|  |  |
| --- | --- |
| HỌC VIỆN NÔNG NGHIỆP VIỆT NAM  KHOA CÔNG NGHỆ THÔNG TIN | CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập - Tự do - Hạnh phúc |



KHÓA LUẬN TỐT NGHIỆP

Đề tài:

Ứng Dụng Mạng Convolutional neural network Nhận dạng khuôn mặt đeo Khẩu trang

Họ và tên sinh viên : Bùi Thọ Kiên

MSV : 621096

Lớp : K62CNPMP

Giảng viên hướng dẫn : Ts. Nguyễn Trọng Khương

Hà Nội - 2023

MỤC LỤC

[CHƯƠNG 1 GIỚI THỆU 4](#_Toc136932979)

[1.1 Tên đề tài 4](#_Toc136932980)

[1.2 Đặt vấn đề 4](#_Toc136932981)

[1.3 Mục đích và yêu cầu 5](#_Toc136932982)

[1.3.1 Mục đích 5](#_Toc136932983)

[1.3.2 Yêu cầu 5](#_Toc136932984)

[1.4 Tình hình nghiên cứu trong và ngoài nước 6](#_Toc136932985)

[1.4.1 Tình hình nghiên cứu trong nước 6](#_Toc136932986)

[1.4.2 Tình hình nghiên cứu ngoài nước 6](#_Toc136932987)

[1.5 Phương pháp nghiên cứu 7](#_Toc136932988)

[1.5.1 Đối tượng 7](#_Toc136932989)

[1.5.2 Phạm vi nghiên cứu 7](#_Toc136932990)

[CHƯƠNG 2 CƠ SỞ LÝ THUYẾT 8](#_Toc136932991)

[2.1 Giới thiệu ngôn ngữ lập trình Python 8](#_Toc136932992)

[2.2 Giới thiệu về thư viện OpenCV 8](#_Toc136932993)

[2.3 Mạng Convolutional Neural Network 10](#_Toc136932994)

[2.3.1 CNN là gì? 10](#_Toc136932995)

[2.3.2 Lớp tích chập(convolution) 11](#_Toc136932996)

[2.3.3 Convolutional layer 12](#_Toc136932997)

[2.3.4 Relu layer 13](#_Toc136932998)

[2.3.5 Lớp tổng hợp (Pooling Layer) 15](#_Toc136932999)

[2.3.6 Lớp kết nối đầy đủ (Fully-connected layer) 16](#_Toc136933000)

[2.4 Thư viện Keras/Tensorflow 16](#_Toc136933001)

[CHƯƠNG 3 PHÂN TÍCH HỆ THỐNG 18](#_Toc136933002)

[3.1 Mô hình tổng quan của hệ thống nghiên cứu 18](#_Toc136933003)

[3.1.1 Luồng hoạt động của hệ thống 18](#_Toc136933004)

[3.2 Phân tích chi tiết luồng hoạt động của hệ thống 19](#_Toc136933005)

[3.2.1 Giai đoạn huấn luyện mô hình 19](#_Toc136933006)

[3.2.2 Giai đoạn áp dụng vào nhận diện khuôn mặt đeo khẩu trang hay không 20](#_Toc136933007)

[3.3 Phân tích tập dữ liệu 21](#_Toc136933008)

[3.3.1 Thu thập dữ liệu 21](#_Toc136933009)

[3.3.2 Xử lý dữ liệu và tiền xử lý 22](#_Toc136933010)

[CHƯƠNG 4 XÂY DỰNG CHƯƠNG TRÌNH 24](#_Toc136933011)

[4.1 Triển khai kịch bản huấn luyện (training) nhận diện khẩu trang 24](#_Toc136933012)

[4.2 Tiến hành nhận diện đeo khẩu trang hay không 31](#_Toc136933013)

[4.3 Kết quả thử nghiệm chương trình 41](#_Toc136933014)

[4.4.1 Kết quả huấn luyện mô hình 41](#_Toc136933015)

[4.4.2 Kết quả ứng dụng mạng CNN nhận diện đeo khẩu trang hay không 46](#_Toc136933016)

[CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 50](#_Toc136933017)

[5.1 Kết luận 50](#_Toc136933018)

[5.2 Hướng phát triển 50](#_Toc136933019)

[CHƯƠNG 6 DANH MỤC THAM KHẢO 51](#_Toc136933020)

LỜI CẢM ƠN

Đầu tiên, em và các bạn xin gửi lời cảm ơn chân thành tới các thầy cô, cho phép em được bày tỏ lòng biết ơn đến tất cả các cá nhân và tổ chức đã tạo điều kiện hỗ trợ, giúp đỡ chúng em trong suốt quá trình học tập và nghiên cứu đề cương này. Trong suốt khoảng thời gian từ khi bắt đầu học tập tại trường đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý thầy cô và bạn bè.

Với lòng biết ơn, Em xin gửi đến quý Thầy Cô khoa Công nghệ thông tin – Học viện Nông Nghiệp Việt Nam đã truyền đạt vốn kiến thức quý báu cho em trong suốt thời gian học tập tại trường. Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc tới giảng viên – thầy giáo Nguyễn Trọng Kương – thầy đã trực tiếp hướng dẫn và chỉ bảo cho em, giúp đỡ em và các bạn rất nhiều trong quá trình tìm hiểu và hoàn thiện đề cương khóa luận tốt nghiệp này .

Bước đầu đi vào thực tế của em còn nhiều hạn chế và còn nhiều bỡ ngỡ nên không tránh khỏi những thiếu sót, cùng với đó là khoảng thời gian còn hạn chế, bài báo cáo đề cương khóa luận tốt nghiệp của em không thể tránh khỏi sai sót. Vì vậy em rất mong nhận được những ý kiến đóng góp quý báu của các thầy cô để em có thể nâng cao kiến thức của bản thân, và hơn nữa là để phục vụ phục vụ cho công việc sau này.

*Chúng em xin chân thành cảm ơn!*

# GIỚI THỆU

## Tên đề tài

*Đề tài:* Ứng dụng mạng Convolutional Neural Network vào nhận diện khuôn mặt đeo khẩu trang.

## Đặt vấn đề

Một trong những thách thức quan trọng trong việc kiểm soát và phòng ngừa dịch bệnh, đặc biệt là trong bối cảnh đại dịch COVID-19. Đeo khẩu trang là một biện pháp cơ bản để ngăn chặn sự lây lan của virus trong không khí và bảo vệ sức khỏe cộng đồng.

Tuy nhiên, việc kiểm tra và đảm bảo rằng mọi người đang thực sự đeo khẩu trang là một nhiệm vụ khó khăn. Nhất là trong các khu vực đông người, như sân bay, nhà ga, bệnh viện, trường học hay khu công cộng, việc giám sát và thực thi việc đeo khẩu trang trở nên phức tạp và tốn kém.

Vì vậy trong đề tài, em sẽ xây dựng hệ thống nhận diện khuôn mặt có đeo khẩu trang hay không đeo khẩu trang, là tiền đề để phát triển “hệ thống cảnh báo mang khẩu trang đối với mọi người” đặt biệt là ở nơi công cộng tập trung đông người như trung tâm thương mại, trường học, bệnh viện, …

Trong đề này nhóm em xây dựng “Ứng dụng Mạng Convolutional Neural Network nhận diện khuôn mặt đeo khẩu trang” với thư viện OpenCV, Keras/Tensorflow, và Mạng Convolutional Neural Network.

## Mục đích và yêu cầu

* 1. Mục đích

Xây dựng một phần mềm nhận diện khuôn mặt có đang đeo khẩu trang hay là không dùng mạng Convolutional Neural Network.

Tìm hiểu về thư viện Keras/Tensorflow/OpenCV khi sử dụng trong môi trường lập trình bằng ngôn ngữ Python.

Tìm hiểu về Convolutional Neural Network: cách hoạt động và ứng dụng của Convolutional Neural Network trong đề tài này.

* 1. Yêu cầu

- Nắm vững kiến thức cơ bản về Convolutional Neural Network, về việc sử dụng Convolutional Neural Network trong bài toán phân loại hình ảnh.

- Có khả năng sử dụng các thư viện và công cụ phân tích dữ liệu, như Python

- Xây dựng mô hình Convolutional Neural Network cho bài toán nhận diện khẩu trang.

