**ĐẠI HỌC ĐÀ NẴNG**

**KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**BÁO CÁO ĐỒ ÁN CƠ SỞ**

**ĐỀ TÀI: NHẬN DIỆN NGƯỜI ĐEO KHẨU TRANG SỬ DỤNG OPENCV VÀ TENSORFLOW/KERAS.**

Sinh viên thực hiện : **TRƯƠNG HỮU MẪN– 17IT075**

Lớp : **17IT1**

Giảng viên hướng dẫn : **THS. NGUYỄN VĂN BÌNH**

***Đà nẵng, tháng 07 năm 2020***

**ĐẠI HỌC ĐÀ NẴNG**

**KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**BÁO CÁO ĐỒ ÁN CƠ SỞ**

**NHẬN DIỆN NGƯỜI ĐEO KHẨU TRANG**

**SỬ DỤNG OPENCV VÀ TENSORFLOW/KERAS**

***Đà Nẵng, tháng 07 năm 2020***

**LỜI MỞ ĐẦU**

Hiện nay, trí tuệ nhân tạo (Artifical Intelligent) hay học máy (Machine Learning) là một lĩnh vực đang được nhắc đến khá nhiều trong thời gian gần đây bởi tính ứng dụng của nó trong thực tiễn. Có rất nhiều ứng dụng của công nghệ học máy đã được áp dụng vào cuộc sống hằng ngày như: google dịch, xe ô tô tự hành, quảng cáo gợi ý mua hàng, hệ thống nhận diện… Đặc biệt, lĩnh vực đáng quan tâm nhất hiện nay đó là “nhận dạng”. Đối tượng cho việc nghiên cứu nhận dạng cực kì phong phú và đa dạng. Cho nên trong khuôn khổ bài tập lớn lần này, nhóm chúng em quyết định chọn đối tượng “khuôn mặt con người” để tiến hành nghiên cứu.

Bên cạnh đó, hiện nay trên thế giới, tình hình dịch bệnh đang có chiều hướng gia tăng. Để đảm bảo an toàn cho người dân, một số thành phố đã ban hành qui định về việc đeo khẩu trang ở nơi công cộng. Và để phát hiện người vi phạm quy định trong thời gian dịch bệnh thì cũng phải tốn rất nhiều sức người để có thể rà soát và túc trực. Đó cũng là lí do để nhóm chọn đề tài: “NHẬN DIỆN NGƯỜI ĐEO KHẨU TRANG SỬ DỤNG OPENCV VÀ TENSORFLOW/KERAS.” Nhằm giảm sức người và đảm bảo phát hiện nhanh kịp thời các trường hợp vi phạm quy định.

**LỜI CẢM ƠN**

Trong thời gian qua, dưới sự hướng dẫn nhiệt tình của THS.NGUYỄN VĂN BÌNH, nhóm đã hoàn thành được ứng dụng “NHẬN DIỆN NGƯỜI ĐEO KHẨU TRANG SỬ DỤNG OPENCV VÀ TENSORFLOW/KERAS” từ đó có cơ hội được tìm hiểu và tiếp thu được rất nhiều kiến thức.

Do thời gian và kiến thức có hạn nên đề tài không tránh khỏi những thiếu sót, chúng em rất mong nhận được lời góp ý từ Thầy (Cô) giáo và các bạn. Nhóm xin bày tỏ lòng biết ơn sâu sắc đến thầy vì đã giúp chúng em hoàn thành đề tài lần này cũng như hoàn thiện được nội dung của bài báo cáo.

Chúng em chân thành cảm ơn!

**NHẬN XÉT**

**(của giáo viên hướng dẫn)**

................................................................................................................................................................................

................................................................................................................................................................................

................................................................................................................................................................................

................................................................................................................................................................................

................................................................................................................................................................................

................................................................................................................................................................................

................................................................................................................................................................................

................................................................................................................................................................................

................................................................................................................................................................................

................................................................................................................................................................................

................................................................................................................................................................................

................................................................................................................................................................................

................................................................................................................................................................................

................................................................................................................................................................................

................................................................................................................................................................................

*Đà Nẵng, tháng 07 năm 2020*

**Giảng viên hướng dẫn**

*THS. NGUYỄN VĂN BÌNH*

**MỤC LỤC**

**Chương 1. MỞ ĐẦU .............................................................................................................. 9**

1.1 Bối cảnh, lý do thực hiện đề tài ............................................................................................ 9

1.2 Bài toán đặt ra .............................................................................................................................. 9

**Chương 2. CƠ SỞ LÝ THUYẾT........................................................................................ 10**

2.1 Thư viện dlib.............................................................................................................................. 10

2.1.1 Giới thiệu............................................................................................................................. 10

2.1.2 Ứng dụng............................................................................................................................. 10

2.2 Thư viện OpenCV ..................................................................................................................... 11

2.2.1 Giới thiệu............................................................................................................................. 11

2.2.2 Ứng dụng............................................................................................................................. 11

2.3 Python........................................................................................................................................... 11

2.3.1 Giới thiệu............................................................................................................................. 11

2.4 Tensorflow ................................................................................................................................. 12

2.4.1 Giới thiệu............................................................................................................................. 12

2.4.2 Một số khái niệm cơ bản trong Tensorflow ........................................................ 12

**Chương 3. XÂY DỰNG MÔ HÌNH THỰC TẾ ................................................................ 13**

3.1 Training Face Mask Detector ............................................................................................. 13

3.1.1 Tập dữ liệu train (dataset) .......................................................................................... 14

3.1.2 Quá trình hình thành tập dữ liệu train (dataset) .............................................. 14

3.2 Apply Face Mask Detector ................................................................................................... 18

**Chương 4. TRIỂN KHAI XÂY DỰNG ............................................................................. 19**

4.1 Môi trường thực hiện dự án: .............................................................................................. 19

4.2 Cấu trúc dự án ........................................................................................................................... 19

4.3 Load tập dữ liệu train ............................................................................................................ 20

4.4 Thực hiện phát hiện người đeo khẩu trang trong ảnh tĩnh .................................. 24

4.5 Thực hiện phát hiện người đeo khẩu trang theo thời gian thực ........................ 27

4.6 Kết quả thực hiện .................................................................................................................... 31

