

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC CẦN THƠ**



## **BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN  
HỆ THỐNG KIỂM TRA VÀ NHẮC NHỞ  
ĐEO KHẨU TRANG  
THS2022-13**

Thuộc nhóm ngành khoa học: Công nghệ Thông tin và Truyền thông

**Cần Thơ, 10/2022**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC CẦN THƠ**



**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN  
HỆ THỐNG KIỂM TRA VÀ NHẮC NHỞ  
ĐEO KHẨU TRANG  
THS2022-13**

Thuộc nhóm ngành khoa học: Công nghệ Thông tin và Truyền thông

Sinh viên thực hiện: Châu Thái Phong                      Nam, Nữ: Nam

Dân tộc: Kinh

Lớp, khoa: DI18V7F1 – K44    Khoa: Công nghệ Thông tin và Truyền thông

Ngành học: Công nghệ thông tin (chất lượng cao)

Giáo viên hướng dẫn: PGS-TS. Đỗ Thanh Nghị

**Cần Thơ, 10/2022**

**DANH SÁCH THÀNH VIÊN THAM GIA**

<b>HỌ VÀ TÊN</b>	<b>ĐƠN VỊ</b>
Châu Thái Phong	MSSV: B1809716 Lớp: CNTT Chất lượng cao F1-K44
Nguyễn Bá An	MSSV: B1812823 Lớp: CNTT Chất lượng cao F1-K44
Tôn Thất Toại	MSSV: B1809725 Lớp: CNTT Chất lượng cao F1-K44

**ĐƠN VỊ PHỐI HỢP CHÍNH**

- ✓ Khoa Công nghệ Thông tin & Truyền thông – Đại học Cần Thơ
- ✓ Đoàn Trường Đại học Cần Thơ
- ✓ Phòng Quản lý Khoa học – Đại học Cần Thơ

**DANH MỤC TỪ VIẾT TẮT**

Từ viết tắt	Từ tiếng Anh	Diễn giải
OpenCV	Open Source Computer Vision Library	Là thư viện nguồn mở hàng đầu cho Computer Vision và Machine Learning
CNN	Convolutional Neural Network	Cấu trúc mạng Convolutional Neural Network
ML	Machine Learning	Máy học

## THÔNG TIN KẾT QUẢ NGHIÊN CỨU CỦA ĐỀ TÀI

## 1. Thông tin chung:

- Tên đề tài: **HỆ THỐNG KIỂM TRA VÀ NHẮC NHỞ ĐEO KHẨU TRANG**
- Sinh viên thực hiện: **Châu Thái Phong**
- Lớp: **CNTT-F1 (chất lượng cao)**
- Khoa: **Công nghệ Thông tin và Truyền thông**
- Năm thứ: **4**                                      - Số năm đào tạo: **4.5**
- Giáo viên hướng dẫn: **PGS-TS. Đỗ Thanh Nghi**

## 2. Mục tiêu đề tài:

- Xây dựng một hệ thống phát hiện một người có đeo khẩu trang hay không.
- Xây dựng một ứng dụng chạy trên thiết bị di động, dễ sử dụng và thao tác.
- Khi phát hiện một người đang không đeo khẩu trang, hệ thống sẽ tự động nhắc nhở người đó đeo khẩu trang vào.

### 3. Tính mới và sáng tạo:

Ứng dụng được chạy trên website của thiết bị di động, dễ thao tác và sử dụng

#### 4. Kết quả nghiên cứu:

Xây dựng thành công hệ thống kiểm tra với chức năng chính là nhận biết được một người có đang đeo khẩu trang không.

**5. Đóng góp về mặt kinh tế - xã hội, giáo dục và đào tạo, an ninh, quốc phòng và khả năng áp dụng của đề tài:**

### 5.1. Về phát triển kinh tế- xã hội

- Hỗ trợ về phát hiện tự động, giúp giảm nguy cơ lây nhiễm của dịch COVID-19 trong thời gian xã hội đang trở về bình thường hóa.
- Hệ thống dễ xài, dễ truy cập.

## 5.2. Về triển khai ứng dụng

Ứng dụng hoàn toàn có khả năng triển khai trong thực tế tại các nơi công cộng, trường học, bệnh viện, quán ăn,...

Ngày tháng năm 2022

**Sinh viên chịu trách nhiệm thực hiện đề tài**  
(ký, họ và tên)

## Châu Thái Phong

**Nhận xét của người hướng dẫn về những đóng góp khoa học của sinh viên thực hiện đề tài** *(phần này do người hướng dẫn ghi):*

**Xác nhận của Trường Đại học Cần Thơ**  
*(ký tên và đóng dấu)*

Ngày    tháng    năm 2022  
**Giáo viên hướng dẫn**  
*(ký, họ và tên)*

**Đỗ Thanh Nghị**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**THÔNG TIN VỀ SINH VIÊN**  
**CHỊU TRÁCH NHIỆM CHÍNH THỰC HIỆN ĐỀ TÀI**

**I. SƠ LƯỢC VỀ SINH VIÊN:**

Họ và tên: Châu Thái Phong  
Sinh ngày: 02 tháng 10 năm 2000  
Nơi sinh: Cần Thơ  
Lớp: CNTT-F1 chất lượng cao Khóa: 44  
Khoa: Công nghệ Thông tin và Truyền Thông  
Địa chỉ liên hệ: 223 đường Trần Quang Diệu, An Thới, Bình Thủy, Cần Thơ  
Điện thoại: 0946686455 Email: phongb1809716@student.ctu.edu.vn

Ảnh 4x6

**II. QUÁ TRÌNH HỌC TẬP** (kê khai thành tích của sinh viên từ năm thứ 1 đến năm đang học):

**\* Năm thứ 1:**

Ngành học: CNTT (chất lượng cao) Khoa: Công nghệ Thông tin và Truyền thông  
Kết quả xếp loại học tập: 2.13

**\* Năm thứ 2:**

Ngành học: CNTT (chất lượng cao) Khoa: Công nghệ Thông tin và Truyền thông  
Kết quả xếp loại học tập: 2.14

**\* Năm thứ 3:**

Ngành học: CNTT (chất lượng cao) Khoa: Công nghệ Thông tin và Truyền thông  
Kết quả xếp loại học tập: 2.38

**\* Năm thứ 4:**

Ngành học: CNTT (chất lượng cao) Khoa: Công nghệ Thông tin và Truyền thông  
Kết quả xếp loại học tập: 2.45

**Xác nhận của Trường Đại học Cần Thơ**  
(ký tên và đóng dấu)

Ngày tháng năm 2022  
**Sinh viên chịu trách nhiệm chính**  
**thực hiện đề tài**  
(ký, họ và tên)

## MỤC LỤC

<b>CHƯƠNG 1: MỞ ĐẦU</b>	1
1. ĐẶT VẤN ĐỀ	1
2. TỔNG QUAN TÌNH HÌNH NGHIÊN CỨU TRONG VÀ NGOÀI NƯỚC	2
3. SỰ CẦN THIẾT CỦA ĐỀ TÀI	3
4. MỤC TIÊU CỦA ĐỀ TÀI	4
5. NỘI DUNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU	4
6. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU	4
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT</b>	5
1. Nhận diện khuôn mặt	5
1.1 Khái niệm	5
1.2 Các bước để nhận diện khuôn mặt	6
2. Thuật toán CNN	7
2.1 Khái niệm	7
2.2 Lớp tích chập – Convolution Layer	8
2.3 Ứng dụng thực tế trong đề tài	8
3. Open CV	9
3.1 Khái niệm	9
3.2 Ứng dụng thực tế trong đề tài	9
4. Tensorflow	10
4.1 Khái niệm	10
4.2 Nguyên lý hoạt động:	10
4.3 Kiến trúc của Tensorflow	10
4.4 Ứng dụng thực tế trong đề tài	11
5. Keras	11
5.1 Khái niệm	11
5.2 Cấu trúc của Keras	12
6. MobileNet V2	12
6.1 Khái niệm	12
6.2 Sự khác nhau giữa v1 và v2	12
7. Python	15
7.1 Khái niệm	15
7.2 Tính năng chính của Python	15
7.3 Lý do chọn Python cho đề tài	16
<b>CHƯƠNG 3: MÔ TẢ HỆ THỐNG</b>	19
1. TỔNG QUAN VỀ HỆ THỐNG	19



2. MÔ TẢ ỨNG DỤNG .....	19
3. NỘI DUNG NGHIÊN CỨU .....	20
3.1 Bài toán phát hiện khẩu trang .....	20
3.2 Tiền xử lý dữ liệu.....	20
3.3 Huấn luyện mô hình.....	21
3.4 Vận hành mô hình.....	22
3.5 Phát hiện khẩu trang trên ảnh .....	23
3.6 Phát hiện khẩu trang ở thời gian thực .....	23
<b>CHƯƠNG 4: XÂY DỰNG HỆ THỐNG NHẬN BIẾT KHUÔN MẶT ĐEO KHẨU TRANG .....</b>	<b>24</b>
1. KẾT QUẢ NGHIÊN CỨU .....	24
2. XÂY DỰNG MÔ HÌNH .....	23
2.1 Các thư viện cần sử dụng .....	23
2.2 Kết nối với Google drive để đọc file và lưu file.....	23
2.3 Khai báo đường dẫn đến thư mục chứa dữ liệu .....	24
2.4 Dán nhãn cho dữ liệu .....	24
2.5 Tiền xử lý dữ liệu .....	25
2.6 Đọc dữ liệu train, validation và gán nhãn tương ứng với cấu trúc của các thư mục con .....	24
2.7 Xây dựng mô hình .....	25
2.8 Thiết lập các tham số để huấn luyện mô hình .....	26
2.9 Huấn luyện mô hình .....	26
2.10 Sử dụng mô hình .....	26
2.11 Giao diện .....	27
2.11.1 Giao diện chính .....	27
2.11.2 Giao diện khi mô hình hoạt động .....	27
3. ĐÁNH GIÁ KIỂM THỬ MÔ HÌNH .....	30
3.1 Kiểm thử 01 .....	28
3.2 Kiểm thử 02 .....	28
3.3 Kiểm thử 03 .....	29
3.4 Kiểm thử 04 .....	30
<b>CHƯƠNG 5: KIỂM THỬ HỆ THỐNG ỨNG DỤNG .....</b>	<b>31</b>
1. MỤC TIÊU KIỂM THỬ .....	31
2. NỘI DUNG KIỂM THỬ .....	31
2.1.Các tính năng được kiểm thử .....	31
2.2 Tiêu chí kiểm thử .....	31
2.3 Sản phẩm bàn giao của kiểm thử .....	31
2.4 Quản lý kiểm thử .....	31
2.4.1 Thiết bị sử dụng .....	31
2.4.2 Môi trường kiểm thử .....	31
2.4.3 Các trường hợp kiểm thử .....	32
2.4.3.1 Sử dụng webcam để kiểm tra .....	32
2.4.3.2 Sử dụng camera trước điện thoại (có kết nối internet) .....	35
2.4.3.3 Sử dụng camera sau điện thoại (có kết nối internet) .....	38
2.5 Kết luận kiểm thử .....	40

<b>CHƯƠNG 6: KẾT LUẬN VÀ KIẾN NGHỊ .....</b>	<b>50</b>
<b>I. KẾT QUẢ ĐÓNG GÓP .....</b>	<b>50</b>
1. Về khoa học và đào tạo .....	50
2. Về phát triển kinh tế - xã hội.....	50
<b>II. TRIỂN KHAI ỨNG DỤNG.....</b>	<b>50</b>
<b>III. HƯỚNG PHÁT TRIỂN.....</b>	<b>50</b>

## **MỤC LỤC HÌNH**

Hình 1.1 .....	3
Hình 2.1 .....	5
Hình 2.2 .....	6
Hình 2.3 .....	7
Hình 2.4 .....	7
Hình 2.5 .....	8
Hình 2.6 .....	8
Hình 2.7 .....	11
Hình 2.8 .....	12
Hình 2.9 .....	13
Hình 2.10 .....	14
Hình 2.11 .....	16
Hình 3.1 .....	18
Hình 3.2 .....	19
Hình 3.3 .....	19
Hình 3.4 .....	20
Hình 3.5 .....	20
Hình 3.6 .....	21
Hình 3.7.....	21
Hình 4.1 .....	26
Hình 4.2 .....	27
Hình 4.3 .....	27
Hình 4.4 .....	28
Hình 4.5 .....	28
Hình 4.6 .....	29
Hình 4.7 .....	30
Hình 4.8 .....	30
Kết quả kiểm thực tế .....	32 - 37

## MỤC LỤC BẢNG

Bảng 2.1 .....	14
Bảng 2.2 .....	14
Bảng 4.1 .....	22
Bảng 4.2 .....	23
Bảng 4.3 .....	23
Bảng 4.4 .....	24
Bảng 4.5 .....	24
Bảng 4.6 .....	24
Bảng 4.7 .....	24
Bảng 4.8 .....	25
Bảng 4.9 .....	26
Bảng 4.10 .....	26
Bảng 4.11 .....	26
Bảng 5.1 .....	39

## **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, nhóm đã nghiên cứu, xây dựng chương trình phát hiện người 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 nơon 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 cao khi tiến hành phát hiện người đeo khẩu trang trên tập dữ liệu thử nghiệm.

# CHƯƠNG 1: MỞ ĐẦU

Trong tình hình dịch bệnh Covid-19 diễn biến phức tạp và đầy thách thức như hiện nay, nhóm chúng tôi đã lựa chọn nghiên cứu đề tài và đặt ra các mục tiêu cho đề tài. Trong chương này chúng tôi sẽ giới thiệu về đề tài, lý do chọn đề tài, tổng quan về tình hình nghiên cứu của đề tài trong nước và ngoài nước, mục tiêu đề tài, các phương pháp nghiên cứu, đối tượng và phạm vi nghiên cứu của đề tài.

## 1. ĐẶT VẤN ĐỀ

Hiện tại, dịch bệnh COVID-19 vẫn có diễn biến phức tạp trên toàn thế giới, nó không chỉ ảnh hưởng đến kinh tế, thương mại, dịch vụ mà còn ảnh hưởng không nhỏ đến tâm lý xã hội của người dân và chưa có vacxin điều trị cho nên cần kiểm soát tốt nguồn lây nhiễm. Leung et al. cho rằng khẩu trang có thể làm giảm sự lây lan của vi-rút Corona vì đa số các ca lây nhiễm COVID-19 do tiếp xúc rất gần và không đeo khẩu trang khi nói chuyện.

Theo trung tâm kiểm soát bệnh tật CDC (Centers for Disease Control and Prevention), khẩu trang được khuyến nghị là một rào chắn đơn giản để giúp ngăn các giọt bắn từ đường hô hấp bay vào không khí và lên người khác. Khuyến nghị này dựa trên nghiên cứu vai trò của các giọt bắn từ đường hô hấp đối với sự lây lan của vi-rút gây ra COVID-19, kết hợp với bằng chứng mới xuất hiện từ các nghiên cứu lâm sàng và trong phòng thí nghiệm cho thấy khẩu trang làm giảm việc phun các giọt bắn khi đeo qua mũi và miệng [2]. Các giọt cũng có thể rơi xuống các bề mặt nơi vi-rút có thể tồn tại. Do đó, môi trường trực tiếp của một cá nhân bị nhiễm bệnh có thể đóng vai trò là nguồn lây truyền do tiếp xúc.

Theo các chuyên gia của Tổ chức Y tế thế giới (WHO), đeo khẩu trang có thể hạn chế sự lây lan của một số bệnh đường hô hấp trong đó có COVID-19. Dịch bệnh này lây lan chủ yếu giữa những người tiếp xúc gần với nhau (trong khoảng 6 feet), vì vậy việc sử dụng khẩu trang đặc biệt quan trọng ở những nơi mọi người gần nhau hoặc nơi khó có thể duy trì cách ly xã hội. WHO cũng khuyến cáo mọi người nên đeo khẩu trang nếu họ có triệu chứng hô hấp, hoặc họ đang chăm sóc những người có triệu chứng. Như vậy, phát hiện người đeo khẩu trang đề cập đến việc phát hiện xem một người đeo khẩu trang hay không và vị trí của khuôn mặt đó trở thành một nhiệm vụ quan trọng nhưng nghiên cứu liên quan đến vấn đề này chưa phổ biến. Trên thế giới, đã có một số nghiên cứu phát hiện người đeo hay không đeo khẩu trang.

Trong đã xây dựng mô hình CNN để phát hiện khuôn mặt đeo khẩu trang, mô hình có độ chính xác 98,86% với bộ huấn luyện và 96,19% với bộ thử nghiệm. Tác giả sử dụng đặc trưng Haar để phát hiện khuôn mặt và CNN với các lớp khác nhau như Conv2D, MaxPooling2D, Flatten, Dropout và Dense. Trong sử dụng PyTorch và CNN để cài đặt học sâu. Các mô hình đã được thử nghiệm với hình ảnh và video thời gian thực với độ chính xác của mô hình là 60%. Trong dùng mô hình học sâu thay vì mô hình Haar-Cascade, CNN sử dụng 4 lớp chập theo sau là 2 lớp tuyến tính, dùng ReLU làm chức năng kích hoạt và MaxPool2 là lớp tổng hợp. Huấn luyện mô hình trong 10 lần lặp và độ chính xác đạt cao nhất tại lần lặp thứ 8 là 99%.

Trong sử dụng deeplearning và bộ dữ liệu của tác giả Prajna Bhandary là bộ dữ liệu đào tạo có phần hạn chế bởi nhãn lớp có khẩu trang được tạo một cách nhân tạo. Cụ thể, chụp ảnh bình thường của khuôn mặt sau đó tùy chỉnh để thêm khẩu trang cho chúng. Phương pháp này dễ hơn nhiều so với áp dụng các mốc trên khuôn mặt. Các mốc trên khuôn mặt cho phép chúng ta tự động suy ra vị trí của các cấu trúc khuôn mặt, bao gồm: Đôi mắt, lông mày, mũi, miệng, hàm. Dựa vào các mốc trên khuôn mặt để xây

dựng bộ dữ liệu khuôn mặt đeo khẩu trang. Ngoài ra, bộ dữ liệu nhân tạo hình ảnh khuôn mặt đeo khẩu trang không chứa các hình ảnh có thể khiến phân loại nhầm. Ở Pháp cũng sử dụng AI để kiểm tra xem mọi người có đeo khẩu trang trên phương tiện giao thông công cộng hay không như tại ga tàu điện ngầm Chatelet-Les Halles. Tại Việt Nam, Robot do Đại học Công nghệ (ĐH Quốc gia) chế tạo để phát hiện người không đeo khẩu trang và nhắc nhở sử dụng những thiết bị, linh kiện có sẵn. Camera tích hợp cảm biến laser để đo khoảng cách, phát hiện người ra vào cửa. Cùng lúc, camera sẽ thu lại hình ảnh và truyền tới máy tính. Các chi tiết chính trên khuôn mặt như mắt, mũi, miệng sẽ được số hóa và xử lý. Nếu máy tính phát hiện mũi và miệng bị che, có nghĩa là người đó đang đeo khẩu trang. Còn trong trường hợp miệng bị che nhưng mũi hở, hoặc cả mũi và miệng đều hở, nghĩa là người đó đeo chưa đúng cách hoặc không đeo. Khi đó máy tính sẽ lập tức kích hoạt loa và phát ra thông báo.

