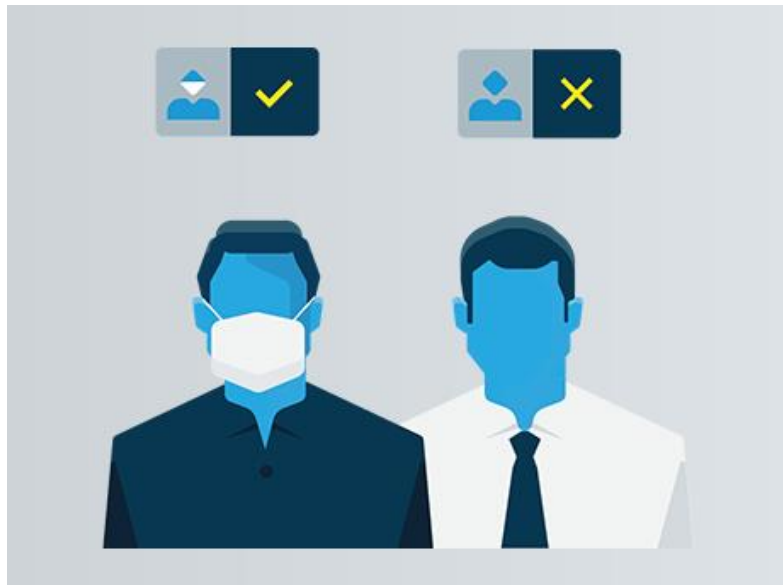


BÁO CÁO TÌM HIỂU

ĐỀ TÀI

# FACE MASK DETECTION



Thực hiện: Phạm Đình Thục

Tháng 11 năm 2021

## I. Đặt vấn đề

Trong tình hình dịch COVID-19, mọi người luôn đeo khẩu trang để bảo vệ mình khỏi bị lây nhiễm, vấn đề đặt ra là một công cụ nhằm phát hiện một cá nhân có đeo khẩu trang hay không. Thông qua công cụ này, nó sẽ giúp phát hiện sớm các trường hợp không đeo khẩu trang hoặc đeo không đúng cách nhằm ngăn ngừa sự lây lan dịch bệnh, góp phần bảo vệ sức khỏe, tính mạng của bản thân và những người xung quanh.

## II. Mục tiêu

Mục tiêu chính của đề tài:

- Hiểu được cách thức vận hành, chạy của một công cụ phát hiện người nào đó có đeo khẩu trang hay không?
- Nâng cấp bộ dữ liệu có sẵn nhằm tăng tính tổng quát cho mô hình khám phá
- Cải tiến một số function để tăng hiệu suất quá trình huấn luyện.

## III. Nội dung tìm hiểu

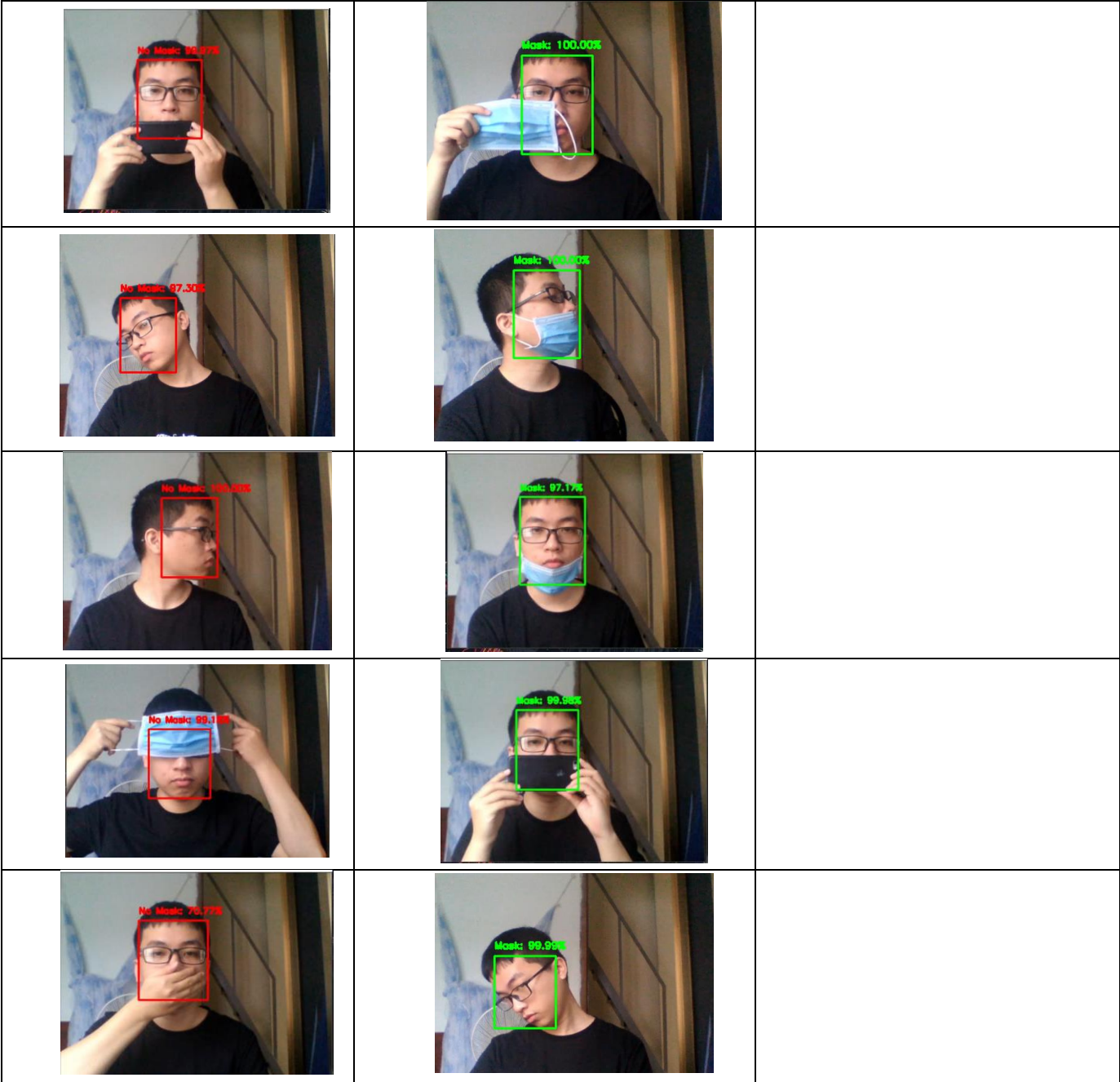
- Đầu tiên là quá trình tìm hiểu bao gồm các bước: Đọc, tìm hiểu tài liệu được yêu cầu.
- Clone Repository từ Github, tiến hành train với bộ dữ liệu có sẵn
  - o Bộ dataset: bao gồm 2 bộ (without\_mask và with\_mask) với hơn 4k Dữ liệu
- Sau khi đã train cho model. Tiến hành test thử nghiệm. Có 2 cách để test:
  - o Thông qua ảnh tính: input đầu vào là một ảnh, output là kết quả có đeo khẩu trang (**khung xanh** - tỉ lệ) hay không (**khung đỏ** - tỉ lệ)
  - o Thông qua webcam: Sẽ sử dụng webcam để ghi lại hình ảnh và trả về kết quả theo thời gian thực
- Tiếp theo, ta tiến hành thu thập thêm dữ liệu nhằm tăng tính tổng quát cho mô hình:  
Sau khi đã tham khảo nhiều nguồn, đã chọn được dataset phù hợp:
  - o Bộ dataset của Kaggle (link: [Face Mask Detection ~12K Images Dataset | Kaggle](#)) với hơn 12k Dữ liệu của 3 tập: train, validation và test.
  - o Các ảnh ở định dạng .png. Nên trước khi merge vào dataset ban đầu, ta đổi định dạng sang .jpg (Dùng Pillow).
  - o Thêm 3000 ảnh with\_mask và 3000 ảnh without\_mask
- Thuật toán sử dụng công cụ TensorFlow nên cần chạy trên card đồ họa NVIDIA, ta upload repo lên Google Colab để có thể train model (vì lý do máy tính dùng card AMD)
- Sau khi đã upload đầy đủ các file cần thiết, ta tiến hành train model. Có 2 cách có thể train model:
  - o Dùng cú pháp **!python** cho file .py
  - o Dùng file .ipynb có sẵn
- Quá trình train diễn ra nhanh hay chậm ngoài phụ thuộc vào thuật toán còn phụ thuộc vào chất lượng phần cứng. Vì hạn chế về thiết bị (Dùng GG Colab bản miễn phí) nên quá trình diễn ra lâu, do vấn đề tràn RAM khi load tất cả ảnh vào RAM trước khi train.

