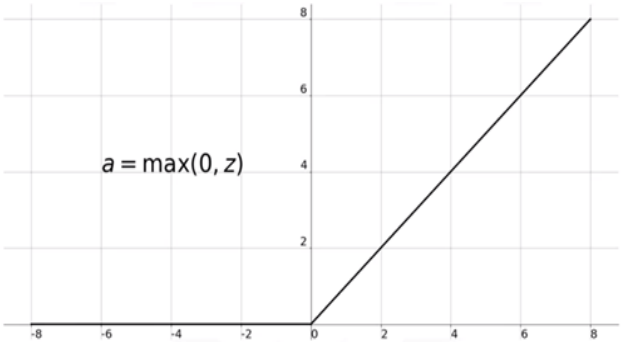
Flatten layer: là lớp chuyển đổi dữ liệu ở lớp AveragePooling2D layer thành một vector một chiều, để nhập dữ liệu vào các lớp tiếp theo [25].

Desen layer [26] : là lớp mạng nơ-ron nhân tạo trong đó mỗi nơ-ron nhận đầu vào từ tất cả nơ-ron của lớp trước đó.

+ Units là số chiều không đầu ra của lớp.

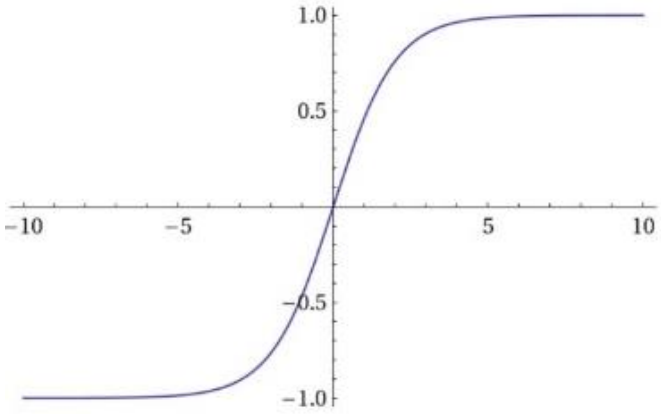
+ Activation là tham số kích hoạt (hàm tính toán) trong việc áp chức năng kích hoạt các phần tử trong vùng thỏa điều kiện tham số kích hoạt.

Relu là một tham số kích hoạt có dạng $a = \max(0, z)$.



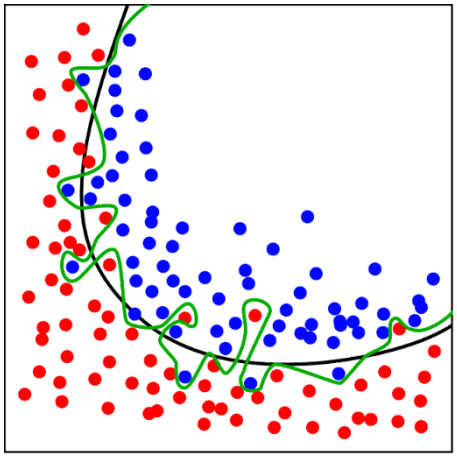
Hình 3-7 Đồ thị của hàm Relu.

Softmax là một tham số kích hoạt có dạng $a_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$



Hình 3-8 Đồ thị của hàm Softmax

Dropout layer [27]: là lớp sẽ bỏ qua một vài unit trong suốt quá trình train trong mô hình, những unit bị bổ qua được lựa chọn ngẫu nhiên, điều này sẽ chống lỗi over-fitting.