Độ chính xác phụ thuộc nhiều vào bối cảnh, chẳng hạn ánh sáng, cách di chuyển của người ra vào cao nhất đã đạt 95%. Như vậy, phát hiện khẩu trang là một nhiệm vụ cần thiết trong tình hình hiện nay, mục tiêu ngoài nhắc nhở những cá nhân không đeo khẩu trang, mà còn tạo ra dữ liệu thống kê giúp chính quyền dự đoán sự bùng phát của COVID-19 trong tương lai. Nghiên cứu này tiếp cận theo phương pháp học sâu sử dụng mô hình MobileNetV2, một mô hình khá nhẹ, số lượng tham số ít, tốc độ tính toán nhanh và dễ triển khai ứng dụng real time trên các thiết bị di động hoặc thống nhúng có khả năng tính toán thấp để tự động phát hiện cùng lúc nhiều người không đeo khẩu trang trên ảnh và video thời gian thực với bộ dữ liệu người đeo khẩu trang thực.

## **2. TỔNG QUAN TÌNH HÌNH NGHIÊN CỨU TRONG VÀ NGOÀI NƯỚC**

- **Trong nước**

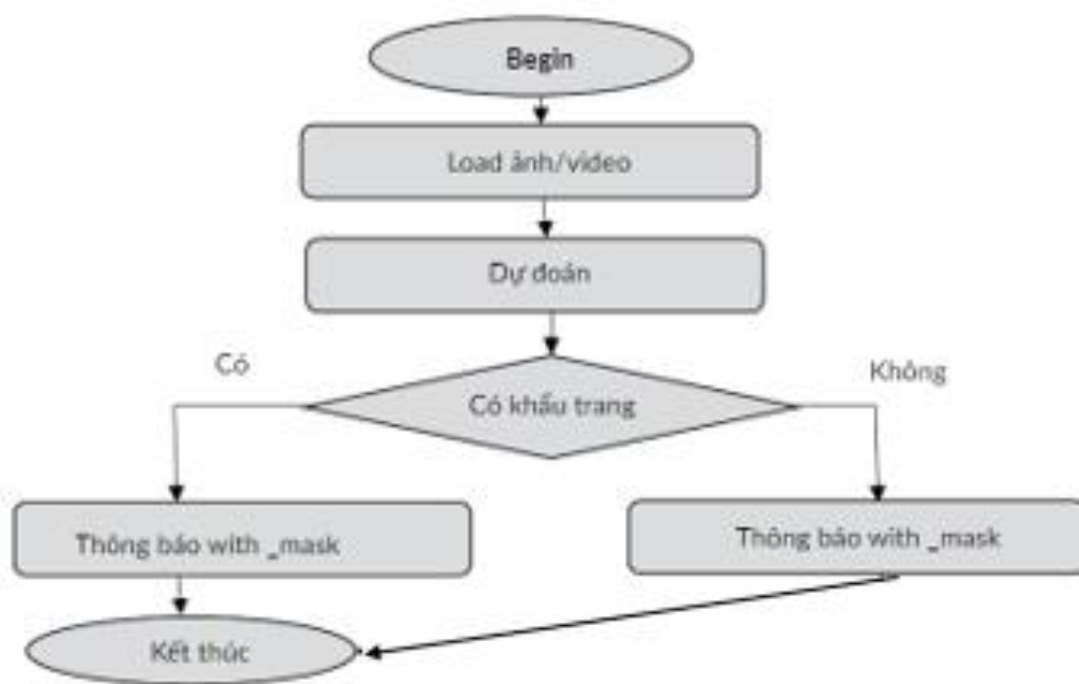
Sự phát triển công nghệ trong nước đang có những bước tiến đáng kể trong thời gian qua và sự cải tiến công nghệ đem lại sự tiện lợi và hiệu quả trong công việc đang được nhiều nhà đầu tư hướng đến. Tại Việt Nam QTSC đã phối hợp với công ty TMA Innovation tại Công viên phần mềm Quang Trung nghiên cứu và phát triển, đưa vào ứng dụng trong thực tế “Ứng dụng kiểm tra thân nhiệt và nhắc đeo khẩu trang tự động” nhằm sàng lọc người nghi nhiễm Covid-19. Khi có người đi qua, hệ thống sẽ tự động đo thân nhiệt và nhắc đeo khẩu trang (nếu người đó không đeo), nếu vượt quá 37,5°C thì máy sẽ đưa ra cảnh báo. Công đo thân nhiệt tự động này được áp dụng tại các tòa nhà, khu vực cách ly cần kiểm soát người ra vào nhằm mục đích thay thế người kiểm tra thân nhiệt như hiện nay, giảm sự tiếp xúc trực tiếp giữa người với người, hạn chế nguy cơ nhiễm bệnh, mang lại tính an toàn cao.

- **Ngoài nước**

Nhiều nước đã triển khai hệ thống camera tích hợp AI để theo dõi việc giãn cách và đeo khẩu trang trên phương tiện công cộng. Không chỉ xe buýt, tàu điện ngầm, một số nước còn áp dụng camera AI trên cả taxi. AI được sử dụng để giám sát việc tuân thủ biện pháp hạn chế lây lan Covid-19. Công nghệ này có thể báo cáo hành động vi phạm giãn cách và đeo khẩu trang không đúng cách, nhờ phân tích video quay từ camera giám sát trên xe.

Với sự phát triển nhanh chóng của học sâu, đặc biệt là mạng nơ-ron 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. Một trong những ứng dụng của mạng nơ-ron 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. 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 nơon 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 để 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.



1.1 Sơ đồ thuật toán phát hiện khẩu trang

### 3. SỰ CẦN THIẾT CỦA ĐỀ TÀI

Do tình hình đại dịch COVID - 19 và với việc học sinh, sinh viên bắt đầu quay lại lớp học trực tiếp, việc kiểm tra và nhắc nhở các học sinh, sinh viên đeo khẩu trang là một việc cấp thiết. Cách duy nhất để kiểm tra và nhắc nhở việc đeo khẩu trang của học sinh, sinh viên khi đến trường là kiểm tra và nhắc nhở trực tiếp, có thể tang nguy cơ lây lan dịch bệnh. Từ những điều trên, nhóm đang nghiên cứu và áp dụng để đưa ra một

hệ thống nhận dạng và kiểm tra để phân biệt được người đeo khẩu trang và người không đeo khẩu trang. Đề tài của nhóm nghiên cứu đặt mục tiêu chính là giúp các trường học có thể kiểm tra và nhắc nhở học sinh, sinh viên về việc đeo khẩu trang một cách tự động hơn và thông minh, gần gũi hơn với các học sinh và sinh viên.

Đại dịch COVID-19 đã, đang và sẽ gây ra nhiều tổn thất nặng nề về kinh tế lẫn sức khỏe và tính mạng của người dân trên đất nước Việt Nam nói riêng và cả thế giới nói chung. Việc phòng tránh để tự bảo vệ bản thân và cho cộng đồng quan trọng hơn bao giờ hết, có thể xuất phát từ những việc đơn giản nhất như rửa tay thường xuyên, đeo khẩu trang nơi công cộng,... Tuy nhiên, việc này tuy đơn giản nhưng không phải ai cũng có thể nhớ và thực hiện thường xuyên.

#### **4. MỤC TIÊU CỦA ĐỀ TÀI**

Tổng quan: giúp tiết kiệm thời gian, nhân lực giảm thiểu rủi ro lây nhiễm do tiếp xúc gần khi phải kiểm tra. Cụ thể:

- Xây dựng một hệ thống phát hiện một người có đang đeo khẩu trang hay không.
- Xây dựng một ứng dụng chạy trên thiết bị di động, dễ sử dụng và thao tác.
- Khi phát hiện một người đang không đeo khẩu trang, hệ thống sẽ tự động nhắc nhở người đó đeo khẩu trang.

#### **5. NỘI DUNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU**

##### **• Nội dung nghiên cứu**

- Dựa trên yêu cầu thực tế, các công nghệ hiện nay có, tiến hành thiết kế và phát triển hệ thống.
- Thông qua khảo sát và góp ý của người dùng để chỉnh sửa, hoàn thiện hệ thống.

##### **• Phương pháp nghiên cứu**

- Tìm kiếm, tìm hiểu và thu thập các thông tin, tài liệu về công nghệ xử lý hình ảnh
- Phân tích: phân tích yêu cầu từ các thông tin đã thu thập, viết tài liệu phân tích.
- Tổng hợp các thông tin, tài liệu sau đó xây dựng và kiểm thử hệ thống.
- Thiết kế:
  - + Phân tích nhu cầu đối với ứng dụng, xây dựng kịch bản hệ thống.
  - + Tìm hiểu, xây dựng phần mềm kiểm tra và nhắc nhở đeo khẩu trang.
  - + Nghiên cứu và thiết kế mô hình thử nghiệm.
  - + Áp dụng và kiểm thử, sửa lỗi trên mô hình.

#### **6. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU**

##### **• Đối tượng nghiên cứu**

- Hệ thống nhận diện khuôn mặt

##### **• Phạm vi nghiên cứu**

- Nhận diện khuôn mặt
- Máy học



## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

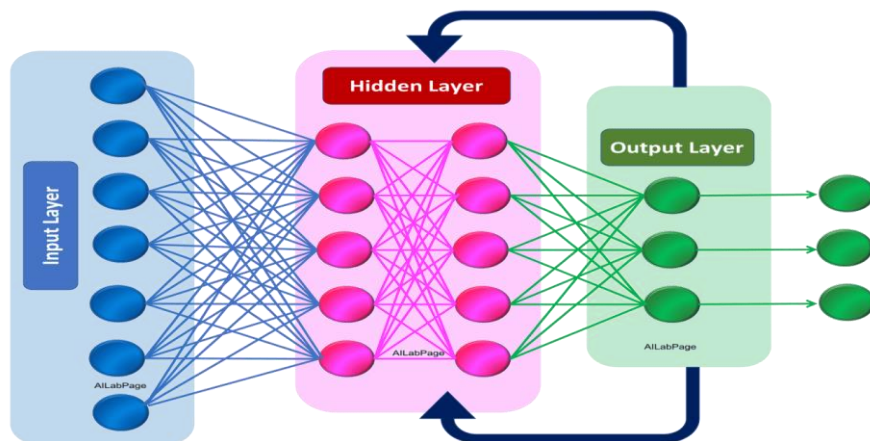
Để thiết kế và xây dựng được một hệ thống, trước hết, chúng ta cần tìm và hiểu một số cơ sở về lý thuyết. Trong chương này, chúng tôi sẽ tập trung một số nội dung về lý thuyết bao gồm các lý thuyết về thị giác máy tính và các công cụ cần để ứng dụng vào mô hình nhận diện khuôn mặt.

### 1. Nhận diện khuôn mặt

#### 1.1 Khái niệm

Học sâu (hay Deep Learning) là một chức năng của trí tuệ nhân tạo, bắt chước hoạt động của bộ não con người trong việc xử lý dữ liệu và tạo ra các mẫu để sử dụng cho việc ra quyết định.

Học sâu là một tập con của Học máy trong Trí tuệ nhân tạo, sử dụng một mạng lưới nơ ron thần kinh dạng lớp để học mà không bị giám sát từ dữ liệu không có cấu trúc. Học sâu không có nghĩa là học máy thêm kiến thức chuyên sâu, nó có nghĩa là máy sử dụng các lớp khác nhau để học hỏi từ dữ liệu. Độ sâu của mô hình biểu thị bằng số lớp trong mô hình.



#### 2.1 Mô phỏng mạng nơ ron nhân tạo của deep learning

Bên cạnh Học sâu, thị giác máy tính (hay Computer vision) lại là một lĩnh vực trong Trí tuệ nhân tạo và Khoa học máy tính nhằm giúp máy tính có được khả năng nhìn và hiểu như con người. Ở đây, thị giác máy tính sẽ mô phỏng thị giác con người qua ba giai đoạn nối tiếp như sau: mô phỏng mắt (thu nhận hình ảnh), mô phỏng vỏ não thị giác (hay là giai đoạn xử lý hình ảnh) và mô phỏng phần còn lại của bộ não (tức giai đoạn phân tích, trích xuất các thông tin từ các hình ảnh thu nhận và nhận diện hình ảnh).

Trong đề tài này, chúng tôi chủ yếu ứng dụng học sâu và thị giác máy tính để huấn luyện tạo ra mô hình nhận diện khuôn mặt của các đối tượng. Trong đó sẽ thực hiện các bước phát hiện khuôn mặt bằng CNN, sau đó trích xuất các đặc trưng trên khuôn mặt bằng mô hình Facenet và cuối cùng là phân lớp khuôn mặt dựa trên các đặc trưng đã trích xuất thông qua SVM. Quá trình này được mô phỏng như hình:



## 2.2 Quá trình nhận diện khuôn mặt

### 1.2 Các bước để nhận diện khuôn mặt

Nhận dạng khuôn mặt hoạt động theo ba bước: phát hiện, phân tích và nhận dạng.

#### Phát hiện:

Phát hiện là quá trình tìm khuôn mặt trong hình ảnh. Nhận dạng khuôn mặt được thị giác máy tính hỗ trợ có thể phát hiện và xác định khuôn mặt của cá nhân từ hình ảnh có một hoặc nhiều khuôn mặt người. Công nghệ này có thể phát hiện dữ liệu khuôn mặt ở cả nét mặt nhìn từ phía trước và từ phía bên.

#### Phân tích:

Sau đó, hệ thống nhận dạng khuôn mặt phân tích hình ảnh khuôn mặt. Hệ thống này ánh xạ và đọc hình dạng khuôn mặt cũng như biểu cảm trên khuôn mặt. Hệ thống xác định các điểm quan trọng trên khuôn mặt, những điểm này đóng vai trò là chìa khóa để phân biệt khuôn mặt với những vật thể khác. Công nghệ nhận dạng khuôn mặt thường tìm kiếm những đặc điểm sau:

- + Khoảng cách giữa hai mắt
- + Khoảng cách từ trán đến cằm
- + Khoảng cách giữa mũi và miệng
- + Độ sâu của hốc mắt
- + Hình dạng của gò má
- + Đường viền môi, tai và cằm

Sau đó, hệ thống sẽ chuyển đổi dữ liệu nhận dạng khuôn mặt thành một dãy số hoặc điểm được gọi là dấu khuôn mặt (faceprint). Mỗi người sẽ có một dấu khuôn mặt độc nhất, tương tự như dấu vân tay. Thông tin được nhận dạng khuôn mặt sử dụng cũng có thể được sử dụng ngược lại để tái cấu trúc khuôn mặt của một người thành phiên bản kỹ thuật số.

#### Nhận dạng:

Nhận dạng khuôn mặt có thể xác định một người bằng cách so sánh các khuôn mặt trong hai hoặc nhiều hình ảnh và đánh giá khả năng khớp khuôn mặt. Ví dụ: nhận dạng khuôn mặt có thể xác minh khuôn mặt có trong ảnh tự chụp bằng camera của di động khớp với khuôn mặt trong hình ảnh giấy tờ tùy thân do chính phủ cấp như giấy phép lái xe hoặc hộ chiếu, cũng như xác minh khuôn mặt có trong ảnh tự chụp không khớp với khuôn mặt trong tập hợp các khuôn mặt được chụp trước đó.



2.3 Kết quả nhận diện khuôn mặt

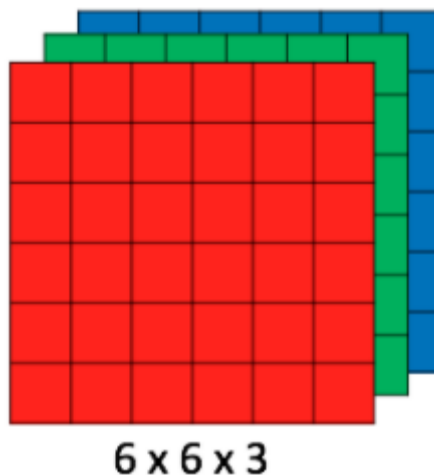
## 2. Thuật toán CNN

### 2.1 Khái niệm

CNN là viết tắt của **Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập)** là một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay. CNN được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh.

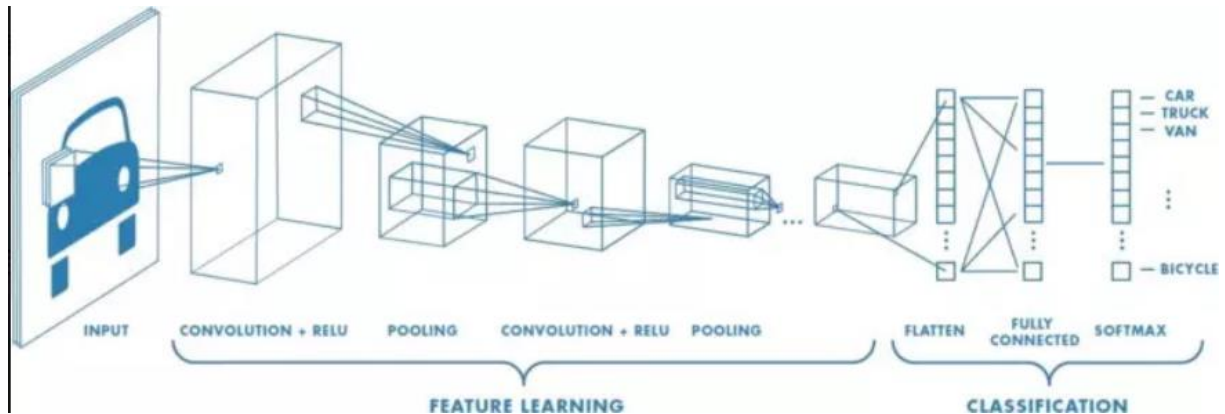
Trong mạng neural, mô hình mạng neural tích chập (CNN) là 1 trong những mô hình để nhận dạng và phân loại hình ảnh. Trong đó, xác định đối tượng và nhận dạng khuôn mặt là 1 trong số những lĩnh vực mà CNN được sử dụng rộng rãi.

CNN phân loại hình ảnh bằng cách lấy 1 hình ảnh đầu vào, xử lý và phân loại nó theo các hạng mục nhất định (Ví dụ: Chó, Mèo, Hổ, ...). Máy tính coi hình ảnh đầu vào là 1 mảng pixel và nó phụ thuộc vào độ phân giải của hình ảnh. Dựa trên độ phân giải hình ảnh, máy tính sẽ thấy  $H \times W \times D$  (H: Chiều cao, W: Chiều rộng, D: Độ dày). Ví dụ: Hình ảnh là mảng ma trận RGB  $6 \times 6 \times 3$  (3 ở đây là giá trị RGB).



2.4 Ma trận RGB  $6 \times 6 \times 3$

Về kỹ thuật, mô hình CNN để training và kiểm tra, mỗi hình ảnh đầu vào sẽ chuyển nó qua 1 loạt các lớp tích chập với các bộ lọc (Kernels), tổng hợp lại các lớp được kết nối đầy đủ (Full Connected) và áp dụng hàm Softmax để phân loại đối tượng có giá trị xác suất giữa 0 và 1. Hình dưới đây là toàn bộ luồng CNN để xử lý hình ảnh đầu vào và phân loại các đối tượng dựa trên giá trị.