4.6.1 Phát hiện người đeo khẩu trang trong ảnh tĩnh ................................................ 31

4.6.2 Phát hiện người đeo khẩu trang theo thời gian thực ...................................... 31

**Chương 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ...................................................... 32**

5.1 Kết luận ........................................................................................................................................ 32

5.2 Hướng phát triển ..................................................................................................................... 32

**DANH MỤC HÌNH**

Hình 1: Nhận diện người đeo khẩu trang ......................................................................................... 9

Hình 2: Tọa độ 68 điểm trên khuôn mặt từ bộ dữ liệu iBUG 300-W ................................ 10

Hình 3: Thư viện OpenCV ..................................................................................................................... 11

Hình 4: Ngôn ngữ Python ..................................................................................................................... 11

Hình 5: Tensorflow .................................................................................................................................. 12

Hình 6: Mô hình ứng dụng nhận diện người đeo khẩu trang sử dụng OpenCV và

Tensorflow/Keras.................................................................................................................................... 13

Hình 7: Tập dữ liệu train....................................................................................................................... 14

Hình 8: Hình ảnh người không đeo khẩu trang. ......................................................................... 15

Hình 9: Áp dụng tính năng nhận diện khuôn mặt...................................................................... 15

Hình 10: Trích xuất vùng quan tâm ................................................................................................. 16

Hình 11: Áp dụng Facial LandMark.................................................................................................. 16

Hình 12: Hình ảnh của chiếc khẩu trang........................................................................................ 17

Hình 13: Kết quả cuối cùng. ................................................................................................................. 17

Hình 14: Tập dữ liệu train nhân tạo. ............................................................................................... 17

Hình 15: with\_mask................................................................................................................................. 18

Hình 16: without\_mask.......................................................................................................................... 18

Hình 17: Cấu trúc dự án ........................................................................................................................ 19

Hình 18: Một số thư viện cần dùng trong train dataset .......................................................... 20

**DANH MỤC CỤM TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **TỪ VIẾT TẮT** | **Ý NGHĨA** |
| SVM | Support Vector Machine |
| OpenCV | Open Computer Vision |
| FC | Fully-connected |
| ROI | Region Of Interest |
| faceROIs | Face Region Of Interests |
|  |  |

**Chương 1. MỞ ĐẦU**

**1.1 Bối cảnh, lý do thực hiện đề tài**

- Hiện nay, tình hình dịch bệnh Covid-19 trên thế giới vẫn đang diễn biến phức tạp, với số ca mắc và số ca tử vong cao ở nhiều nước trên thế giới. Vì vậy, để giúp các lực lượng chức năng giảm bớt gánh nặng trong việc giám sát an ninh góp phần vào việc ngăn chặn đại dịch Covid-19. Nhóm chúng em đã tìm hiểu và xây dựng một ứng dụng giúp nhận diện người đeo khẩu trang theo thời gian thực.

**1.2 Bài toán đặt ra**

- Đầu vào : Hình ảnh cần phát hiện/nhận diện.

- Đầu ra : Ảnh kết quả sẽ khoanh vùng khuôn mặt đeo khẩu trang đeo khẩu trang bằng ô vuông xanh, ngược lại không đeo khẩu trang thì sẽ khoanh vùng bằng ô vuông đỏ.

- Đối với bài toán thực tế này bài toán sẽ được chia làm 2 bài toán nhỏ hơn:

1. Bài toán xác định (Detection): Xử dụng OpenCV/dlib thực hiện các biện pháp như giảm nhiễu, tách biên,... Từ đó xác định được vị trí của vật thể trong ảnh nhằm làm đầu vào cho bài toán tiếp theo.

2. Bài toán nhận dạng (Classification): Sử dụng các mô hình Deep Learning hoặc SVM (Support Vector Machine) để xác định/ nhận dạng các vật thể vừa được xác định ở bài toán đầu tiên nhằm đưa ra kết quả cuối cùng.

- Trong khuôn khổ môn học này, chúng em sẽ sử dụng thư viện OpenCV,dlib để

thực hiện bài toán đầu tiên, và sử dụng Tensorflow/Keras để xây dựng mạng



Deep Learning cho bài toán thứ 2.

*Hình 1: Nhận diện người đeo khẩu trang*

**Chương 2. CƠ SỞ LÝ THUYẾT**

**2.1 Thư viện dlib**

**2.1.1 Giới thiệu**

Thư viện Dlib được viết bằng ngôn ngữ C++ do ***Davis King*** tạo ra vào năm 2012 và được sử dụng rộng rãi trong lĩnh vực Computer Vision đặc biệt là nhận dạng đối tượng và nhận dạng khuôn mặt.

**2.1.2 Ứng dụng**

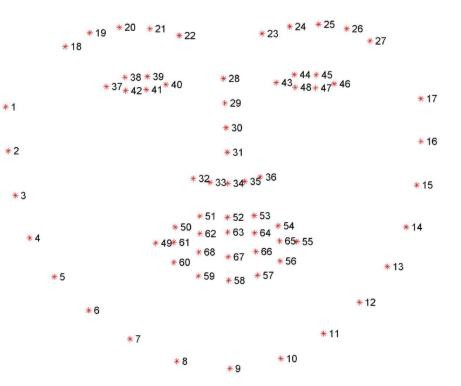
Sử dụng dlib để ước tính vị trí 68 điểm chính theo tọa độ (x,y) tạo nên khuôn mặt người như hình dưới đây.

Để có thể xác định được 68 điểm này trên khuôn mặt người, bộ xác

định facial landmark của dlib được huấn luyện với bộ dữ liệu iBUG 300-W.

Càng nhiều điểm landmard được xác định thì khuôn mặt người càng cụ

thể, rõ ràng và chính xác hơn.



*Hình 2: Tọa độ 68 điểm trên khuôn mặt từ bộ dữ liệu iBUG 300-W*

Trong khuôn khổ đề tài lần này, nhóm em sử dụng Ficial LandMark 68

điểm để trích xuất các vùng cụ thể của khuôn mặt, áp dụng căn chỉnh từ đó

xây dựng ứng dụng nhận diện người đeo khẩu trang.

