**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**VIỆT - HÀN**

**KHOA KHOA HỌC MÁY TÍNH**

****

**ĐỒ ÁN CƠ SỞ 4**

**ĐỀ TÀI: PHÁT TRIỂN ỨNG DỤNG NHẬN DIỆN KHUÔN MẶT**

Sinh viên thực hiện: **NGUYỄN THỊ THU HƯƠNG**

**TRẦN NGỌC QUANG**

Lớp: **19IT2**

Giảng viên hướng dẫn: TS.LÊ TÂN

***Đà Nẵng , tháng 12 năm 2021***

**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**VIỆT - HÀN**

**KHOA KHOA HỌC MÁY TÍNH**



****

**ĐỒ ÁN CƠ SỞ 4**

**ĐỀ TÀI: PHÁT TRIỂN ỨNG DỤNG NHẬN DIỆN KHUÔN MẶT**

Sinh viên thực hiện: **NGUYỄN THỊ THU HƯƠNG**

**TRẦN NGỌC QUANG**

Lớp: **19IT2**

Giảng viên hướng dẫn: TS.LÊ TÂN

***Đà Nẵng , tháng 12 năm 21***

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………… Đà Nẵng, ngày….tháng….năm 2021

Giảng viên hướng dẫn

TS.Lê Tân

**LỜI CẢM ƠN**

Chúng em xin chân thành cảm ơn sự giúp đỡ nhiệt tình của thầy Lê Tân, người đã định hướng, hướng dẫn cũng như hỗ trợ cho chúng em trong quá trình chuẩn bị và tiến hành để chúng em có thể hoàn thành tốt đồ án này.

Chúng em cũng xin gửi lời cảm ơn tới các thầy cô giáo, giảng viên trong Đại học Công nghệ thông tin và Truyền thông Việt – Hàn, Khoa Khoa học máy tính đã cung cấp cho chúng em những kiến thức cần thiết để thực hiện đồ án này.

Chúng em xin gửi lời cảm ơn đến gia đình, bạn bè luôn động viên giúp đỡ chúng em trong suốt thời gian học tập và nghiên cứu, đóng góp những kinh nghiệm quý báu trong thời gian thực hiện đồ án này.

Kính chúc quý thầy cô mạnh khỏe, công tác tốt, tiếp tục giảng dạy và đào tạo thế hệ trẻ thành công.

Một lần nữa, chúng em xin chân thành cảm ơn!

*Sinh viên,*

Nguyễn Thị Thu Hương

Trần Ngọc Quang

**MỤC LỤC**

Contents

[DANH MỤC HÌNH ẢNH 7](#_Toc89639258)

[DANH MỤC CÁC TỪ VIẾT TẮT 8](#_Toc89639259)

[MỞ ĐẦU 9](#_Toc89639260)

[**1.** **Giới thiệu** 9](#_Toc89639261)

[**2.** **Mục tiêu, nhiệm vụ đề tài** 9](#_Toc89639262)

[**3.** **Đối tượng, phạm vi và phương pháp tiếp cận** 9](#_Toc89639263)

[**4. Đóng góp đề tài** 9](#_Toc89639264)

[**5. Bố cục đề tài** 10](#_Toc89639265)

[Chương 1. TỔNG QUAN VỀ ĐỀ TÀI ĐỀ TÀI 11](#_Toc89639266)

[1.1 Tên dự án 11](#_Toc89639267)

[1.2 Đặt vấn đề 11](#_Toc89639268)

[1.3 Phương hướng giải quyết/ Ý tưởng thực hiện 11](#_Toc89639269)

[1.4 Cấu trúc dự án 11](#_Toc89639270)

[1.5 Cơ sở lý thuyết 12](#_Toc89639271)

[1.5.1 Tìm hiểu về các ngôn ngữ lập trình 12](#_Toc89639272)

[1.5.2 Các thuật toán Object Detection 13](#_Toc89639273)

[1.5.3 Lớp các mô hình họ R-CNN 14](#_Toc89639274)

[1.5.4 Lớp các mô hình họ YOLO 16](#_Toc89639275)

[1.6 Kết quả cần đạt được 17](#_Toc89639276)

[Chương 2. TRIỂN KHAI XÂY DỰNG 18](#_Toc89639277)

[2.1 Chuẩn bị dữ liệu 18](#_Toc89639278)

[2.1.1 Tổng quan bộ dữ liệu 18](#_Toc89639279)