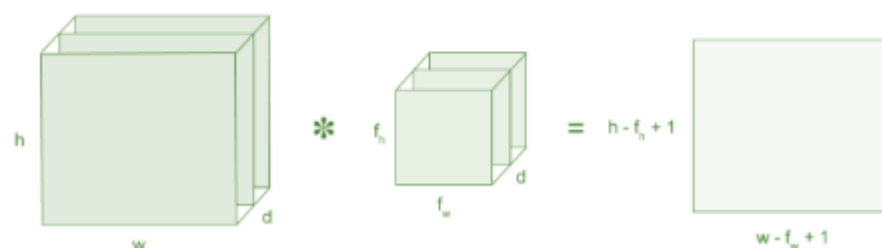


2.5 CNN xử lý đầu vào và phân loại đối tượng

## 2.2 Lớp tích chập – Convolution Layer

Tích chập là lớp đầu tiên để trích xuất các tính năng từ hình ảnh đầu vào. Tích chập duy trì mối quan hệ giữa các pixel bằng cách tìm hiểu các tính năng hình ảnh bằng cách sử dụng các ô vuông nhỏ của dữ liệu đầu vào. Nó là 1 phép toán có 2 đầu vào như ma trận hình ảnh và 1 bộ lọc hoặc hạt nhân.

- An image matrix (volume) of dimension  **$(h \times w \times d)$**
- A filter  **$(f_h \times f_w \times d)$**
- Outputs a volume dimension  **$(h - f_h + 1) \times (w - f_w + 1) \times 1$**



2.6 Lớp tích chập

## 2.3 Ứng dụng thực tế trong đề tài

Dùng CNN để huấn luyện mô hình sử dụng dữ liệu đã có ở 1.1.2:

### a. Ưu điểm của CNN:

- Độ chính xác cao

### 3. Open CV

#### 3.1 Khái niệm

OpenCV là tên viết tắt của **open source computer vision library** – có thể được hiểu là một thư viện nguồn mở cho máy tính. Cụ thể hơn OpenCV là kho lưu trữ các mã nguồn mở được dùng để xử lý hình ảnh, phát triển các ứng dụng đồ họa trong thời gian thực. OpenCV cho phép cải thiện tốc độ của CPU khi thực hiện các hoạt động real time. Nó còn cung cấp một số lượng lớn các mã xử lý phục vụ cho quy trình của thị giác máy tính hay các learning machine khác.

Thư viện OpenCV được phát hành với giấy phép BDS. Do đó các dịch vụ nó cung cấp là hoàn toàn miễn phí và được hạn chế tối đa các rào cản thông thường. Cụ thể, bạn được phép sử dụng phần mềm này cho cả hoạt động thương mại lẫn phi thương mại. OpenCV sở hữu giao diện thiên thiện với mọi loại ngôn ngữ lập trình, ví dụ như C++, C, Python hay Java... Ngoài ra, nó cũng dễ dàng tương thích với các hệ điều hành khác nhau, bao gồm từ Windows, Linux, Mac OS, iOS cho đến cả Android.

Kể từ lần đầu xuất hiện từ năm 1999, giờ đây OpenCV đã sở hữu đội ngũ người dùng hùng hậu, con số ước tính có thể lên tới 47.000 người. Tất cả là nhờ những ưu điểm vượt trội của OpenCV.

Các modules được dùng trong OpenCV:

- **Core functionality (core):** Module này sở hữu cơ chế rất nhỏ gọn. Nó được dùng để định hình các cấu trúc của cơ sở dữ liệu cơ bản, bao gồm cả những mảng đa chiều. Ngoài ra nó còn xác định các chức năng của những module đi kèm khác nữa.
- **Image Processing (imgproc):** Đây là module được dùng cho quá trình xử lý hình ảnh. Nó cho phép người dùng thực hiện các hoạt động như lọc hình ảnh tuyến tính và phi tuyến, thực hiện phép biến hình, thay đổi không gian màu, xây dựng biểu đồ và rất nhiều thao tác khác liên quan.
- **Video Analysis (video):** Giống như tên gọi của nó, module này cho phép phân tích các video. Kết quả được trả về bao gồm các ước tính chuyển động, thực hiện tách nền và các phép toán theo dõi vật thể.
- **Camera Calibration and 3D Reconstruction (calib3d):** Module này cung cấp các thuật toán hình học đa chiều cơ bản và hiệu chuẩn máy ảnh single và stereo. Ngoài ra nó còn đưa ra các dự đoán kiểu dáng của đối tượng và sử dụng thuật toán thư tín âm thanh nội cùng các yếu tố tái tạo 3D.
- **2D Features Framework (features2d):** Module này giúp phát hiện các tính năng nổi trội của bộ nhận diện, bộ truy xuất thông số và thông số đối chọi.
- Ngoài ra còn có rất nhiều module khác với đa dạng tính năng, ví dụ như: FLANN, Google test wrapper...

#### 3.2 Ứng dụng thực tế trong đề tài

Dùng OpenCV để nhận dạng được khuôn mặt:

- Người có đeo khẩu trang
- Người không đeo khẩu trang

##### a) Ưu điểm của OpenCV:

- Dễ tiếp cận cho người mới bắt đầu.
- OpenCV hoạt động tốt, có các thuật toán rộng.

- Là mã nguồn mở và miễn phí, được sử dụng rộng rãi, có cộng đồng hỗ trợ lớn.

#### **b. Nhược điểm**

- Tốn nhiều thời gian cho việc huấn luyện.
- Các thư viện chỉ hỗ trợ một tính năng cụ thể và duy nhất cho một công việc của người dùng. Đây là điểm hạn chế của OpenCV khi các công cụ hiện nay luôn ưu tiên cho việc đa dạng hóa các tính năng, giúp người thuận tiện hơn trong việc phát triển.

### **4. Tensorflow**

#### **4.1 Khái niệm**

Tensorflow cơ bản là một thư viện dạng nguồn mở được sử dụng rất phổ biến trong lĩnh vực học máy - Machine Learning giúp gia tăng tốc độ nhanh chóng và dễ dàng hơn. Một khi trí tuệ nhân tạo được đưa vào trong giảng dạy tại các trường đại học thì sẽ trở nên nổi tiếng và được dùng phổ biến trong giáo dục. Tensorflow được tạo ra và phát triển bởi đội ngũ chuyên viên của Google mà cụ thể là Google Brain. Nó được tạo ra với mục đích chính là sử dụng cho các nhu cầu nghiên cứu và áp dụng trong sản xuất sao cho hiệu quả nhất.

#### **4.2 Nguyên lý hoạt động:**

TensorFlow sẽ giúp các lập trình viên tạo ra một biểu đồ hướng xử lý dữ liệu ban đầu, và quy trình xử lý các thuật toán là một chuỗi các nốt (node) có trên biểu đồ đó. Ngoài ra, 1 đường liên kết giữa các nốt sẽ tương đương với 1 mảng dữ liệu đa chiều (1 tensor).

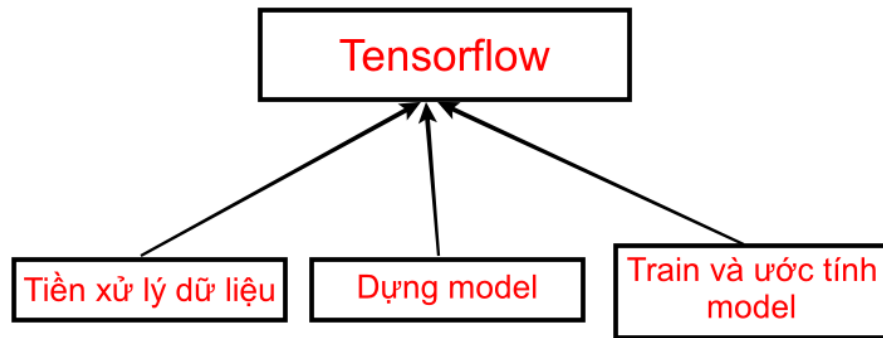
TensorFlow hỗ trợ cho ngôn ngữ Python bởi vì đây là ngôn ngữ đơn giản, dễ để học và dễ ứng dụng vào công việc. Python ở đây đóng góp vai trò điều phối các luồng công việc và kết nối chúng lại với nhau. Còn đối với các phép toán có độ phức tạp cao sẽ được viết dưới dạng ngôn ngữ C++ có hiệu suất cao để khắc phục hạn chế tốc độ của Python.

#### **4.3 Kiến trúc của Tensorflow**

Giúp tạo ra một biểu đồ hướng xử lý dữ liệu ban đầu, và quy trình xử lý các thuật toán là một chuỗi các nốt (node) có trên biểu đồ

Hỗ trợ ngôn ngữ Python, nhưng đối với các phép toán có mức độ phức tạp cao sẽ được viết dưới dạng C++ để cải thiện tốc độ

Các ứng dụng của Tensorflow có thể chạy hầu hết trên các đối tượng thông dụng: máy local, cluster trong các đám mây, thiết bị di động IOS và Android, CPU hay GPU



2.7 Kiến trúc Tensorflow

#### 4.4 Ứng dụng thực tế trong đề tài

Dùng Tensorflow giúp xử lý thuật toán

##### a. Ưu điểm của Tensorflow:

- Cho phép bạn đánh giá cũng như sửa đổi các operation của biểu đồ theo một cách minh bạch và riêng biệt với chế độ eager execution. Vì vậy, thay vì xây dựng biểu đồ dưới đối tượng độc lập vốn rất mơ hồ thì nên đưa ra những đánh giá chung tổng thể.
- Cung cấp thêm những tiện ích bổ sung khác để các lập trình viên có thể debug hoặc suy xét các ứng dụng Tensorflow.
- Chỉ cần tập trung thực hiện các phần logic tổng thể của ứng dụng, những phần còn lại như triển khai thuật toán, chuyển output của chức năng này sang input của chức năng kia sẽ do Tensorflow đối phó và tìm biện pháp xử lý.

##### b. Nhược điểm

- Khó tìm ra lỗi và khó gỡ lỗi
- Thiếu vòng lặp tự động trung
- Tốc độ tính toán

## 5. Keras

### 5.1 Khái niệm

Keras là một open source cho Neural Network được viết bởi ngôn ngữ Python. Nó là một thư viện 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:

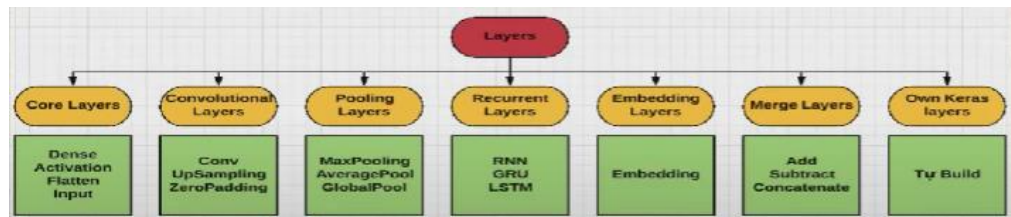
- Dễ sử dụng, dùng đơn giản hơn Tensor, xây dựng model nhanh.
- Chạy được trên cả CPU và GPU.

- 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.

## 5.2 Cấu trúc của Keras

Gồm 3 phần chính:

- Function dùng để xây dựng bộ xương cho model: Keras hỗ trợ 2 cách dựng models là Sequential model và Function API
- Function ở trong bộ xương của model:



- Function dùng để tiền xử lý dữ liệu:
  - + Sequence Preprocessing: Tiền xử lý chuỗi
  - + Text Preprocessing: Tiền xử lý văn bản
  - + Image Preprocessing: Tiền xử lý ảnh
  - + Loss\_function, Metrics, Optimizers, Callbacks

## 5.3 Ứng dụng thực tế trong đề tài

Hỗ trợ xây dựng CNN

### a. Ưu điểm của Tensorflow:

Graphs

- TensorFlow có hình ảnh hóa đồ thị tính toán tốt hơn. Những thứ vốn có khi so sánh với các thư viện khác như Torch và Theano.

Library management

- Google ủng hộ nó. Và có lợi thế về hiệu suất liên mạch, cập nhật nhanh chóng và các bản phát hành mới thường xuyên với các tính năng mới.

Debugging

- Nó giúp chúng tôi thực thi phần con của biểu đồ, phần này giúp nó có ưu thế hơn vì chúng tôi có thể giới thiệu và truy xuất dữ liệu rời rạc

Scalability

- Các thư viện được triển khai trên một máy phần cứng, là một thiết bị di động cho máy tính có thiết lập phức tạp.



Pipelining:

- TensorFlow được thiết kế để sử dụng nhiều phần mềm phụ trợ khác nhau (GPU, ASIC), v.v. và cũng có tính song song cao.
- Nó có một cách tiếp cận độc đáo cho phép theo dõi tiến độ đào tạo của các mô hình của chúng tôi và theo dõi một số chỉ số.

TensorFlow có sự hỗ trợ cộng đồng tuyệt vời.

Hiệu suất của nó cao và phù hợp với những gì tốt nhất trong ngành.

## **b. Nhược điểm**

- Thiếu vòng lặp tượng trưng
- Không hỗ trợ cho các cửa sổ
- Kiểm tra Benchmark
- Không hỗ trợ GPU cho Nvidia và chỉ hỗ trợ ngôn ngữ
- Tốc độ tính toán
- Không hỗ trợ OpenCL

## **6. MobileNet V2**

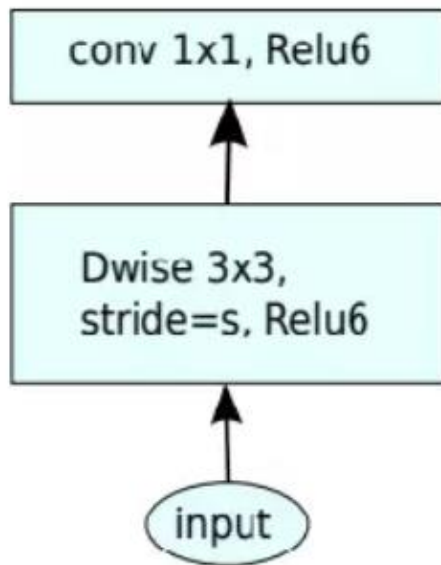
### **6.1 Khái niệm**

MobileNet là một trong những kiến trúc được ưa chuộng và sử dụng phổ biến bởi độ chính xác và hiệu năng tính toán. Qua bài này mình đã phân tích cho các bạn đặc điểm kiến trúc của một block layer MobileNet. Điểm mấu chốt giúp cho các mô hình MobileNet giảm thiểu số lượng tính toán đó là áp dụng tích chập tách biệt chiều sâu. Đồng thời qua thời gian, nhóm tác giả MobileNet đã lồng ghép các ưu điểm từ những kiến trúc CNN khác vào mô hình của mình để ngày càng cải thiện hơn về độ chính xác và hiệu năng.

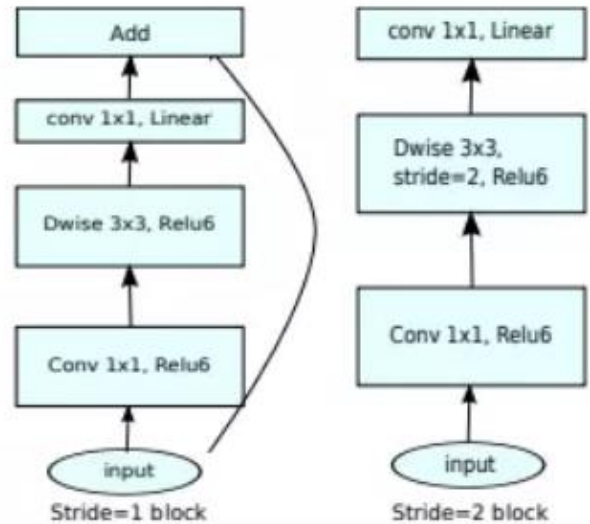
MobileNet v2 tiếp tục sử dụng Depthwise Separable Convolutions, ngoài ra còn đề xuất thêm:

- **Linear bottlenecks**
- **Inverted Residual Block** (shortcut connections giữa các bottlenecks)

### **6.2 Sự khác nhau giữa v1 và v2**



MobileNet



MobileNet V2

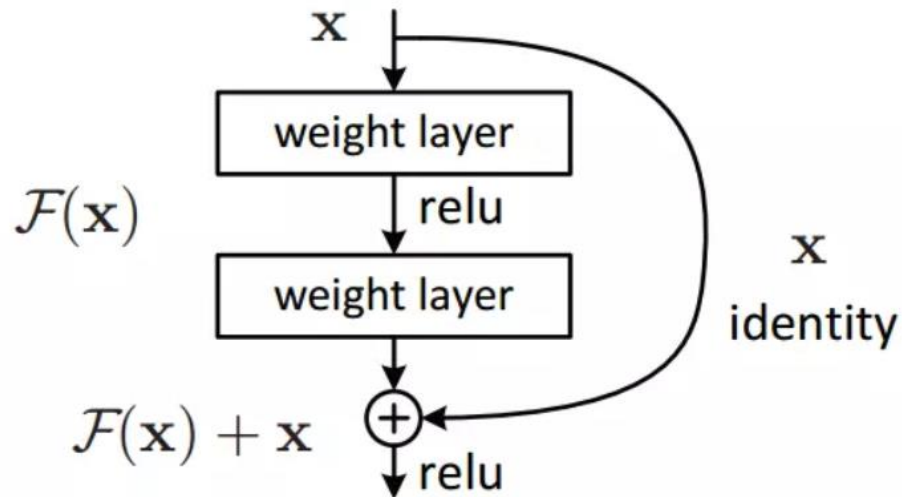
## 2.8 Khác nhau giữa mobileNet V1 và V2

Phân tích sự khác biệt trong bức ảnh trên:

- **MobileNet** gồm sử dụng 1 loại blocks gồm 2 phần, Deepwise và Pointwise.
- **MobileNet v2** sử dụng 2 loại blocks, bao gồm các blocks: residual block với stride = 1 và block với stride = 2 phục vụ downsizing.

Có 3 phần đối với mỗi block:

- Layer đầu là  $1 \times 1$  convolution với ReLU6.
- Layer thứ hai, như cũ, là depthwise convolution.
- Layer thứ 3 tiếp tục là  $1 \times 1$  convolution nhưng không có activation function. Linear được sử dụng thay vì ReLU như bình thường.