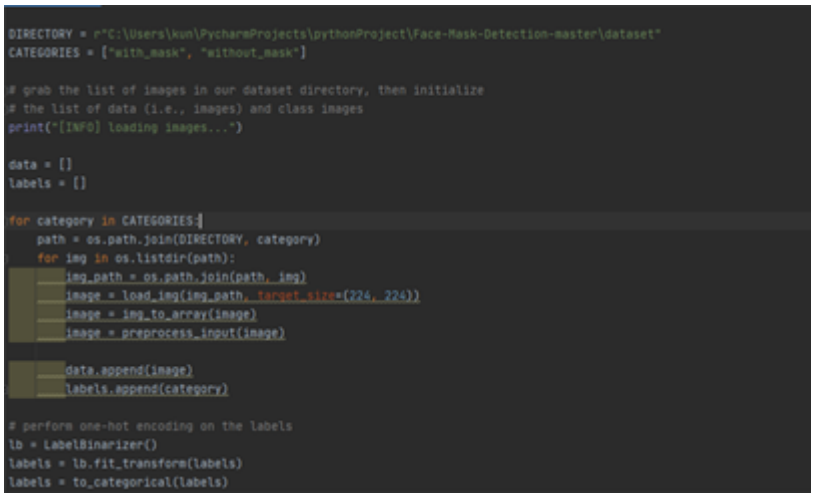


Hình 3-9 Đường màu đen là mô hình không bị lỗi Overfitting, đường màu xanh là mô hình bị lỗi Overfitting.

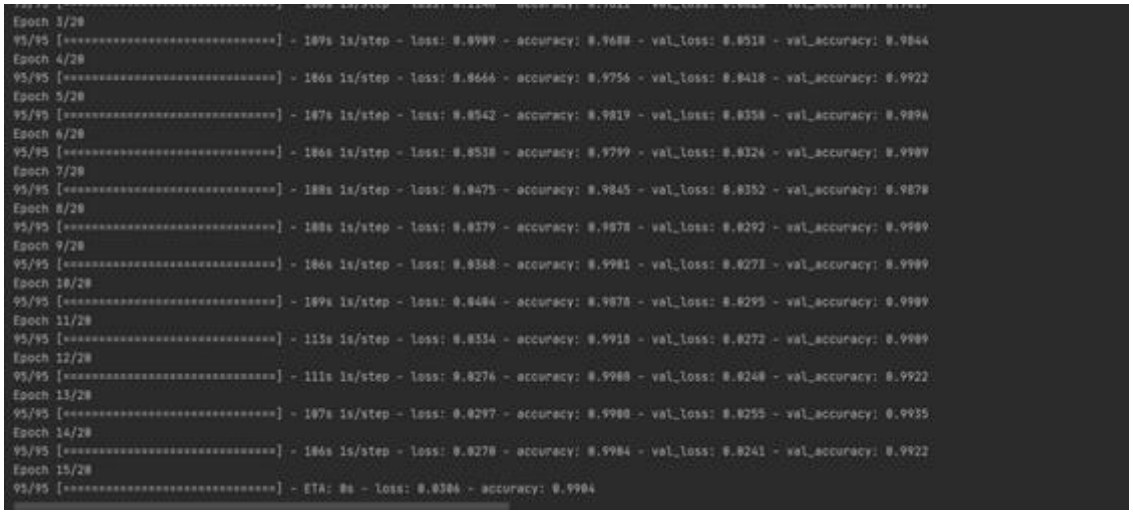
Overfitting là một lỗi mô hình hóa xảy ra khi một hàm quá phù hợp với một tập hợp giới hạn các điểm dữ liệu. Việc trang bị quá mức mô hình thường có hình thức tạo ra một mô hình quá phức tạp để giải thích các đặc điểm riêng trong dữ liệu đang nghiên cứu.

3.2.3.2. Hướng dẫn chạy mô hình

Bước 1: Mở file “train_mask_detector.py”: để huấn luyện.



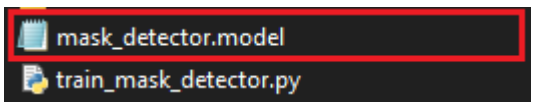
Hình 3-10 Giao diện khi mở file train_mask_detector.py bằng PyCharm (Python 3.8.2).



Hình 3-11 File " train_mask_detector.py " đang được thực thi.