**2.2 Thư viện OpenCV**

**2.2.1 Giới thiệu**

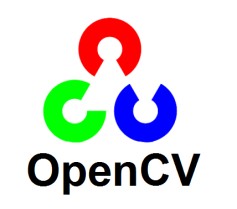
OpenCV (Open Computer Vision) là một thư viện mã nguồn mở hàng đầu cho xử lý về thị giác máy tính, machine learning, xử lý ảnh. OpenCV được viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực. OpenCV có các interface giành cho C/C++, Python và Java vì vậy nó hỗ trợ được cho cả Windows, Linux, MacOS thậm chí cả Android và Ios.

**2.2.2 Ứng dụng**

OpenCV có rất nhiều ứng dụng nhưng chủ yếu nó được áp dụng vào các công việc sau:

Nhận dạng ảnh. Xử lý ảnh.

Phục hồi ảnh/video. Thực tế ảo.



**2.3 Python**

**2.3.1 Giới thiệu**

*Hình 3: Thư viện OpenCV*

Python là ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, được phát triển bởi ***Guido van Rossum***, nó trở nên phổ biến rất nhanh, chủ yếu bởi có ưu điểm mạnh về dễ đọc, dễ học và dễ nhớ. Python cũng là

ngôn ngữ có hình thức rất “sáng sủa”, cấu trúc rõ ràng và thuận tiện cho người mới lập trình.

*Hình 4: Ngôn ngữ Python*



**2.4 Tensorflow**

**2.4.1 Giới thiệu**

Nói một cách dễ hiểu thì Tensorflow là một thư viện mã nguồn mở cung cấp khả năng xử lý tính toán số học dựa trên biểu đồ mô tả sự thay đổi của dữ liệu.

**2.4.2 Một số khái niệm cơ bản trong Tensorflow**

**Node:**

Vì Tensorflow mô tả lại dòng chảy của dữ liệu thông qua graph nên mỗi một điểm giao cắt trong graph thì được gọi là Node. Tại sao điều này quan trọng thì là vì các Node chính là điểm đại diện cho việc thay đổi của dữ liệu nên việc lưu trữ lại tham chiếu của các Node này là rất quan trọng.

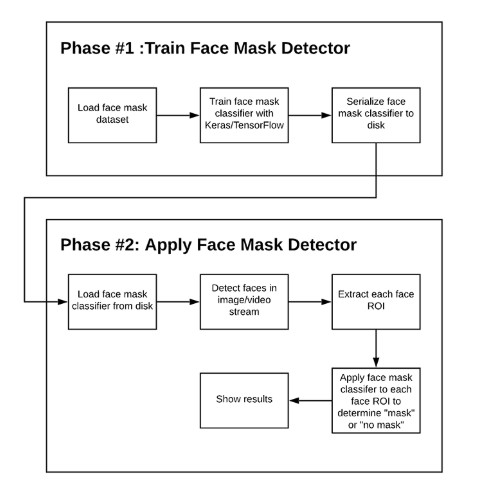
**Tensor:**

Để giải quyết được các bài toán Machine Learning, cần phải làm cho máy tính hiểu được dữ liệu của tập nguồn và dữ liệu của tập đích. Tensorflow cung cấp một loại dữ liệu mới được gọi là *Tensor*. Trong “thế giới” của Tensorflow, mọi kiểu dữ liệu đều quy về một mối được gọi là Tensor hay trong Tensorflow, tất cả dữ liệu đề là Tensor. Vậy nên có thể hiểu Tensorflow chính là một thư viện có chức năng mô tả, điều chỉnh dòng chảy của Tensor.



*Hình 5: Tensorflow*

**Chương 3. XÂY DỰNG MÔ HÌNH THỰC TẾ**



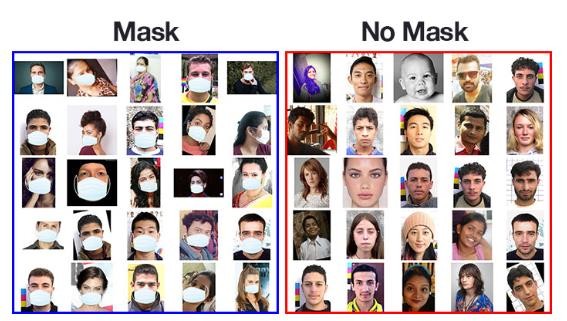
*Hình 6: Mô hình ứng dụng nhận diện người đeo khẩu trang sử dụng OpenCV và Tensorflow/Keras*

Để train dữ liệu cho ứng dụng nhận diện này, chúng ta chia dự án thành hai

giai đoạn riêng biệt, mỗi giai đoạn có các bước phụ tương ứng như hình trên.

**3.1 Training Face Mask Detector**

Ở giai đoạn này, nhóm sẽ tập trung vào việc tải tập dữ liệu (dataset) từ ổ đĩa. Sau đó tiến hành train mô hình (model) sử dụng Tensorflow/Keras trên tập dữ liệu này. Cuối cùng tiến hành lưu file model vào ổ đĩa nhằm phục vụ cho giai đoạn tiếp theo.



**3.1.1 Tập dữ liệu train (dataset)**

*Hình 7: Tập dữ liệu train.*

* Tập dữ liệu mà chúng ta sử dụng trong dự án này được tạo ra bởi

Prajna Bhandary. Bao gồm 1,376 hình ảnh chia làm 2 nhóm:

1. With\_mask: 690 hình ảnh.

2. Without\_mask: 686 hình ảnh.

**3.1.2 Quá trình hình thành tập dữ liệu train (dataset)**

* Để tạo ra tập dữ liệu này, Prajna tiến hành hai giai đoạn chính:

1. Thu thập ảnh chụp bình thường của khuôn mặt.

2. Sau đó, viết script Python để thêm khẩu trang cho những khuôn mặt ở giai đoạn 1. Từ đó tạo ra được bộ dữ liệu nhân tạo (nhưng vẫn có thể áp dụng trong thế giới thực).