[2.1.2 Khai phá dữ liệu 18](#_Toc89639280)

[2.2 Cấu trúc dự án 23](#_Toc89639281)

[2.3 Huấn luyện mô hình 24](#_Toc89639282)

[3.4 Triển khai dự án 25](#_Toc89639283)

[3.5 Xây dựng API 25](#_Toc89639284)

[Chương 3. THIẾT KẾ PHÁT TRIỂN ỨNG DỤNG 26](#_Toc89639285)

[**1.** **Kết quả** 26](#_Toc89639286)

[KẾT LUẬN 27](#_Toc89639287)

[1. Đánh giá kết quả 27](#_Toc89639288)

[2. Hướng phát triển 27](#_Toc89639289)

[TÀI LIỆU THAM KHẢO 28](#_Toc89639290)

# DANH MỤC HÌNH ẢNH

[*Hình 1. Sơ đồ mối liên hệ giữa các tác vụ trong Computer Vision* 14](#_Toc89640107)

[*Hình 2. Kiến trúc của các mô hình YOLOv5* 17](#_Toc89640108)

[*Hình 3/ Kiến trúc chung của mô hình YOLOv5* 17](#_Toc89640109)

[*Hình 4. Tập dữ liệu phát hiện mặt nạ* 18](#_Toc89640110)

[*Hình 5. Bức ảnh của một người không đeo mặt.* 19](#_Toc89640111)

[*Hình 6. Bước tiếp theo là áp dụng tính năng nhận diện khuôn mặt. Ở đây, chúng tôi đã sử dụng phương pháp học sâu để thực hiện nhận diện khuôn mặt với OpenCV* 20](#_Toc89640112)

[*Hình 7. Bước tiếp theo là trích xuất ROI khuôn mặt với OpenCV và NumPy cắt.* 20](#_Toc89640113)

[*Hình 8. Sau đó, chúng tôi phát hiện các điểm mốc trên khuôn mặt bằng cách sử dụng dlib để chúng tôi biết vị trí đặt mặt nạ trên khuôn mặt.* 21](#_Toc89640114)

[*Hình 9. Ví dụ về khẩu trang / lá chắn COVID-19.* 21](#_Toc89640115)

[*Hình 10. Trong hình này, mặt nạ được đặt trên khuôn mặt của người đó trong khung hình ban đầu.* 22](#_Toc89640116)

[*Hình 11. Hình ảnh mặt nạ COVID-19 nhân tạo được hiển thị.* 23](#_Toc89640117)

[*Hình 12. Một số câu lệnh để khởi chạy mô hình* 24](#_Toc89640118)

[*Hình 13. Các đường cong đào tạo về độ chính xác / tổn thất của máy dò mặt nạ COVID-19* 25](#_Toc89640119)

[*Hình 14. Kết quả khi nhận diện được khẩu trang* 26](#_Toc89640120)

[*Hình 15. Kết quả khi nhận diện được khẩu trang* 26](#_Toc89640121)

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **KÝ HIỆU CHỮ VIẾT TẮT** | **CHỮ VIẾT ĐẦY ĐỦ** |
| 1 | CAD | Computer-Aided Diagnosis |
| 2 | CSDL | Cơ sở dữ liệu |
| 3 | CNN | Convolutional Neural Networks |
| 4 | CT Scan | Computed Tomography Scan |
| 5 | MRI Scan | Magnetic Resonance Imaging |
| 6 | NMS | Non-maximum Suppression |
| 7 | PET Scan | Positron Emission Tomography Scan |
| 8 | R-CNN | Region-based Convolutional Neural Networks |
| 9 | WBF | Weighted Box |
| 10 | YOLO | You Only Look Once |

# MỞ ĐẦU

1. **Giới thiệu**

Công nghệ đang ngày càng phát triển và có những vai trò, chức năng quan trọng trong nhiều lĩnh vực dịch vụ và nghiên cứu. Với tình hình ngày nay, đại dịch Covid-19 đang căng thẳng và công tác bảo vệ an toàn của các cơ quan chức năng còn nhiều hạn chế.

Nhận thấy điều đó, chúng em quyết định phát triển một ứng dụng cho phép quét thấy những người có đeo khẩu trang và cảnh báo những người không đeo khẩu trang để tiện cho công tác xử lý và ngăn chặn kịp thời của các cơ quan chức năng.