Tiến hành chạy file “train_mask_detector.py”:

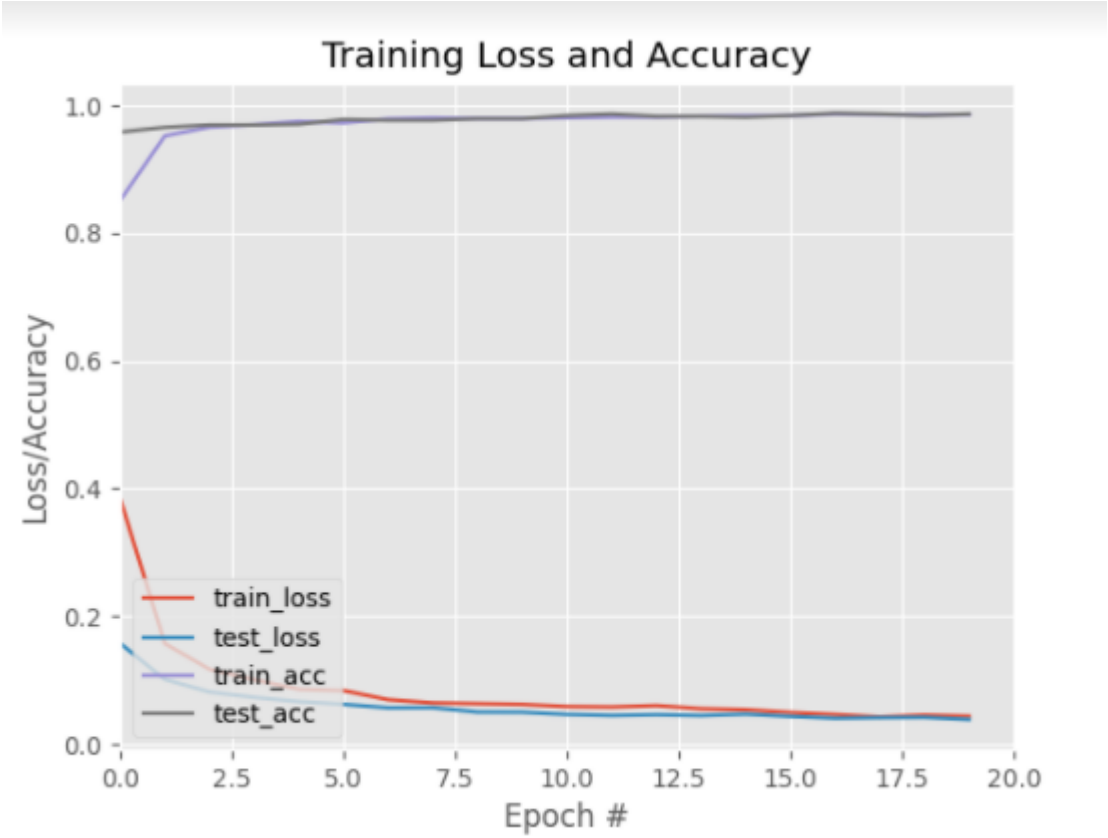
Sau khi chương trình thực thi xong, chương trình sẽ tạo một file là “mask_detector.model”.



3.2.3.3. Đánh giá bộ đo training.

	precision	recall	f1-score	support
with_mask	0.99	0.99	0.99	412
without_mask	0.99	0.99	0.99	400
accuracy			0.99	812
macro avg	0.99	0.99	0.99	812
weighted avg	0.99	0.99	0.99	812

Hình 3-12 Kết quả sau training



Hình 3-13 Kết quả sau khi traning.

Trong đó:

train_loss: dự đoán sai khi huấn luyện.

train_acc: dự đoán đúng khi huấn luyện.

test_loss: dự đoán sai khi kiểm tra.

test_acc: dự đoán sai khi kiểm tra.

Khi quan sát kết quả của bộ đo training cho thấy đường *train_acc và test_acc* gần tiến về 1.0 đồng thời *train_lost và test_lost* cũng gần tiến về 0.0, nghĩa là kết quả của việc nhận diện khuôn mặt đeo khẩu trang gấn đúng 100%.

Ngoài ra, từ kết quả ta nhận thấy kể từ epoch thứ 17 trở đi thì đường *train_acc* và *test_acc* trùng nhau tại vị trí 1.0 đồng thời đường *train_loss* và *test_loss* cũng trùng nhau tại vị trí 0.0, như vậy với mô hình này thì chạy epoch bằng 17 là đủ.

