

PHÁT HIỆN NGƯỜI ĐEO KHẨU TRANG TRONG THỜI GIAN THỰC REAL TIME FACE MASK DETECTION

ĐỖ THỊ CHANG¹, NGUYỄN KHẮC KHIÊM², TRẦN SINH BIÊN^{1*}

¹Khoa Điện - Điện tử, Trường Đại học Hàng hải Việt Nam

²Trường Đại học Hàng hải Việt Nam

*Email liên hệ: transinhbien@vamaru.edu.vn

Tóm tắt

Việc đeo khẩu trang nơi công cộng đã góp phần hạn chế được sự lây lan của dịch bệnh Covid 19 trên toàn cầu. Giám sát người dân thực hiện đúng chủ trương, chính sách của Nhà nước trong việc đeo khẩu trang nơi công cộng hoàn toàn có thể được thực hiện một cách tự động. Bằng việc ứng dụng công nghệ máy học và xử lý ảnh, bài báo đã xây dựng chương trình phát hiện người dân có đeo khẩu trang hay không và nhắc nhở những người không đeo khẩu trang bằng giọng nói. Trong đó, nhóm tác giả đã thực hiện thu thập dữ liệu để xây dựng mô hình đào tạo dựa trên mạng nơron tích chập (CNN). Chương trình triển khai trên ngôn ngữ Python và sử dụng một số thư viện mã nguồn mở như OpenCV, Tensorflow, Keras,... Mô hình được đào tạo đạt độ chính xác 98,28% khi tiến hành phát hiện người đeo khẩu trang trên tập dữ liệu thử nghiệm.

Từ khóa: Covid 19, phát hiện người đeo khẩu trang, mạng nơron tích chập, CNN, Python, Tensorflow, Keras.

Abstract

Wearing face masks in public has contributed to reducing the global spread of Covid 19. And the supervision of people in accordance with the Government's guidelines and policies in wearing face masks in public can be done automatically. By using machine learning and image processing, we have developed a program to detect whether someone is wearing a face mask or not and remind people who are not wearing face masks by the voice. In particular, the authors collected data to build up a training model based on the convolutional neural network model (CNN). The program is implemented in Python language and uses some open source libraries such as OpenCV, Tensorflow, Keras,... The trained model achieved 98,28% accuracy when detecting the face mask on the test data set.

Keywords: Covid 19, face mask detection, convolutional neural network, CNN, Python, Tensorflow, Keras.

1. Đặt vấn đề

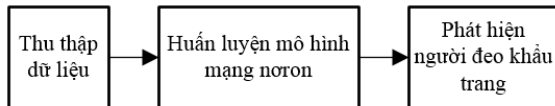
Ngày 11 tháng 3 năm 2020, Tổ chức Y tế Thế giới (WHO) ra tuyên bố gọi "COVID-19" là "Đại dịch toàn cầu" [1]. Để ngăn chặn tình trạng lây lan nhanh chóng của đại dịch, bên cạnh khuyến nghị mà WHO đưa ra về việc đeo khẩu trang ở nơi đông người, Chính phủ Việt Nam cũng đã yêu cầu người dân phải đeo khẩu trang nơi công cộng để hạn chế sự lây lan của dịch bệnh. Tuy nhiên, để giám sát người dân thực hiện theo đúng chỉ đạo của Chính phủ với những hình thức cũ là khá khó khăn và tốn kém vì thiếu nguồn lực. Nhằm hỗ trợ và nâng cao công tác giám sát và nhắc nhở người dân, nhóm tác giả đã xây dựng chương trình phát hiện người không đeo khẩu trang trong thời gian thực một cách tự động.