Liên hệ tới đề tài, hệ thống nhận dạng mặt người là một hệ thống nhận vào là một ảnh hoặc một đoạn video (một chuỗi các ảnh). Qua xử lý tính toán hệ thống xác định được vị trí mặt người trong ảnh (nếu có) và xác định là người nào trong số những người hệ thống đã được biết (qua quá trình học) hoặc là người lạ.

Đặc biệt trong tình trạng dịch bệnh đang có xu hướng gia tăng đáng kể trong những năm gần đây, mục tiêu của đề tài càng trở nên ý nghĩa hơn trong hỗ trợ và thúc đẩy quá trình phát hiện và ngăn chặn dịch bệnh.

1. **Mục tiêu, nhiệm vụ đề tài**

* Rèn luyện kỹ năng nghiên cứu và tìm hiểu tài liệu.
* Tìm hiểu sâu phần mềm lập trình Visual studio code.
* Nâng cao kỹ năng thiết kế và lập trình bằng ngôn ngữ Python.
* Ứng dụng OpenCv vào đề tài nghiên cứu.
* Áp dụng phân tích thiết kế hệ thống vào ứng dụng.
* Tìm hiểu về ứng dụng nhận diện khuôn mặt, từ đó phát triển lên ứng dụng nhận diện mặt nạ trong bối cảnh dịch bệnh hiện nay.

1. **Đối tượng, phạm vi và phương pháp tiếp cận**

* Đối tượng: Dành cho những tất cả mọi người có nhu cầu kiểm tra nhận diện để kiểm soát người đang đeo và không đeo mặt nạ.
* Phương pháp tiếp cực: Quảng bá trên phạm vi internet qua các ứng dụng con người thường xuyên tiếp cận như facebook, instagram, báo…

**4. Đóng góp đề tài**

* Nghiên cứu và tìm hiểu qua internet.
* Tham khảo ý kiến của các bạn cùng học lập trình.
* Tìm hiểu về các tài liệu liên quan đến trí tuệ nhân tạo, đặc biệt OpenCV.

**5. Bố cục đề tài**

Sau phần Mở đầu, báo cáo được trình bày trong ba chương, cụ thể như sau:

Chương 1. Tổng quan về đề tài.Trong chương này, báo cáo trình bày về đặt vấn đề, giải pháp, cơ sở lý thuyết và mục đích xây dựng ứng dụng.

Chương 2. Nghiên cứu tổng quan. Chương này mô tả thuật toán.

Chương 3. Thiết kế phát triển ứng dụng. Chương này thể hiện chi tiết các trang ứng dụng.

Cuối cùng là *Kết luận*, *Tài liệu tham khảo* liên quan đến đề tài.

# Chương 1. TỔNG QUAN VỀ ĐỀ TÀI ĐỀ TÀI

## 1.1 Tên dự án

Phát triển ứng dụng nhận diện khuôn mặt.

## 1.2 Đặt vấn đề

Công nghệ đang ngày càng phát triển và có những vai trò, chức năng quan trọng trong nhiều lĩnh vực dịch vụ và nghiên cứu. Với tình hình ngày nay, đại dịch Covid-19 đang căng thẳng và công tác bảo vệ an toàn của các cơ quan chức năng còn nhiều hạn chế.

Nhận thấy điều đó, chúng em quyết định phát triển một ứng dụng cho phép quét thấy những người có đeo khẩu trang và cảnh báo những người không đeo khẩu trang để tiện cho công tác xử lý và ngăn chặn kịp thời của các cơ quan chức năng.

Liên hệ tới đề tài, hệ thống nhận dạng mặt người là một hệ thống nhận vào là một ảnh hoặc một đoạn video (một chuỗi các ảnh). Qua xử lý tính toán hệ thống xác định được vị trí mặt người trong ảnh (nếu có) và xác định là người nào trong số những người hệ thống đã được biết (qua quá trình học) hoặc là người lạ.

Đặc biệt trong tình trạng dịch bệnh đang có xu hướng gia tăng đáng kể trong những năm gần đây, mục tiêu của đề tài càng trở nên ý nghĩa hơn trong hỗ trợ và thúc đẩy quá trình phát hiện và ngăn chặn dịch bệnh.

## 1.3 Phương hướng giải quyết/ Ý tưởng thực hiện

Để thực hiện quá trình chẩn đoán, mô hình sẽ nhận input là 1 ảnh chụp. Sau đó ảnh sẽ được xử lý thông qua mô hình deep learning, dựa trên kỹ thuật xử lý ảnh và nhận diện vật thể. Quá trình xử lý sẽ xác định được người đó có đeo khẩu trang hay không và từ đó đưa ra những kết quả thu được.