* Việc áp dụng Facial LandMark vào giai đoạn 2 sẽ giúp việc tạo ra bộ dữ liệu nhân tạo dễ dàng hơn.
* Facial LandMark như đã được đề cập ở **CHƯƠNG 2**, nó tự động xác định vị trí các bộ phận trên khuôn mặt như:

o Mắt

o Lông mày

o Mũi

o Miệng

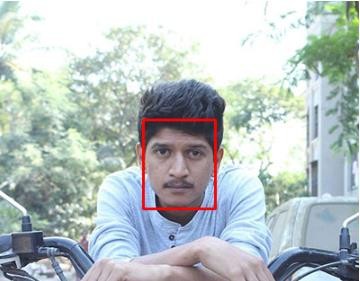
o Hàm

* Sử dụng Facial LandMark để xây dựng tập dữ liệu về khuôn mặt người đeo khẩu trang, trước tiên chúng ta cần bắt đầu với một hình ảnh người không đeo khẩu trang.



*Hình 8: Hình ảnh người không đeo khẩu trang.*

* Từ đó, áp dụng tính năng nhận diện khuôn mặt để tính toán vị trí khung giới hạn khuôn mặt trên ảnh.



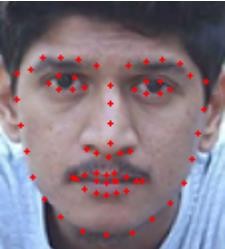
*Hình 9: Áp dụng tính năng nhận diện khuôn mặt.*

* Khi đã biết được vị trí khuôn mặt trên bức ảnh, chúng ta sẽ tiến hành trích xuất vùng quan tâm (Region of Interest - ROI)



*Hình 10: Trích xuất vùng quan tâm*

* Và từ đây, áp dụng Facial LandMark để xác định vị trí của mắt, mũi, miệng, hàm trên khuôn mặt.



*Hình 11: Áp dụng Facial LandMark*

* Tiếp theo, chúng ta cần hình ảnh của một chiếc khẩu trang (nền trong suốt – transparent background) như dưới đây:



*Hình 12: Hình ảnh của chiếc khẩu trang.*

Chiếc khẩu trang này sẽ tự động ghép với khuôn mặt bằng cách sử dụng facial landmark (cụ thể là các điểm dọc theo cằm – hàm và mũi) để tính toán vị trí đặt khẩu trang



*Hình 13: Kết quả cuối cùng.*

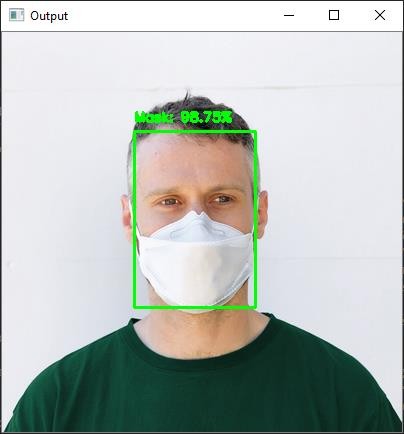
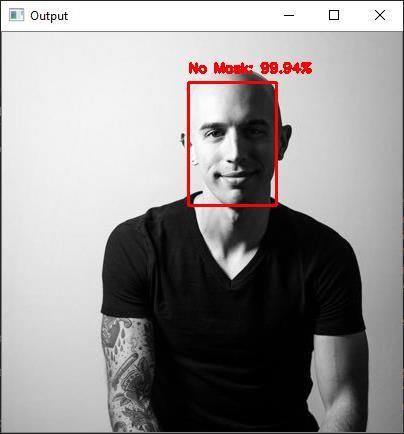
* Lặp lại quá trình này nhiều lần với nhiều ảnh đầu vào. Từ đó có thể tạo ra tập dữ liệu “khuôn mặt người đeo khẩu trang” nhân tạo.



*Hình 14: Tập dữ liệu train nhân tạo.*

**3.2 Apply Face Mask Detector**

Sau khi đã train tập dữ liệu, chúng ta có thể tiếp tục tiến đến bước load file model, phát hiện khuôn mặt và cuối cùng là phân loại từng khuôn mặt là with\_mask hay without\_mark.



*Hình 15: with\_mask Hình 16: without\_mask*

**Chương 4. TRIỂN KHAI XÂY DỰNG**

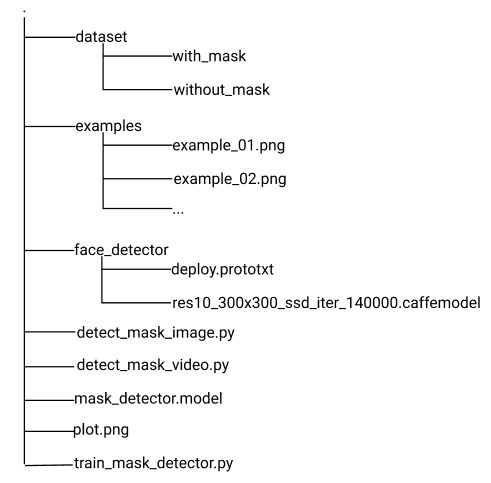
**4.1 Môi trường thực hiện dự án:**

1. Windows 10.

2. Python version 3.7

3. Pip version 19.2

**4.2 Cấu trúc dự án**



*Hình 17: Cấu trúc dự án*

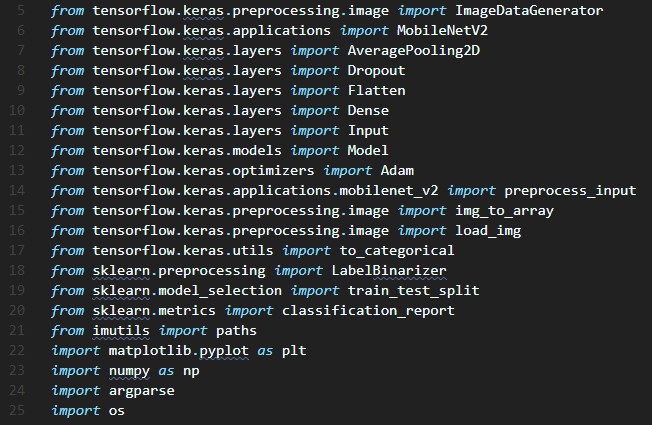
- Thư mục *dataset/* chứa dữ liệu được mô tả trong phần Training Face Mask

Detector.

- Một số hình ảnh trong thư mục *example/* được sử dụng trong việc chạy thử ứng dụng phát hiện người đeo khẩu trang trong ảnh tĩnh.