Với sự phát triển nhanh chóng của học sâu, đặc biệt là mạng nơron tích chập (CNN), thị giác máy tính đã đạt được những tiến bộ đáng kể trong những năm gần đây về nhận dạng và phát hiện đối tượng [2][3]. Một trong những ứng dụng của mạng nơron tích chập trong tình hình dịch bệnh đang diễn biến hết sức phức tạp trên toàn thế giới đó là việc xây dựng những hệ thống nhằm giám sát việc đeo khẩu trang của mọi người. Việc phát hiện người đeo khẩu trang bằng kỹ thuật học sâu đã được công bố rất nhiều trong thời gian gần đây. Một nghiên cứu về việc phát hiện người đeo khẩu trang nhằm hạn chế đại dịch Covid 19 trong mạng lưới thành phố thông minh đã được công bố gần đây [4]. Nhóm tác giả Mohammad Marufur Rahman và Md. Motaleb Hossen Manik đã đề cập một hệ thống phát hiện người không đeo khẩu trang một cách tự động và thông báo cho cơ quan có thẩm quyền tương ứng trong mạng thành phố thông minh. Hệ thống sử dụng camera CCTV của thành phố để trích xuất hình ảnh khuôn mặt và những hình ảnh này được sử dụng để xác định mặt nạ trên khuôn mặt. Sau đó, mạng nơron tích chập (CNN) được sử dụng để trích xuất các tính năng từ các hình ảnh đó và đưa ra dự đoán xem người đó có đeo khẩu trang hay không, nếu người đó không đeo khẩu

trang, hệ thống sẽ tự động gửi tin nhắn tới các cơ quan có thẩm quyền đề giải quyết.

Trong phạm vi nghiên cứu của bài báo, nhóm tác giả sẽ tiến hành phát hiện người không đeo khẩu trang cũng dựa trên mạng nơ ron tích chập, nhưng thay vì gửi thông tin tới cơ quan giám sát thì hệ thống sẽ đưa ra cảnh báo, trực tiếp nhắc nhở người dân thực hiện đúng việc đeo khẩu trang nơi công cộng bằng giọng nói.

Cấu trúc chương trình gồm 3 bước được thể hiện trên Hình 1.



Hình 1. Cấu trúc chương trình

Bước 1: Thu thập dữ liệu: Sử dụng mã nguồn mở OpenCV để phát hiện, lưu trữ 500 bức ảnh của người đeo khẩu trang và 500 bức ảnh của người không đeo khẩu trang.

Bước 2: Sử dụng nguồn dữ liệu đã thu thập được ở bước 1 để huấn luyện mô hình dựa trên mô hình mạng nơron tích chập CNN.

Bước 3: Phát hiện người đeo khẩu trang hay không. Bước này sẽ tiến hành so sánh dữ liệu được trích xuất từ camera (sau khi đã được xử lý dữ liệu đầu vào) với dữ liệu đã được huấn luyện để trích xuất kết quả và cảnh báo bằng giọng nói.

2. Thuật toán của chương trình

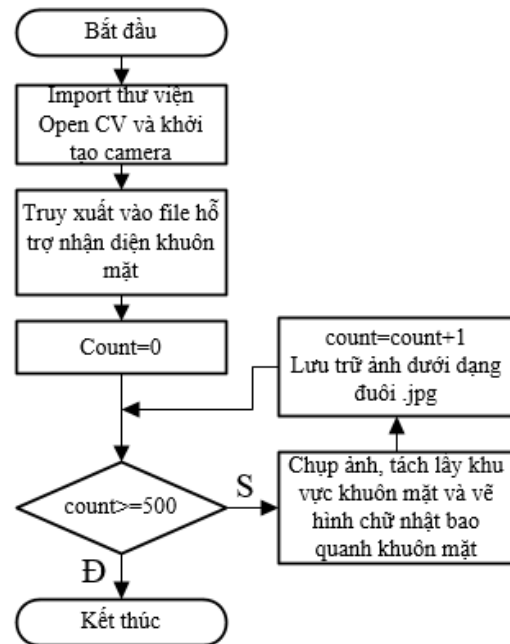
2.1. Thuật toán thu thập dữ liệu

Để thu thập dữ liệu, thay vì lấy nguồn dữ liệu trên mạng, nhóm tác giả đã thực hiện thu thập bằng chương trình được viết trên ngôn ngữ python, sử dụng thư viện mã nguồn mở OpenCV để phát hiện khuôn mặt người. Hình 2 thể hiện thuật toán thu thập dữ liệu.