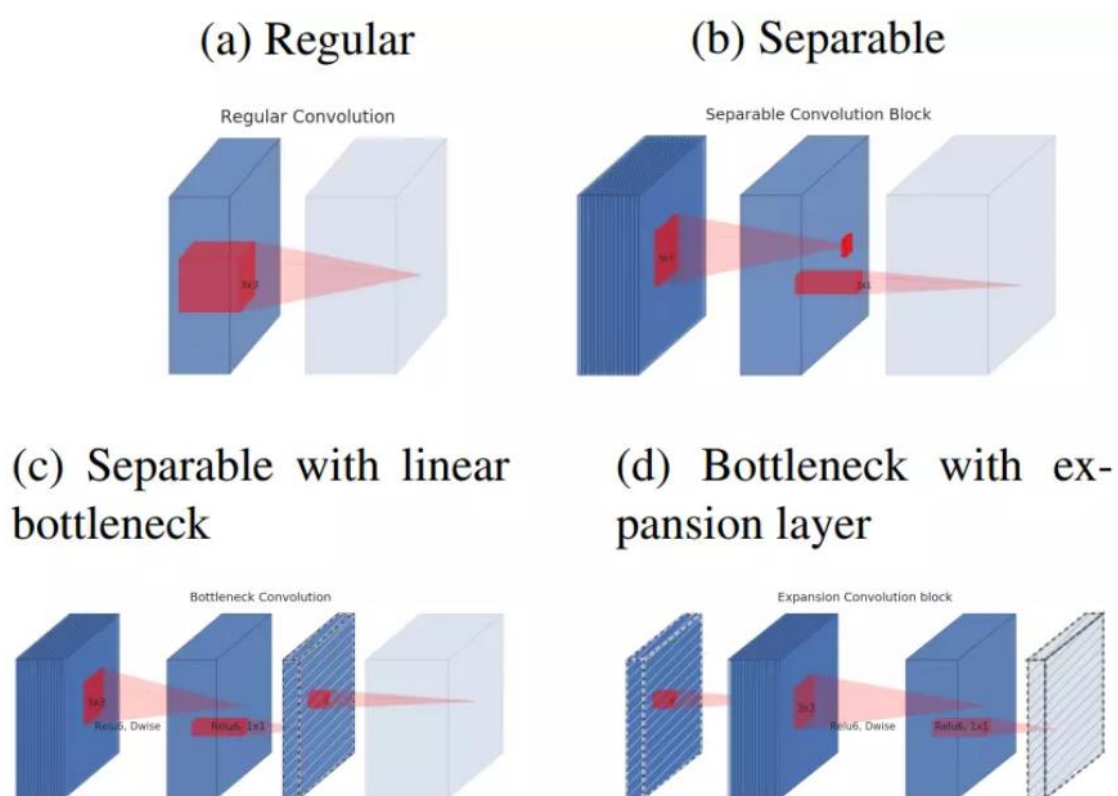
### 2.9 Residual Block

Kết nối tắt ở MobileNetV2 được điều chỉnh sao cho số input và output channels của mỗi block residual được thắt hẹp lại. Chính vì thế nó được gọi là các bottleneck layers.

Residual block của MobileNet v2 ngược lại so với các kiến trúc residual truyền thống, vì kiến trúc residual truyền thống có số lượng kênh ở input và output của một block lớn hơn so với các layer trung gian. Chính vì vậy nó còn được gọi là Inverted residual block.

Paper cho rằng các layer trung gian trong một block sẽ làm nhiệm vụ biến đổi phi tuyến nên cần dày hơn để tạo ra nhiều phép biến đổi hơn. Kết nối tắt giữa các block được thực hiện trên những bottleneck input và output chứ không thực hiện trên các layer trung gian. Do đó các layer bottleneck input và output chỉ cần ghi nhận kết quả và không cần thực hiện biến đổi phi tuyến (linear f).

Ở giữa các layer trong một block inverted residual block chúng ta cũng sử dụng những biến đổi tích chập tách biệt chiều sâu để giảm thiểu số lượng tham số của mô hình. Đây cũng chính là bí quyết giúp họ các model MobileNet có kích thước giảm nhẹ.



### 2.10 Mô tả các layer trung gian

MobileNet v2 có 2 đóng góp vô cùng lớn:

- SSDLite+MobileNetV2 cho bài toán object detection.

Model	Params	Multiply-Adds	mAP	Mobile CPU
MobileNetV1 + SSDLite	5.1M	1.3B	22.2%	270ms
MobileNetV2 + SSDLite	4.3M	0.8B	22.1%	200ms

### 2.1 Bài toán OB

- DeepLabV3 + MobileNetV2 cho bài toán semantic segmentation.

Model	Params	Multiply-Adds	mIOU
MobileNetV1 + DeepLabV3	11.15M	14.25B	75.29%
MobileNetV2 + DeepLabV3	2.11M	2.75B	75.32%

### 2.2 Bài toán SS

## 7. Python

### 7.1 Khái niệm

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ, được tạo ra bởi Guido van Rossum. Python có cú pháp đơn giản và dễ sử dụng, do đó nó trở thành một trong những ngôn ngữ lập trình nhập môn tốt nhất cho người mới bắt đầu.

Python có cấu trúc dữ liệu cao cấp mạnh mẽ và cách tiếp cận đơn giản nhưng hiệu quả đối với lập trình hướng đối tượng. Cú pháp lệnh của Python là điểm cộng vô cùng lớn vì sự rõ ràng, dễ hiểu và linh động, làm cho nó nhanh chóng trở thành một ngôn ngữ lý tưởng để viết script và phát triển ứng dụng trong nhiều lĩnh vực, ở hầu hết các nền tảng nghiên cứu.

Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.

### 7.2 Tính năng chính của Python

Ngôn ngữ lập trình đơn giản, dễ học:

Python có cú pháp rất đơn giản, rõ ràng. Nó dễ đọc và viết hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như C++, Java, C#. Python làm cho việc lập trình trở nên thú vị, cho phép bạn tập trung vào những giải pháp chứ không phải cú pháp.

Miễn phí, mã nguồn mở:

Bạn có thể tự do sử dụng và phân phối Python, thậm chí là dùng nó cho mục đích thương mại. Vì Python là ứng dụng mã nguồn mở, bạn không những có thể sử dụng các phần mềm, chương trình được viết trong Python mà còn có thể thay đổi mã nguồn của nó. Python có một cộng đồng rộng lớn, không ngừng cải thiện nó mỗi lần cập nhật.

Khả năng di chuyển:

Các chương trình Python có thể di chuyển từ nền tảng này sang nền tảng khác và chạy nó mà không có bất kỳ thay đổi nào. Nó chạy liền mạch trên hầu hết tất cả các nền tảng như Windows, macOS, Linux.

Khả năng mở rộng và có thể nhúng:

Giả sử một ứng dụng đòi hỏi sự phức tạp rất lớn, bạn có thể dễ dàng kết hợp các phần code bằng C, C++ và những ngôn ngữ khác (có thể gọi được từ C) vào code Python. Điều này sẽ cung cấp cho ứng dụng của bạn những tính năng tốt hơn cũng như khả năng scripting mà những ngôn ngữ lập trình khác khó có thể làm được.

Ngôn ngữ thông dịch cấp cao:

Không giống như C/C++, với Python, bạn không phải lo lắng những nhiệm vụ khó khăn như quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa,... Khi chạy code Python,

nó sẽ tự động chuyển đổi code sang ngôn ngữ máy tính có thể hiểu. Bạn không cần lo lắng về bất kỳ hoạt động ở cấp thấp nào.

Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến:

Python có một số lượng lớn thư viện tiêu chuẩn giúp cho công việc lập trình của bạn trở nên dễ thở hơn rất nhiều, đơn giản vì không phải tự viết tất cả code.

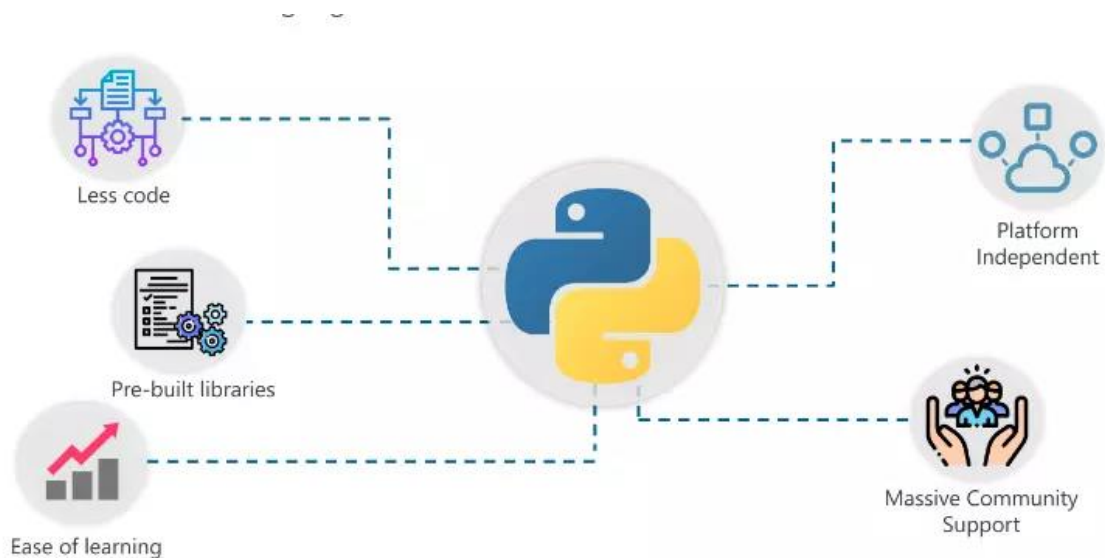
Ví dụ: Bạn cần kết nối cơ sở dữ liệu MySQL trên Web server? Bạn có thể nhập thư viện MySQLdb và sử dụng nó. Những thư viện này được kiểm tra kỹ lưỡng và được sử dụng bởi hàng trăm người. Vì vậy, bạn có thể chắc chắn rằng nó sẽ không làm hỏng code hay ứng dụng của mình.

Hướng đối tượng:

Mọi thứ trong Python đều là hướng đối tượng. Lập trình hướng đối tượng (OOP) giúp giải quyết những vấn đề phức tạp một cách trực quan. Với OOP, bạn có thể phân chia những vấn đề phức tạp thành những tập nhỏ hơn bằng cách tạo ra các đối tượng.

### 7.3 Lý do chọn Python cho đề tài

- Tính đơn giản và nhất quán
- Cho phép truy cập vào các thư viện và framework tuyệt vời cho AI và học máy (ML)
- Tính linh hoạt
- Độc lập nền tảng
- Và cộng đồng rộng lớn.



### 2.11 Lợi ích của Python trong một dự án AI

## CHƯƠNG 3: MÔ TẢ HỆ THỐNG

Trong chương này, chúng tôi sẽ trình bày tổng quan về hệ thống, mô tả cách hoạt động của ứng dụng, các yêu cầu chức năng và các yêu cầu phi chức năng.

### 1. TỔNG QUAN VỀ HỆ THỐNG

Ứng dụng gồm 2 phần chính: mô hình nhận diện khuôn mặt có đeo khẩu trang hay không và kiểm tra đưa ra kết quả ở màn hình.

Đầu tiên, về mô hình nhận diện khuôn mặt, chúng tôi thực hiện huấn luyện để xây dựng một mô hình nhận diện khuôn mặt theo dataset có sẵn. Với đầu vào là tập ảnh chứa khuôn mặt của những người có và không đeo khẩu trang được đưa vào và huấn luyện với sự hỗ trợ của thư viện Kera/Tensorflow. Mô hình sau khi huấn luyện thành công sẽ được lưu lại và dùng cho việc điểm danh trong tương lai.

Phần đưa ra cảnh báo đeo khẩu trang sẽ được thực hiện sau khi đã xác định được khuôn mặt có đeo khẩu trang hay chưa. Khi đã nhận diện được khuôn mặt trong vùng tọa độ được vẽ khung trên frame hiện ảnh từ phía camera (camera điện thoại).

Hệ thống được chạy trên nền tảng web. Sử dụng như một ứng dụng web sau khi tải về từ server web, chương trình được thực thi trên trình duyệt Chrome của thiết bị di động.

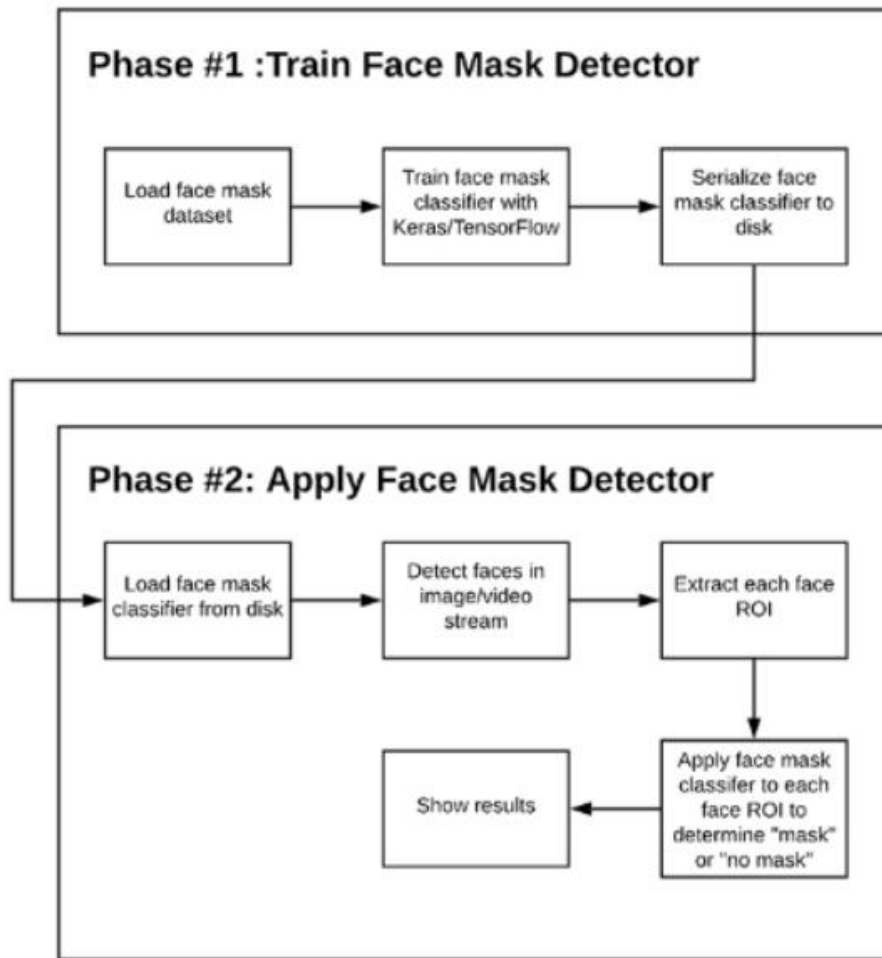
### 2. MÔ TẢ ỨNG DỤNG

Ứng dụng được chạy website và có thể sử dụng trên điện thoại di động không cần các thiết bị lắp ráp, camera dễ dàng xây dựng một thiết bị di động. Khi người dùng bật camera hệ thống sẽ tự động xác định khuôn mặt và đưa ra thông báo là có đeo khẩu trang hay không.

Ứng dụng nhận diện này chỉ sử dụng thông tin từ các camera thường và không cần đến các cảm biến phụ (như hồng ngoại hoặc cảm biến chiều sâu). Đây là yếu tố quan trọng giúp giá thành sản phẩm không cao, có thể dễ dàng tích hợp vào các hệ thống camera đã có sẵn. Sản phẩm có thể áp dụng rộng rãi vào các hệ thống chấm công, hệ thống xác thực cho tổ chức và doanh nghiệp.

Ứng dụng chia thành 2 giai đoạn chính:

- Về phần đào tạo:
  - + Bước 1: Tải, tạo và load datasheet
  - + Bước 2: Bắt đầu huấn luyện các lớp với sự hỗ trợ của Keras/Tensorflow
  - + Bước 3: Xuất Model
- Về phần kiểm tra:
  - + Bước 1: Mở livestream video (camera điện thoại)
  - + Bước 2: Load Model
  - + Bước 3: Nhận diện khuôn mặt qua live stream video
  - + Bước 4: Phân loại (khuôn mặt đang có/không đeo khẩu trang)
  - + Bước 5: Show kết quả, vẽ kết quả lên màn hình



3.1 Sơ đồ khối hệ thống kiểm tra & nhắc nhở đeo khẩu trang

### 3. NỘI DUNG NGHIÊN CỨU

#### 3.1 Bài toán phát hiện khẩu trang

Phát hiện đối tượng là để trả lời câu hỏi: “Đối tượng cần tìm có ở trong ảnh hay video hay không?” và “Nếu có thì nằm ở vị trí nào?” Trong bài toán phát hiện người đeo khẩu trang cần trả lời câu hỏi người trong ảnh/video có hay không đeo khẩu trang? Bài toán phân loại này gồm hai bước là xây dựng mô hình và vận hành mô hình. Cho sẵn một tập dữ liệu các khuôn mặt được gán nhãn là không đeo khẩu trang hay có đeo khẩu trang gọi tắt là lớp không khẩu trang và lớp có khẩu trang. Cần một phương pháp huấn luyện để xây dựng một mô hình phân lớp từ tập dữ liệu mẫu đó, sau đó dùng mô hình này dự đoán lớp của những khuôn mặt mới chưa biết nhãn.

#### 3.2 Tiền xử lý dữ liệu

Cơ sở dữ liệu hình ảnh thu thập tại Real World Masked Face Dataset (RMFD) chứa 5.000 khuôn mặt đeo khẩu trang của 525 người và 90.000 khuôn mặt bình thường, sau đó tiến hành tiền xử lý để đưa vào mô hình huấn luyện [11]. Các bước xử lý trước bao gồm thay đổi kích thước thành  $224 \times 224$  pixel, chuyển đổi sang định dạng mảng, chuyển kênh màu BGR sang RGB và chia tỷ lệ cường độ pixel trong hình ảnh đầu vào thành phạm vi  $[-1, 1]$ . Sau đó sử dụng scikit-learn One-Hot-Encoding để tạo nhãn lớp cho mỗi hình ảnh. Trong chiến lược này, mỗi véc-tơ giá trị nhãn đầu ra được chuyển đổi thành dạng mới, trong đó chỉ có 1 đầu ra bằng “1” ứng với mã phân loại của véc-tơ đầu



vào tương ứng còn các đầu ra khác đều bằng “0”. Tập hình ảnh này gọi là tập huấn luyện. Trong nghiên cứu này tác giả sử dụng 686 hình ảnh khuôn mặt không đeo khẩu trang và 3.905 hình ảnh đeo khẩu trang làm dữ liệu huấn luyện mô hình.



3.2 (a) Mặt có khẩu trang và (b) mặt không có khẩu trang

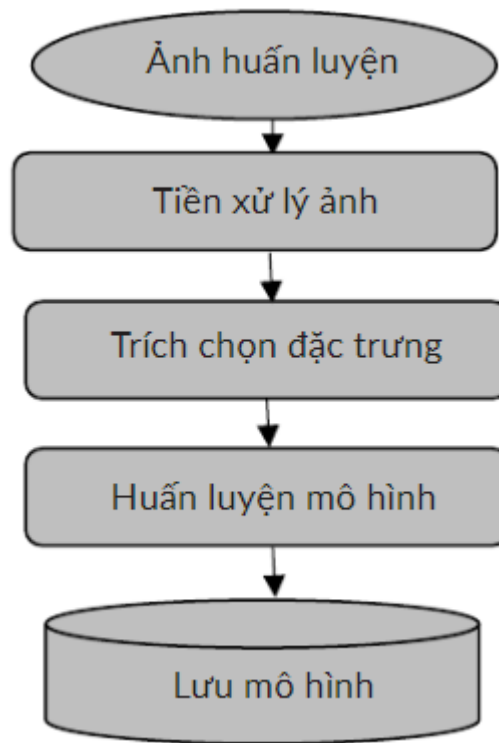


3.3 (a),(b),(c),(d) Minh họa các ảnh trong bộ dữ liệu phát hiện khẩu trang

### 3.3 Huấn luyện mô hình

CNN (Mạng nơron tích chập) có nhiều phiên bản của mạng được đào tạo trước và kiến trúc tốt như AlexNet, ResNet, Inception, LeNet, MobileNet,... Trong trường hợp này sử dụng MobileNetV2 là mô hình hướng di động, nhẹ và hiệu quả do số lượng tham số ít. MobileNetV2 xây dựng dựa trên các ý tưởng từ MobileNetV1 [10], sử dụng tích chập có thể phân tách theo chiều sâu với các đặc trưng mới cho kiến trúc. Trong bước này, chia dữ liệu thành tập huấn luyện chứa các hình ảnh mà mô hình CNN sẽ được huấn luyện và tập kiểm tra với các hình ảnh mà mô hình sẽ kiểm tra. Cụ thể lấy split\_size = 0.8, có nghĩa là 80% tổng số hình ảnh cho huấn luyện và 20% còn lại của hình ảnh cho kiểm tra một cách ngẫu nhiên. Sau đó, xây dựng mô hình CNN với các lớp khác nhau như AveragePooling2D với trọng số  $7 \times 7$ , Flatten, Dropout và Dense. Trong lớp

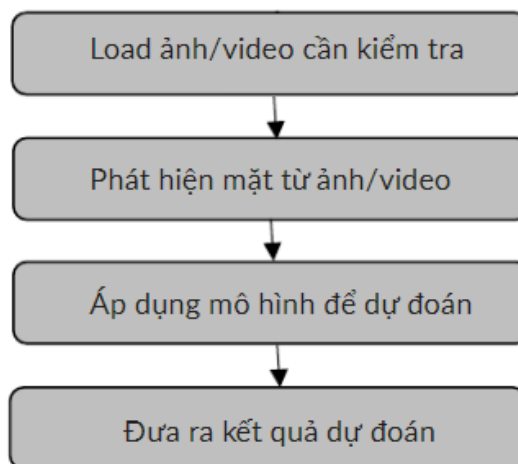
Dense cuối cùng, sử dụng hàm softmax để xuất ra một véc - tơ thể hiện xác suất của mỗi lớp.