- Tiếp theo là 3 file Python script:

1. Train\_mask\_detector.py: Load tập dữ liệu train và cuối cùng tạo ra file mask\_detector.model



|  |  |  |
| --- | --- | --- |
|  | 2. | Detect\_mask\_image.py: Thực hiện phát hiện người đeo khẩu trang |
|  | trong ảnh tĩnh. |
| 3. | Detect\_mask\_video.py: Sử dụng webcam, phát hiện người đeo khẩu |
| **4.3** |  | trang cho mọi frame trong stream.  **Load tập dữ liệu train** |

Trong file ***train\_mask\_detector.py***.

*Hình 18: Một số thư viện cần dùng trong train dataset*

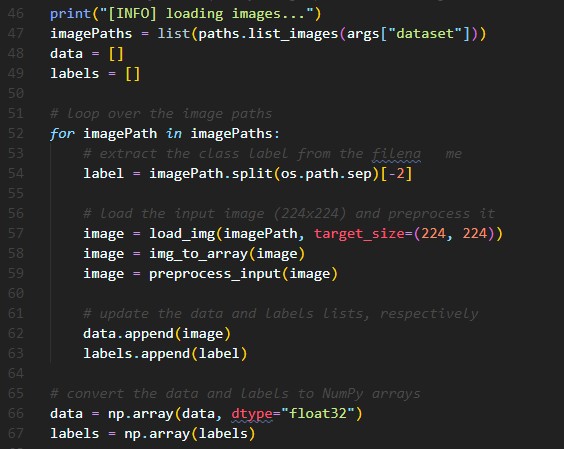
- Import một số thư viện cần thiết:

1. Một tập các import từ ***tensorflow.keras*** cho phép:

* Data augmentation ( Tổng hợp data mới bằng cách sửa đổi data hiện có mà target không thay đổi, hoặc được thay đổi theo cách đã biết.)
* Load trình phân loại MobilNetV2
* Building fully-connected (FC) head
* Pre-processing
* Loading image data

2. Sử dụng ***scikit-learn(sklearn)*** để tạo label cho lớp, phân đoạn dữ liệu và in report phân loại.

3. Thư viện ***imutils*** giúp tìm kiếm và liệt kê hình ảnh trong tập dữ liệu train. Và sử dụng ***matplotlib*** để vẽ đồ thị đường cong thể hiện lịch sử train.(tỉ lệ accuracy/loss)



- Lấy tất cả ***imagePaths*** trong tập dữ liệu (dataset) – L47

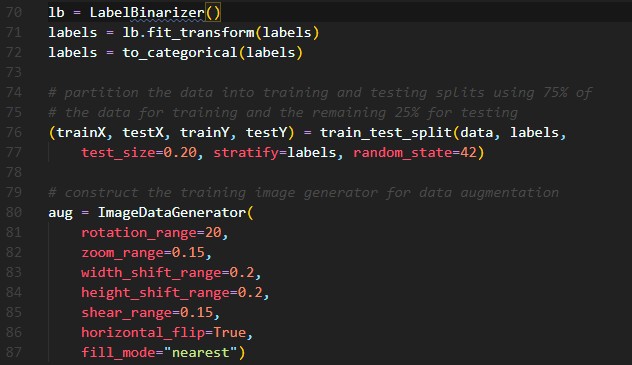
- Khởi tạo 2 list có tên là ***data*** và ***labels*** – L48

- Thực hiện vòng lặp trên ***imagePath*** sau đó load và tiền xử lý ảnh (L52-63).

Tiền xử lý ảnh bao gồm re-size ảnh thành 224x224 pixel, chuyển sang định dạng mảng, chia tỉ lệ cường độ điểm ảnh từ ảnh đầu vào thành [-1,-1] thông qua ***preprocess\_input*** function.

- Lần lượt đẩy ảnh đã qua tiền xử lý và label liên quan vào list data và labels. (L62-63)

- Chuyển đổi ***data*** và ***labels list*** sang NumPy arrays để đảm bảo dữ liệu train ở định dạng đúng. (L66-67)

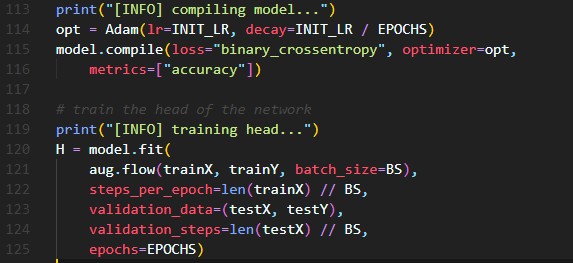


- Mã hóa các labels, phân vùng dữ liệu và chuẩn bị cho phần data augmentation.

- Mã hóa labels(L70-72)

- Sử dụng ***scikit-learn*** để phân đoạn data thành 2 phần: 80% cho phần training và 20% còn lại dùng cho việc testing.(L76-77)

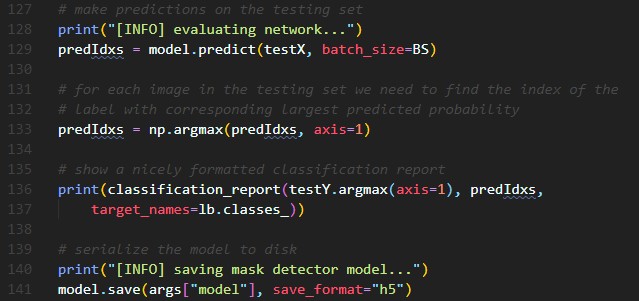
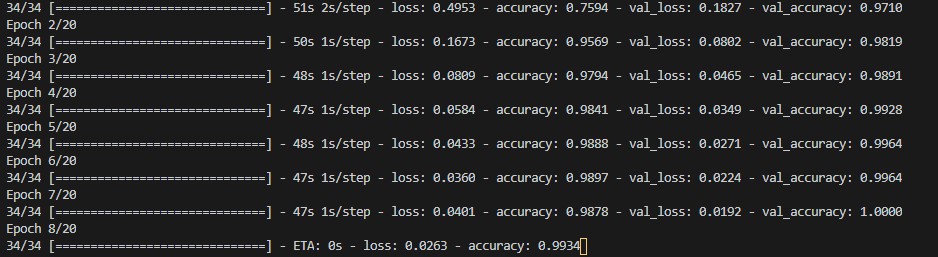
- Để tăng số lượng data. Chúng ta sử dụng phương pháp data augmentation, trong đó các tham số như xoay, phóng, cắt, dịch chuyển… được mô tả ở L80-



87

- Với dữ liệu đã chuẩn bị phía trên, bây giờ có thể tiến hành complie và train netwrok của chúng ta.