Dữ liệu sau khi thu thập dưới dạng file ảnh .jpg sẽ được lưu trữ ở hai file riêng biệt: Một file chứa 500 bức ảnh mô tả khuôn mặt đeo khẩu trang và file còn lại chứa 500 bức ảnh mô tả khuôn mặt không đeo khẩu trang. Mỗi file dữ liệu trên sẽ được chia làm hai phần chính nhờ hàm split trong python, bao gồm:

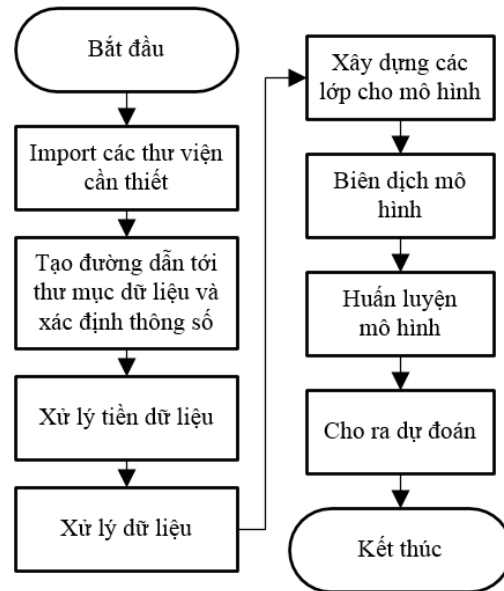
- Tập dữ liệu phục vụ quá trình thử nghiệm cuối cùng - Testing set (chiếm 20%);
- Tập dữ liệu phục vụ quá trình huấn luyện (chiếm 80%), trong đó tập dữ liệu huấn luyện này sẽ tiếp tục được chia thành 2 thành phần trai Training set (chiếm 80% của tập huấn luyện) và Validation set (chiếm 20% của tập huấn luyện) để kiểm thử độ chính xác của mô hình trong quá trình huấn luyện. Trong quá trình thu thập dữ liệu, điều kiện về ánh

sáng của bức ảnh cũng làm ảnh hưởng tới chất lượng tạo mô hình sau này.



Hình 2. Thuật toán thu thập dữ liệu

2.2. Thuật toán huấn luyện mô hình



Hình 3. Thuật toán huấn luyện mô hình

Mô hình huấn luyện được xây dựng dựa trên mô hình mạng nơron tích chập CNN với thuật toán trên Hình 3.

2.2.1. Import các thư viện cần thiết

Nhập các thư viện cần thiết như OpenCV, Keras, Numpy (để xử lý dữ liệu), Os (để quản lý file),...

2.2.2. Xử lý tiền dữ liệu và xử lý dữ liệu

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 bước sau.

2.2.3. Xây dựng các lớp cho mô hình

Các lớp được xây dựng dựa trên mô hình mạng nơron tích chập CNN (mạng nơron tích chập được thiết kế đặc biệt để xử lý dữ liệu đầu vào). Kiến trúc của CNN có ba phần chính:

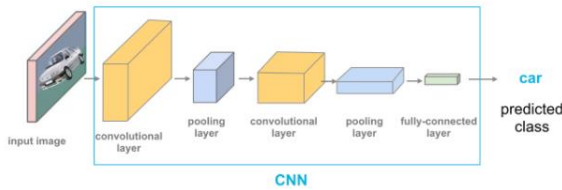
a) Một lớp phức hợp trích xuất các đặc trưng từ một hình ảnh nguồn. Để trích xuất đặc tính từ ảnh ban đầu, ta tiến hành tích chập hai ma trận bao gồm ma trận ảnh nguồn và bộ lọc với kích thước 3x3. Giả sử, Ma trận ảnh đầu vào có kích thước là H1 x W1 x D (H = height, W = width, D = dimension) và bộ lọc (kernel) là H2 x W2 x D thì ma trận đặc điểm ảnh sẽ được xác định theo công thức:

$$\left(\frac{H1-H2+2P}{S} + 1\right) \times \left(\frac{W1-W2+2P}{S} + 1\right) \times K \quad (1)$$

Trong đó: K là số bộ lọc;
 P là bước đệm;
 S là bước nhảy của bộ lọc.

b) Một lớp tổng hợp để giảm kích thước của nơron đi và tập trung vào các phần tử quan trọng nhất. Trong phần này ta sử dụng lớp pooling có kích thước (2x2) và sử dụng kiểu max pooling hay lấy giá trị lớn nhất.

c) Một lớp được kết nối đầy đủ làm phẳng các đối tượng được xác định trong các lớp trước đó thành một vector và dự đoán xác suất mà hình ảnh có thể thuộc về nhãn nào đó.