- **Giải pháp:** Ta chia nhỏ tập train và tiến hành load từng tập ảnh nhỏ vào train. Ở đây ta dùng các thuật toán liên quan đến Data Generator.  
`loss: 0.0053 - accuracy: 0.9989 - val_loss: 0.0271 - val_accuracy: 0.9911`
- Sau khi đã có kết quả từ quá trình trên ta thu được file **mask\_detector.h5** lưu các thông số của mô hình học được.
- Để kiểm tra quá trình train, ta cũng test trên 2 phương diện là ảnh tĩnh và thông qua webcam.

#### IV. Kiểm tra mô hình

- Ta tiến hành kiểm tra trên webcam để có thể phát hiện được một số trường hợp mà mô hình không thể dự đoán chính xác.
- Từ các trường hợp đó, ta có thể xác định và tìm kiếm thêm bộ dữ liệu liên quan đến trường hợp đó để nhằm tăng độ chính xác cho mô hình.

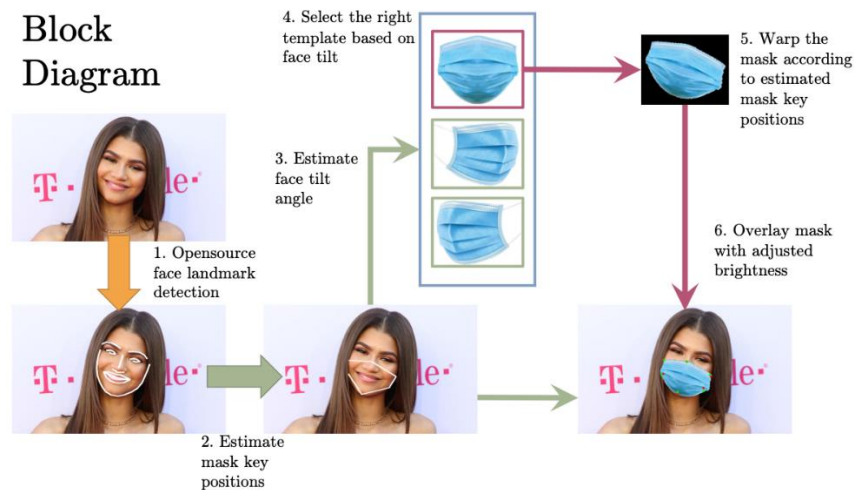
Không đeo Khẩu trang	Đeo khẩu trang	Không nhận diện được
		
		
		



## V. Phương pháp Mask the Face



Nó sử dụng một máy dò các cạnh khuôn mặt dựa trên dlib để xác định độ nghiêng của khuôn mặt và các đặc điểm chính của khuôn mặt cần thiết để đeo khẩu trang vào (khẩu trang ảo). Khẩu trang làm mẫu sẽ được biến đổi theo các hướng của khuôn mặt sao cho khớp nhất có thể. Tham khảo mô hình MaskTheFace



### NOTE:

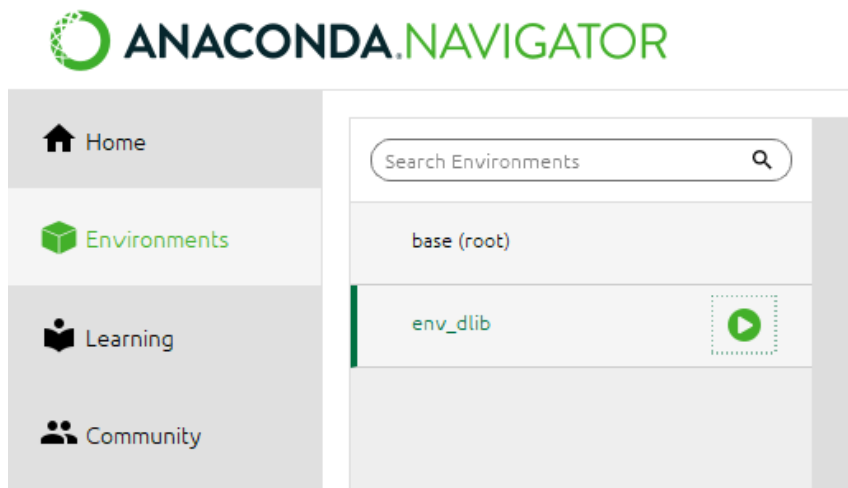
Vì thư viện **DLIB** gặp một số rắc rối trong quá trình cài đặt, ta không thể chạy code trực tiếp trên command prompt. Giải pháp ở đây là sử dụng anaconda.

Ta tiến hành download và cài đặt anaconda.

Tạo môi trường cho Dlib và cài đặt các lib cần thiết theo yêu cầu của repo.

Cài đặt Dlib như sau: **conda install -c conda-forge dlib**

Để có thể dùng ta mở terminal của môi trường vừa cài đặt để sử dụng.



## CÁCH CHẠY CODE

Để có thể thực thi chương trình Mask The Face:

Cú pháp:

**python mask\_the\_face.py --path <path-to-file-or-dir> --mask\_type <type-of-mask> --verbose --write\_original\_image**

Với các thông số có thể bổ sung thêm:

Argument	Giải thích
Path	Chứa địa chỉ của dataset, có thể là địa chỉ 1 ảnh hoặc 1 tập dữ liệu ảnh.
mask_type	Kiểu khẩu trang, lựa chọn mẫu khẩu trang trong các loại được đề cập như: 'N95', 'surgical_blue', 'surgical_green', 'cloth', 'empty' and 'inpaint' hoặc có thể 'all' để thử tất cả các mẫu.
pattern	Chọn họa tiết cho khẩu trang, có thể tham khảo cá mẫu tại masks/textures
pattern_weight	Chọn cường độ của mẫu sẽ được áp dụng. Giá trị phải nằm trong khoảng từ 0 (không có) đến 1 (cường độ tối đa)
color	Chọn màu sẽ được áp dụng cho loại mặt nạ đã chọn ở trên. Các màu được cung cấp dưới dạng giá trị hex. Danh sách các màu như sau: "#fc1c1a", "#177ABC", "#94B6D2", "#A5AB81", "#DD8047", "#6b425e", "#e26d5a", "#c92c48", "6a506d", "#ffc900", "#ffffff", "#000000", "#49ff00".
color_weight	Chọn cường độ của màu sẽ được áp dụng. Giá trị phải nằm trong khoảng từ 0 (không có) đến 1 (cường độ tối đa)
code	áp dụng ngẫu nhiên các loại mặt nạ khác nhau một cách ngẫu nhiên cho các hình ảnh trong tập dữ liệu



verbose	Nếu được đặt thành True, sẽ được sử dụng để hiển thị các thông báo hữu ích trong quá trình tạo mặt nạ
write_original_image	Nếu được sử dụng, hình ảnh unmasked ban đầu cũng sẽ được lưu trong masked image cùng với hình ảnh có mặt nạ đã xử lý

### Mask Types:

Currently MaskTheFace supports the following 4 mask types

1. Surgical
2. N95
3. KN95
4. Cloth
5. Gas



Figure 1: Các mẫu khẩu trang

### 1. Textures/Patterns variations:

MaskTheFace provides 24 existing patterns that can be applied to mask types above to create more variations of the graph. Moreover, users can easily add custom patterns following the guidelines provided.



Figure 2: Họa tiết khẩu trang

## 2. Color variations:

MaskTheFace provided script to modify existing mask types in terms of colors to generate variations of existing graphs.



Figure 3: Màu sắc

## 3. Intensity variations:

MaskTheFace provided script to modify existing mask types in terms of intensity to generate variations of existing graphs.



Figure 4: Cường độ

Thực nghiệm: ta tiến hành thực hành mô hình Mask The Face trên tập dữ liệu VN-Celeb: VN-Celeb là bộ dữ liệu được xây dựng dành riêng cho mục đích nghiên cứu, tối ưu hóa bài toán nhận diện khuôn mặt cho người Việt. Với hơn 23k hình ảnh của hơn 1000 người Việt nổi tiếng.