- Compile model của chúng ta với Adam optimizer. (L114-116)



- Quá trình training được bắt đầu từ L120-125.

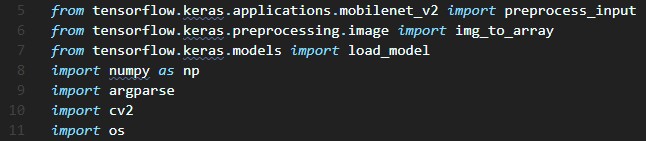
- Sau khi quá trình training kết thúc, ta sẽ tiến hành đánh giá kết quả của model trên tập dữ liệu test.

- L129-133 đưa ra dự đoán trên tập kiểm tra. Sau đó print classfication\_report

ra terminal.

- Cuối cùng là lưu model train vào ổ đĩa. (L141)

- Quá trình thực hiện train:



**4.4 Thực hiện phát hiện người đeo khẩu trang trong ảnh tĩnh**

- Đến phần này, ứng dụng của chúng ta đã được train. Tiếp theo ta sẽ thực hiện:

1. Load ảnh input từ ổ đĩa.

2. Phát hiện khuôn mặt trong ảnh.

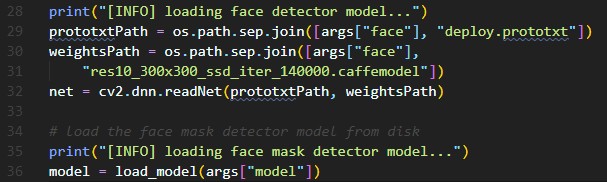
3. Áp dụng face mask detector model để phân loại with\_mask hay without\_mask.

- Trong file detect\_mask\_image.py:

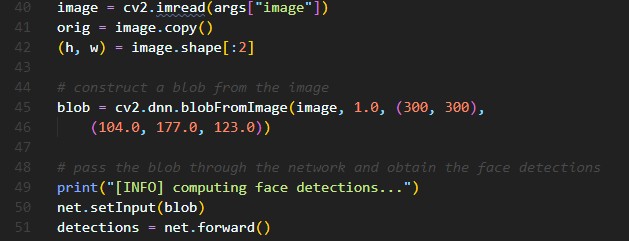
- Sử dụng 03 Tensorflow/Keras import để load model và tiền xử lý hình ảnh

đầu vào.

- OpenCV được sử dụng vào việc hiển thị và xử lý hình ảnh.



- Tiếp theo, chúng ta sẽ load model phát hiện khuôn mặt và nhận diện khuôn mặt đeo khẩu trang.



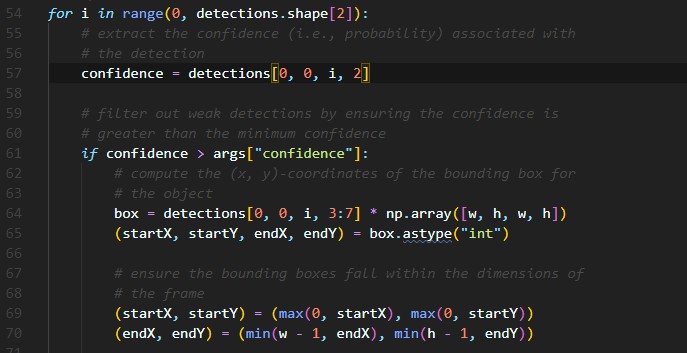
- Load và tiền xử lý ảnh đầu vào.

- Load ảnh và tạo một bản sao, lấy kích thước khung phục vụ cho việc hiển thị

và chia tỉ lệ ảnh trong phần tiếp theo. (L41-42)

- Tiền xử lý ảnh đầu vào được xử lý bằng OpenCV’s blobFromImage Function(L45-46). Như các tham số được hiển thị ở trên, ảnh được resize thành 300x300 pixel và thực hiện mean subtraction.

- Thực hiện phát hiện khuôn mặt để tìm ra tất cả khuôn mặt có trong bức ảnh. (L50-51)

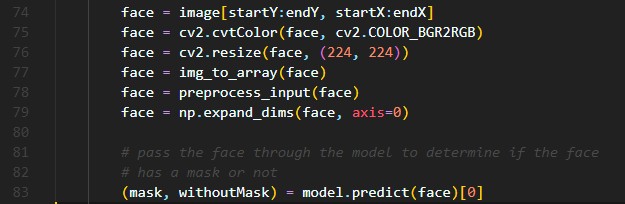


- Khi đã đưa ra được dự đoán về vị trí khuôn mặt trong ảnh input, ta cần chắc chắn rằng nó đạt đúng ***ngưỡng tin cậy*** trước khi trích xuất faceROIs.

- Ở đây từ L54-61, thực hiện vòng lặp qua tất cả những detections sau đó trích

xuất giá trị của confidence nhằm so sánh với confidence định ra từ trước.

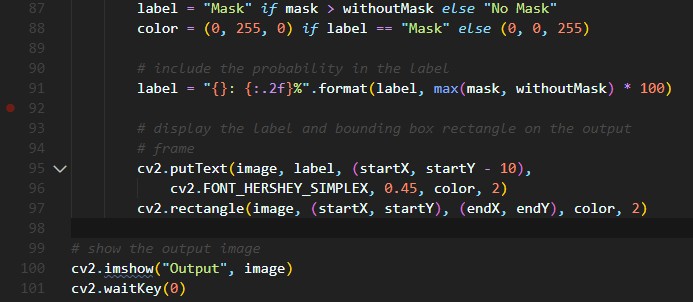
- Cuối cùng, L64-70 thực hiện việc tính toán giá trị cho box bounding khuôn mặt được phát hiện và kèm theo đó là đảm bảo bounding không bị vượt quá biên ảnh input.