Hình 4. Cấu trúc mạng CNN [5]

Ngoài ra, để kích hoạt cho nơron hoạt động ta cần phải xác định hàm kích hoạt cho nó. Trong chương trình được xây dựng, nhóm tác giả đã sử dụng cả hai hàm kích hoạt là ReLu [6] và Softmax [7] với thuật toán được xác định:

ReLU:
 $f(x)=\max(0,x) \quad (2)$

Softmax:
 $a(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ với } j = 1, \dots, k \quad (3)$

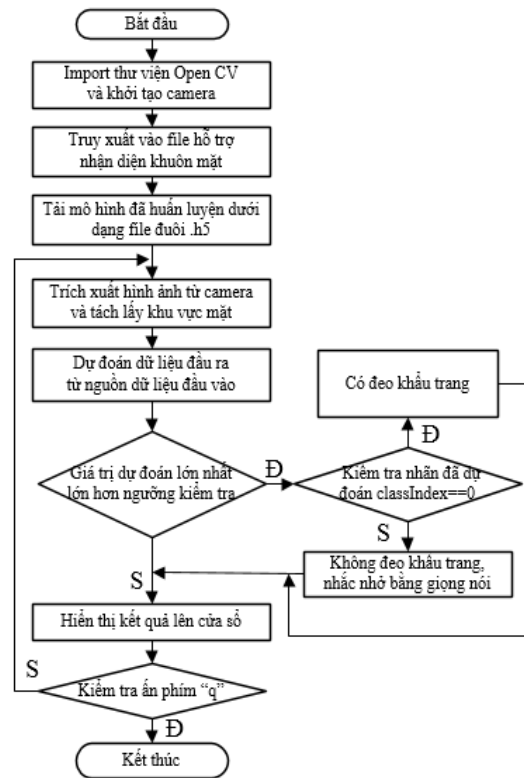
Bảng 1. Các lớp Layer trong mô hình

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 30, 30, 32)	320
conv2d_2 (Conv2D)	(None, 28, 28, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 32)	0
conv2d_3 (Conv2D)	(None, 12, 12, 64)	18496
conv2d_4 (Conv2D)	(None, 10, 10, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 5, 5, 64)	0
dropout_1 (Dropout)	(None, 5, 5, 64)	0
flatten_1 (Flatten)	(None, 1600)	0
dense_1 (Dense)	(None, 64)	102464
dropout_2 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 2)	130

2.2.4. Biên dịch mô hình

Thực hiện các thuật toán tối ưu và biên dịch chương trình. Để tối ưu mô hình, nhóm tác giả đã sử dụng thuật toán Adam [8] vì yêu cầu ít bộ nhớ, các tham số được biến thiên một cách hiệu quả và ít phải điều chỉnh.

2.3. Thuật toán phát hiện người đeo khẩu trang trong thời gian thực



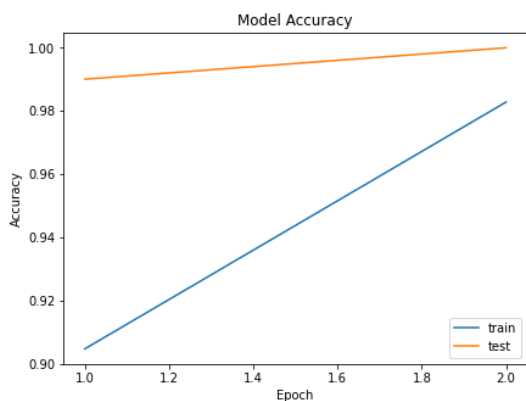
Hình 5. Thuật toán phát hiện người đeo khẩu trang trong thời gian thực

Để phát hiện khuôn mặt, nhóm tác giả sử dụng thuật toán phát hiện khuôn mặt Viola-Jones [9] giống như trong chương trình thu thập dữ liệu.