VN-Celeb dataset



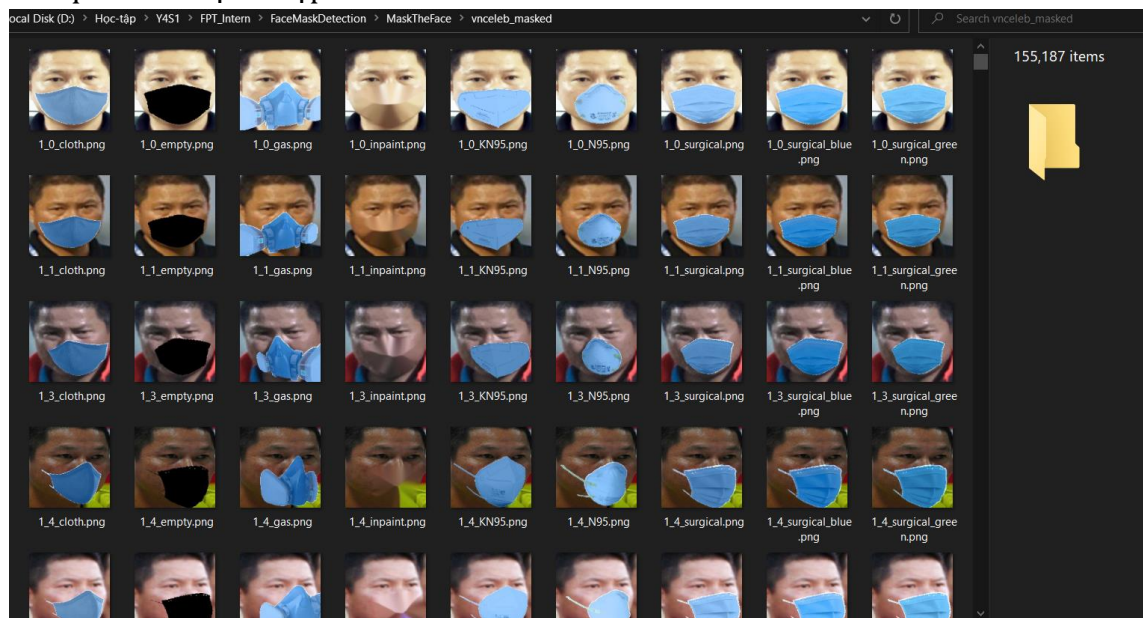
Áp dụng mô hình Mask the Face giới thiệu ở trên với các thông số như path, mask\_type, pattern, pattern\_weight, color, color\_weight, code, verbose, write\_original\_image. để tiến hành đeo khẩu trang cho các hình ảnh trong tập dữ liệu này, ta thu được kết quả là

```
C:\WINDOWS\system32\cmd.exe
Processing: vnceleb/9_19.png
Faces found: 0
Processing: vnceleb/9_2.png
Faces found: 1
Processing: vnceleb/9_20.png
Faces found: 1
Processing: vnceleb/9_21.png
Faces found: 0
Processing: vnceleb/9_3.png
Faces found: 0
Processing: vnceleb/9_4.png
Faces found: 1
Processing: vnceleb/9_5.png
Faces found: 1
Processing: vnceleb/9_6.png
Faces found: 0
Processing: vnceleb/9_7.png
Faces found: 1
Processing: vnceleb/9_8.png
Faces found: 1
Processing: vnceleb/9_9.png
Faces found: 1
100% | 23105/23105 [1:54:41<00:00, 3.36it/s]
----- Masking image directories -----
0it [00:00, ?it/s]
Processing Done

(env_dlib) D:\Hoc-tap\Y4S1\FPT_Intern\FaceMaskDetection\MaskTheFace>
```

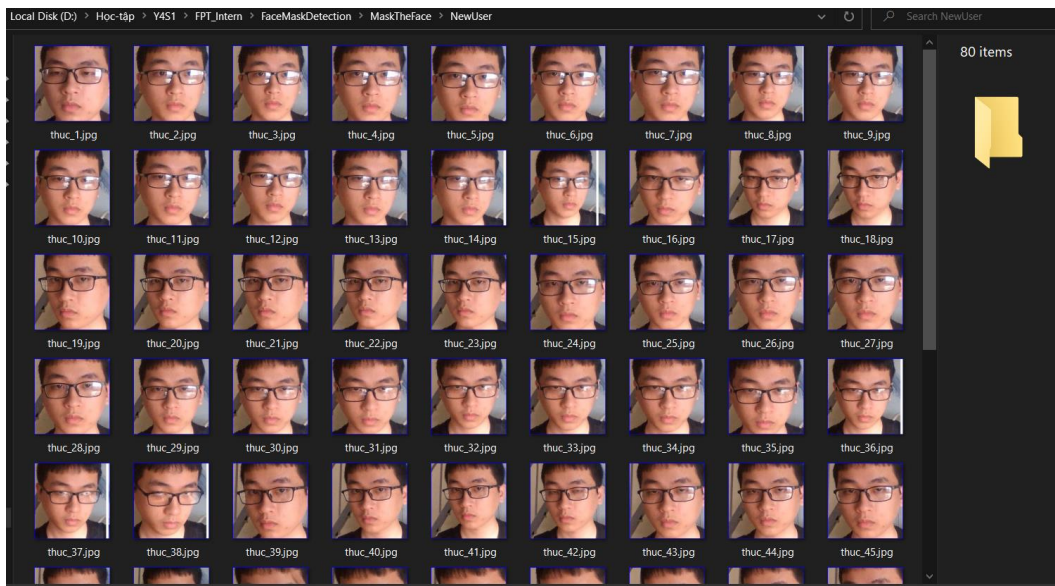
Quá trình kéo dài khá lâu (~2h) với công việc đeo khẩu trang cho hơn 20k ảnh (23105 ảnh) cho hơn 1000 cá nhân (1020 người).

Kết quả thu được là tập ảnh với **155.187** hình ảnh



Thu thập dữ liệu ảnh cá nhân:

Ta chụp ảnh khuôn mặt cá nhân thông qua webcam và OpenCV:



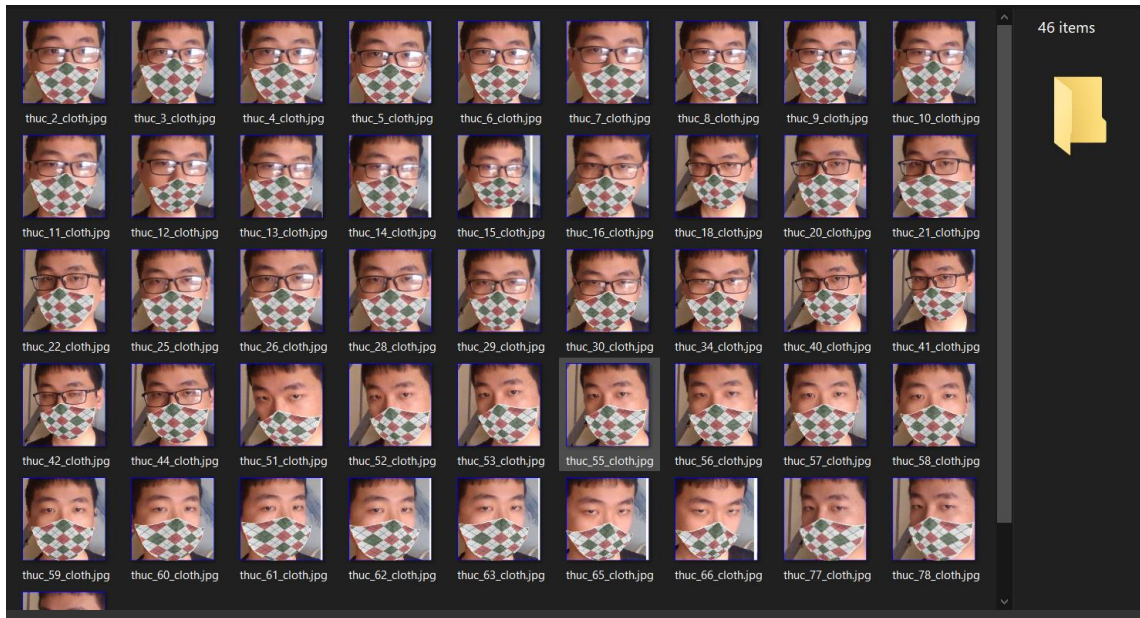
## Quá trình chạy

```
(env_dlib) D:\Học-tập\Y4S1\FPT_Intern\FaceMaskDetection\MaskTheFace>python mask_the_face.py --path NewUser
--mask_type cloth --pattern masks/textures/check/check_5.jpg --color_weight 0 --verbose

MaskTheFace
----- Masking image files -----
Processing: NewUser/thuc_1.jpg
Faces found: 0
Processing: NewUser/thuc_10.jpg
Faces found: 1
Processing: NewUser/thuc_11.jpg
Faces found: 1
Processing: NewUser/thuc_12.jpg
Faces found: 1
```

## Kết quả thu được:

Trong quá trình chạy, sẽ có một số hình ảnh mà mô hình sẽ không nhận dạng được khuôn mặt nên sẽ không tiến hành đeo khẩu trang cho chúng, do đó ta có thể thấy output chỉ có 46 ảnh trong khi input có 80 ảnh.