Mô hình xử lý ảnh được xây dựng trên mô hình YOLO v5 và được huấn luyện bằng bộ dữ liệu VinDr-CXR.

## 1.4 Cấu trúc dự án

* Web client: Giao diện người dùng sử dụng để đăng tải ảnh chụp và nhận kết quả chẩn đoán
* Web server: Máy chủ. Dùng để giao tiếp với client, mô hình xử lý, có nhiệm vụ tiếp nhận dữ liệu (ảnh), đưa qua mô hình xử lý và trả về kết quả
* Mô hình xử lý ảnh và nhận diện vật thể: Được huấn luyện để thực hiện nhiệm vụ xử lý và nhận diện ảnh

## Cơ sở lý thuyết

### Tìm hiểu về các ngôn ngữ lập trình

* **Ngôn ngữ lập trình Python**

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ, được tạo ra bởi Guido van Rossum. Nó dễ dàng để tìm hiểu và đang nổi lên như một trong những ngôn ngữ lập trình nhập môn tốt nhất cho người lần đầu tiếp xúc với ngôn ngữ lập trình. Python hoàn toàn tạo kiểu động và sử dụng cơ chế cấp phát bộ nhớ tự động. Python có cấu trúc cấp cao 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 cách gõ 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.

* **Framework Pytorch**

Pytorch là 1 trong số framework hỗ trợ deep learning phổ biến, có lượng người dùng đông đảo, cộng đồng lớn mạnh. Đây là package sử dụng các thư viện của CUDA và C/C++ hỗ trợ các tính toán trên GPU nhằm gia tăng tốc độ xử lý của mô hình. 2 mục tiêu chủ đạo của package này hướng tới là:

* Một sự thay thế cho bộ thư viện numpy để tận dụng sức mạnh tính toán của GPU.
* Một platform Deep learning phục vụ trong nghiên cứu, mang lại sự linh hoạt và tốc độ.
* **Framework Flask**

Flask là một framework web của Python, còn được coi là một microframework bởi nó không yêu cầu các công cụ hoặc thư viện cụ thể. Được phát triển bởi Armin Ronacher, người dẫn đầu một nhóm những người đam mê Python quốc tế có tên là Poocco. Flask phát triển dựa trên bộ công cụ Werkzeug WSGI và công cụ mẫu Jinja2. Flask cung cấp tất cả các công cụ, thư viện và công nghệ cho phép bạn xây dựng một ứng dụng web.

* **Javascript**

JavaScript, theo phiên bản hiện hành, là một ngôn ngữ lập trình thông dịch được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web (phía người dùng) cũng như phía máy chủ (với Nodejs). Nó vốn được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên là Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. “.js” là phần mở rộng thường được dùng cho tập tin JavaScript.

Phiên bản mới nhất của JavaScript là ECMAScript 7. ECMAScript là phiên bản chuẩn hóa của JavaScript. Trình duyệt Mozilla phiên bản 1.8 beta 1 có hỗ trợ không đầy đủ cho E4X - phần mở rộng cho JavaScript hỗ trợ làm việc với XML, được chuẩn hóa trong ECMA-357.

### Các thuật toán Object Detection

Nhận dạng đối tượng là một thuật ngữ chung để mô tả một tập hợp các nhiệm vụ thị giác máy tính có liên quan liên quan đến việc xác định các đối tượng trong ảnh kỹ thuật số.

Phân loại hình ảnh liên quan đến việc dự đoán lớp của một đối tượng trong một hình ảnh. Định vị vật thể đề cập đến việc xác định vị trí của một hoặc nhiều đối tượng trong một hình ảnh và vẽ bounding box xung quanh chúng. Phát hiện đối tượng kết hợp hai nhiệm vụ trên và thực hiện cho một hoặc nhiều đối tượng trong hình ảnh. Chúng ta có thể phân biệt giữa ba nhiệm vụ thị giác máy tính cơ bản trên thông qua input và output của chúng như sau:

* Phân loại hình ảnh: Dự đoán nhãn của một đối tượng trong một hình ảnh.
* Input: Một hình ảnh với một đối tượng, chẳng hạn như một bức ảnh.
* Output: Nhãn lớp (ví dụ: một hoặc nhiều số nguyên được ánh xạ tới nhãn lớp).
* Định vị đối tượng: Xác định vị trí hiện diện của các đối tượng trong ảnh và cho biết vị trí của chúng bằng bounding box.
* Input: Một hình ảnh có một hoặc nhiều đối tượng, chẳng hạn như một bức ảnh.
* Output: Một hoặc nhiều bounding box được xác định bởi tọa độ tâm, chiều rộng và chiều cao.
* Phát hiện đối tượng: Xác định vị trí hiện diện của các đối tượng trong bounding box và nhãn của các đối tượng nằm trong một hình ảnh.
* Input: Một hình ảnh có một hoặc nhiều đối tượng, chẳng hạn như một bức ảnh.
* Output: Một hoặc nhiều bounding box và nhãn cho mỗi bounding box.

Diagram

Description automatically generated

*Hình 1. Sơ đồ mối liên hệ giữa các tác vụ trong Computer Vision*

### 1.5.3 Lớp các mô hình họ R-CNN

R-CNN (Regions with CNN features) là lớp các mô hình xác định vùng đặc trưng dựa trên các mạng CNN được phát triển bởi Ross Girshick và các cộng sự. Lớp các mô hình này gồm 3 mô hình chính là R-CNN, Fast R-CNN và Faster-RCNN được thiết kế cho các nhiệm vụ định vị vật thể và nhận diện vật thể.

**R-CNN (2014)**

R-CNN được giới thiệu lần đầu vào 2014. Nó có thể là một trong những ứng dụng nền móng đầu tiên của mạng nơ-ron tích chập đối với vấn đề định vị, phát hiện và phân đoạn đối tượng. Cách tiếp cận đã được chứng minh trên các bộ dữ liệu điểm chuẩn, đạt được kết quả tốt nhất trên bộ dữ liệu VOC-2012 và bộ dữ liệu phát hiện đối tượng ILSVRC-2013 gồm 200 lớp.

Kiến trúc của R-CNN gồm 3 thành phần đó là:

* Vùng đề xuất hình ảnh (Region proposal): Có tác dụng tạo và trích xuất các vùng đề xuất chứa vật thể được bao bởi các bounding box.
* Trích lọc đặc trưng (Feature Extractor): Trích xuất các đặc trưng giúp nhận diện hình ảnh từ các region proposal thông qua các mạng deep convolutional neural network.
* Phân loại (classifier): Dựa vào input là các features ở phần trước để phân loại hình ảnh chứa trong region proposal về đúng nhãn.

Một kỹ thuật được sử dụng để đề xuất các region proposal hoặc các bounding boxes chứa các đối tượng tiềm năng trong hình ảnh được gọi là “selective search”, các region proposal có thể được phát hiện bởi đa dạng những thuật toán khác nhau. Nhưng điểm chung là đều dựa trên tỷ lệ IoU giữa bounding box và ground truth box.

**Fast R-CNN (2015)**

Điểm đột phá của Fast R-CNN là sử dụng một single model thay vì pipeline để phát hiện region và classification cùng lúc.

Kiến trúc của mô hình trích xuất từ bức ảnh một tập hợp các region proposals làm đầu vào được truyền qua mạng deep CNN. Một pretrained-CNN, chẳng hạn VGG-16, được sử dụng để trích lọc features. Phần cuối của deep-CNN là một custom layer được gọi là layer vùng quan tâm (Region of Interest Pooling - RoI Pooling) có tác dụng trích xuất các features cho một vùng ảnh input nhất định.

Sau đó các features được kết bởi một lớp fully connected. Cuối cùng mô hình chia thành hai đầu ra, một đầu ra cho dự đoán nhãn thông qua một softmax layer và một đầu ra khác dự đoán bounding box (kí hiệu là bbox) dựa trên hồi qui tuyến tính. Quá trình này sau đó được lặp lại nhiều lần cho mỗi vùng RoI trong một hình ảnh.

Mô hình này nhanh hơn đáng kể cả về huấn luyện và dự đoán, tuy nhiên vẫn cần một tập hợp các region proposal được đề xuất cùng với mỗi hình ảnh đầu vào.

**Faster R-CNN (2016)**

Kiến trúc này mang lại độ chính xác cao nhất đạt được trên cả hai nhiệm vụ phát hiện và nhận dạng đối tượng tại các cuộc thi ILSVRC-2015 và MS COCO-2015.