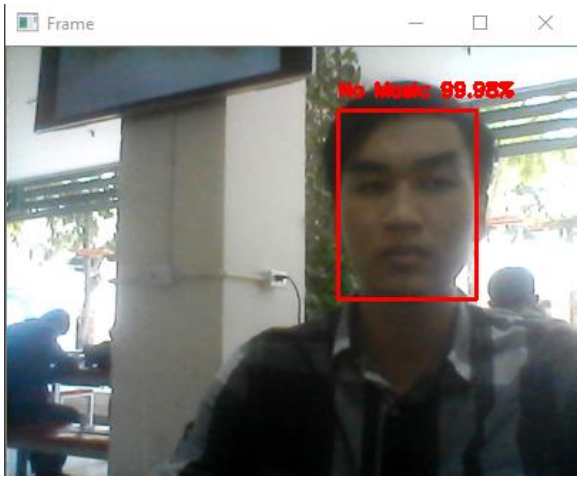
3.3. Thực thi ngoài thực tế.

Mở file "detect_mask_video.py", tiến hành thực thi chương trình trong file "detect_mask_video.py".

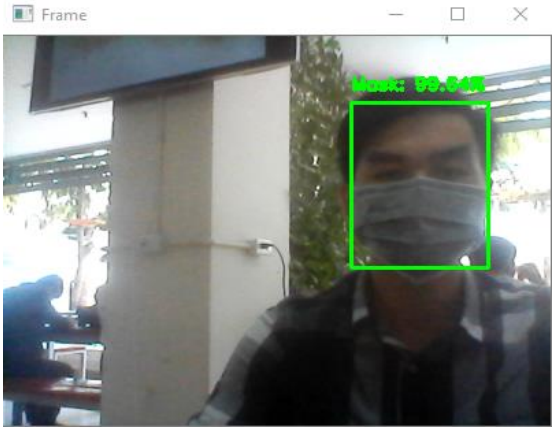
```
def detect_and_predict_mask(frame, faceNet, maskNet):
    # grab the dimensions of the frame and then construct a blob
    # from it
    (h, w) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),
        (104.0, 177.0, 123.0))

    # pass the blob through the network and obtain the face detections
    faceNet.setInput(blob)
    detections = faceNet.forward()
    print(detections.shape)

    # initialize our list of faces, their corresponding locations,
    # and the list of predictions from our face mask network
    faces = []
    locs = []
    preds = []
```

Hình 3-14 Giao diện khi mở file train_mask_detector.py bằng PyCharm (Python 3.8.2).

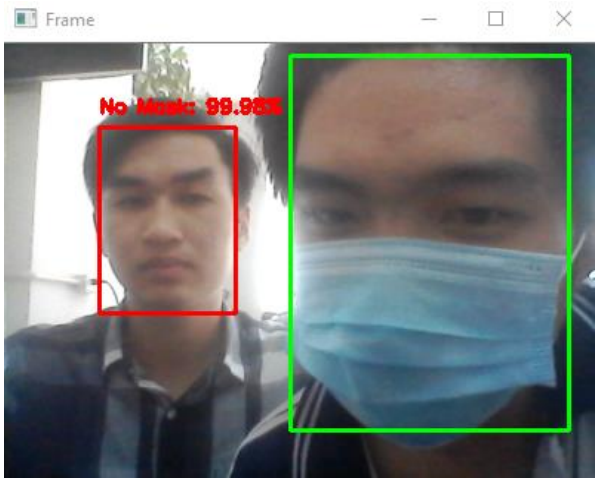
Khi chạy chương trình thì camera trước của thiết bị chạy chương trình sẽ được bật lên, và nhận dạng khuôn mặt có sử dụng khẩu trang hay không trên không gian thực.