### Tạo bộ dữ liệu vn-celeb hoàn chỉnh:

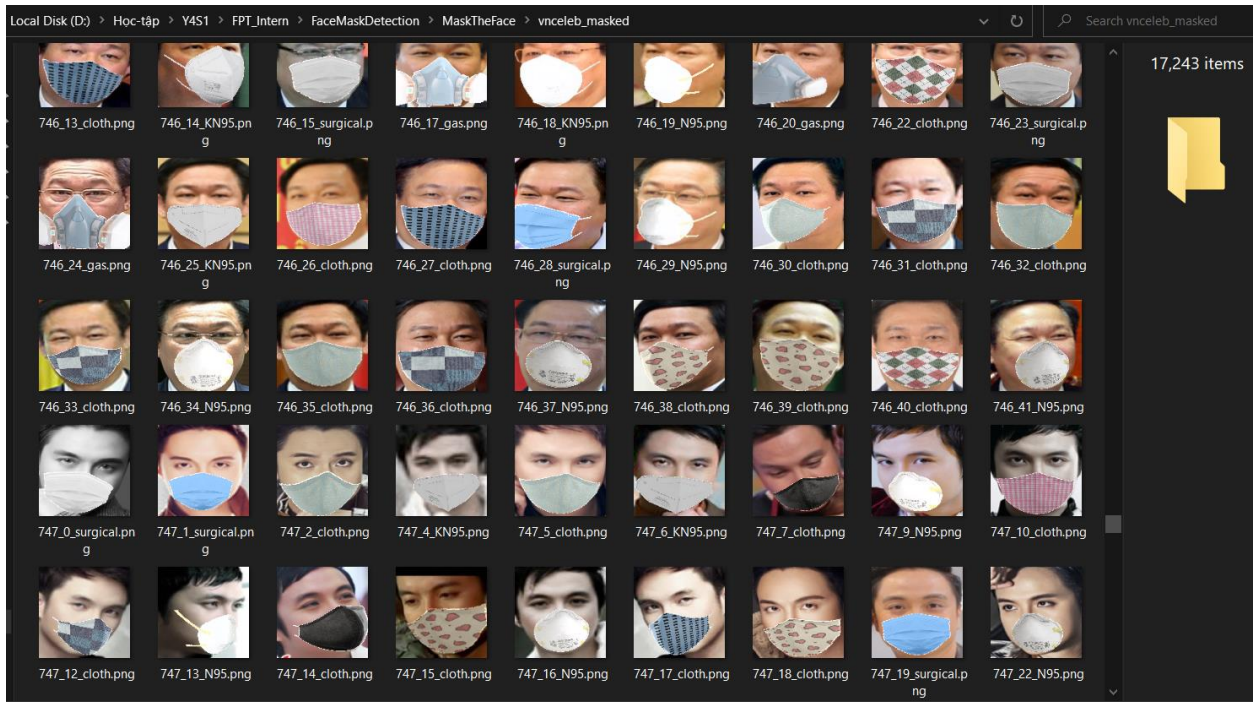
- Từ bộ dữ liệu gốc ban đầu, ta gán nhãn cho tập này là **without\_mask**.
  - Sử dụng phương pháp Mask The Face để tạo bộ dữ liệu **with\_mask**.
- ❖ Cấu hình các thông tin cho Mask The Face: Sử dụng phương thức “-code” với các loại khẩu trang khác nhau để có thể tạo ra các mẫu khẩu trang ngẫu nhiên khác nhau cho tập dữ liệu **with\_mask**.

```
(env_dlib) D:\Học-tập\Y4S1\FPT_Intern\FaceMaskDetection\MaskTheFace>
(env_dlib) D:\Học-tập\Y4S1\FPT_Intern\FaceMaskDetection\MaskTheFace>python mask_the_face.py --path vnceleb --code cloth-masks/textures/check/
check_4.jpg,surgical-#FFFFFF,surgical-#808080,KN95-#FFFFFF,KN95-#808080,N95-#FFFFFF,N95-#808080,gas-#FFFFFF,gas-#808080,surgical-#0473E2,clot
h-masks/textures/check/check_3.pngcloth-masks/textures/floral/floral_3.jpg,cloth-masks/textures/check/check_2.jpg,cloth-masks/textures/check/
check_1.png,cloth-masks/textures/check/check_5.jpg,cloth-masks/textures/fruits/bananas.png,cloth-masks/textures/others/heart_1.png

MaskTheFace
----- Masking image files -----
0%|  | 63/23105 [00:04<26:28, 14.50it/s]D:\Học-tập\Y4S1\FPT_In
tern\FaceMaskDetection\MaskTheFace\utils\fit_ellipse.py:38: RuntimeWarning: invalid value encountered in sqrt
  res1=np.sqrt(up/down1)
2%|  | 534/23105 [00:36<22:19, 16.85it/s]D:\Học-tập\Y4S1\FPT_In
tern\FaceMaskDetection\MaskTheFace\utils\fit_ellipse.py:39: RuntimeWarning: invalid value encountered in sqrt
  res2=np.sqrt(up/down2)
75%|  | 17406/23105 [25:21<06:25, 14.79it/s]D:\Học-tập\Y4S1\FPT_In
tern\FaceMaskDetection\MaskTheFace\utils\aux_functions.py:131: RankWarning: Polyfit may be poorly conditioned
  perp_line = fit_line(x, y, image)
100%|  | 23105/23105 [32:41<00:00, 11.78it/s]
----- Masking image directories -----
0it [00:00, ?it/s]
Processing Done
(env_dlib) D:\Học-tập\Y4S1\FPT_Intern\FaceMaskDetection\MaskTheFace>
```



## ❖ Kết quả thu được



➤ Tổng kết lại ta thu được tập dữ liệu **vn-celeb** với thông tin như sau:

- with\_mask: 17243 images
- without\_mask: 23105 images


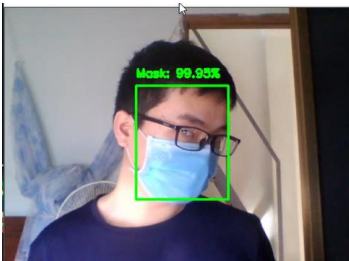
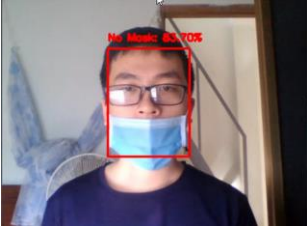
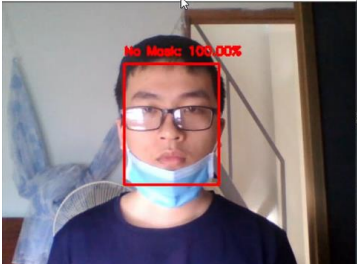
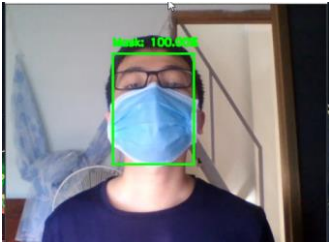
## VI. Train mô hình với dữ liệu mới