- Có khả năng trình bày kết quả nghiên cứu và đánh giá một các rõ ràng và logic.

## Tình hình nghiên cứu trong và ngoài nước

Tình hình nghiên cứu về nhận diện đeo khẩu trang trong và ngoài nước đã thu hút sự quan tâm đáng kể, đặc biệt trong bối cảnh đại dịch COVID-19

### Tình hình nghiên cứu trong nước

* Trường Đại học Hàng hải Việt Nam cũng đã tiến hành nghiên cứu về nhận diện đeo khẩu trang bằng cách sử dụng các kỹ thuật học sâu và mạng Convolutional Neural Network (10).
* Viện Nghiên cứu Trí tuệ Nhân tạo VinAI Research (thuộc 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 (9).

### Tình hình nghiên cứu ngoài nước

* "Face Mask Detection and Recognition Using Convolutional Neural Network" - Nghiên cứu này được công bố trong IEEE Transactions on Image Processing và tác giả là Abdul Ahad Sami, Zeeshan Iqbal, Muhammad Shahid, Syed Tariq Hussain, và Imran Usman. Bài báo này trình bày một mô hình CNN để nhận diện và phân loại việc đeo khẩu trang trên các hình ảnh kỹ thuật số (8).
* "Real-Time Face Mask Detection System Using Convolutional Neural Networks" - Nghiên cứu này được thực hiện bởi Xinyi Guo, Ziyue Feng, Xianglin Zeng, Jiayu Wang, và Songlin Xu. Họ đã phát triển một hệ thống nhận diện khẩu trang thời gian thực sử dụng mạng CNN. Hệ thống này sử dụng camera để thu thập hình ảnh và sau đó sử dụng mạng CNN để nhận diện việc đeo khẩu trang (9).

## Phương pháp nghiên cứu

### Đối tượng

* Mạng Convolutional Neural Network

### Phạm vi nghiên cứu

* Tập trung tìm hiểu Mạng Convolutional Neural Network
* Huấn luyện để nhận diện từ bộ dữ liệu đầu vào có sẵn.
* Đưa ra số ảnh được dự đoán đúng và tỷ lệ phần tram dự đoán đúng

# CƠ SỞ LÝ THUYẾT

Để giải quyết bài toán nhận diện khuôn mặt và nhận diện đeo khẩu trang, sử dụng thư việc xư lý ảnh OpenCV, thư viện Keras/Tensorflow và sử dụng Convolutional Neural Network trong quá trình thực hiện đề tài này.

Trong đồ án này ngôn ngữ lập trình được sử dụng là Python, được viết trên môi trường của Python là IDLE(Python 3.11)

### Giới thiệu ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu. (2)

Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý. (2)

Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, Python dần mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python (2).

## Giới thiệu về thư viện OpenCV

- OpenCV ( Open Source Computer Vision Library ) là một [thư viện](https://en.wikipedia.org/wiki/Library_(computing)) các chức năng lập trình chủ yếu dành cho [thị giác máy tính](https://en.wikipedia.org/wiki/Computer_vision" \o "Tầm nhìn máy tính)[thời gian thực](https://en.wikipedia.org/wiki/Real-time_computing) . Ban đầu được phát triển bởi [Intel](https://en.wikipedia.org/wiki/Intel" \o "thông minh) , sau đó được hỗ trợ bởi [Willow Garage](https://en.wikipedia.org/wiki/Willow_Garage" \o "nhà xe liễu) , sau đó là Itseez (sau này được Intel mua lại ). Thư viện [đa nền tảng](https://en.wikipedia.org/wiki/Cross-platform) và được cấp phép dưới dạng [phần mềm mã nguồn mở miễn phí](https://en.wikipedia.org/wiki/Free_and_open-source_software" \o "Phần mềm mã nguồn mở và miễn phí) theo [Giấy phép Apache](https://en.wikipedia.org/wiki/Apache_License" \o "Giấy phép Apache) 2. Bắt đầu từ năm 2011, OpenCV có tính năng tăng tốc GPU cho các hoạt động thời gian thực. (3)

Chính thức ra mắt vào năm 1999, dự án OpenCV ban đầu là một sáng kiến [​​của Nghiên cứu Intel](https://en.wikipedia.org/wiki/Intel_Research_Lablets) nhằm nâng cao các ứng dụng sử dụng nhiều [CPU](https://en.wikipedia.org/wiki/Central_processing_unit) , một phần của loạt dự án bao gồm [dò tia](https://en.wikipedia.org/wiki/Ray_tracing_(graphics)" \o "Dò tia (đồ họa))[thời gian thực](https://en.wikipedia.org/wiki/Real-time_computing) và tường [hiển thị 3D .](https://en.wikipedia.org/wiki/3D_Display" \o "Hiển thị 3D) Những người đóng góp chính cho dự án bao gồm một số chuyên gia tối ưu hóa ở Intel Nga, cũng như Nhóm Thư viện Hiệu suất của Intel. Trong những ngày đầu của OpenCV, các mục tiêu của dự án đã được mô tả như sau:

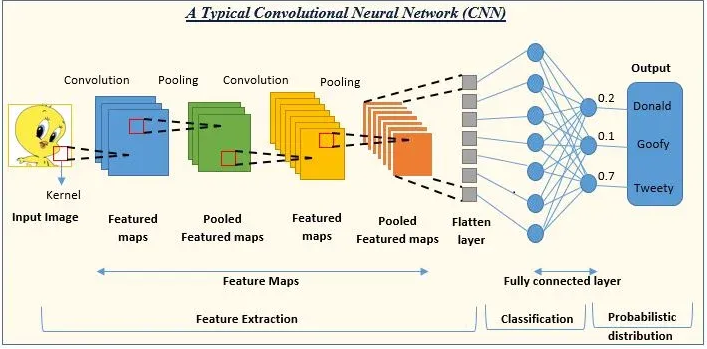
* Nâng cao nghiên cứu tầm nhìn bằng cách cung cấp không chỉ mã mở mà còn [được tối ưu hóa](https://en.wikipedia.org/wiki/Code_optimization" \o "tối ưu hóa mã) cho cơ sở hạ tầng tầm nhìn cơ bản. Không còn [phát minh lại bánh xe](https://en.wikipedia.org/wiki/Reinventing_the_wheel" \o "Phát minh lại bánh xe) .
* Phổ biến kiến ​​thức tầm nhìn bằng cách cung cấp một cơ sở hạ tầng chung mà các nhà phát triển có thể xây dựng trên đó, để mã có thể dễ đọc và dễ chuyển nhượng hơn.
* Nâng cao các ứng dụng thương mại dựa trên tầm nhìn bằng cách cung cấp miễn phí mã [di động](https://en.wikipedia.org/wiki/Portability_(computer_science)) , được tối ưu hóa hiệu suất – với giấy phép không yêu cầu bản thân mã phải mở hoặc miễn phí. (3)

## Mạng Convolutional Neural Network

### CNN là gì?

- CNN được viết tắt của Convolutional Neural Network hay còn được gọi là CNNS mang nơ-ron tích chập, là một trong những [mô hình Deep Learning](https://vietnix.vn/deep-learning-la-gi/) cực kỳ tiên tiến, bởi chúng cho phép bạn xây dựng những hệ thống có độ chính xác cao và thông minh. Nhờ khả năng đó, Convolutional Neural Network có rất nhiều ứng dụng, đặc biệt là những bài toán cần nhận dạng vật thể (object) trong ảnh (6).

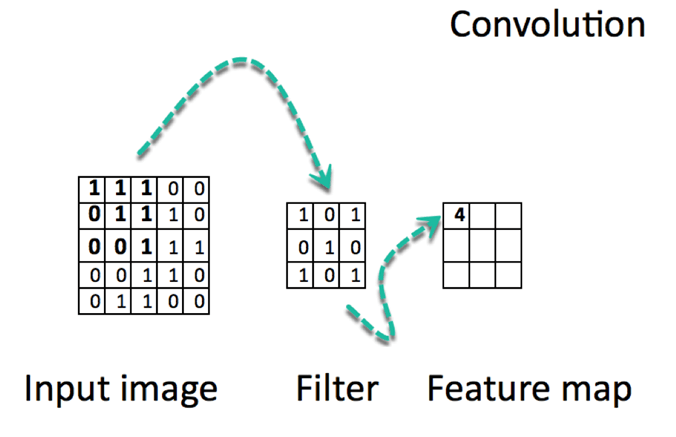
- Convolutional Neural Network bao gồm: lớp tích chập(convolution), lớp tổng hợp(Pooling),lớp kết nối đầy đủ(Fully-Connected).