- Trích xuất faceROIs thông qua NumPy (L74)

- Tiền xử lý faceROIs.(L75-L79)

- Thực hiện mask detection để dự đoán with\_mask hay without\_mask. (L80)



- Cuối cùng là hiển thị kết quả:

1. Đầu tiên, xác định label dựa vào xác suất được trả về từ mask detector model. (L87) và gán màu liên quan cho phần chú thích(L88). Màu xanh lá cho with\_mask và màu đỏ cho without\_mask.

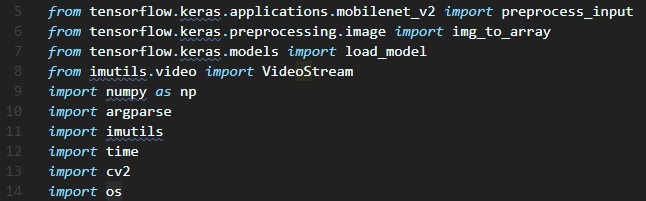
2. Sau đó, tiến hành vẽ label text và hình chữ nhật giới hạn khu vực khuôn

mặt sử dụng OpenCV function.(L95-97)

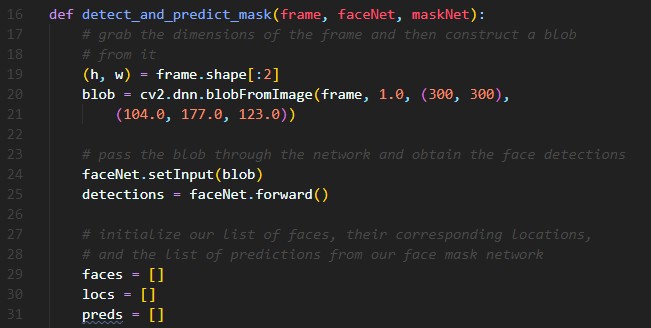
3. L100-101 hiển thị ảnh kết quả.

**4.5 Thực hiện phát hiện người đeo khẩu trang theo thời gian thực**

- Trong file **detect\_mask\_video.py**:



- Tương tự như phát hiện người đeo khẩu trang trên ảnh tĩnh. Điểm khác biệt duy nhất đó chính là cần VideoStream và time class để giúp chúng ta có thể làm việc với stream data.



- Trong function này, define thuật toán face detection/ mask detection của chúng ta.

- Function này phát hiện khuôn mặt sau đó áp dụng face mask classifier cho

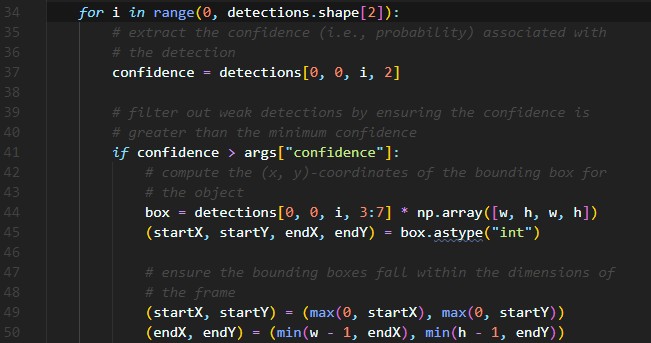
từng faceROIs.

- Detect\_and\_predict\_mask có 3 tham số truyền vào bao gồm:

1. Frame.

2. faceNet: model sử dụng để phát hiện vị trí khuôn mặt.

3. maskNet: face mask classifier model của chúng ta.

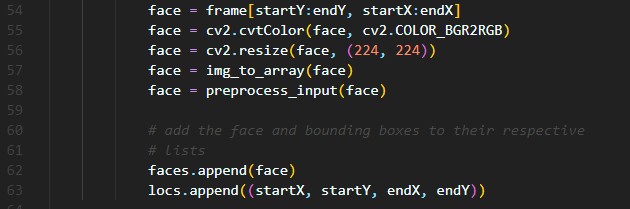


- Từ đó, tiến hành vòng lặp qua tất cả face detection.

- Trong vòng lặp for:

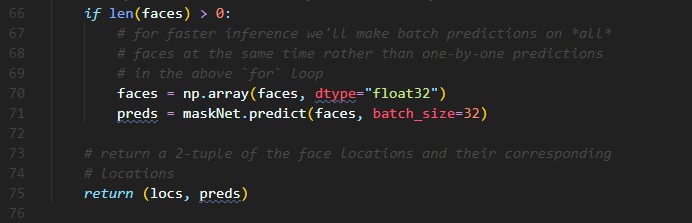
1. Tiến hành lọc các detection “yếu” (L37-41)

2. Trích xuất bounding box trong khi đảm bảo được tọa độ box không nằm ngoài biên của ảnh.(L44-50)



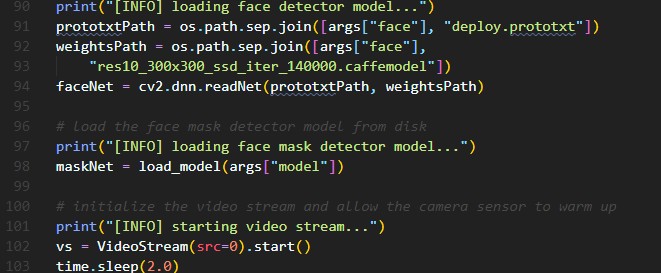
- Trích xuất và tiền xử lý faceROIs.(L54-58)

- Thêm box bounding.(L59-60)



- Đầu tiên, chúng ta cần chắc chắn rằng có ít nhất 1 khuôn mặt được phát hiện trên frame(L66) – nếu không thì sẽ trả về giá trị preds.

- L76 trả về vị trí của box giới hạn khuôn mặt.