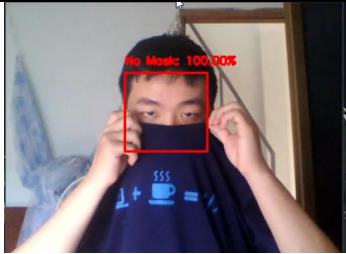

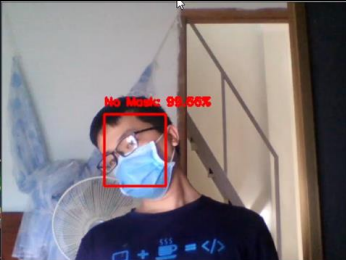
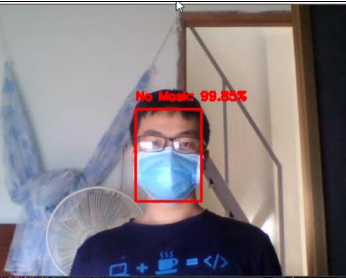
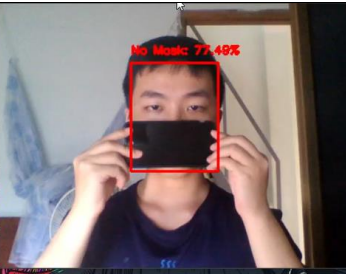
Ta sử dụng bộ dữ liệu vn-celeb thu được ở trên cho quá trình training.

```
Epoch 15/20
1009/1009 [=====] - 241s 239ms/step - loss: 9.7692e-05 - accuracy: 1.0000 - val_loss: 2.6268e-04 - val_accuracy: 0.9999
Epoch 16/20
1009/1009 [=====] - 246s 244ms/step - loss: 6.1288e-05 - accuracy: 1.0000 - val_loss: 2.5381e-04 - val_accuracy: 0.9998
Epoch 17/20
1009/1009 [=====] - 248s 245ms/step - loss: 1.0237e-04 - accuracy: 1.0000 - val_loss: 4.1307e-04 - val_accuracy: 0.9996
Epoch 18/20
1009/1009 [=====] - 276s 273ms/step - loss: 6.1666e-05 - accuracy: 1.0000 - val_loss: 8.4659e-04 - val_accuracy: 0.9996
Epoch 19/20
1009/1009 [=====] - 235s 232ms/step - loss: 2.7153e-05 - accuracy: 1.0000 - val_loss: 5.8815e-04 - val_accuracy: 0.9998
Epoch 20/20
1009/1009 [=====] - 273s 270ms/step - loss: 5.7318e-05 - accuracy: 1.0000 - val_loss: 8.5902e-04 - val_accuracy: 0.9996
```

Sau khi đã training xong, chạy thuật toán phát hiện đeo khẩu trang trên webcam để kiểm tra.



Không đeo Khẩu trang	Đeo khẩu trang	Không nhận diện được
		
		
		
		
		
		

## VII. Các trường hợp đeo khẩu trang không nghiêm túc

Để đảm bảo an toàn cho sức khỏe của chính mình và mọi người xung quanh thì việc đeo khẩu trang đúng cách rất quan trọng và không phải ai cũng đeo nghiêm túc. Để giải quyết vấn đề này ta cần liệt kê và bổ sung thêm các trường hợp đeo khẩu trang không nghiêm túc.



Ở đây ta sẽ làm 2 trường hợp là: **under nose** và **only nose**.

Với dataset ban đầu cùng thuật toán mask the face đã tìm hiểu, ta tiến hành điều chỉnh các thông số của mô hình để có thể đeo khẩu trang một cách “không nghiêm túc” nhằm làm data cho quá trình huấn luyện.

## Giới thiệu Dlib

Dlib là thư viện mã nguồn mở về Machine Learning, trong đó nổi tiếng nhất là Deep Learning. Thư viện Dlib cung cấp hai giải thuật cho bài toán nhận dạng khuôn mặt, một giải thuật dùng phương pháp học máy cổ điển và giải thuật mới hơn dùng deep learning. Phương pháp cổ điển có ưu điểm là cần ít phép tính toán hơn và do đó không yêu cầu phần cứng cấu hình cao, tuy nhiên độ chính xác thì kém hơn phương pháp dùng deep learning.



Ta sẽ dùng thuật toán **Facial landmark** xác định được những điểm chính tạo nên hình dạng của khuôn mặt đối tượng trong 1 bức ảnh. Facial landmark là đầu vào cho nhiều bài toán khác như dự đoán tư thế đầu, tráo đổi khuôn mặt, phát hiện nháy mắt, xoay chỉnh lại khuôn mặt và điển hình là công nghệ nhận dạng khuôn mặt FaceID được Apple trang bị trên iPhone X

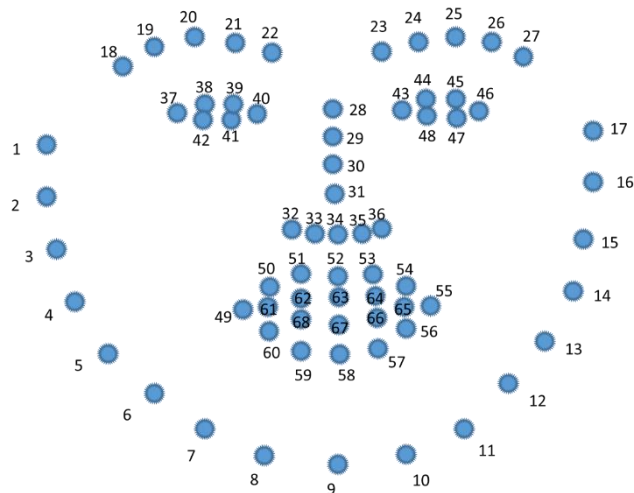
Việc xác định facial landmark gồm có hai bước:

- Bước 1: Xác định được vị trí khuôn mặt trong bức ảnh
- Bước 2: Xác định được các điểm tạo nên cấu trúc của khuôn mặt

Trong bước 1 ta sẽ dùng các thuật toán face detection để thu được một được xác định bởi tọa độ (x,y) bao quanh khuôn mặt trong bức ảnh.

Sau khi xác định được khuôn mặt trong bức ảnh, chúng ta sẽ xác định cấu trúc của khuôn mặt. Bộ xác định facial landmark này sẽ xác định 68 điểm chính theo tọa độ (x,y) cấu tạo nên khuôn mặt người như hình bên dưới.:

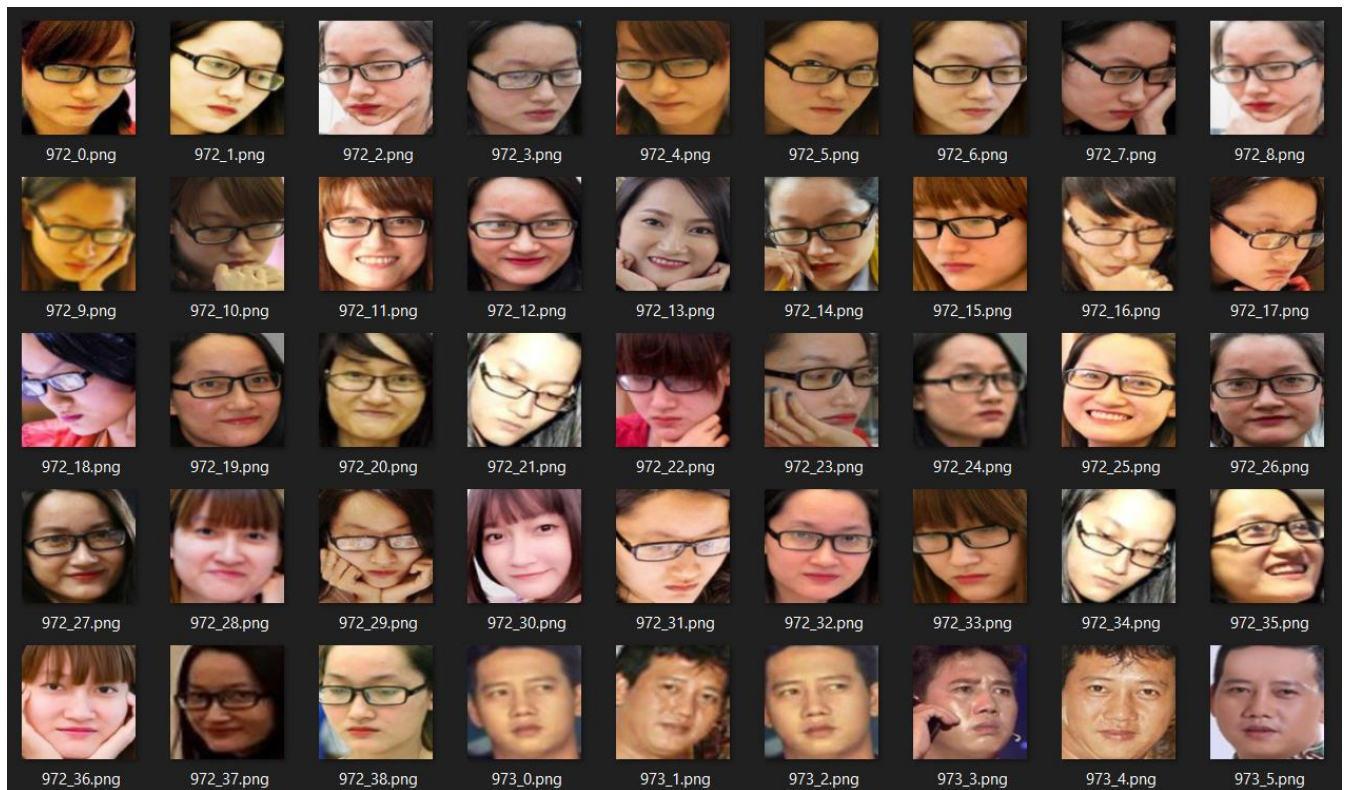
- Lông mày phải
- Lông mày trái
- Mắt phải
- Mắt trái
- Sống Mũi
- Đầu mũi
- Môi trên
- Môi dưới
- Cằm