Cấu trúc của mạng CNN (1)

### Lớp tích chập(convolution)

Convolutional là một loại cửa sổ dạng trượt nằm trên một ma trận. Các convolutional layer sẽ chứa các parameter có khả năng tự học, qua đó sẽ điều chỉnh và tìm ra cách lấy những thông tin chính xác nhất trong khi không cần chọn feature (6).



1. Convolutional layer

Đây chính là lớp đóng vai trò mấu chốt của CNN, khi layer này đảm nhiệm việc thực hiện mọi tính toán. Stride, padding, filter map, feature map là những yếu tố quan trọng nhất của convolutional layer (6).

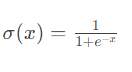
* Cơ chế của CNN là tạo ra các filter áp dụng vào từng vùng hình ảnh. Các filter map này được gọi là ma trận 3 chiều, bên trong chứa các parameter dưới dạng những con số.
* Stride là sự dịch chuyển filter map theo pixel dựa trên giá trị từ trái sang phải.
* Padding: Là các giá trị 0 được thêm cùng lớp input.
* Feature map: Sau mỗi lần quét, một quá trình tính toán sẽ được thực hiện. Feature map sẽ thể hiện kết quả sau mỗi lần filter map quét qua input.

1. Relu layer

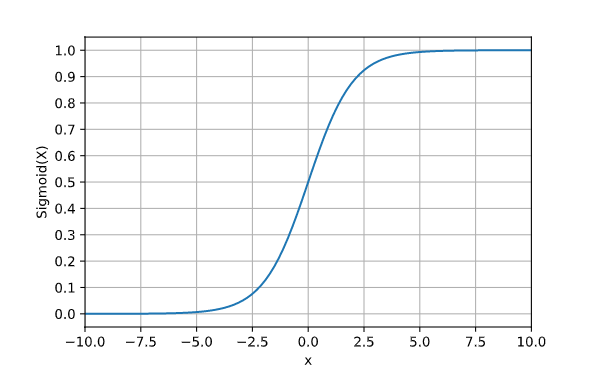
Còn có tên gọi khác là activation function, đây là một hàm được kích hoạt trong [neural network](https://vietnix.vn/neural-network/). Nó có tác dụng mô phỏng các neuron có tỷ lệ truyền xung qua axon. Trong activation function chúng còn có hàm nghĩa là: Relu, Tanh, Sigmoid, Maxout, Leaky,… Relu layer được ứng dụng phổ biến trong việc huấn luyện nơ-ron do sở hữu nhiều ưu điểm tiên tiến (6).

* Hàm Sigmoid

Công thức:



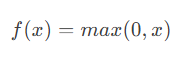
Phân tích:



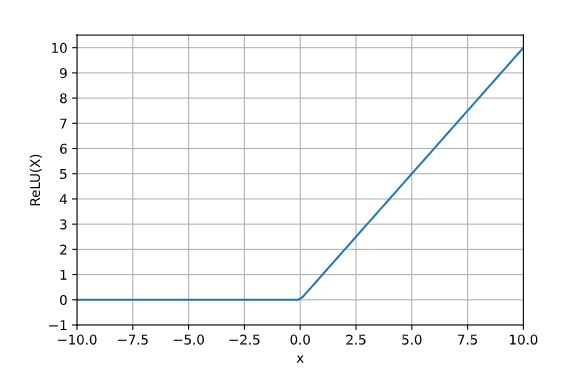
Hàm Sigmoid nhận đầu vào là một số thực và chuyển thành một giá trị trong khoảng (0;1) (xem đồ thị phía trên). Đầu vào là số thực âm rất nhỏ sẽ cho đầu ra tiệm cận với 0, ngược lại, nếu đầu vào là một số thực dương lớn sẽ cho đầu ra là một số tiệm cận với 1. Trong quá khứ hàm Sigmoid hay được dùng vì có đạo hàm rất đẹp (7).

* Hàm ReLU

Công thức:



Phân tích:

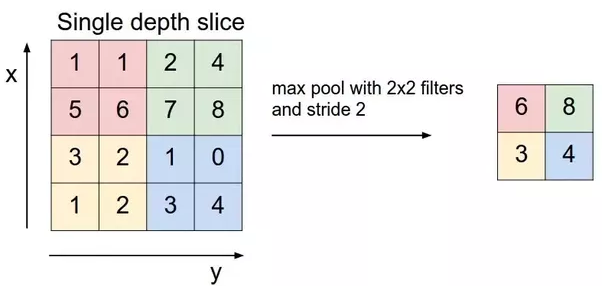


Hàm ReLU đang được sử dụng khá nhiều trong những năm gần đây khi huấn luyện các mạng neuron. ReLU đơn giản lọc các giá trị < 0. Nhìn vào công thức chúng ta dễ dàng hiểu được cách hoạt động của nó (7).

1. Lớp tổng hợp (Pooling Layer)

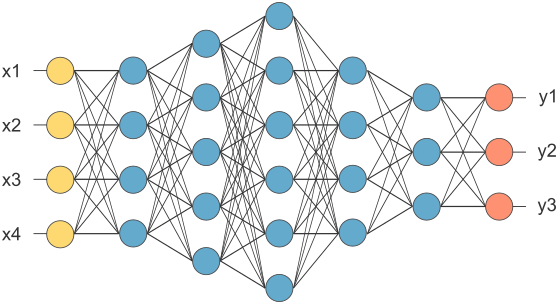
Lớp pooling sẽ giảm bớt số lượng tham số khi hình ảnh quá lớn. Không gian pooling còn được gọi là lấy mẫu con hoặc lấy mẫu xuống làm giảm kích thước của mỗi map nhưng vẫn giữ lại thông tin quan trọng. Các pooling có thể có nhiều loại khác nhau: Max Pooling, Average Pooling, Sum Pooling

Max pooling lấy phần tử lớn nhất từ ma trận đối tượng, hoặc lấy tổng trung bình. Tổng tất cả các phần tử trong map gọi là sum pooling (1)



1. Lớp kết nối đầy đủ (Fully-connected layer)

Khi 2 lớp convolutional layer và pooling layer nhận được ảnh truyền, lớp này sẽ có nhiệm vụ xuất kết quả. Khi ta nhận được kết quả là model đọc được thông tin ảnh, ta cần phải tạo sự liên kết để cho ra nhiều output hơn (6).



## Thư viện Keras/Tensorflow

Thư viện Keras:

Keras là một open source cho Neural Network được viết bởi ngôn ngữ Python. Nó là một library được phát triển vào năm 2005 bởi Francois Chollet, là một kỹ sư nghiên cứu Deep Learning. Keras có thể sử dụng chung với các thư viện nổi tiếng như Tensorflow, CNTK, Theano. Một số ưu điểm của Keras như:

* Dễ sử dụng, dùng đơn giản hơn Tensor, xây dựng model nhanh.
* Run được trên cả CPU và GPU.
* Hỗ trợ xây dựng CNN, RNN hoặc cả hai. Với những người mới tiếp cận đến Deep như mình thì mình chọn sử dụng Keras để build model vì nó đơn giản,dễ nắm bắt hơn các thư viện khác. Dưới đây mình xin giới thiệu một chút về API này (7).

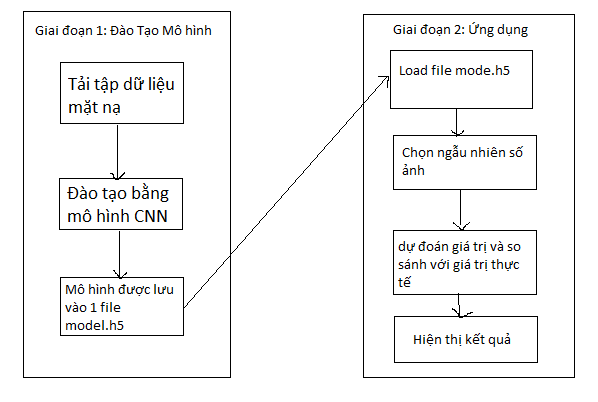
Thư viện Tensorflow:

Với sự bùng nổ của lĩnh vực [Trí Tuệ Nhân Tạo – A.I.](https://topdev.vn/it-jobs/ai-kt1555) trong thập kỷ vừa qua, machine learning và deep learning rõ ràng cũng phát triển theo cùng. Và ở thời điểm hiện tại, [TensorFlow](https://www.tensorflow.org/) chính là thư viện mã nguồn mở cho machine learning nổi tiếng nhất thế giới, được phát triển bởi các nhà nghiên cứu từ Google. Việc hỗ trợ mạnh mẽ các phép toán học để tính toán trong machine learning và deep learning đã giúp việc tiếp cận các bài toán trở nên đơn giản, nhanh chóng và tiện lợi hơn nhiều (8).