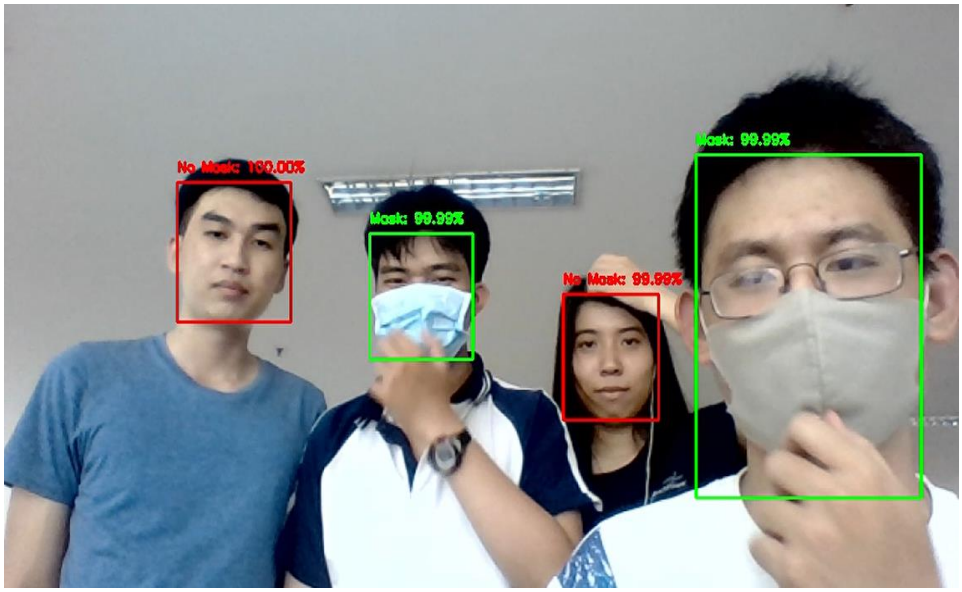
Hình 3-15 Đối tượng không sử dụng khẩu trang



Hình 3-16 Đối tượng sử dụng khẩu trang



Hình 3-17 Một đối tượng sử dụng khẩu trang và một đối tượng không sử dụng khẩu trang.



Hình 3-18 Thực nghiệm kiểm tra đeo khẩu trang một lúc nhiều đối tượng..

CHƯƠNG 4. KẾT LUẬN

4.1. Kết luận.

Nghiên cứu đã trình bày về vấn đề nhận dạng đối tượng sử dụng khẩu trang, một bài toán quan trọng trong lĩnh vực xử lý hình ảnh. Luận án tập trung nghiên cứu, phát triển về lý thuyết và ứng dụng đối với bài toán nhận dạng đối tượng, đề xuất một số mô hình và giải pháp nhằm nâng cao hiệu quả nhận dạng đối tượng sử dụng khẩu trang và đưa ra một số khung làm việc phục vụ cho quá trình nhận dạng đối tượng.

Nâng cao chất lượng nhận dạng đối tượng liên quan tới khuôn mặt sử dụng khẩu trang bằng cách thu nhập số lượng dữ liệu hình ảnh đủ lớn, tiến hành thực nghiệm ngoài thực tế để ghi chép những những sai sót, lập bảng thống kê về các tham số để chương trình được tối ưu hơn.

Tuy nhiên, hệ thống trong đề tài mà chúng em đang nghiên cứu vẫn còn một số hạn chế như không thể nhận diện hoặc nhân diện sai trong điều kiện thiếu ánh sáng, hay là dùng một vật thể khác mà không phải là khẩu trang che khuôn mặt. Mặc dù nghiên cứu còn những hạn chế và thiếu sót nhất định, nhưng chúng em sẽ tiếp tục cố gắng hoàn thiện trong thời gian tới.

4.2. Hướng phát triển.

Hiện tại, do hạn chế về thời gian, nghiên cứu dừng lại thử nghiệm với tập dữ liệu có kích cỡ vừa. Sau này chúng em sẽ mở rộng thử nghiệm trên tập dữ liệu lớn hơn

Mở rộng và thử nghiệm trên các kiểu thực thể và mối quan hệ thực thể khác. Cải tiến áp dụng các dạng biểu đồ trực quan khác nhau trong việc phân tích các thực thể.

Sau khi trải qua mùa dich COVID thì hệ thống vẫn có thể đổi lại cài đặt giống như lúc trước dịch. Vì việc đeo khẩu trang lúc này không còn bắt buộc nữa

Ngoài ra, nhóm chúng em còn sẽ có định hướng đến việc kết hợp hệ thống nhận diện khẩu trang với một hệ thống nhận biết danh tính của một người đó trên không gian thực, nhằm mục đích là có thể nhận biết danh tính của một người, kể cả khi họ đang đeo khẩu trang, với hệ thống này có thể được áp dụng trong hệ thống điểm danh ở các công ty, ...