Sau khi đã xác định được các điểm để cấu tạo nên khuôn mặt, ta sẽ ghép khẩu trang vào khuôn mặt bằng các điều chỉnh các thông số cho phù hợp với mỗi đối tượng. Để thể hiện các trường hợp đeo không nghiêm ngặt, ta sẽ thêm các hàm và chỉnh sửa các thông số nhằm tạo ra các dataset với các trường hợp đeo khẩu trang không đúng (under nose: dưới mũi và only nose: chỉ đeo khẩu trang ở mũi) và liệt kê chúng vào tập without\_mask (xem như không đeo khẩu trang).

Ví dụ một tập dữ liệu nhỏ:























Các dataset thu được:

Name	Date modified	Type
NewUser	12/15/2021 10:56 AM	File folder
NewUser_masked	12/15/2021 10:56 AM	File folder
NewUser_masked_only_nose	12/15/2021 10:56 AM	File folder
NewUser_masked_under_nose	12/15/2021 10:56 AM	File folder








Ví dụ:

Ảnh gốc	Masked	Under_nose	Ony_nose
			
			
			
			
			

Tiếp theo, ta sẽ ghép ngẫu nhiên các phương thức như: masked, under\_nose, only\_nose và unmasked vào dataset. Với tỷ lệ lần lượt là (3 : 1 : 1 : 1) ~ (8k : 3k : 3k : 3k )

Kết quả thu được

 vnceleb	12/17/2021 8:30 AM	File folder
 vnceleb_masked	12/19/2021 8:19 AM	File folder
 vnceleb_masked_only_nose	12/19/2021 8:19 AM	File folder
 vnceleb_masked_under_nose	12/19/2021 8:19 AM	File folder
 vnceleb_unmasked	12/19/2021 8:19 AM	File folder

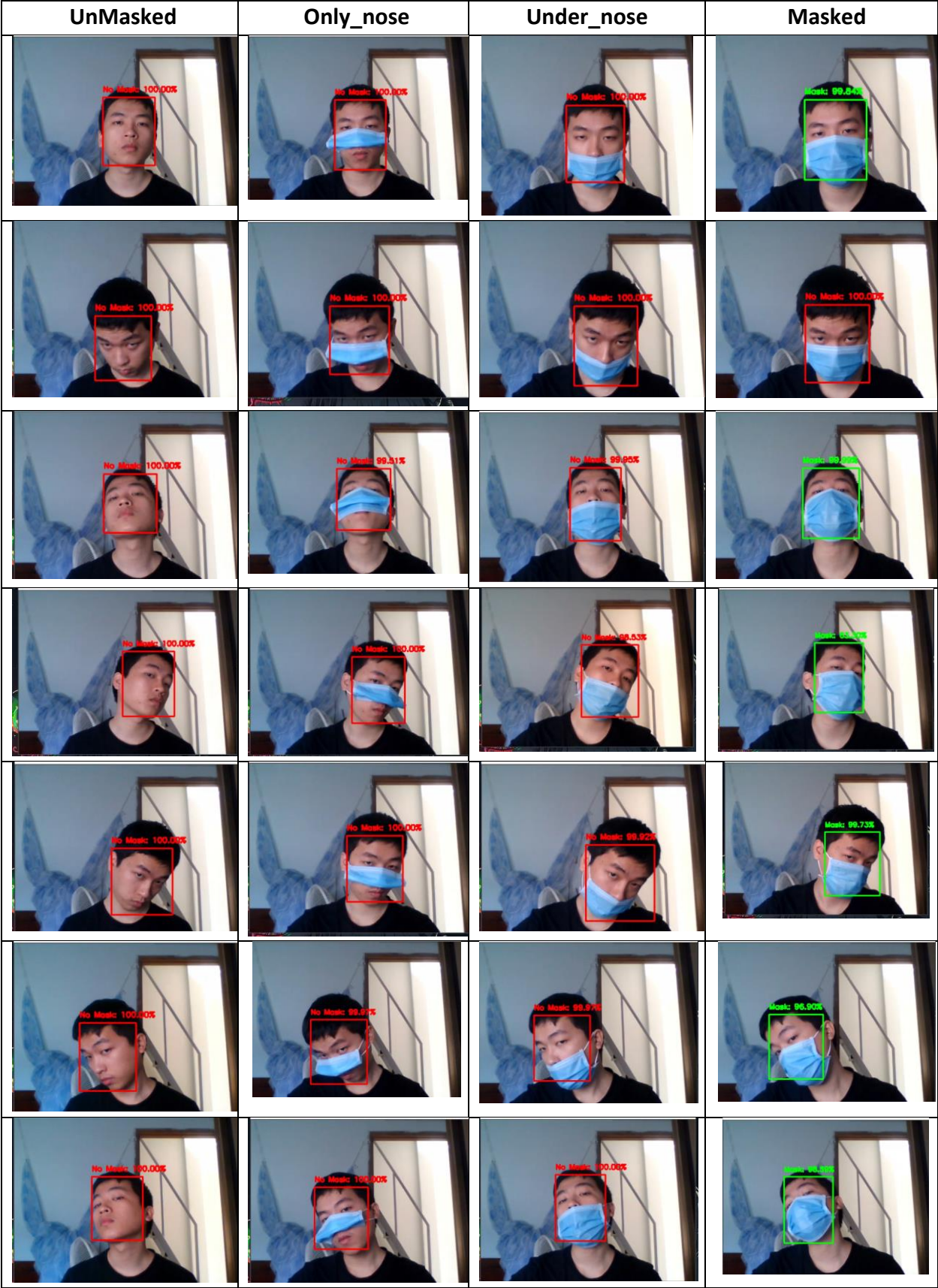
Sau đó ta gộp 3 phương thức (only\_nose, under\_nose và unmasked) thành tập without\_mask (xem như không đeo đúng cách) và tập masked thành tập with\_mask. Để tăng tính đa dạng của dữ liệu, ta tiến hành lặp lại ghép khẩu trang nhằm tạo một bộ các dữ liệu mới.