# PHÂN TÍCH HỆ THỐNG

## Mô hình tổng quan của hệ thống nghiên cứu

1. Luồng hoạt động của hệ thống



Để có thể tùy chỉnh được việc huấn luyện trong nhận diện khuôn mặt đeo khẩu trang ta chia ra gồm 2 giai đoạn riêng biệt, mỗi giai đoạn có các bước nhiệm vụ riêng

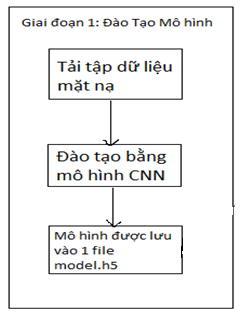
Ta có 2 giai đoạn chính như trên sơ đồ:

Giai đoạn 1: Đào tạo mô hình

Giai đoạn 2: Ứng dụng thực tế

## Phân tích chi tiết luồng hoạt động của hệ thống

### Giai đoạn huấn luyện mô hình



Chúng ta sẽ tập trung vào việc tải tập dữ liệu phát hiện khẩu trang của mình từ bộ dữ liệu, đào tạo một mô hình trên tập dữ liệu này

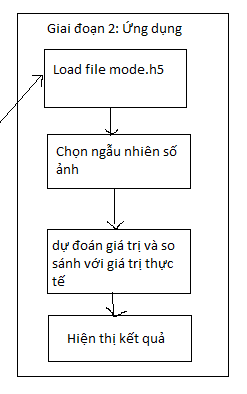
Phân tích luồng hoạt động của giai đoạn Train dữ liệu qua thứ tự các bước:

- Tải tập dữ liệu : bước đầu tiên, hệ thống sẽ bắt đầu được load dữ liệu để dữ liệu được đưa vào, dữ liệu được lấy từ tập dữ liệu (traindata).

- Đào tạo bằng mô hình CNN: Việc huấn luyện lúc này có nhiệm vụ phân loại ra là khuôn mặt có đeo khẩu trang và không đao khẩu trang để khi việc phân tích sẽ đưa hai kết quả cho việc nhận diện.

- Mô hình được lưu vào một file mode.h5: lúc này mô hình được lưu vào 1

### Giai đoạn áp dụng vào nhận diện khuôn mặt đeo khẩu trang hay không



Sau khi trình phát hiện khẩu trang được huấn luyện (train), chúng ta có thể chuyển sang ứng dụng xem có .đeo khẩu trang hay không

Phân tích luồng hoạt động của giai đoạn áp dụng vào nhận diện khuôn mặt đeo khẩu trang hay không:

- Load file mode.h5: Trước khi phân tích, chúng ta cho hệ thống load qua mô hình CNN trong file mode.h5.

- Chọn ảnh ngẫu nhiên : sau khi được load load qua mô hình CNN trong file mode.h5 ta chọn ngẫu nhiên một số ảnh để dự đoán.

- Dự đoán và so sánh giá trị thực tế của ảnh: sau khi nhận dạng được khuôn mặt từ ảnh tự động dự đoán giá trị và so sánh với giá trị thực tế của ảnh để xem có nhận diện khuôn mặt đeo khẩu trang hay không.

- Hiện thị kết quả: sau khi so sánh hệ thống sẽ hiển thị kết quả xem có nhận diện khuôn mặt đeo khẩu trang hay không, tổng số ảnh có đeo và tổng số ảnh không đeo, tỷ lệ phần trăm ảnh được dự đoán đúng, số ảnh được đoán đúng

## Phân tích tập dữ liệu

1. Thu thập dữ liệu

Nhóm thu một loạt hình ảnh đeo và không đeo của nhiều người khác nhau

Để xây dựng tập dữ liệu cho việc nhận diện khẩu trang, nhóm đã tập hợp và được sự cho phép của những cá nhân, người quen,... để thu thập các dữ liệu chuẩn bị cho việc thực hiện đề tài. Vì các dữ liệu ban đầu là một tập các đoạn video ngắn có chứa từng góc quay, từ nhìn chính diện, nhìn sang trái và sang phải, nhìn hướng lên trên và cúi xuống dưới, đến việc quay khi đeo khẩu trang và riêng khuôn mặt dẫn đến vài trường hợp video quay vội góc quay sang ngang không thể xoay đến mức yêu cầu của đề tài làm đoạn hình ảnh đó khuôn mặt chỉ tối đa quay nghiêng đối với góc nhìn chính diện. Hoặc có vài trường hợp khi quay do người thực hiện quay gần để lấy nét cho hình ảnh dữ liệu dẫn đến việc khi quay sang ngang, người thực hiện bị mất hình ảnh ở viền của camera khiến đoạn video không thể sử dụng được và phải quay lại từ đầu, làm cho việc thu thập dữ liệu tốn thêm thời gian và cho hai bên chán nản không thể hợp tác một cách suôn sẻ.

Khi đã có một tập các dữ liệu video đầy đủ và đồng nhất, nhóm cũng cần phải cắt từng đoạn của video thành tập hình ảnh của mỗi người với các góc quay và khi đeo khẩu trang với ảnh của khuôn mặt bởi máy tính không thể nào học được với các dữ liệu lớn bằng video vì phải có một “siêu máy tính” rất mạnh mới có thể train và test từng đoạn video một với lượng lớn độ pixel (điểm ảnh), dải màu của mỗi một giây có thể cho ra bao nhiêu là hình ảnh và mỗi hình ảnh cho ra rất nhiều các phần tử pixel của chiều dài và chiều rộng ứng với mỗi hàng và cột của ma trận và phần tử của nó là 1 pixel, chưa kể đến việc thời gian để máy tính học và nhận diện từng một video có thể mất đến hàng tiếng hoặc nhiều ngày mới có thể đào tạo xong cho máy tính.

Việc đào tạo còn gặp phải nhiều rủi ro nếu như máy tính không thể tải được dữ liệu gặp lỗi hoặc đến thời gian nhất định máy tính có thể quá tải làm cho việc train và test của mô hình gián đoạn phải làm lại từ đầu khiến công việc tốn rất nhiều thời gian và công sức mới hoàn thành xong được công việc vốn chỉ mới bắt đầu.

Việc thu thập dữ liệu hình ảnh về khuôn mặt sẽ tạo ra nhiều lợi ích lớn đối với mô hình của đề tài. Các hình ảnh sẽ có một số lượng pixel nhất định đảm bảo dữ liệu đó không quá lớn mà máy tính cũng có thể nhận biết được, cũng như không để máy tính học hay nhận diện một bức ảnh nhiều lần vì một video mỗi giây máy tính sẽ quét rất nhiều lần gây lãng phí bộ nhớ khi xử lý các vấn đề khác. Trong khi đó với các máy tính có GPU (đơn vị xử lý đồ họa) thấp việc để máy tính train và test bằng hình ảnh của mô hình sẽ dễ dàng hơn mà việc nhận diện còn tốt hơn rất nhiều.

Kết quả: nhóm đã thu thập dữ liệu video của 30 đối tượng gồm

1. Xử lý dữ liệu và tiền xử lý

Việc tiền xử lý dữ liệu nhằm đưa tất cả các ảnh về cùng kích thước, sau đó các ảnh này sẽ được chuyển đổi để phục vụ cho quá trình xử lý ảnh ở các bước sau.

Để có thể nhận diện tốt khuôn mặt đeo khẩu trang và không đeo khẩu trang, ta cần mô hình trích xuất hiệu quả được các điểm đặc trưng trên khuôn mặt cho nhiều người khác nhau. Đồng thời cũng nên kèm theo các quá trình tiền xử lý để cải thiện khả năng nhận diện gương mặt chính xác của mô hình.