3.4 Huấn luyện mô hình

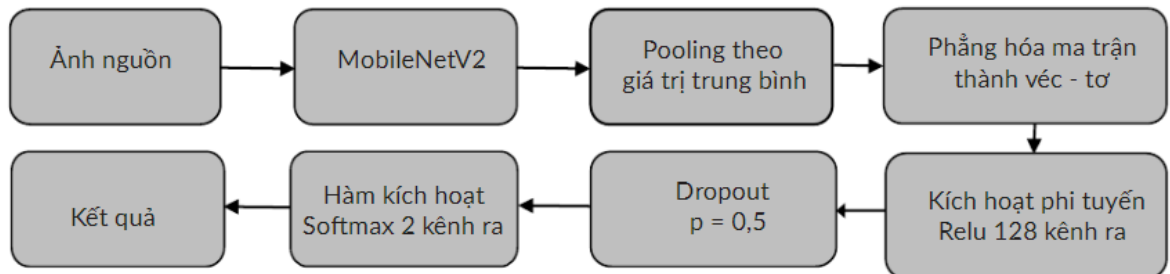
### 3.4 Vận hành mô hình

Tải hình ảnh, video đầu vào sau đó phát hiện khuôn mặt trong ảnh. Áp dụng bộ phát hiện khẩu trang để phân loại khuôn mặt đeo khẩu trang (with mask) hoặc là không đeo khẩu trang (without\_mask). Tiền xử lý được thực hiện là chuẩn hóa kích thước và hoán đổi kênh màu. Để giảm nhiễu cho ảnh do chiếu sáng cần chuyển ảnh sang dạng ảnh blob thông qua hàm blobFromImage của OpenCV.



### 3.5 Phát hiện khẩu trang trên ảnh

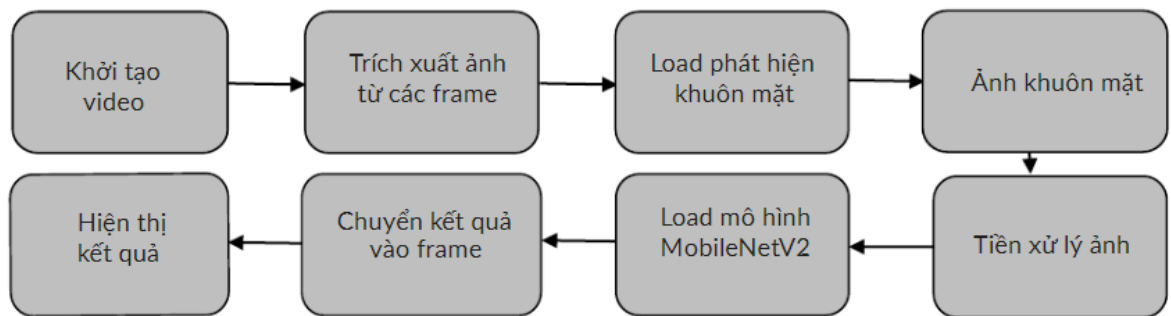
Với mỗi hình ảnh trong bộ kiểm tra, cần tìm chỉ mục của nhãn với xác suất dự đoán lớn nhất tương ứng, đánh giá và lưu mô hình. Sơ đồ phát hiện khẩu trang trên ảnh như trên hình.



3.6 Sơ đồ phát hiện khẩu trang trên ảnh

### 3.6 Phát hiện khẩu trang ở thời gian thực

Đối với vấn đề xác định người trong video/webcam có đeo khẩu trang hay không cần xác định các khuôn mặt trong webcam và phân loại khuôn mặt đeo khẩu trang. Đầu tiên, lặp qua các frame từ video và thay đổi kích thước để có chiều rộng tối đa 400 pixel, phát hiện khuôn mặt trong frame và xác định xem họ có đang đeo khẩu trang hay không? Quy trình chi tiết được thể hiện trên hình



3.7 Sơ đồ phát hiện khẩu trang trên webcam

## CHƯƠNG 4: XÂY DỰNG HỆ THỐNG NHẬN BIẾT KHUÔN MẶT ĐEO KHẨU TRANG

Trong chương này, chúng tôi sẽ trình bày các bước thiết kế và xây dựng nên hệ thống. Chủ yếu tập trung vào cách xây dựng mô hình huấn luyện nhận diện khuôn mặt, các xử lý nhiệt độ bằng camera nhiệt, sử dụng phần cứng và cuối cùng là cơ sở dữ liệu cho hệ thống.

### 1. KẾT QUẢ NGHIÊN CỨU

Dữ liệu thực hiện huấn luyện được lấy tại Real World Masked Face Dataset với nhiều ánh sáng, tư thế, sự che khuất khác nhau, một số khuôn mặt được che bằng tay hoặc các vật thể khác thay vì khẩu trang thực. Dữ liệu bao gồm các mặt có khẩu trang, mặt không có khẩu trang, mặt có và không có khẩu trang trong một hình ảnh và hình ảnh nhằm là khẩu trang nhưng thực tế thì không có. Để đánh giá tập trọng số cần xác định lỗi cho cả huấn luyện (loss) và kiểm tra (val\_loss) ta sử dụng hàm Cross Entropy là nhị phân chéo binary\_cross-entropy. Cụ thể, tính toán loss của mỗi trường hợp bằng cách tính giá trị trung bình như sau:

$$loss(y, \hat{y}) = - \left( \frac{1}{n} \cdot \sum_{i=1}^n y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i) \right) \quad (1)$$

Với  $n$  là số lượng giá trị vô hướng trong đầu ra của mô hình, hàm loss trả về một số thực không âm thể hiện sự chênh lệch giữa hai đại lượng  $\hat{y}$  là xác suất nhận được dự đoán và  $y$  là xác suất của nhãn đúng. Sau đó sử dụng thuật toán gradient descent “adam” (Adaptive Moment Estimator) để tối ưu [13]. Hơn nữa, để kiểm định hiệu năng của mô hình phân loại, cần tính toán tỷ lệ chính xác trung bình trên tất cả các dự đoán sử dụng thang đo ma trận nhầm lẫn như sau:

	Dự đoán là Positive	Dự đoán là Negative
Thực tế là Positive	TP	FN
Thực tế là Negative	FP	TN

4.1 Bảng ma trận nhầm lẫn

Trong đó: Các hàng của ma trận là nhãn lớp thực tế, các cột của ma trận là nhãn lớp dự đoán.

- TN: Số lượng khuôn mặt không đeo khẩu trang được phân loại chính xác
- FN: Số lượng khuôn mặt đeo khẩu trang bị phân loại nhầm là khuôn mặt không đeo khẩu trang.
- TP: Số lượng khuôn mặt đeo khẩu trang được phân loại chính xác.
- FP: Số lượng khuôn mặt không đeo khẩu trang bị phân loại nhầm là khuôn mặt đeo khẩu trang. Từ đó, độ chính xác của mô hình được tính theo công thức sau:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2)$$

Đây là tỉ lệ của tất cả trường hợp phân loại đúng (không phân biệt negative/positive) trên toàn bộ trường hợp trong mẫu kiểm tra.

Một độ đo cũng thường được dùng để đánh giá mô hình phân lớp đó là F-measure hay F-core được tính dựa trên 2 độ đo khác là precision và recall, và được tính như sau:

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$F_{score} = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2TP}{TP + FN + FP} \quad (5)$$

## 2. XÂY DỰNG MÔ HÌNH

### 2.1 Các thư viện cần sử dụng

Thư viện	Công dụng
Tensorflow	Hỗ trợ các phép toán trong học máy, giúp tiếp cận bài toán 1 cách đơn giản, nhanh chóng, tiện lợi
Matplotlib	Là thư viện dùng để vẽ biểu đồ
Numpy	Là thư viện dùng để tính toán số học

4.2 Bảng các thư viện cần sử dụng

### 2.2 Kết nối với Google drive để đọc file và lưu file

Nội dung	Chi tiết
Lệnh kết nối	from google.colab import drive drive.mount("face_mask_dataset", force-remount=True)

4.3 Kết nối với google drive

## 2.3 Khai báo đường dẫn đến thư mục chứa dữ liệu

Nội dung	Chi tiết
<ul style="list-style-type: none"><li>Khai báo đường dẫn đến thư mục chứa ảnh để huấn luyện và kiểm định mô hình bằng lệnh</li></ul>	<pre>Import os train_image_files_path = “...” train_image_files_path = “...”</pre>

### 4.4 Khai báo đường dẫn

## 2.4 Dán nhãn cho dữ liệu

Nội dung	Chi tiết
<ul style="list-style-type: none"><li>Phân loại ảnh là bài toán học có giám sát, do đó dữ liệu huấn luyện và kiểm định phải được gán nhãn. Ảnh được gán nhãn dựa theo tên thư mục chứa nó, ta có tên và thứ tự các nhãn tương ứng với tên và thứ tự các thư mục chứa ảnh huấn luyện và kiểm định,</li><li>Nhóm đưa ra 2 nhãn để gán cho dữ liệu là: mask (đối tượng có đeo khẩu trang) và nomask (đối tượng không đeo khẩu trang).</li></ul>	<pre>label=[‘mask’,’nomask’]</pre>

### 4.5 Dán nhãn cho dữ liệu

## 2.5 Tiền xử lý dữ liệu

Nội dung	Chi tiết
Một file ảnh được lưu trong máy tính dưới dạng một ma trận dữ liệu số có giá trị trong khoản [0,255]	<pre>from tensorflow.keras.preprocessing.image import ImageDataGenerator train_data_gen = ImageDataGenerator(rescale=1/255) validation_data_gen = ImageDataGenerator(rescale=1/255)</pre>

### 4.6 Tiền xử lý dữ liệu

## 2.6 Đọc dữ liệu train, validation và gán nhãn tương ứng với cấu trúc của các thư mục con

<b>Nội dung</b>	Ở đây, nhóm chọn biến đổi ảnh về cùng một kích thước là [100x100]
<b>Chi tiết</b>	<pre> Train_generator= train_data_gen.flow_flow_fromdirectory (train_image_files_path,target_size(100,100), batch_size=32,class_mode='categorical')  Validation_generator = validation_data_gen.flow_from_director( Valid_image_files_path, Target_size=(100,100), Batch_szie=32, Class_mode='categorical') </pre>

### 4.7 Đọc dữ liệu train, validation và gán nhãn tương ứng

## 2.7 Xây dựng mô hình

<b>Nội dung</b>	<ul style="list-style-type: none"> <li>– Mô hình gồm 6 tầng: Input image -&gt; CNN 1 -&gt; CNN 2 -&gt; CNN 3 -&gt; Fully connected layer -&gt; Output</li> <li>– Tầng CNN 1: gồm 64 bộ lọc kích thước 3x3. Tầng CNN1 kết nối với đầu vào nên phải mô tả thông tin của đầu vào (input_shape). Với mỗi bộ lọc khác nhau sẽ học được những đặc trưng khác nhau của ảnh, do đó mỗi tầng convolutional ta sẽ dùng nhiều bộ lọc (CNN 1 sử dụng 64 bộ lọc) để học được nhiều đặc trưng của ảnh (ví dụ biên ngang, biên dọc,...)</li> <li>– Tầng CNN 2: gồm 64 bộ lọc kích thước 3x3.</li> <li>– Tầng CNN 3: gồm 64 bộ lọc kích thước 3x3.</li> <li>– MaxPooling2D: lớp Pooling thường được dùng giữa các lớp convolutional, để giảm kích thước dữ liệu nhưng vẫn giữ được các thuộc tính quan trọng. Kích thước dữ liệu giảm sẽ giúp giảm việc tính toán trong model.</li> <li>– Hầu hết khi dùng pooling layer thì sẽ dùng cửa sổ trượt size = (2,2), bước dịch chuyển stride=2. Khi đó kích thước của dữ liệu sẽ giảm đi một nửa.</li> <li>– Có 2 loại pooling layer là: max pooling và average pooling. Ở đây nhóm chọn loại max pooling.</li> <li>– Hàm kích hoạt Relu để loại các giá trị âm.</li> <li>– Flatten: chuyển ảnh từ dạng ma trận về mảng 1 chiều.</li> <li>– Sau khi ảnh được truyền qua nhiều lớp CNN thì mô hình đã học được các đặc điểm của ảnh, khi đó output của lớp CNN cuối cùng là một ma trận, sẽ được chuyển về 1 vector một chiều.</li> <li>–</li> </ul>
-----------------	--

<b>Chi tiết</b>	<pre> model=tf.keras.models.Sequential([     keras.layers.Conv2D(64,(3,3),         activation='relu',input_shape=(100,100,3))     keras.layers.MaxPooling2D(2,2),      keras.layers.Conv2D(64,(3,3), activation='relu'),     keras.layers.MaxPooling2D(2,2),      keras.layers.Conv2D(64,(3,3), activation='relu',     keras.layers.MaxPooling2D(2,2),      keras.layers.Flatten(),     keras.layers.Dense(512, activation=tf.nn.relu),     keras.layers.Dense(2, activation=tf.nn.softmax) ]) </pre>
-----------------	---

#### 4.8 Xây dựng mô hình

## 2.8 Thiết lập các tham số để huấn luyện mô hình

Nội dung	Chi tiết
Có rất nhiều thuật toán tối ưu Optimizer khác nhau. Xem xét ưu, nhược điểm từng loại, nhóm quyết định chọn Adam để huấn luyện mô hình	<pre> from tensorflow.keras.optimizers import Adam model.compile(optimizer = Adam(lr=0.001),                loss='categorical_crossentropy',                metrics=['acc']) </pre>

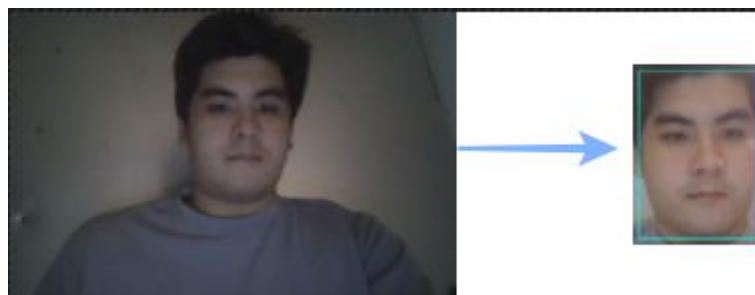
#### 4.9 Thiết lập tham số

## 2.9 Huấn luyện mô hình

Nội dung	Chi tiết
Với số vòng lặp Epochs=300, có thể thấy khi huấn luyện đến epoch từ 200 thì độ chính xác của mô hình đã ở mức ổn định	<pre> EPOCHS=300 history=model.fit(     train_generator,     steps_per_epoch=2,     epochs=EPOCHS,     verbose=1,     validation_data = validation_generator,     validation_steps=2) </pre>

#### 4.10 Huấn luyện mô hình





4.1 Xác định được khuôn mặt ở camera

## 2.10 Sử dụng mô hình

Nội dung	Chi tiết
Với mục tiêu xây dựng hệ thống chạy trên thiết bị di động nhóm đã đưa ra giải pháp là để mô hình chạy trên nền tảng web, dễ truy cập, thao tác sử dụng trên thiết bị di động từ đó tiết kiệm chi phí lắp ráp và vận hành.	Push mô hình lên github

4.11 Sử dụng mô hình

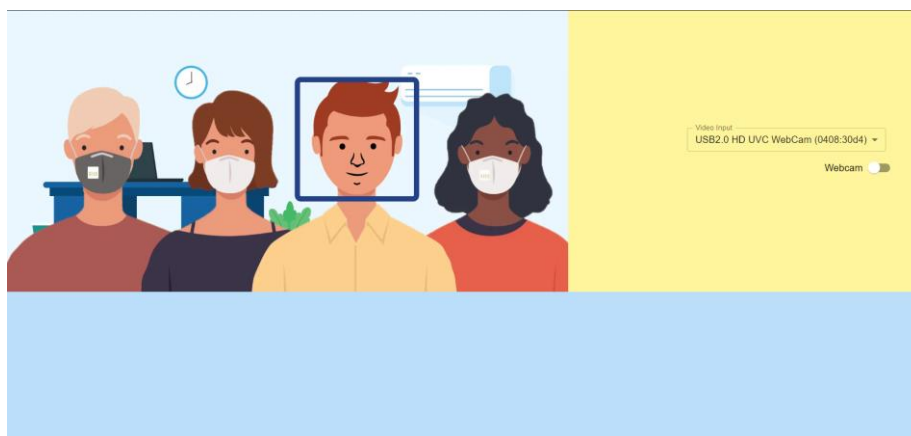


4.2 Kết quả sử dụng mô hình

## 2.11 Giao diện

### 2.11.1 Giao diện chính:

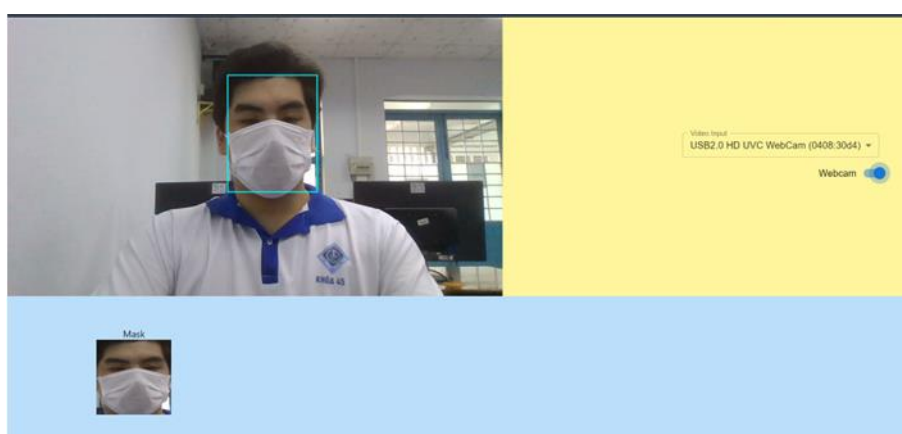
- + Nút xin quyền truy cập camera (chưa được cấp quyền)
- + Box hiển thị thông tin camera đang sử dụng
- + Banner



4.3 Giao diện chính của mô hình

### 2.11.2 Giao diện khi mô hình hoạt động:

- + Nút xin quyền truy cập camera (đã được cấp quyền)
- + Box hiển thị thông tin camera đang sử dụng
- + Ô hiển thị hình ảnh camera
- + Ô hiển thị kết quả
- + Dòng Text hiển thị kết quả



4.4 Giao diện khi mô hình hoạt động

## 3. ĐÁNH GIÁ KIỂM THỬ MÔ HÌNH

### 3.1. Kiểm thử 01:

#### Mục tiêu:

⇒ Kiểm tra độ sai sót của mô hình trên tập dữ liệu huấn luyện.