Kết quả tập dữ liệu:

- **with\_mask: khoảng 17k ảnh (của masked)**
- **without\_mask: khoảng 18k ảnh (của only\_nose, under\_nose và unmasked)**

Tiến hành train model cho bộ dữ liệu này để kiểm tra kết quả.

```
Epoch 15/20
896/896 [=====] - 197s 220ms/step - loss: 0.0347 - accuracy: 0.9886 - val_loss: 0.0394 - val_accuracy: 0.9856
Epoch 16/20
896/896 [=====] - 201s 224ms/step - loss: 0.0325 - accuracy: 0.9895 - val_loss: 0.0423 - val_accuracy: 0.9844
Epoch 17/20
896/896 [=====] - 199s 222ms/step - loss: 0.0302 - accuracy: 0.9900 - val_loss: 0.0368 - val_accuracy: 0.9865
Epoch 18/20
896/896 [=====] - 201s 224ms/step - loss: 0.0280 - accuracy: 0.9912 - val_loss: 0.0378 - val_accuracy: 0.9860
Epoch 19/20
896/896 [=====] - 197s 220ms/step - loss: 0.0268 - accuracy: 0.9912 - val_loss: 0.0360 - val_accuracy: 0.9866
Epoch 20/20
896/896 [=====] - 199s 222ms/step - loss: 0.0247 - accuracy: 0.9922 - val_loss: 0.0366 - val_accuracy: 0.9865
```



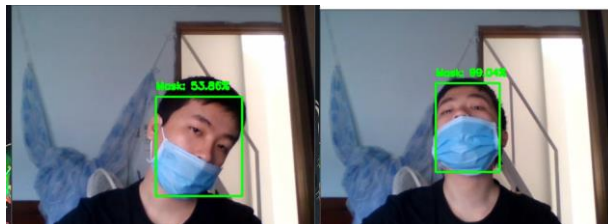
Một số ngoại lệ, lỗi cần chỉnh sửa

Đúng nhận thành sai:

- Mặt cúi thấp
- Có đeo kính
- Xa Camera



Sai nhận thành đúng:



## VIII. Multiple Labels

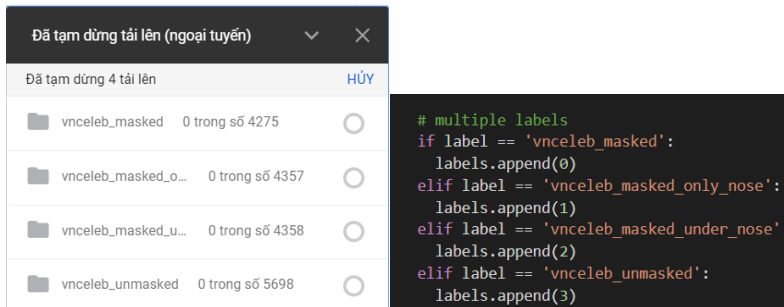
Để có thể xác định rõ từng trường hợp đeo khẩu trang, ta tiến hành huấn luyện mô hình từ việc chỉ phân biệt việc có hay không đeo khẩu trang thành các trường hợp cụ thể như sau:

- Đeo khẩu trang đúng cách
- Chỉ đeo khẩu trang ở mũi
- Đeo khẩu trang dưới mũi
- Không đeo khẩu trang

Dataset:

Từ tập dataset vnceleb, ta tiến hành ghép khẩu trang (mask the face) một cách ngẫu nhiên theo nhiều phương thức và chia chúng ngẫu nhiên vào các tập được chỉnh định (4 tập) với số lượng mẫu gần như nhau





Sau khi đã chia bộ dữ liệu xong, tiến hành cấu hình các thông số trong model.

## IX. Cải tiến mô hình

Để tăng độ chính xác cho mô hình, ta tiến hành các phương pháp cải tiến nhằm đem lại hiệu suất tốt, giảm sai số của mô hình.

Các Phương pháp cải tiến bao gồm:

- Thêm data thật
- Thay đổi model mạnh hơn
- Augmentation
- ...

### 1. Phương pháp 1: Thêm data thật

Ta xem xét các bộ data trong thực tế và áp dụng chúng vào mô hình để “học” thực tế.

#### Bộ data 1:

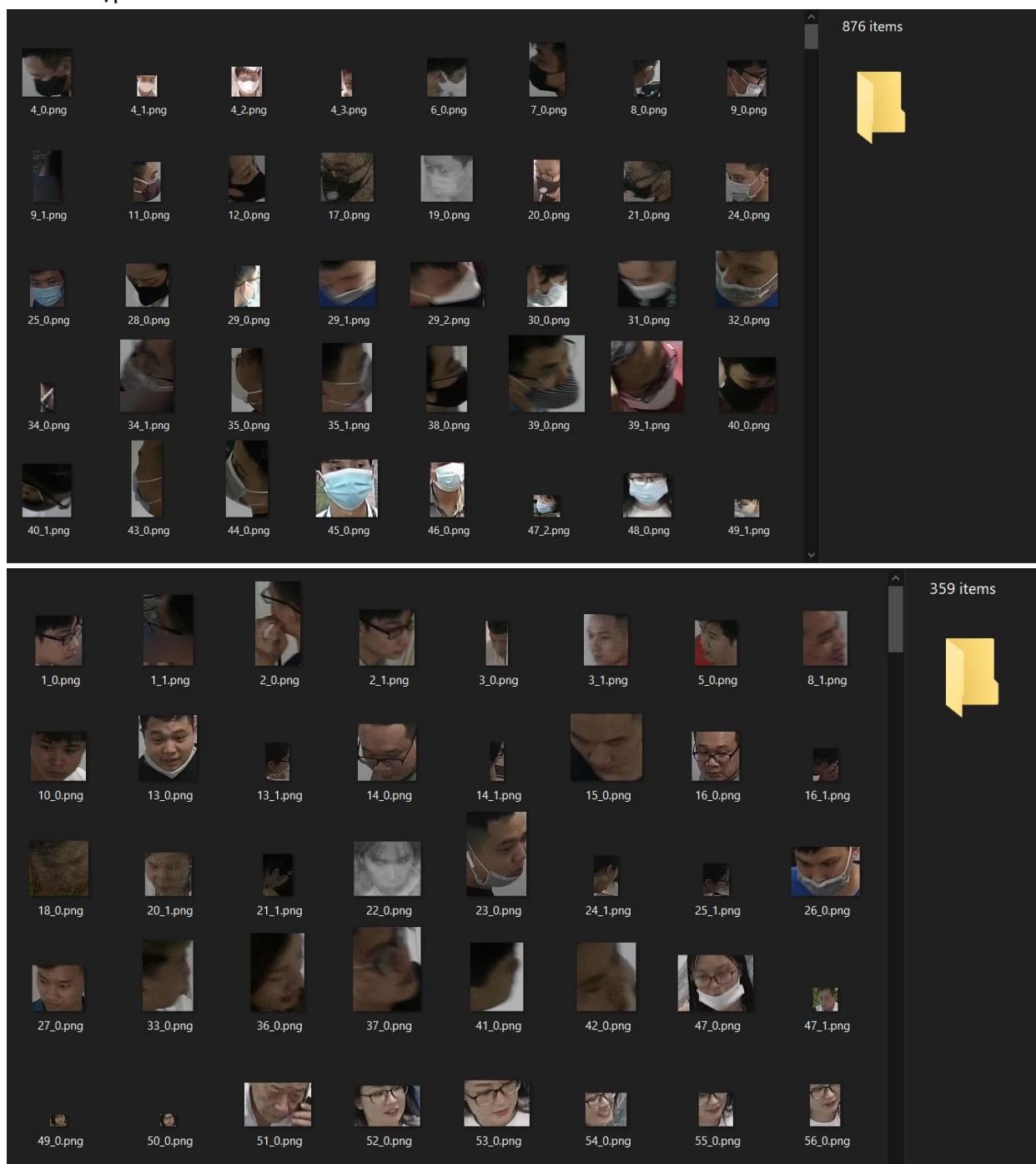
- Kích thước: 849 MB
- Gồm: file Images và Labels
  - Images: gồm 3 tập public\_test, train và val với các ảnh (jpg) được thu thập từ camera giám sát tại các vị trí phòng ban ở 1 công ty.
    - Public\_test: 88 ảnh
    - Train: 792 ảnh
    - Val: 184 ảnh
  - Labels: 3 tập public\_test, train và val với các tập tin (txt) lưu trữ labels của từng ảnh tương ứng ở thư mục images.
- Ảnh minh họa



Đối với dataset này, ta tiến hành rút trích các khuôn mặt và gán label. Với các thông số box và label cho các ảnh có sẵn. Dùng một số thuật toán của OpenCV để cắt khuôn mặt theo box và tiến hành gán label cho các ảnh đó.