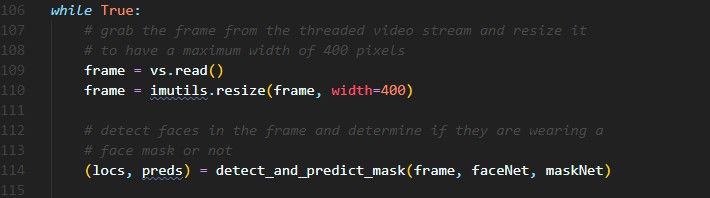


- Insert thêm một số khởi tạo cần thiết như:

1. Face detector.

2. Face mask detector.

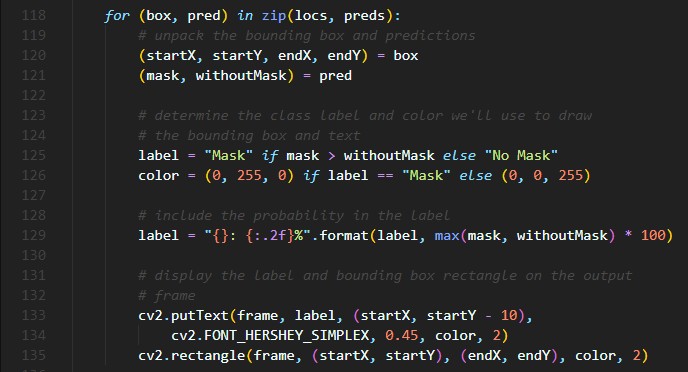
3. Webcam video stream.



- Tiến hành vòng lặp qua tất cả các frame trên stream:

1. Lấy 1 frame từ stream sau đó resize (L109-110)

2. Phát hiện và dự đoán xem người có đeo mặt nạ hay không. (L114)



- Trong vòng lặp qua tất cả các kết quả dự đoán(bắt đầu từ L118), chúng ta sẽ:

1. Xác định label và màu(L125-126)

2. Chú thích label và face bounding box.(L133-135)

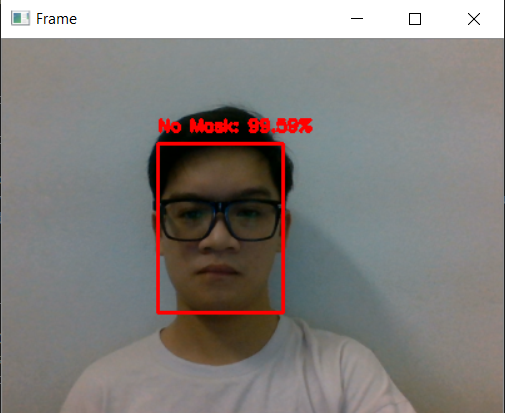


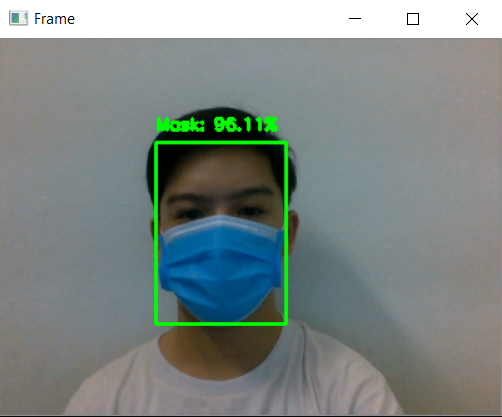
- Và cuối cùng đó là hiển thị kết quả; Sau khi hiển thị frame nếu người dùng ấn

phím “q” – quit thì chúng ta dừng vòng lặp

**4.6 Kết quả thực hiện**

**4.6.1 Phát hiện người đeo khẩu trang trong ảnh tĩnh**





**4.6.2 Phát hiện người đeo khẩu trang theo thời gian thực**

Video kết quả thực hiện: <https://www.youtube.com/watch?v=geobTvlpMyU&feature=youtu.be>

**Chương 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

**5.1 Kết luận**

* Trong khuôn khổ bài tập lớn lần này, nhóm em gồm có 2 thành viên đã tìm hiểu, nghiên cứu và xây dựng ứng dụng “NHẬN DIỆN NGƯỜI ĐEO KHẨU TRANG SỬ DỤNG OPENCV VÀ TENSORFLOW/KERAS”. Để tạo ra face mask detector, chúng ta cần train một model gồm có 2 lớp gồm: những người đeo khẩu trang và những người không đeo khẩu trang.
* Tinh chỉnh MobileNetV2 trên bộ dữ liệu (dataset) và từ đó việc phân loại có độ chính xác rất cao ~97%.
* Sau đó, tiếp tục sử dụng face mask classifier áp dụng vào cho cả hình ảnh và video stream real-time bằng cách:

1. Phát hiện khuôn mặt trong hình ảnh/video.

2. Trích xuất từng khuôn mặt.

3. Áp dụng face mask classifier

* Face mask detector được xây dựng ở bài tập này có độ chính xác cao, và nhờ vào việc sử dụng MobileNetV2 nên cũng có tính hiệu quả, giúp triển khai mô hình này một cách dễ dàng hơn cho các hệ thống nhúng như: Raspberry, Google Coral…
* Nếu được áp dụng rộng rãi thì sẽ giảm bớt được gánh nặng trong việc giám

sát an ninh trong thời kì dịch bệnh ngày nay.

**5.2 Hướng phát triển**

Có thể dễ dàng nhận thấy từ phần kết quả ở phía trên, face mask detector

đang hoạt động khá tốt mặc dù:

* Dữ liệu training còn hạn chế.
* Bộ dữ liệu về những người đeo khẩu trang được tạo ra một cách nhân tạo.

Do đó, để cải tiến mô hình này, chúng ta nên thu thập dữ liệu hình ảnh thực tế(thay vì sử dụng ảnh nhân tạo) của những người đeo khẩu trang.

Cũng nên thu thập thêm hình ảnh của những khuôn mặt có thể làm cho bộ phân loại của chúng ta nhầm lẩn họ đang đeo khẩu trang nhưng thực tế thì không. Ví dụ như hình ảnh người đó đang sử dụng khăn choàng….

Với một nhược điểm nữa đó là bộ phân loại của chúng ta chỉ hoạt động khi bộ phát hiện khuôn mặt hoạt động tốt. Nếu như không thể phát hiện khuôn mặt trong bức ảnh/video thì việc nhận diện người đó có đeo khẩu trang hay không là điều bất khả thi. Một hướng phát triển nữa giành cho đề tài này đó là cải tiến bộ phát hiện khuôn mặt trong ảnh tĩnh/video.