TÀI LIỆU THAM KHẢO

[1]. Solomon, C.J., Breckon, T.P. (2010). Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab. Wiley-Blackwell

[2]. Valueva, M.V.; Nagornov, N.N.; Lyakhov, P.A.; Valuev, G.V.; Chervyakov, N.I. (2020). "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation". Mathematics and Computers in Simulation. Elsevier BV. 177: 232–243. doi:10.1016/j.matcom.2020.04.031. ISSN 0378-4754. Convolutional neural networks are a promising tool for solving the problem of pattern recognition.

[3]. Mạng nơ ron tích chập cheatsheet : stanford.edu.

[4]. . Sanders, J.G., Jenkins, (2018). R Individual differences in hyper-realistic mask detection. Cogn. Research 3, 24

[5]. A. Zaliha Abd Aziz, H. Wei and J. Ferryman, "Face anti-spoofing countermeasure: Efficient 2D materials classification using polarization imaging," 2017 5th International Workshop on Biometrics and Forensics (IWBF), Coventry, UK, 2017, pp. 1-6.

[6]. Sikandar, T., Ghazali, K.H. & Rabbi, M.F. ATM crime detection using image processing integrated video surveillance: a systematic review. Multimedia Systems 25, 229–251 (2019).

[7]. Xiaobai Li, J. Komulainen, G. Zhao, Pong-Chi Yuen and M. Pietikäinen, "Generalized face anti-spoofing by detecting pulse from face videos," 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 2016, pp. 4244-4249

[8]. E. M. Nowara, A. Sabharwal and A. Veeraraghavan, "PPGSecure: Biometric Presentation Attack Detection Using Photopletysmograms," 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), Washington, DC, USA, 2017, pp. 56-62

[9]. R. Shao, X. Lan and P. C. Yuen, "Deep convolutional dynamic texture learning with adaptive channel-discriminability for 3D mask face anti-spoofing," 2017 IEEE International Joint Conference on Biometrics (IJCB), Denver, CO, USA, 2017, pp. 748-755

[10]. COVID-19: Lưu Ý Về Việc Đeo Khẩu Trang | CDC.

[11]. S. A. Sanjaya and S. Adi Rakhmawan, "Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic," 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), Sakheer, Bahrain, 2020, pp. 1-5,

[12]. Facial recognition identifies people wearing masks - BBC News
[13]. de Leeuw, Karl; Bergstra, Jan (2007). The History of Information Security: A Comprehensive Handbook. Amsterdam: Elsevier. pp. 264–265
[14]. Face Recognition Grand Challenge (FRGC) | NIST
[15]. History of Face Recognition & Facial recognition software
[16]. Haar Cascade là gì? Luận về một kỹ thuật chuyên dùng để nhận biết các khuôn mặt trong ảnh : viblo.asia.
[17]. COVID-19: Face Mask Detector with OpenCV, Keras/TensorFlow, and Deep Learning – PyImageSearch
[18]. Face Mask Detection Using Transfer Learning of InceptionV3 | SpringerLink
[19]. SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2 - ScienceDirect
[20]. Y. Chen et al., "Face Mask Assistant: Detection of Face Mask Service Stage Based on Mobile Phone," in IEEE Sensors Journal, vol. 21, no. 9, pp. 11084-11093, 1 May1, 2021, doi: 10.1109/JSEN.2021.3061178.
[21]. Hệ thống cảnh báo đeo khẩu trang phòng chống dịch COVID-19 đơn giản với OpenCV và Keras (pivietnam.com.vn)
[22]. Làm thử hệ thống phát hiện đeo khẩu trang bằng OpenCV - Mì AI (miai.vn)
[23]. Việt Nam nghiên cứu thành công công nghệ nhận diện khi đeo khẩu trang - Báo Nhân Dân
[24]. Convolutional neural network | Deep Learning cơ bản: nttuan8.com.
[25]. The Flatten Layer, Explained: youtube.com.
[26]. Keras Dense Layer Explain for Beginners | MLK – Machine Learning Knowlrdge.
[27]. Tìm hiểu về dropout trong deep learning, machine learning : phamduytung.com