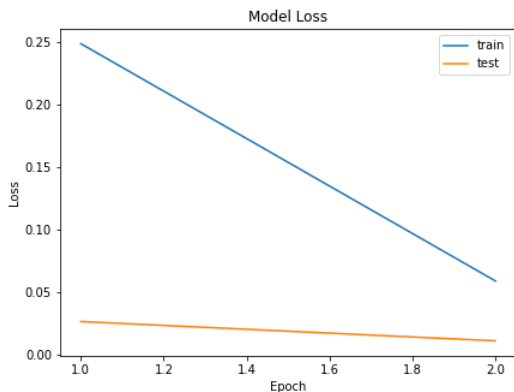
Sau khi ảnh được cắt từ video, nó sẽ được định dạng lại kích thước giống với kích thước của dữ liệu trong mô hình huấn luyện. Sau đó, ảnh này sẽ được chuyển đổi thành những tham số có định dạng giống mô hình mẫu. Khi đó các tham số sinh ra được từ tập dữ liệu huấn luyện sẽ được sử dụng để thẩm định lại tính thích hợp của mô hình trên tập dữ liệu của hình ảnh vừa được trích xuất. Dựa vào kết quả thu được ta tiến hành hiển thị lên màn hình kết quả người dân có đeo khẩu trang hay không. Nếu người đó không đeo khẩu trang thì sẽ lập tức nhắc nhở người đó đeo khẩu trang thông qua lời nói trực tiếp. Việc nhắc nhở này sẽ được thực thi nhờ sự hỗ trợ của thư viện “pyttsx3” - thư viện hỗ trợ chuyển đổi văn bản thành giọng nói trong python. Thuật toán xây dựng như trong Hình 7.

3. Kết quả

Sau 2 lần training mô hình, độ chính xác (accuracy) của mô hình đã đạt tới 98,28%, độ mất mát (loss) giảm còn 5,88%. Hình 6 và 7 cho thấy dạng đồ thị của độ chính xác và sự mất mát của mô hình.

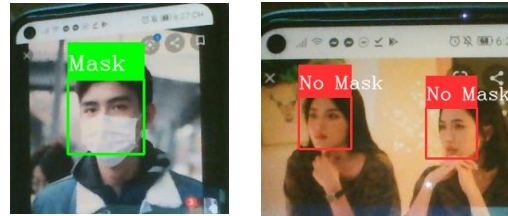


Hình 6. Đồ thị độ chính xác của mô hình

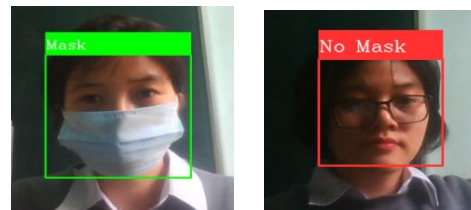


Hình 7. Đồ thị độ mất mát của mô hình

Chương trình đã thực hiện kiểm tra trong nhiều điều kiện khác nhau, bao gồm cả những hình ảnh khuôn mặt được hiển thị trên điện thoại thì vẫn cho ra kết quả khá chính xác như Hình 8 và Hình 9.



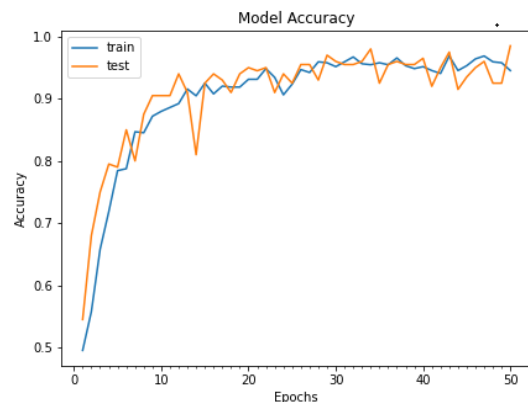
Hình 8. Kết quả khi phát hiện khẩu trang với khuôn mặt người trên điện thoại



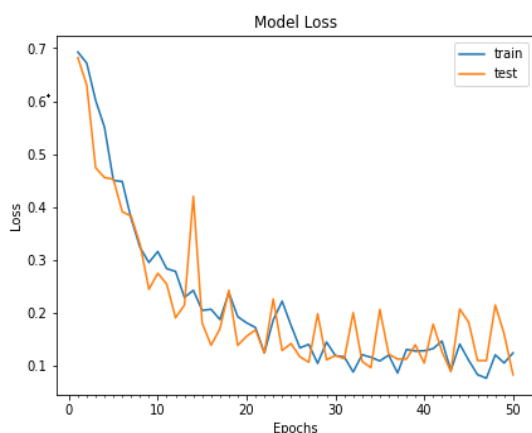
Hình 9. Kết quả khi phát hiện khẩu trang trong thời gian thực

Tuy nhiên, khi điều kiện ánh sáng kém việc phát hiện khẩu trang vẫn chưa thực sự hiệu quả. Chương trình chưa phân biệt được người đeo khẩu trang hay không khi không phát hiện được mặt của chủ thể.