Việc căn chỉnh hình ảnh khuôn mặt thẳng hàng có nhiều lợi ích đối với việc train và test cho mô hình, cũng như kiểm tra từng hình ảnh có thể đồng nhất với dữ liệu khi kiểm thử ứng dụng.

Khi căn chỉnh dữ liệu hình ảnh, các khuôn mặt sẽ được đưa về cùng một kích thước và hướng nhìn, giúp cho các thuật toán nhận diện khuôn mặt của người đeo và không đeo khẩu trang có thể hoạt động tốt hơn và đưa ra độ chính xác cao hơn. Các khuôn mặt khi đó sẽ được đưa về đúng vị trí, giúp giảm thiểu sai số trong quá trình máy tính nhận diện và kiểm tra. Tốc độ xử lý trong các hình ảnh được cải thiện rõ rệt, các thuật toán để nhận diện khuôn mặt có thể hoạt động nhanh hơn bởi vì nó không cần xử lý các khuôn mặt ở vị trí khác nhau hoặc ở góc quay khác nhau.

Từ các dữ liệu thô được thu thập được, việc xử lý từng nguồn dữ liệu được hỗ trợ bằng chương trình cắt ảnh tự động sử dụng thư viện face recognition để nhận diện và đóng khung của khuôn mặt rồi cắt phần viền khung mà có chứa khuôn mặt rồi giảm kích thước của hình ảnh gương mặt đó sát với kích thước để khi xuất ra dữ liệu không quá to hoặc quá nhỏ so với màn hình máy tính có thể thấy được và đánh giá bằng con người.

Hình ảnh được xử lý được chia làm 2 phần. Những ảnh cắt được từ dữ liệu thô sẽ được đặt trong 2 thư mục riêng biệt là khẩu trang và không khẩu trang với từng hình ảnh mỗi người từ những góc độ khác nhau từ chính diện, nhìn sang trái, phải và nhìn hướng lên trên rồi hướng xuống dưới. Mỗi người sẽ cho ra ảnh với từng góc độ như trên khi đeo khẩu trang và không đeo khẩu trang.

Kết quả: Dữ liệu video của 30 đối tượng gồm 494 đã được cắt ảnh về cùng một kích thước 250x250 phân loại trong trong 1 thư mục lớn du\_lieu chứa 2 thư mục con “co deo”,”khong deo”

# XÂY DỰNG CHƯƠNG TRÌNH

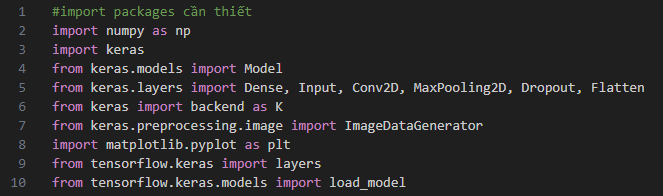
## Triển khai kịch bản huấn luyện (training) nhận diện khẩu trang

Chúng ta tiến hành xem xét tập dữ liệu, tìm hiểu cách để có thể sử dụng Keras và TensorFlow để đào tạo một bộ phân loại nhằm tự động phát hiện một người có đeo khẩu trang hay không.

Để hoàn thành nhiệm vụ này, chúng ta sẽ tinh chỉnh kiến ​​trúc mạng CNN.

Tiến hành triển khai hệ thống:

Chèn các thư viện có liên quan Mạng CNN vào file khautrang.py



Tensorflow/Keras cho phép:

* Data augmentation
* Loading the CNN classifier (chúng ta sẽ tinh chỉnh mô hình này với các trọng số Image được đào tạo trước)
* Pre-processing
* Loading image data

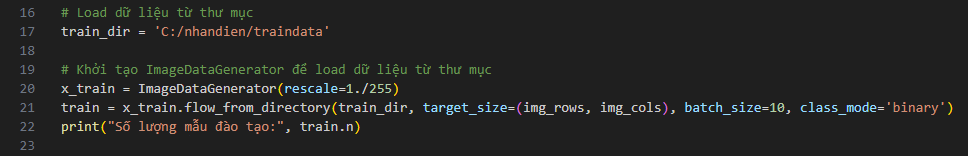
Việc triển khai các đường dẫn của ta sẽ giúp chúng ta tìm và liệt kê các hình ảnh trong tập dữ liệu của mình. Và chúng ta sử dụng matplotlib để vẽ các đường cong đào tạo của chúng ta.

Xác định các tham số ở một nơi:



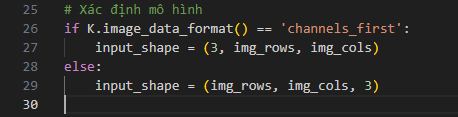
Chỉ định các hằng số tham số bao gồm kích thước ảnh

Tại thời điểm này, chúng ta đã sẵn sàng tải và xử lý trước dữ liệu đào tạo của mình:

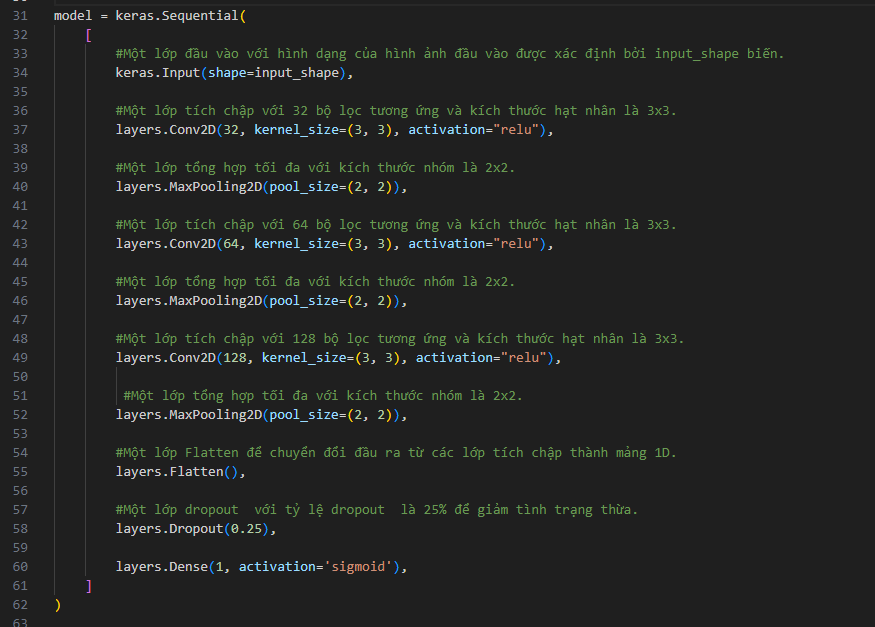


Trong khối lệnh, ta có :

* Load dữ liệu từ thư mục.
* Chuẩn hóa giá trị pixel của ảnh trong khoảng từ 0 đếnm 1.
* ImageDataGenerator.flow\_from\_directory() phương pháp Keras để tạo trình tạo dữ liệu để đào tạo mô hình.
* In ra màn hình Số lượng mẫu đào tạo.

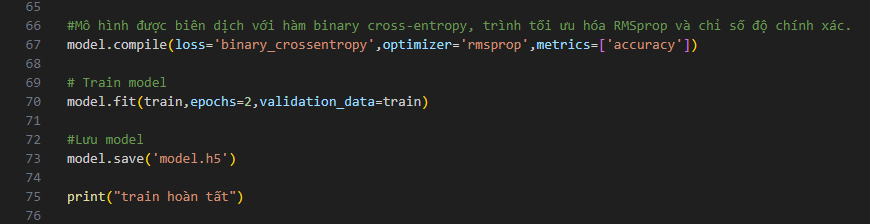


Từ dòng 26 đến 29 là xác định kích thức ảnh đầu và ảnh cuối với 3 màu RGB



Ta sử dụng mô hình này có kiến trúc "Convolutional Neural Network" (CNN) sử dụng Sequential trong thư viện Keras. Mô hình này bao gồm các lớp sau:

* Lớp đầu vào (keras.Input): Đây là lớp đầu tiên của mô hình và có hình dạng của hình ảnh đầu vào xác định bởi biến input\_shape.
* Lớp tích chập (layers\_Conv2D): Đây là một lớp tích chập 2D với 32 bộ lọc, kích thước hạt nhân 3x3 và hàm kích hoạt ReLU. Lớp này giúp trích xuất các đặc trưng từ hình ảnh đầu vào.
* Lớp tổng hợp tối đa (layers\_MaxPooling2D): Đây là một lớp tổng hợp tối đa với kích thước nhóm là 2x2. Nó giúp giảm kích thước không gian của đầu ra và tạo ra sự không gian trừu tượng.
* Các lớp tích chập và tổng hợp tối đa lặp lại (Conv2D và MaxPooling2D): Mô hình tiếp tục có thêm hai cặp lớp tích chập và tổng hợp tối đa với số bộ lọc và kích thước hạt nhân tương ứng là 64 và 128. Điều này giúp tăng cường khả năng trích xuất đặc trưng của mô hình.
* Lớp Flatten (layers\_Flatten): Lớp này chuyển đổi đầu ra từ các lớp tích chập trước thành một mảng 1D, sẵn sàng cho việc đưa vào lớp kết nối đầy đủ.
* Lớp dropout (layers\_Dropout): Lớp dropout với tỷ lệ dropout 0.25 được sử dụng để giảm tình trạng quá khớp trong mô hình.
* Lớp kết nối đầy đủ (layers\_Dense): Cuối cùng, có một lớp kết nối đầy đủ với 1 đơn vị đầu ra và hàm kích hoạt sigmoid. Điều này phù hợp cho bài toán nhị phân, trong đó chúng ta muốn dự đoán một lớp duy nhất (ví dụ: 0 hoặc 1).



Tiếp theo đoạn trên biên dịch, huấn luyện và lưu một mô hình sử dụng thư viện Keras.



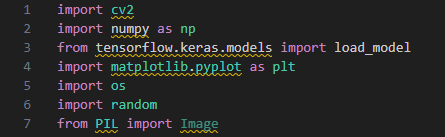
Bước cuối cùng của chúng ta là vẽ các đường cong chính xác và mất mát của chúng ta:

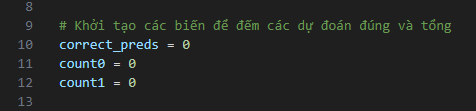
Khi đồ thị của chúng ta đã sẵn sàng, dòng 88 lưu hình vào đĩa bằng cách sử dụng đường dẫn tệp.

## Tiến hành nhận diện đeo khẩu trang hay không

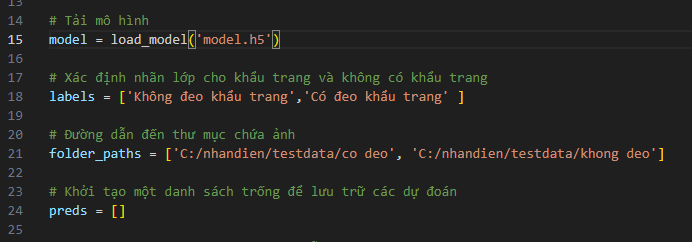
Ta tiến hành xây dựng nhận diện hay không

Sử dụng file ungdung.py để xây dựng cấu trúc tập lệnh nhận diện, ta tiến hành import các thư viện cần thiết như bên dưới:

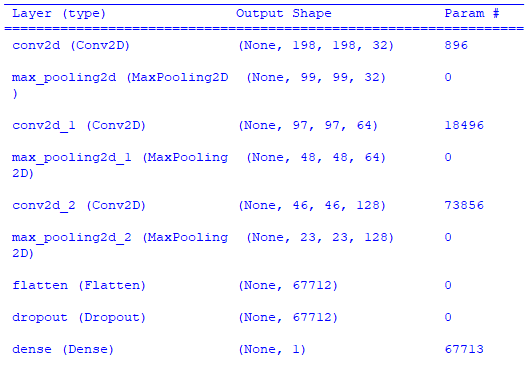




Khởi tạo các biến để đếm các dự đoán đúng và tổng



Hàm tải mô hình: em đã sử dụng hàm load\_model từ tensorflow.keras.models để tải mô hình từ tệp model.h5.



Cấu trúc Mô hình CNN

Bảng mô hình Convolutional Neural Network:

|  |  |  |
| --- | --- | --- |
| **Layer** | **Output Shape** | **Param** |
| **Conv2d** | **(None, 198, 198, 32)** | **896** |
| **Maxpooling2d** | **(None, 99, 99, 32)** | **0** |
| **Conv2d\_1** | **(None, 97, 97, 64)** | **18496** |
| **Maxpooling2d\_1** | **(None, 48, 48, 64)** | **0** |
| **Conv2d\_2** | **(None, 46, 46, 128)** | **73856** |
| **Maxpooling2d\_2** | **(None, 23, 23, 128)** | **0** |
| **Flatten** | **(None, 67712)** | **0** |
| **Dropout** | **(None, 67712)** | **0** |
| **Dense** | **(None, 1)** | **67713** |

Trong cấu trúc này em sửa dụng 2 lần đi vào 3 lần lớp Conv2d và Max\_pooling2d với Kích thước kernel là 2x2 nên kích thước giảm đi 2

Đầu tiên dữ liệu ảnh được đi vào lớp Conv2d với kích thước 198x198, số kênh là 32 và số lượng tham số là 896

Tiếp theo dữ liệu ảnh được đi vào lớp Max\_pooling2d với kích thước 99x99, số kênh là 32 và số lượng tham số là 0

Tiếp tục dữ liệu ảnh được đi vào lớp Conv2d lần 2 với kích thước 97x97, số kênh là 64 và số lượng tham số là 18496

Tiếp theo dữ liệu ảnh được đi vào lớp Max\_poling2d lần 2 với kích thước 48x48, số kênh là 64 và số lượng tham số là 0

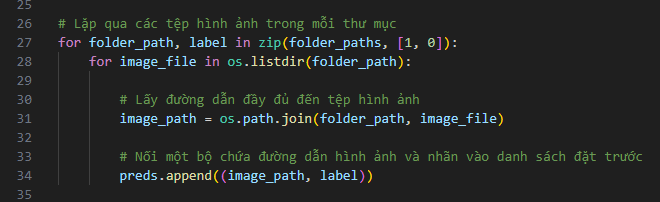
Tiếp tục dữ liệu ảnh được đi vào lớp Conv2d lần 3 với kích thước 46x46, số kênh là 128 và số lượng tham số là 73856

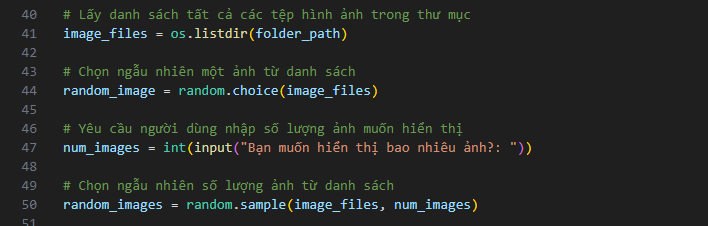
Tiếp theo dữ liệu ảnh được đi vào lớp Max\_poling2d lần 3 với kích thước 23x23, số kênh là 128 và số lượng tham số là 0

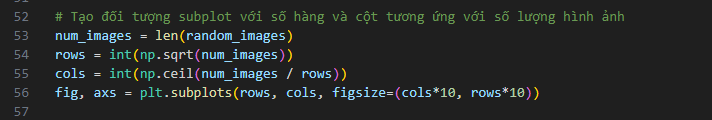
Tiếp theo dữ liệu ảnh được đi vào lớp flatten với kích thước 67712 và số lượng tham số là 0

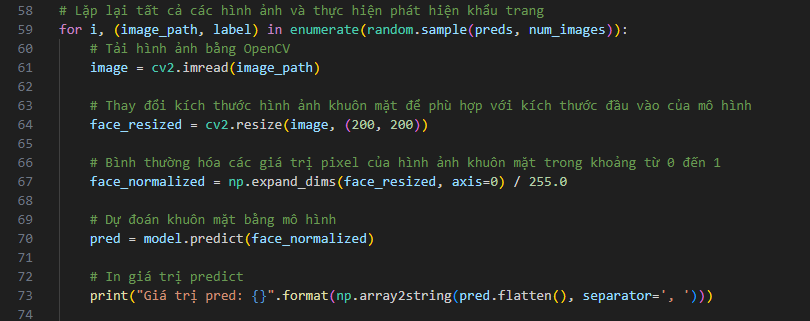
Tiếp theo dữ liệu ảnh được đi vào lớp dropout với kích thước 67712 và số lượng tham số là 0