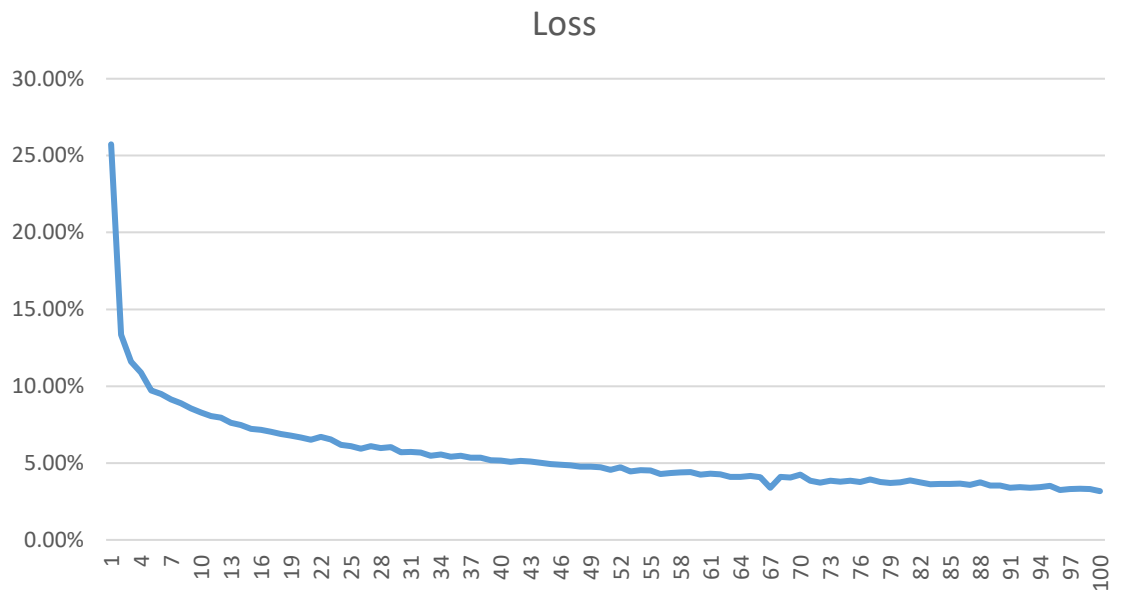
#### Tiến hành:

– Bước 1: Kiểm tra giá trị “loss”.

#### Kết quả mong đợi:

⇒ Giá trị “loss” sẽ giảm dần về dưới 10%.

#### Kết quả kiểm thử:



*4.5 Giá trị loss*

**=>Kết luận: Đạt**

### 3.2.Kiểm thử 02:

#### Mục tiêu:

⇒ Kiểm tra độ chính xác của mô hình trên tập dữ liệu huấn luyện.

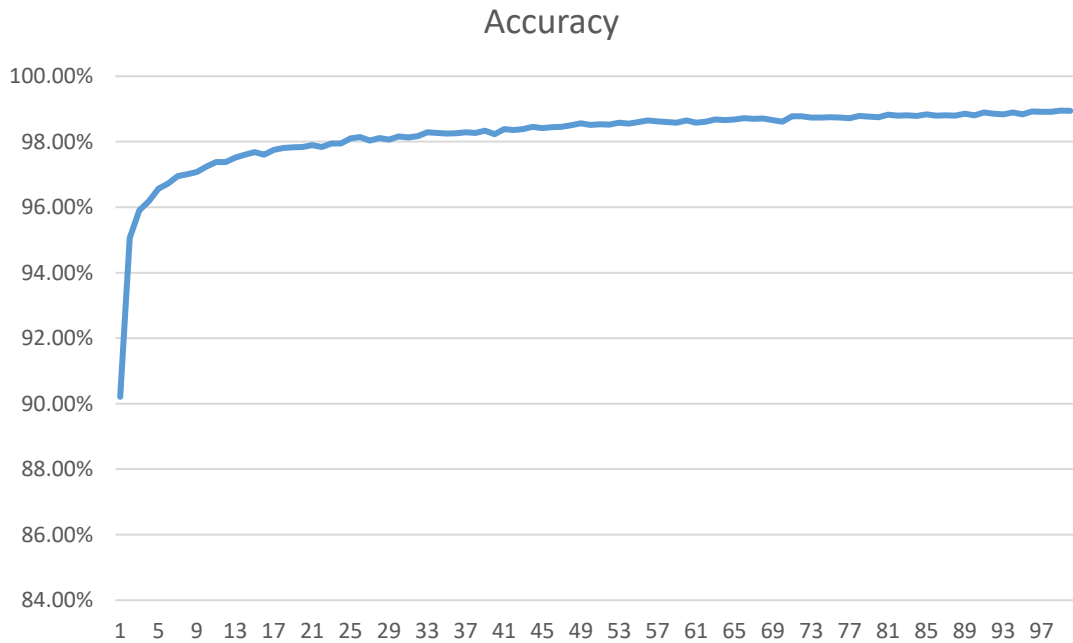
#### Tiến hành:

- Bước 1: Kiểm tra giá trị “accuracy”.

#### Kết quả mong đợi:

⇒ Giá trị “accuracy” sẽ tăng dần đến trên 95%.

#### Kết quả kiểm thử:



#### 4.6 Giá trị accuracy

⇒**Kết luận: Đạt**

### 3.3.Kiểm thử 03:

#### Mục tiêu:

⇒ Kiểm tra độ sai sót của mô hình trên tập dữ liệu kiểm định.

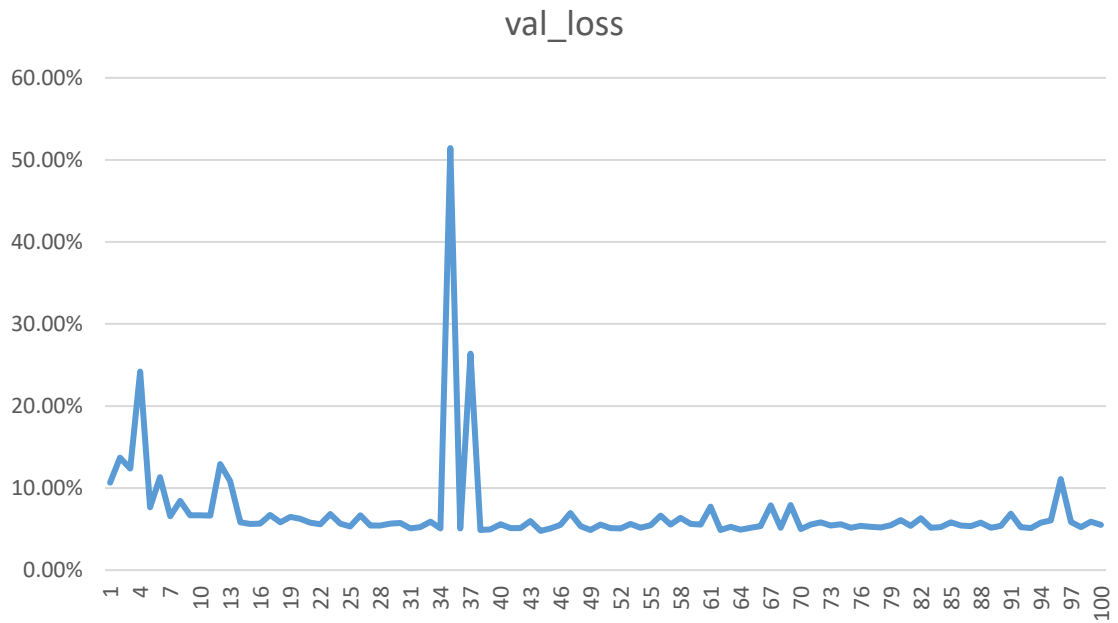
#### Tiến hành:

- Bước 1: Kiểm tra giá trị “val\_loss”.

#### Kết quả mong đợi:

⇒ Giá trị “val\_loss” sẽ giảm dần xuống dưới 10%.

#### Kết quả kiểm thử:



4.7 Giá trị val\_loss

⇒**Kết luận: Đạt**

### 3.4. Kiểm thử 04:

#### Mục tiêu:

⇒ Kiểm tra độ chính xác của mô hình trên tập dữ liệu kiểm định.

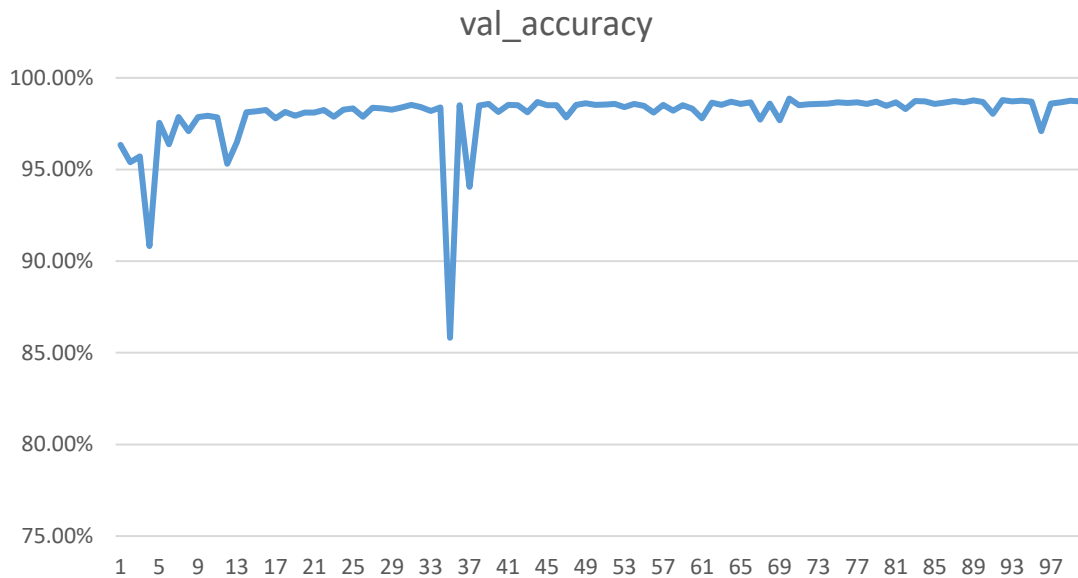
#### Tiến hành:

- Bước 1: Kiểm tra giá trị “val\_accuracy”.

#### Kết quả mong đợi:

⇒ Giá trị “val\_accuracy” sẽ tăng dần đến trên 95%.

#### Kết quả kiểm thử:



4.8 Giá trị val\_accuracy

## CHƯƠNG 5: KIỂM THỬ HỆ THỐNG ỨNG DỤNG

Trong chương này, chúng tôi sẽ tiến hành kiểm thử các chức năng của hệ thống để đánh giá mức độ hoàn thiện và đạt yêu cầu của hệ thống cũng như là đáp ứng được nhu cầu đã đặt ra.

### 1. MỤC TIÊU KIỂM THỬ

Mục tiêu kiểm thử đối với hệ thống của chúng tôi nhằm thiết kế các trường hợp kiểm thử để đánh giá mức độ hoàn thiện và đạt yêu cầu của hệ thống. Từ đó, giúp mọi người có cái nhìn độc lập và đánh giá được những rủi ro trong quá trình phát triển và bảo trì hệ thống.

### 2. NỘI DUNG KIỂM THỬ

#### 2.1 Các tính năng được kiểm thử

- + Nhận biết gương mặt có đeo khẩu trang không ở ảnh.
- + Nhận biết gương mặt có đeo khẩu trang không ở webcam điện thoại.
- + Hiện thị kết quả: Mask và No Mask.
- + Truy cập và load được mô hình trên môi trường trình duyệt.

#### 2.2 Tiêu chí kiểm thử

- **Tiêu chí kiểm thử thành công:**
  - + Vận hành được ứng dụng trên trình duyệt.
  - + Nhận biết được gương mặt người có/không đeo khẩu trang.
  - + Đưa ra cảnh báo.
- **Tiêu chí kiểm thử thất bại:**
  - + Không vận hành được ứng dụng
  - + Không nhận biết được gương mặt người có/không đeo khẩu trang.
  - + Xuất hiện lỗi không mong muốn làm sai kết quả.

#### 2.3 Sản phẩm bàn giao của kiểm thử

- Tài liệu kế hoạch kiểm thử
- Các trường hợp kiểm thử
- Báo cáo kiểm thử

#### 2.4 Quản lý kiểm thử

##### 2.4.1 Thiết bị sử dụng

- Camera điện thoại Xiaomi Note 10 pro
- Webcam Laptop Asus vivoBook X513UA\_M513UA

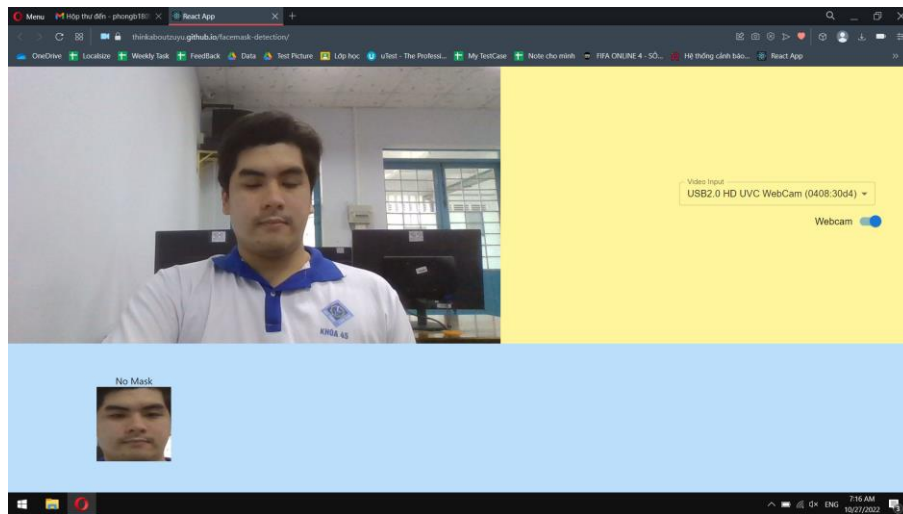
##### 2.4.2 Môi trường kiểm thử

- + Trình duyệt GoogleChrome: phiên bản 106.0.5249.126
- + Webcam laptop.
- + Camera điện thoại:
  - + Camera trước: 16MP
  - + Camera sau: 108MP & 8MP, 5MP, 2MP

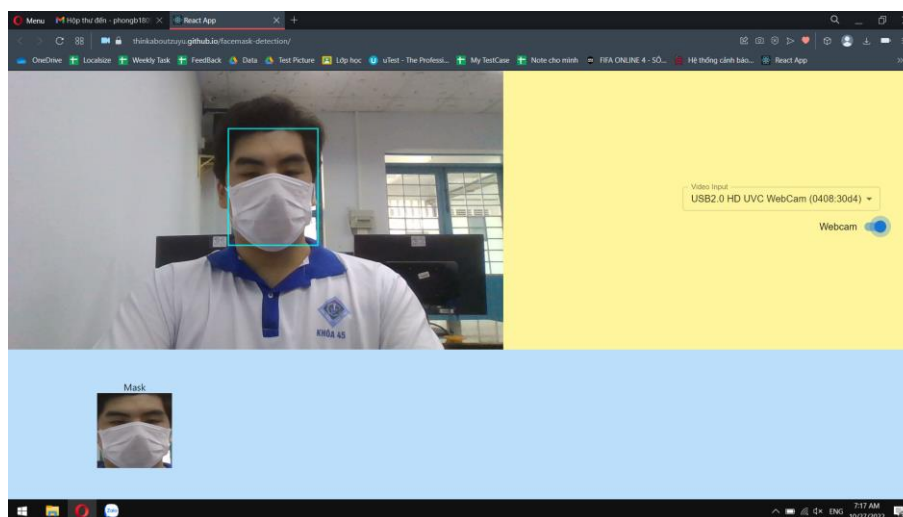
## 2.4.3 Các trường hợp kiểm thử

### 2.4.3.1 Sử dụng webcam để kiểm tra

- Sử dụng webcam với 1 người:



*No Mask*



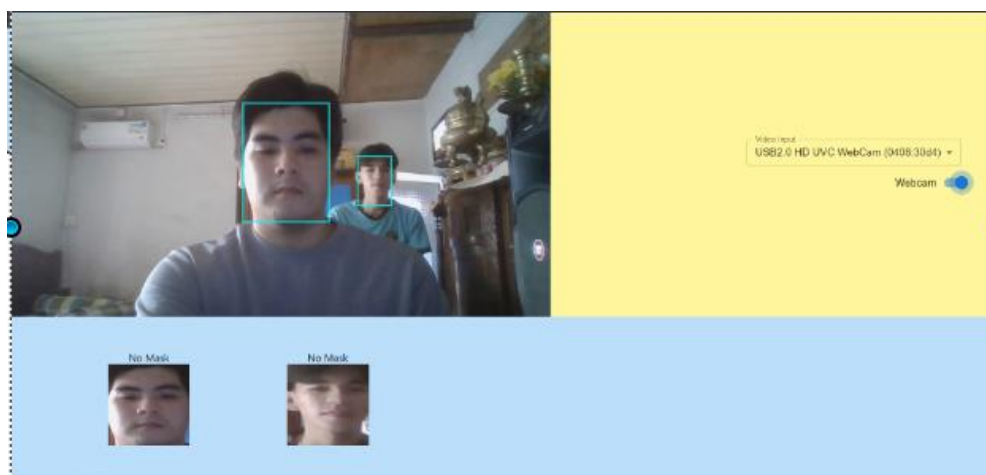
*Mask*



- Sử dụng webcam với >1 người:



*Mask*



*No Mask*

- Sử dụng webcam ở điều kiện ngược sáng:

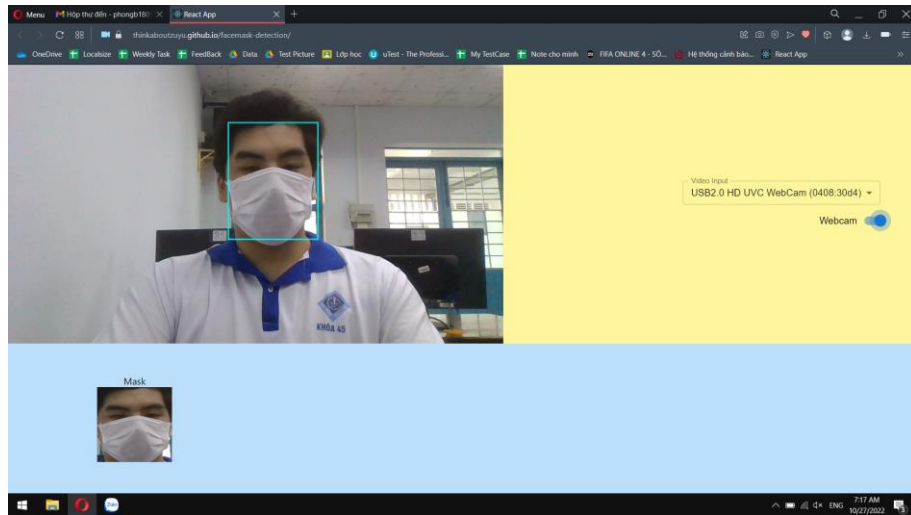


*Masked*

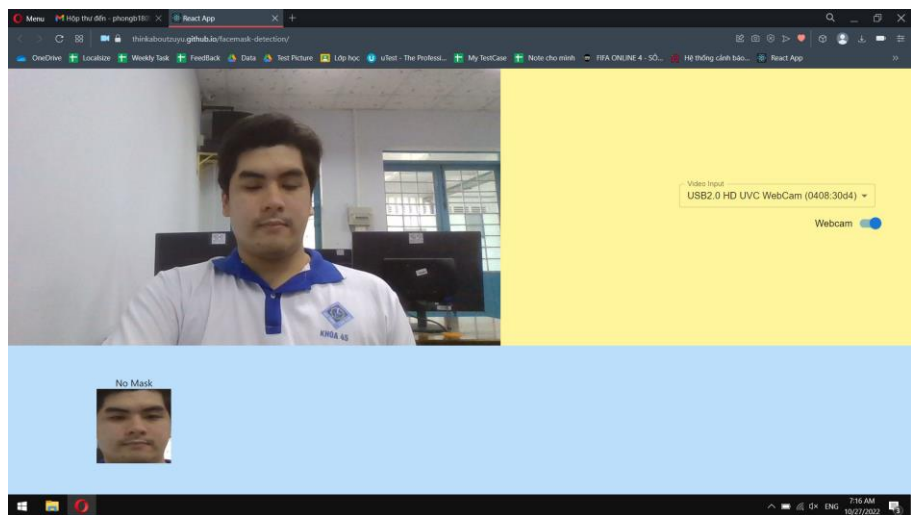


*No Mask*

- Sử dụng webcam ở điều kiện bình thường:



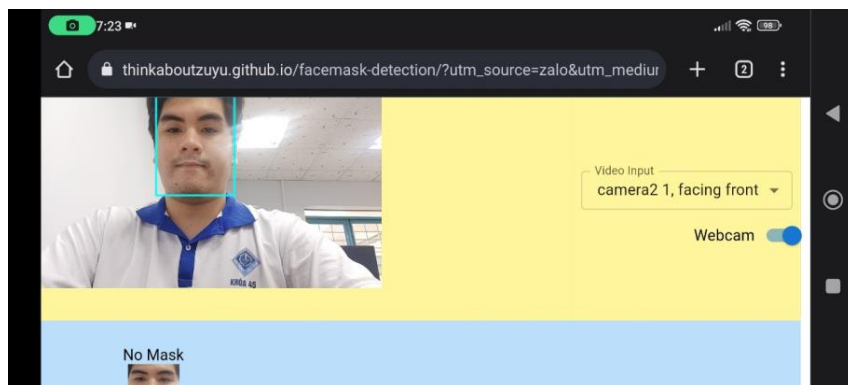
*Mask*



*No Mask*

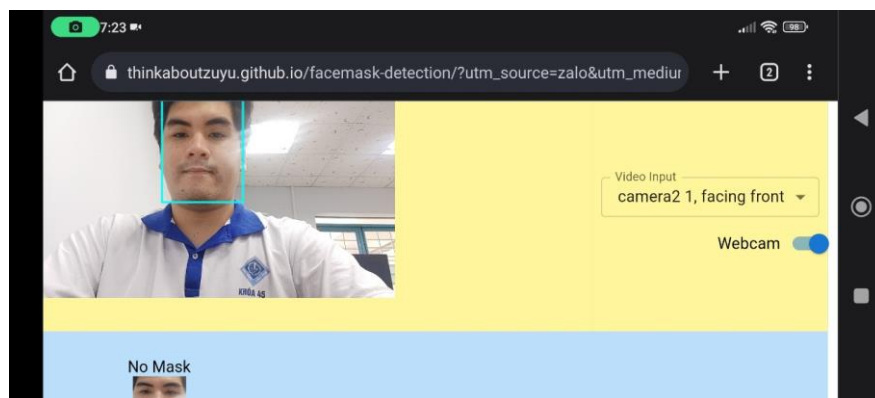
### 2.4.3.2 Sử dụng camera trước điện thoại (có kết nối internet)

- Truy cập được vào web và load được mô hình:

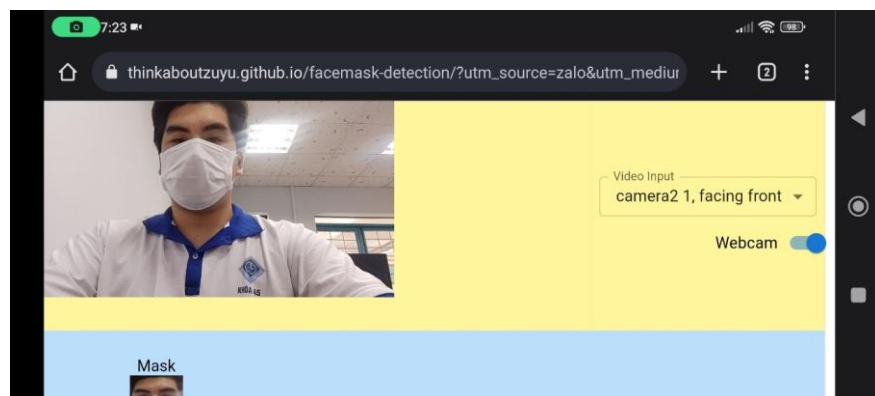


*Truy cập và load mô hình thành công*

- Sử dụng với 1 đối tượng:

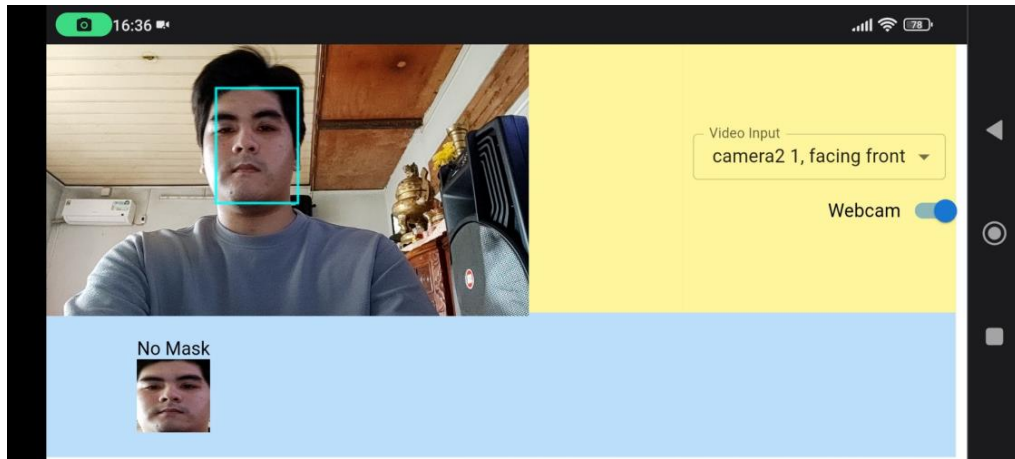


*No Mask*

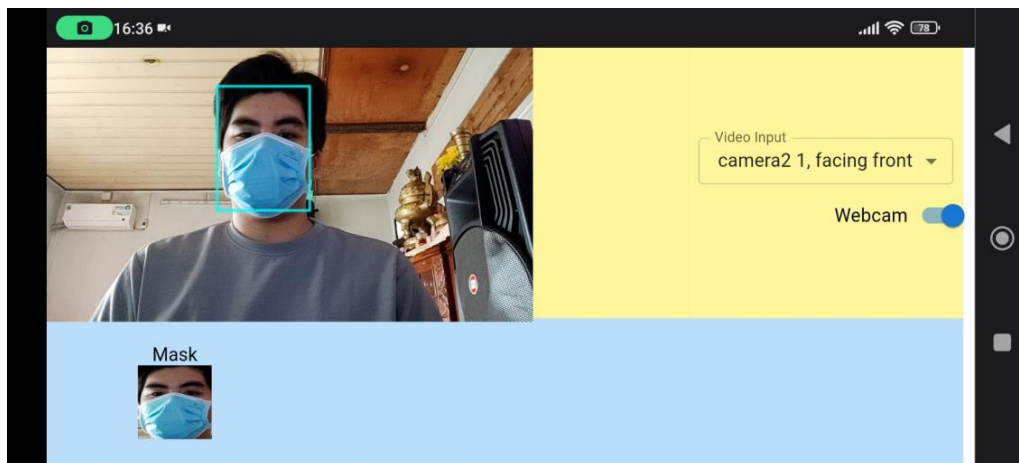


*Mask*

- Sử dụng với >1 đối tượng:

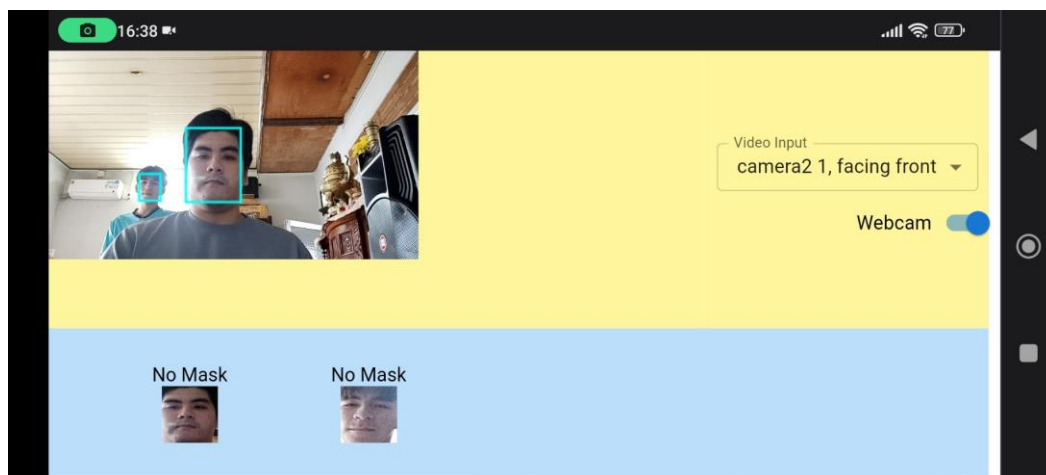


*No mask*

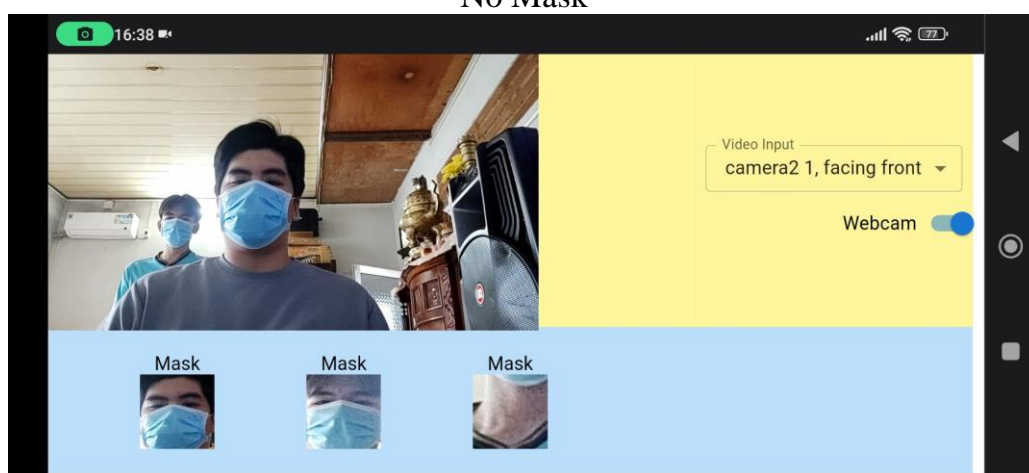


*Masked*

- Sử dụng ở điều kiện ngược sáng:

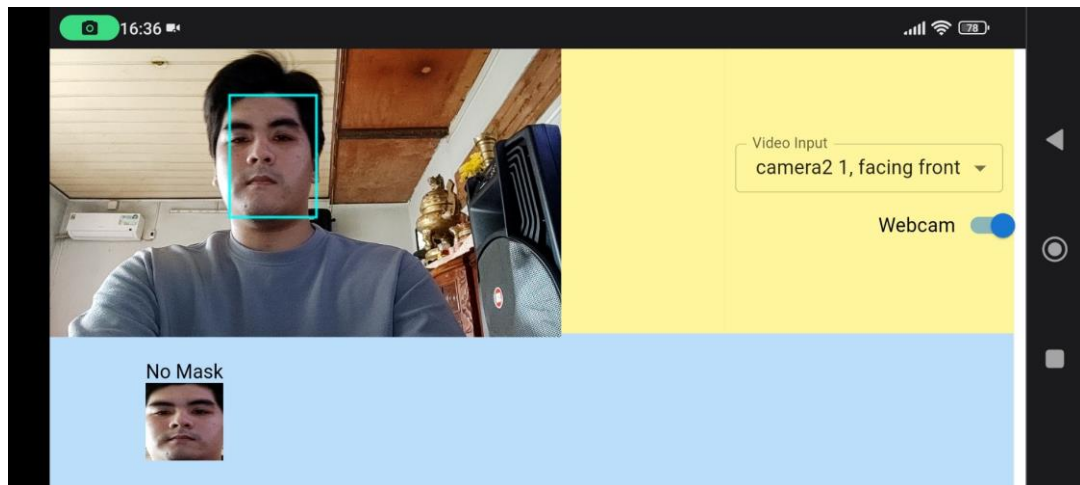


No Mask

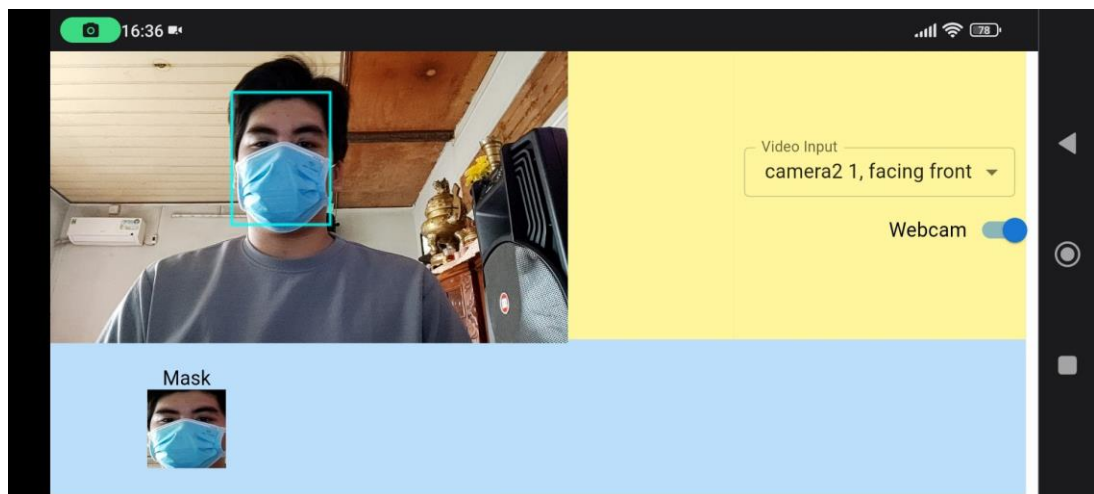


Mask

- Sử dụng ở điều kiện bình thường:



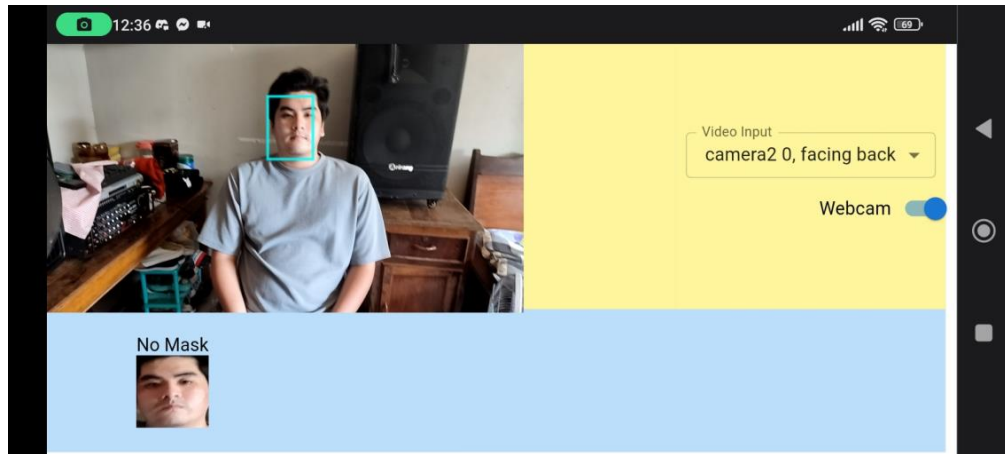
No Mask



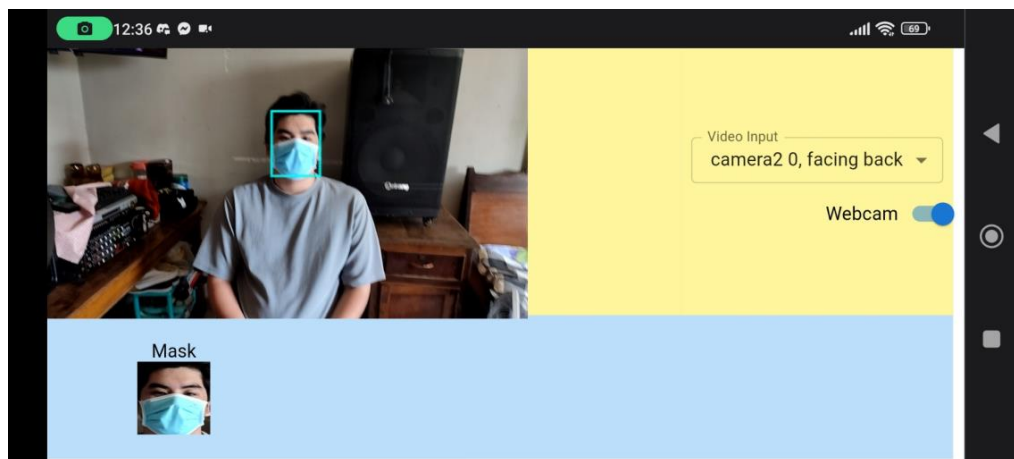
Mask

### 2.4.3.3 Sử dụng camera sau điện thoại (có kết nối internet)

- Sử dụng với 1 đối tượng:



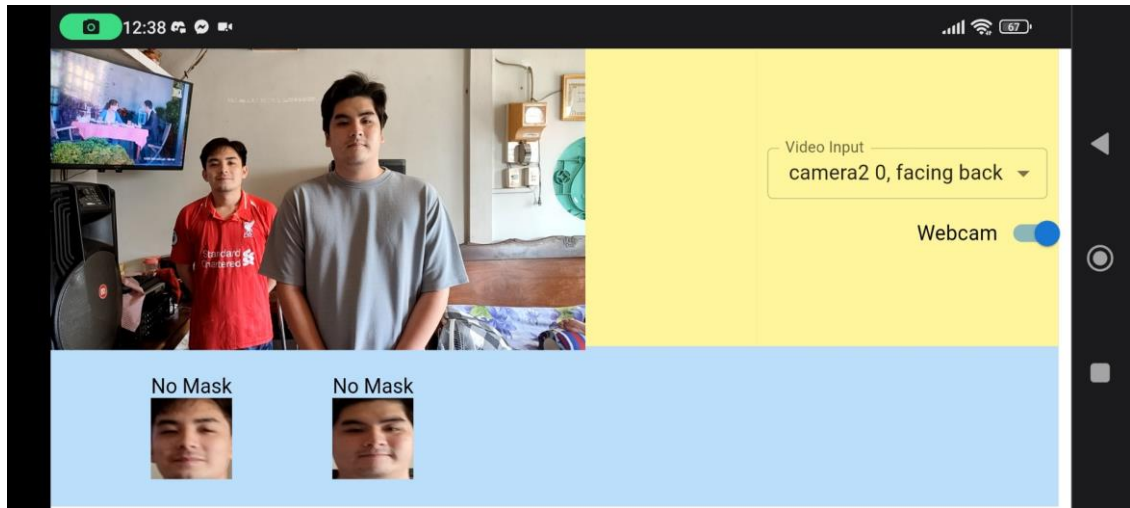
*No Mask*



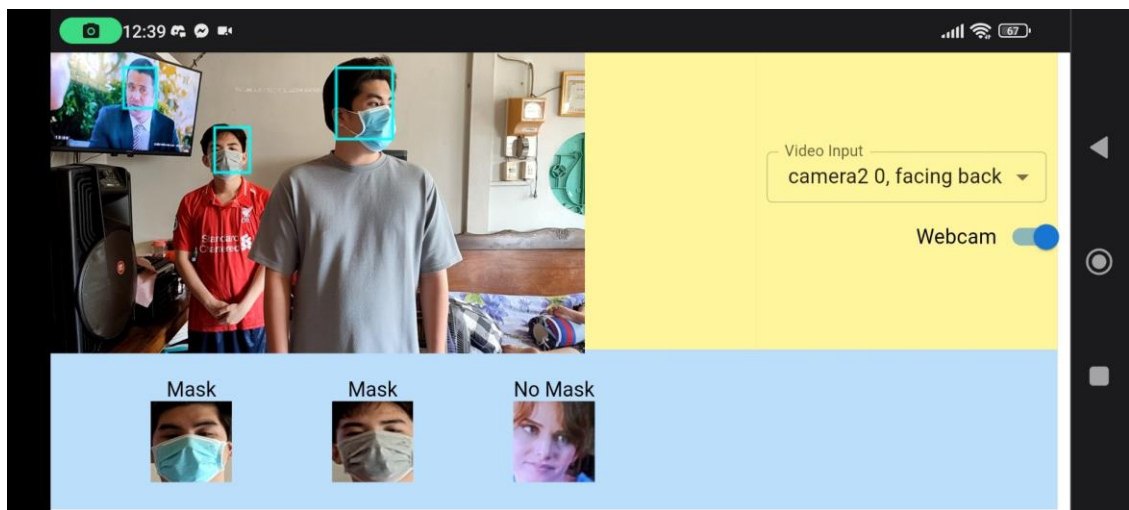
*Mask*



- Sử dụng với >1 đối tượng:

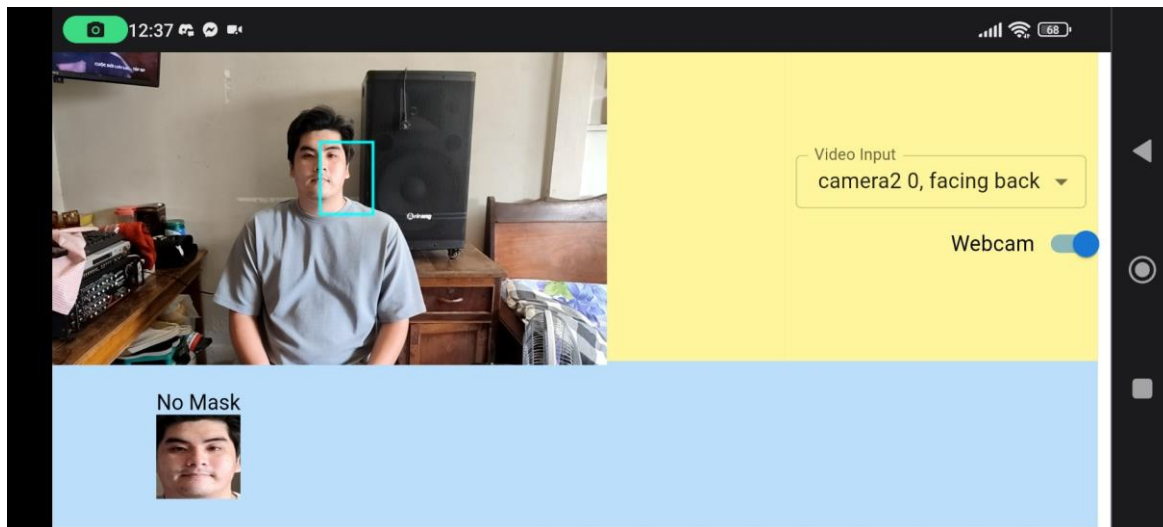


*No Mask*

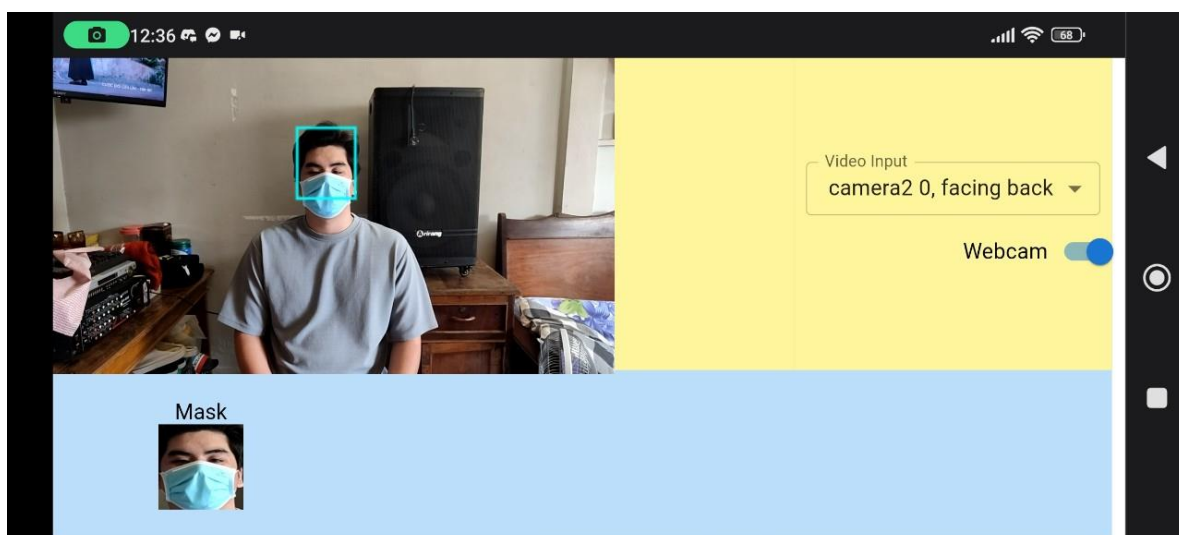


*Mask*

- Sử dụng ở điều kiện ngược sáng:

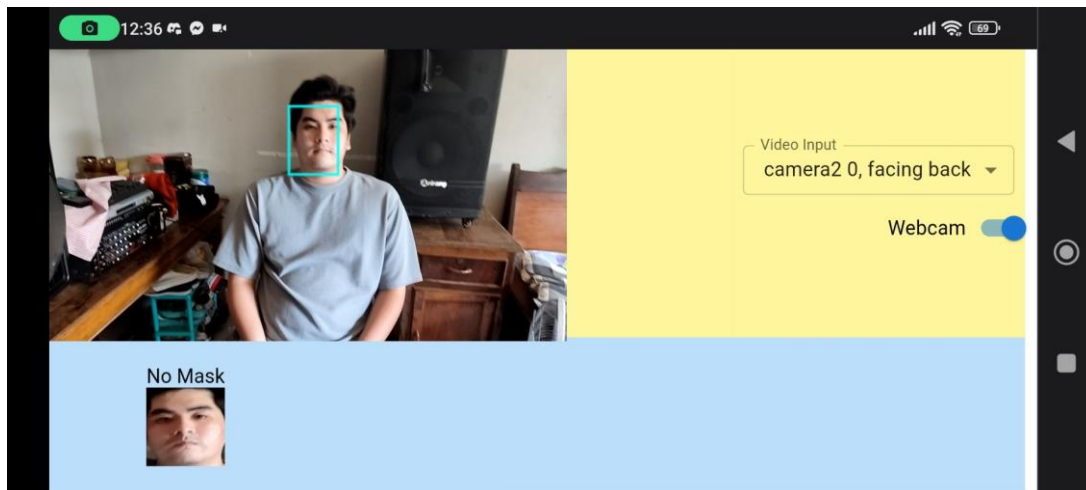


*No Mask*

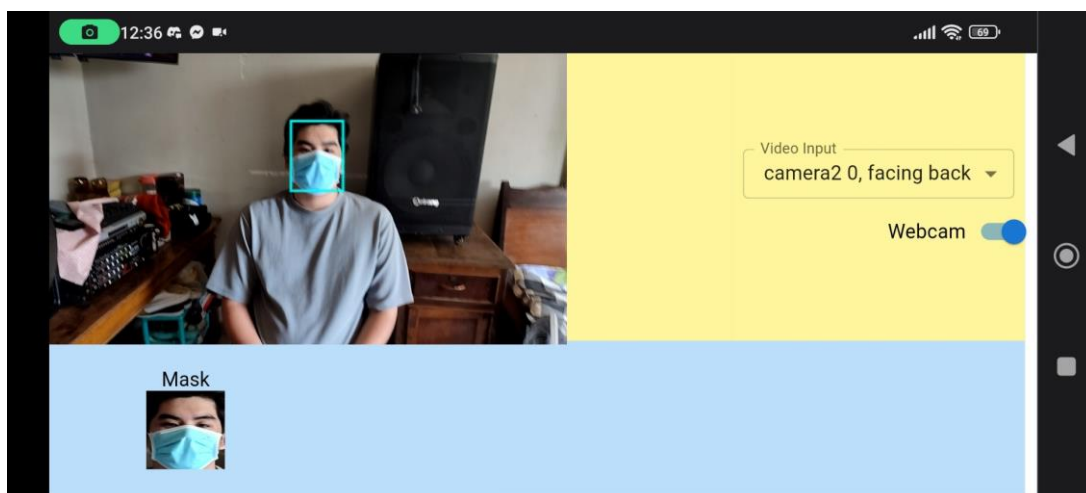


*Mask*

- Sử dụng ở điều kiện bình thường:



No Mask



Mask

ID	Nội dung	Kết quả mong đợi	Kết quả thực tế	Đánh giá
Trên Webcam				
1	Dùng webcam để xác định khẩu trang trên 1 đối tượng	Xác định đúng khi đối tượng có/không đeo khẩu trang và đưa ra thông báo	Pass	Đạt
2	Dùng webcam để xác định khẩu trang trên nhiều đối tượng	Xác định đúng khi đối tượng có/không đeo khẩu trang và đưa ra thông báo	Pass	Đạt
3	Dùng webcam để xác định khẩu trang ở điều kiện ngược sáng	Xác định đúng khi đối tượng có/không đeo khẩu trang và đưa ra thông báo	Pass	Đạt
4	Dùng webcam để xác định khẩu trang ở điều kiện bình thường	Xác định đúng khi đối tượng có/không đeo khẩu trang và đưa ra thông báo	Pass	Đạt
Trên camera trước của điện thoại				
5	Truy cập và sử dụng được mô hình trên nền tảng web	Truy cập được vào trang web và load được mô hình	Pass	Đạt
6	Dùng camera trước để xác định khẩu trang trên 1 đối tượng	Xác định đúng khi đối tượng có/không đeo khẩu trang và đưa ra thông báo	Pass	Đạt
7	Dùng camera trước để xác định khẩu trang trên nhiều đối tượng	Xác định đúng khi đối tượng có/không đeo khẩu trang và đưa ra thông báo	Pass	Đạt
8	Dùng camera trước để xác định khẩu trang ở điều kiện thiếu sáng	Xác định đúng khi đối tượng có/không đeo khẩu trang và đưa ra thông báo	Pass	Đạt
9	Dùng camera trước để xác định khẩu trang ở điều kiện bình thường	Xác định đúng khi đối tượng có/không đeo khẩu trang và đưa ra thông báo	Pass	Đạt
Trên camera sau của điện thoại				
10	Dùng camera sau để xác định khẩu trang trên 1 đối tượng	Xác định đúng khi đối tượng có/không đeo khẩu trang và đưa ra thông báo	Pass	Đạt

11	Dùng camera sau để xác định khẩu trang trên nhiều đối tượng	Xác định đúng khi đối tượng có/không đeo khẩu trang và đưa ra thông báo	Pass	Đạt
12	Dùng camera sau để xác định khẩu trang ở điều kiện thiếu sáng	Xác định đúng khi đối tượng có/không đeo khẩu trang và đưa ra thông báo	Pass	Đạt
13	Dùng camera sau để xác định khẩu trang ở điều kiện bình thường	Xác định đúng khi đối tượng có/không đeo khẩu trang và đưa ra thông báo	Pass	Đạt

5.1 Bảng kết quả kiểm thử mô hình

## 2.5 Kết luận kiểm thử:

### Thành công:

- + Đạt được kì vọng nhận biết được khuôn mặt có/không đeo khẩu trang ở thời gian thực.
- + Nhận biết được khẩu trang ở các điều kiện thiếu sáng, nhiều đối tượng.
- + Chạy thành công mô hình trên nền tảng web, có thể sử dụng trên thiết bị di động, tiện lợi và dễ thao tác.

### Cần cải thiện:

- + Thời gian load mô hình trên thiết bị di động chưa đủ nhanh.
- + Chưa phân biệt được người dùng đeo khẩu trang sai cách.

## **CHƯƠNG 6: KẾT LUẬN VÀ KIẾN NGHỊ**

### **I. KẾT QUẢ ĐÓNG GÓP**

#### **1. Về khoa học và đào tạo**

Sau quá trình thực hiện nghiên cứu, nhóm chúng tôi thu được các kết quả:

- Ứng dụng kỹ thuật học sâu và thị giác máy tính để xây dựng thành công mô hình nhận diện được khuôn mặt của 1 hoặc nhiều đối tượng có đang đeo khẩu trang hay không tương đối chính xác trong điều kiện ánh sáng đủ.
- Thiết kế và xây dựng được hệ thống kiểm tra và nhắc nhở đeo khẩu trang cho sinh viên ngành Công nghệ thông tin Chương trình Chất lượng cao – Khoa Công nghệ Thông tin & Truyền thông với tính năng cần thiết.
- Phát triển hệ thống chạy trên nền tảng web như một ứng dụng cho mục đích dễ thao tác sử dụng, lắp đặt.
- Hệ thống kiểm tra và nhắc nhở đeo khẩu trang có thể giúp các cơ sở công cộng, dịch vụ kiểm soát được lượng người sử dụng khẩu trang giúp phần hạn chế tối đa sự lây nhiễm của COVID-19 trong thời gian bình thường mới.

#### **2. Về phát triển kinh tế - xã hội**

##### **2.1 Về kinh tế**

Giảm thiểu được kinh phí cần để xây dựng mô hình, các thao tác dễ sử dụng, thân thiện với người dùng.

##### **2.2 Về xã hội**

Giúp kiểm soát được lượng người chấp hành đeo khẩu trang nơi công cộng một cách tự động, giảm tối thiểu các tiếp xúc gần giữa người và người trong thời gian dịch COVID-19.

### **II. TRIỂN KHAI ỨNG DỤNG**

- Sau khi hoàn thành ứng dụng sẽ được triển khai ở các cổng vào các cơ sở công cộng như lớp học, bệnh viện, tiệm thuốc, quán ăn, ...
- Ứng dụng được xây dựng miễn phí, không gặp trở ngại về bản quyền, chi phí.
- Ứng dụng được chạy trên nền tảng website nên dễ sử dụng, không cần lắp ráp nhiều thiết bị, có tính di động, dễ sử dụng, dễ thao tác.

### **III. HƯỚNG PHÁT TRIỂN**

Ngoài các kết quả đạt được, nhóm chúng tôi cũng có một số phương hướng phát triển đề tài trong tương lai:

- Phát triển bài toán nhận dạng khuôn mặt để xây dựng thêm các hệ thống phát hiện và định danh đối tượng trên các thiết bị như camera giám sát, ...
- Phát triển mô hình điểm danh bằng khuôn mặt và tránh các trường hợp gian lận khi điểm danh.
- Cải tiến website hệ thống để người dùng có thêm nhiều tính năng.
- Nghiên cứu và tích hợp hệ thống thông tin của nhà trường để lưu tên và báo cáo khi phát hiện đối tượng (là sinh viên, nhân viên) thuộc quản lý của nhà trường vi phạm an toàn 5K không đeo khẩu trang nơi công cộng) để giúp vận hành vào thực tế dễ dàng và phù hợp hơn.

- Nghiên cứu và phát triển hệ thống để nhận diện được những đối tượng đeo khẩu trang sai cách (chừa mũi, miệng) hoặc lấy tay che miệng, mũi để cải thiện độ chính xác của hệ thống.

## TÀI LIỆU THAM KHẢO

- [1] <https://viblo.asia/p/cnn-architecture-series-1-mobilenets-mo-hinh-gon-nhe-cho-mobile-applications-1VgZvJV1ZAw>
- [2] <https://teky.edu.vn/blog/opencv-la-gi/>
- [3] <https://www.kaggle.com/datasets/andrewmvd/face-mask-detection>
- [4] [https://www.researchgate.net/publication/356535764\\_Face\\_Mask\\_Recognition\\_Using\\_MobileNetV2](https://www.researchgate.net/publication/356535764_Face_Mask_Recognition_Using_MobileNetV2)
- [5] <https://topdev.vn/blog/thuat-toan-cnn-convolutional-neural-network/>
- [6] <https://www.johnsoncontrols.com/insights/2021/thought-leadership/understanding-facemaskdetectiontechnology#:~:text=Face%20mask%20detection%20is%20an,who%20is%20not%20wearing%20one>
- [7] <https://phamdinhhkhanh.github.io/2020/09/19/MobileNet.html>
- [8] <https://miai.vn/2020/03/11/chung-tay-chong-covid-19-lam-thu-he-thong-nhac-deo-khau-trang-bang-opencv/>
- [9] <https://www.run.ai/guides/deep-learning-for-computer-vision>