Để đảm bảo tính khách quan, nhóm tác giả cũng đồng thời xây dựng mô hình huấn luyện khác với tập dữ liệu công cộng từ nguồn [10]. Nhóm tác giả chỉ trích xuất lấy lượng dữ liệu với số lượng 500 ảnh người đeo khẩu trang và 500 ảnh người không đeo khẩu trang từ nguồn dữ liệu công cộng. Độ chính xác của mô hình dựa trên tập dữ liệu trích xuất này cũng lên tới 94,53%, độ mất mát giảm còn 12,39% sau 50 epochs. Kết quả này cũng được biểu diễn trên biểu đồ đường như Hình 10 và 11.



Hình 10. Độ chính xác của mô hình dựa trên tập dữ liệu công cộng



Hình 11. Độ mất mát của mô hình dựa trên tập dữ liệu công cộng

Một mô hình phát hiện người đeo khẩu trang dựa trên tập dữ liệu công cộng này (gồm 4095 hình ảnh) cũng đã được công bố gần đây [11] bởi Chandrika Deb với độ chính xác lên tới 98% chỉ sau 20 epochs.

4. Kết luận

Bài báo đã xây dựng thành công chương trình phát hiện và nhắc nhở người không đeo khẩu trang nhằm hỗ trợ công tác giám sát người dân thực hiện đúng quy định của Chính phủ về việc đeo khẩu trang nơi công cộng. Nhất là vào thời điểm hiện tại, khi tốc độ truyền nhiễm của dịch bệnh chưa có dấu hiệu suy giảm.

Chương trình được viết bằng ngôn ngữ Python và sử dụng một số thư viện mã nguồn mở như OpenCV, Tensorflow, Keras,... Dựa vào những kết quả thu được, ta có thể kết hợp chương trình với các thiết bị phần cứng như Raspberry[12], Arduino[13],... để xây dựng hệ thống giám sát và nhắc nhở người dân đeo khẩu trang ở những nơi đông người như: trung tâm thương mại, trường học,...

TÀI LIỆU THAM KHẢO

[1] VnExpress, *Đại dịch thay đổi thế giới năm 2020*, 16/12/2021.
 <<https://vnexpress.net/dai-dich-thay-doi-the-gioi-nam-2020-4207154.html>>

[2] Liu Y, Sun P, Highsmith M R, et al. *Performance comparison of deep learning techniques for recognizing birds in aerial images*, 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC). IEEE, pp.317-324, 2018.

[3] Wenxuan Han; Zitong Huang; Alifu. kuerban; Meng Yan; Haitang Fu, *A Mask Detection Method for Shoppers Under the Threat of COVID-19 Coronavirus*, 2020 International Conference on

Computer Vision, Image and Deep Learning (CVIDL), 2020.

[4] Mohammad Marufur Rahman, Md. Motaleb Hossen Manik, *An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network*, 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2020.

[5] Cezanne Camacho, 2018, *Convolutional Neural Networks*.
 <https://cezannec.github.io/Convolutional_Neural_Networks/>

[6] Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung, *Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit*. Nature 405, 6789, 2000.

[7] Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron. 6.2.2.3 *Softmax Units for Multinoulli Output Distributions*. Deep Learning. MIT Press. pp.180-184. ISBN 978-0-26203561-3. 2016.

[8] Diederik P. Kingma, Jimmy Ba, *Adam: A method for Stochastic Optimization*, at the 3rd International Conference for Learning Representations, San Diego, 2015.

[9] P. Viola and M. J. Jones, *Robust real-time face detection*, Int. J. Comput. Vision, Vol.57, No.2, pp.137-154, May 2004.

[10] GitHub-Chandrika Deb. [Online]. Available: <https://github.com/chandrikadeb7/Face-Mask-Detection/tree/master/dataset>, accessed on: Jan. 7, 2021.

[11] GitHub-Chandrika Deb. [Online]. Available: <https://github.com/chandrikadeb7/Face-Mask-Detection#face-mask-detection>.

[12] Gay, Warren, Book *Raspberry Pi Hardware Reference*, Apress, 2014.

[13] Leo Louis, *Working Principle of Arduino and Using it as a Tool for Study and Research*, International Journal of Control, Automation, Communication and Systems 1, 2016.

Ngày nhận bài:	19/01/2021
Ngày nhận bản sửa:	28/01/2021
Ngày duyệt đăng:	18/02/2021