Cuối cùng dữ liệu ảnh được đi vào lớp dense với kích thước 1 và số lượng tham số là 0

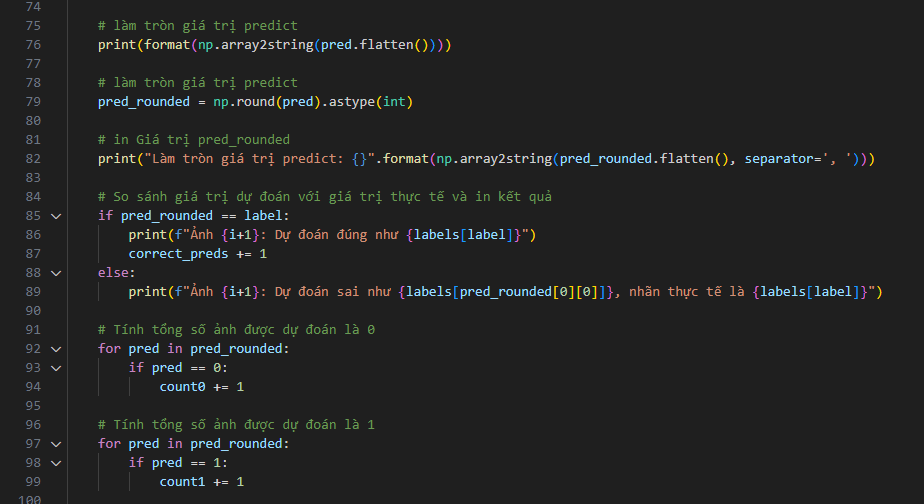




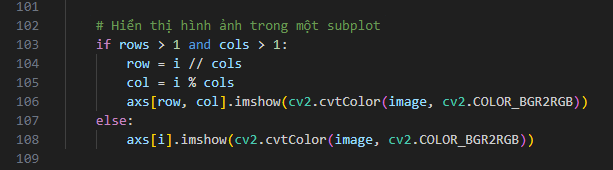


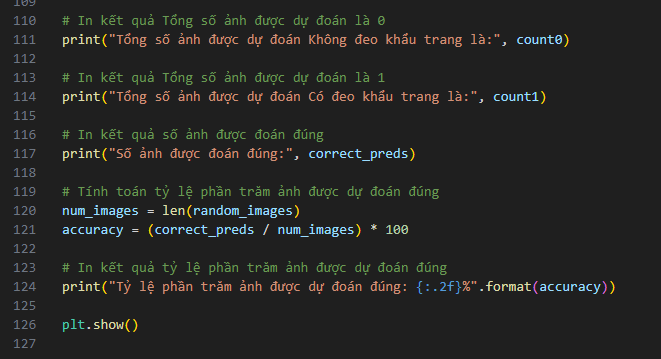


Các tham số hình ảnh: Em đã sử dụng OpenCV để đọc và xử lý các hình ảnh, thay đổi kích thước của chúng thành 200x200 pixel và chuẩn hóa các giá trị pixel trong khoảng từ 0 đến 1.



So sánh dự đoán và nhãn: em đã so sánh giá trị dự đoán với nhãn thực tế và in kết quả tương ứng. Bạn cũng đã tính toán số lượng dự đoán đúng và cập nhật các biến đếm tương ứng.





Hiển thị kết quả: em đã in kết quả về số lượng ảnh được dự đoán là lớp 0 và lớp 1, số lượng dự đoán đúng và tỷ lệ phần trăm dự đoán đúng. em cũng đã vẽ biểu đồ để trực quan hóa giá trị dự đoán của mô hình.

1. Kết quả thử nghiệm chương trình

Kết quả huấn luyện mô hình

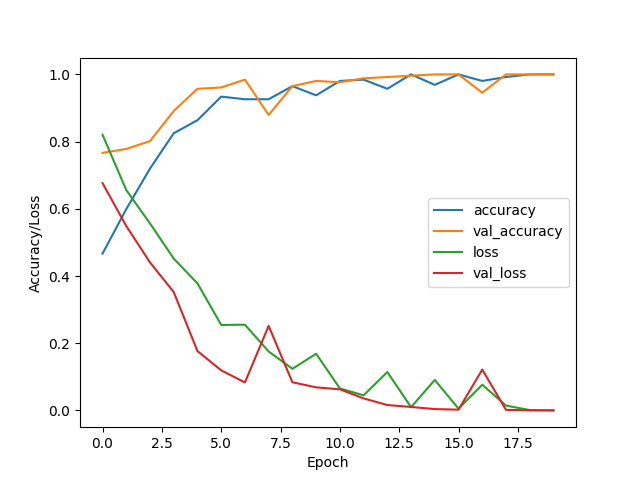
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epoch | Loss | Val\_loss | Accuracy | Val\_ Accuracy |
| 1 | 0.8203 | 0.6764 | 0.4669 | 0.7665 |
| 2 | 0.6558 | 0.5489 | 0.5992 | 0.7782 |
| 3 | 0.5565 | 0.4406 | 0.7198 | 0.8016 |
| 4 | 0.4520 | 0.3524 | 0.8249 | 0.8911 |
| 5 | 0.3781 | 0.1768 | 0.8638 | 0.9572 |
| 6 | 0.25450 | 0.9339 | 0.1195 | 0.9611 |
| 7 | 0.2552 | 0.0834 | 0.9261 | 0.9844 |
| 8 | 0.1756 | 0.2517 | 0.9261 | 0.8794 |
| 9 | 0.1241 | 0.0840 | 0.9650 | 0.9650 |
| 10 | 0.1689 | 0.0689 | 0.9377 | 0.9805 |
| 11 | 0.0657 | 0.0630 | 0.9805 | 0.9767 |
| 12 | 0.0451 | 0.0359 | 0.9844 | 0.9883 |
| 13 | 0.1145 | 0.0163 | 0.9572 | 0.9922 |
| 14 | 0.0099 | 0.0106 | 1.0000 | 0.9961 |
| 15 | 0.0909 | 0.0042 | 0.9689 | 1.0000 |
| 16 | 0.0054 | 0.0042 | 1.0000 | 1.0000 |
| 17 | 0.0767 | 0.1217 | 0.9805 | 0.9455 |
| 18 | 0.0146 | 0.0019 | 0.9922 | 1.0000 |
| 19 | 0.0013 | 3.6120e-04 | 1.0000 | 1.0000 |
| 20 | 3.6383e-04 | 2.8931e-04 | 1.0000 | 1.0000 |

Bảng huấn luyện 20 lần với tệp dữ liệu là 257 ảnh

Kết quả được huấn luyện từ lần 1 với sự mất mát (loss) là 0.8203 với tỷ lệ mất mát (val\_loss) lên đến 67,64% và sự chính xác (accuracy) là 0.4649 với tỷ lệ độ chính xác xác thực cao (val\_accuracy) là 76.65%.

sau 20 lần huấn luyện kết quả đã tăng với với sự mất mát (loss) là 3.6383e-04 với tỷ lệ mất mát (val\_loss) là 2.8931e-04 và sự chính xác (accuracy) là 1.0000 với tỷ lệ độ chính xác xác thực cao (val\_accuracy) là 1.0000.