Kết quả:

- Tập Mask: 876 ảnh
- Tập No Mask: 359 ảnh





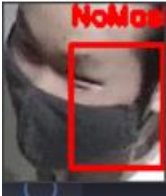


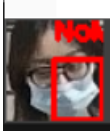
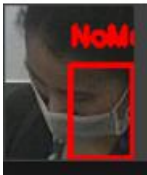
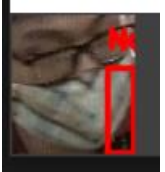
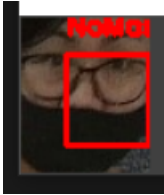
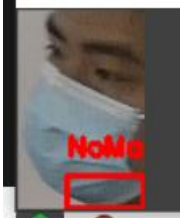
**Test Mô hình cho bộ dữ liệu FPT:**

Tổng ảnh Input	876 (Mask) + 359 (NoMask) = 1235
Số lượng ảnh nhận diện được	125 (Mask) + 95 (NoMask) = 220
Số lượng ảnh không nhận diện được	352 (Mask) + 201 (NoMask) = 553
Số lượng ảnh đeo khẩu trang được nhận diện đúng label	1
Số lượng ảnh không đeo khẩu trang được nhận diện đúng label	95
Độ chính xác	$(1 + 95) / 220$ = 0.4364

**Các giá trị không nhận diện được:**

<b>Mask</b>	 ảnh bị che, góc từ phía sau	 Góc chụp từ phía sau	 Thiếu sáng	 Thiếu sáng, mất thông tin
	 Góc nghiêng	 Thiếu sáng	 Thiếu sáng, góc từ phía sau	
<b>NoMask</b>	 Góc cúi đầu	 Độ phân giải thấp, bị cắt hình	 Nghiêng, cúi đầu	 ảnh mờ, đen trắng
	 Nghiêng, cúi đầu	 ảnh mờ, đen trắng		

Các giá trị nhận diện sai:

Mask				
				
				
No Mask	Nhận diện được 95/95 ảnh detect được			

**Bộ data 2:** [mirror] zalo ai challenge 2021 - 5k, tập data được lấy từ cuộc thi Zalo AI challenge 2021.

- Kích thước: 573 MB
- Gồm: public\_test, sample\_submission, train, mask.tflite
  - public\_test: 542 ảnh các hoạt động được tổ chức trong đại dịch covid. Với file csv đánh số thứ tự ảnh.
  - sample\_submission: 3 submission mẫu, các ảnh được phân loại theo 0 hoặc 1.
  - Train: gồm 4175 ảnh.
  - File mask.tflite



- Ảnh minh họa

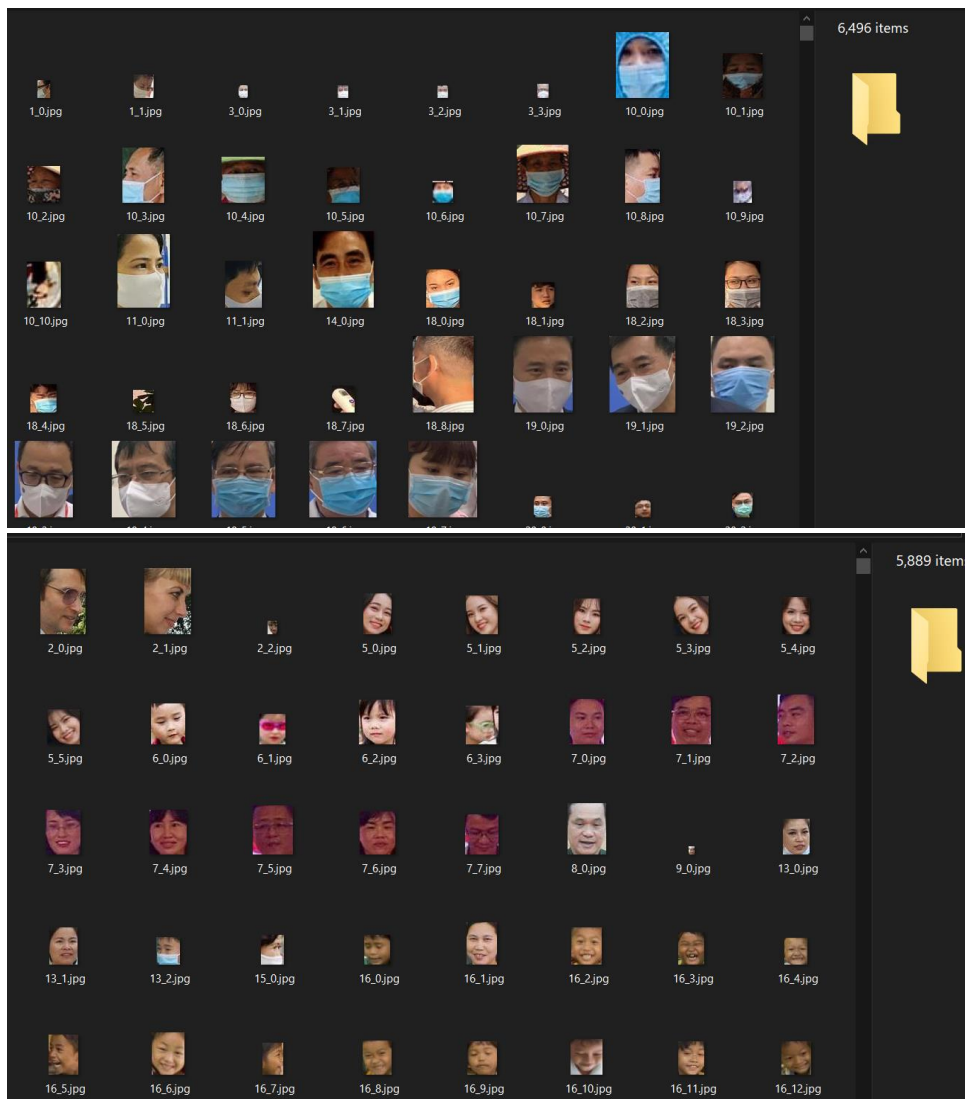


Đối với dataset này, ta tiến hành rút trích các khuôn mặt và gán label cho chúng.

Dùng một số thuật toán của OpenCV, MTCNN để xác định và rút trích khuôn mặt trong ảnh và gán label cho chúng.

Kết quả:

- Tập Mask: 6496 ảnh
- Tập No Mask: 5889 ảnh





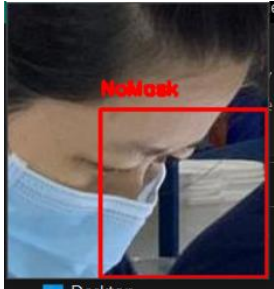
Tổng ảnh Input	6496 (Mask) + 5889 (NoMask) = 12385
Số lượng ảnh nhận diện được	472 (Mask) + 386 (NoMask) = 858
Số lượng ảnh không nhận diện được	4290 (Mask) + 3703 (NoMask) = 7993
Số lượng ảnh đeo khẩu trang được nhận diện đúng label	4
Số lượng ảnh không đeo khẩu trang được nhận diện đúng label	386
Độ chính xác	$(4 + 386) / 858$ = 0.4545



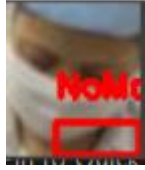

**Các giá trị không nhận diện được:**

<b>Mask</b>	 Góc nghiêng			
	 Ảnh mờ		 Bị vật cản che	
		 Độ phân giải thấp	 Góc nghiêng	
<b>NoMask</b>		 Góc nghiêng	 Độ phân giải thấp	

				 Độ phân giải thấp
	 Độ phân giải thấp, góc nghiêng		 góc nghiêng	 góc nghiêng

Các giá trị nhận diện sai:

Mask				
				
				

				
NoMask	Nhận diện đúng 386/386 ảnh			

## X. Kết luận

Qua đề tài tìm hiểu lần này, em đã học được cách để train một model face mask detection cũng như tìm hiểu được các cách làm tăng, đa dạng bộ dữ liệu cho quá trình huấn luyện.

Tuy nhiên, trong quá trình tìm hiểu cũng đã gặp nhiều khó khăn do vấn đề phần cứng, kiến thức nhưng cũng đã được anh Khang hướng dẫn khắc phục.

## XI. Danh mục tham khảo

- [1] <https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset>
- [2] <https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset>
- [3] <https://www.sciencedirect.com/sdfe/reader/pii/S1532046421001775/pdf>