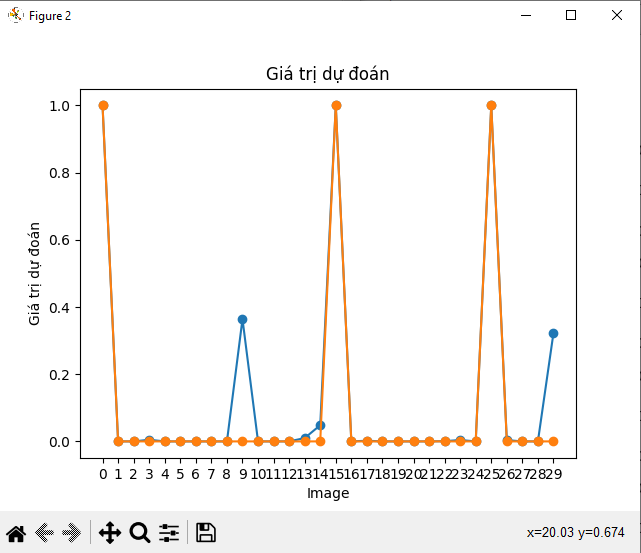
Biểu đồ hiển thị kết quả của sự mất mát (loss), sự chính xác (accuracy), tỷ lệ mất mát xác thực thấp (val\_loss) và tỷ lệ độ chính xác xác thực cao (val\_accuracy):



1. Kết quả ứng dụng mạng CNN nhận diện đeo khẩu trang hay không

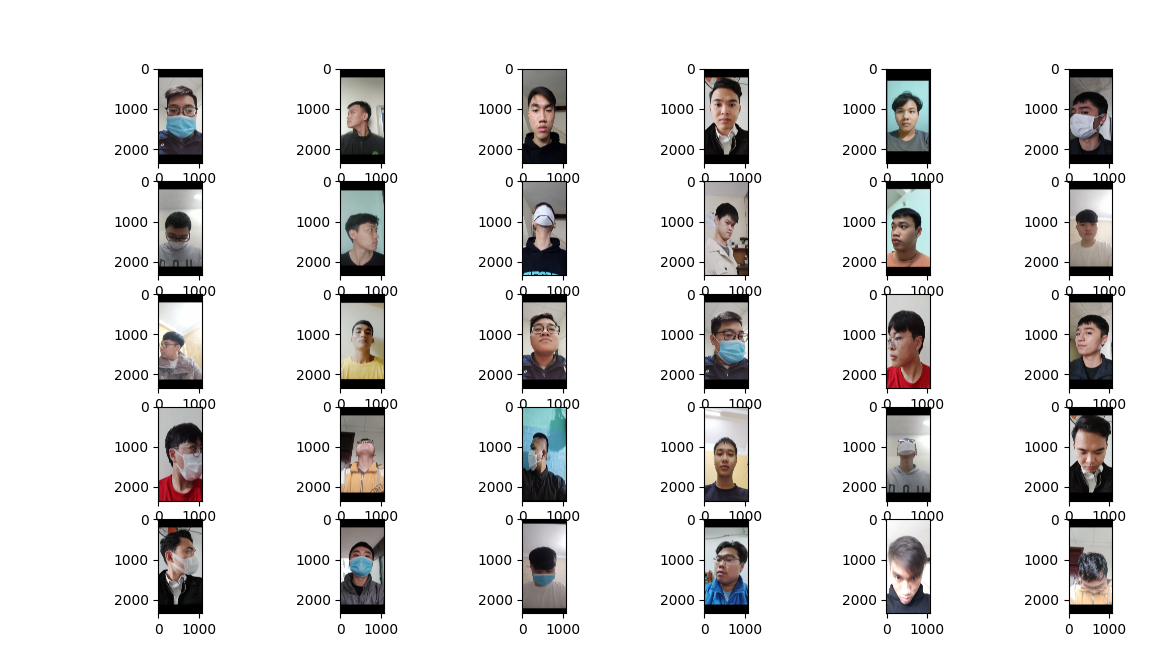
Kết quả ứng dụng mạng CNN nhận diện đeo khẩu trang hay không

Biểu đồ so sánh 30 ảnh được dự đoán:

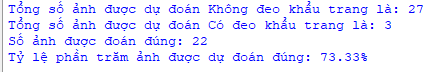


* Dòng cam là ảnh được dự đoán nhãn dãn thực tế.
* Dòng xanh là ảnh được dự đoán bằng mạng CNN.
* Nếu trùng nhau thì ảnh được dự đoán đúng.
* Nếu không trùng nhau thì ảnh được dự đoán sai.

30 ảnh được đưa vào để test:



Bảng Tổng số ảnh được dự đoán Không đeo khẩu trang,Tổng số ảnh được dự đoán Có đeo khẩu trang là,Số ảnh được đoán đúng và Tỷ lệ phần trăm ảnh được dự đoán đúng.



|  |  |  |  |
| --- | --- | --- | --- |
| Tổng số ảnh được dự đoán Không đeo khẩu trang | Tổng số ảnh được dự đoán Có đeo khẩu trang | Số ảnh được đoán đúng | Tỷ lệ phần trăm ảnh được dự đoán đúng |
| 27 | 3 | 22 | 73.33% |

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Độ chính xác: Sử dụng mạng CNN trong nhận dạng khuôn mặt đeo khẩu trang đã đạt được độ chính xác đáng kể. Dựa trên việc huấn luyện trên một tập dữ liệu lớn, mạng CNN có khả năng phân loại đúng các khuôn mặt đeo khẩu trang và nhận dạng một cách chính xác như trong lần test của em thì Tỷ lệ phần trăm ảnh được dự đoán đúng lên đến 73%.

## Hướng phát triển

So sánh giữa mạng CNN và các mạng học sâu khác:

Độ chính xác: CNN thường có khả năng nhận dạng ảnh và đạt được độ chính xác cao. Điều này là do cấu trúc đặc biệt của CNN, bao gồm các lớp Convolutional và Pooling, giúp nắm bắt các đặc trưng quan trọng trong ảnh.

Khả năng học đặc trưng: CNN có khả năng tự động học các đặc trưng từ dữ liệu, mà không cần sự can thiệp thủ công. Các lớp Convolutional trong CNN cho phép mô hình học các bộ lọc đặc trưng có thể nhận dạng các mẫu trong ảnh.

Tính hiệu quả tính toán: CNN được thiết kế để tận dụng tính chất cục bộ và chia sẻ trọng số, giúp giảm số lượng tham số cần học và giảm lượng tính toán trong quá trình huấn luyện và thử nghiệm.

Vì hệ thống còn một số hạn chế nên nhóm em sẽ tiếp tục cải thiện hệ thống về mặt nhận diện.

# DANH MỤC THAM KHẢO

1. (Hải, 2023) *Thuật toán CNN là gì? Hướng dẫn chọn tham số cho CNN.* Được truy lục từ Thuật toán CNN là gì? Hướng dẫn chọn tham số cho CNN: https://meeypage.com/tin-tuc/thuat-toan-cnn-la-gi-huong-dan-chon-tham-so-cho-cnn/
2. (Rossum, 2023) *Python (ngôn ngữ lập trình).* Được truy lục từ Python (ngôn ngữ lập trình): https://vi.wikipedia.org/wiki/Python\_(ng%C3%B4n\_ng%E1%BB%AF\_l%E1%BA%ADp\_tr%C3%ACnh)
3. (openCV): <https://en.wikipedia.org/wiki/OpenCV#cite_note-3>
4. Làm quen với Keras: <https://viblo.asia/p/lam-quen-voi-keras-gGJ59mxJ5X2>
5. TensorFlow là gì? Tìm hiểu về TensoFlow từ A đến Z: <https://topdev.vn/blog/tensorflow-la-gi/>
6. Thuật toán CNN là gì? Tìm hiểu về Convolutional Neural Network: <https://vietnix.vn/cnn-la-gi/>
7. Các hàm kích hoạt (activation function) trong neural network : <https://aicurious.io/blog/2019-09-23-cac-ham-kich-hoat-activation-function-trong-neural-networks>
8. Sami, A. A., Iqbal, Z., Shahid, M., Hussain, S. T., & Usman, I. (2020). Face Mask Detection and Recognition Using Convolutional Neural Network. IEEE Transactions on Image Processing, 29, 9808-9820: https://ieeexplore.ieee.org/document/9088597
9. Guo, X., Feng, Z., Zeng, X., Wang, J., & Xu, S. (2020). Real-Time Face Mask Detection System Using Convolutional Neural Networks. IEEE Access, 8, 195085-195094: <https://ieeexplore.ieee.org/document/9264437>
10. PHÁT HIỆN NG¯ỜI ĐEO KHẨU TRANG TRONG THỜI GIAN THỰC: <https://www.studocu.com/vn/document/hoc-vien-ngan-hang/kinh-doanh-quoc-te/56310-article-text-160692-1-10-20210507/29292380>
11. Việt Nam có công nghệ nhận diện cả khi đeo khẩu trang: https://sokhcn.camau.gov.vn/wps/portal/?1dmy&page=trangchitiet&urile=wcm%3Apath%3A/sokhvcnlibrary/siteofkhohocvacongnghe/tintucsukien/khoahoccongnghe/vncocongnghenhandangkhautrang