Kiến trúc được thiết kế để đề xuất và tinh chỉnh các region proposals như là một phần của quá trình huấn luyện, được gọi là mạng đề xuất khu vực (Region Proposal Network), hoặc RPN. Các vùng này sau đó được sử dụng cùng với mô hình Fast R-CNN trong một thiết kế mô hình duy nhất. Những cải tiến này vừa làm giảm số lượng region proposal vừa tăng tốc hoạt động trong thời gian thử nghiệm mô hình lên gần thời gian thực với hiệu suất tốt nhất. Tốc độ là 5fps trên một GPU.

Mặc dù là một mô hình đơn lẻ duy nhất, kiến trúc này là kết hợp của hai modules:

* Mạng đề xuất khu vực (Region Proposal Network, viết tắt là RPN). Mạng CNN để đề xuất các vùng và loại đối tượng cần xem xét trong vùng.
* Fast R-CNN: Mạng CNN để trích xuất các features từ các region proposal và trả ra các bounding boxes và nhãn.

Cả hai modules hoạt động trên cùng một output của một mạng deep CNN. Mạng RPN hoạt động như một cơ chế attention cho mạng Fast R-CNN, thông báo cho mạng thứ hai về nơi cần xem hoặc chú ý.

RPN hoạt động bằng cách lấy đầu ra của một mạng pretrained deep CNN, chẳng hạn như VGG-16, và truyền feature map vào một mạng nhỏ và đưa ra nhiều region proposals và nhãn dự đoán cho chúng. Region proposals là các bounding boxes, dựa trên các anchor boxes hoặc hình dạng được xác định trước được thiết kế để tăng tốc và cải thiện khả năng đề xuất vùng. Dự đoán của nhãn được thể hiện dưới dạng nhị phân cho biết region proposal có xuất hiện vật thể hoặc không.

Một quy trình huấn luyện xen kẽ được sử dụng trong đó cả hai mạng con được đào tạo cùng một lúc. Điều này cho phép các tham số trong feature dectector của deep CNN được tinh chỉnh cho cả hai tác vụ cùng một lúc.

### 1.5.4 Lớp các mô hình họ YOLO

Một họ mô hình nhận dạng đối tượng phổ biến khác được gọi chung là YOLO (You Only Look One), được phát triển bởi Joseph Redmon, và các cộng sự. Các mô hình R-CNN nói chung có thể chính xác hơn, tuy nhiên họ mô hình YOLO nhanh hơn rất rất nhiều so với R-CNN, và thậm chí đạt được việc phát hiện đối tượng trong thời gian thực.

YOLOv5 là mô hình đầu tiên của YOLO được viết bằng Pytorch thay vì PJ Reddie's Darknet. Nhờ đó, YOLOv5 có được lợi các lợi ích nhờ hệ sinh thái của Pytorch, hỗ trợ đơn giản và triển khai dễ dàng hơn.

Khác với các bản YOLO trước, YOLOv5 cung cấp luôn 4 phiên bản với kiến trúc mạng khác nhau:

* YOLOv5-s : small version
* YOLOv5-m : medium version
* YOLOv5-l : large version
* YOLOv5-x: extra large version

*A picture containing text, watch

Description automatically generated*

*Hình 2. Kiến trúc của các mô hình YOLOv5*

*Diagram

Description automatically generated*

*Hình 3/ Kiến trúc chung của mô hình YOLOv5*

## Kết quả cần đạt được

* Dựa trên ứng dụng phát triển khuôn mặt để tạo ra ứng dụng có thể nhận diện, phát hiện được những.
* Mô hình xử lý ảnh được huấn luyện để đưa ra kết quả có tỷ lệ chính xác tốt nhất.
* Web app để giao tiếp với người dùng.

# Chương 2. TRIỂN KHAI XÂY DỰNG

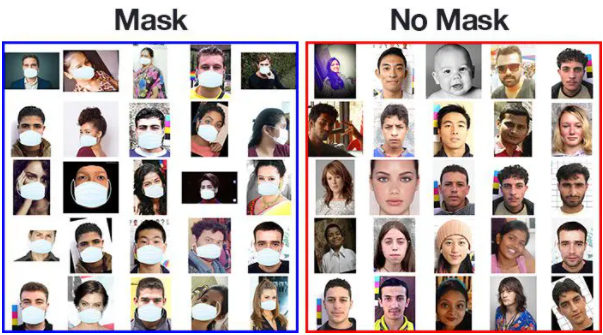
## 2.1 Chuẩn bị dữ liệu

### 2.1.1 Tổng quan bộ dữ liệu

Tập dữ liệu phát hiện mặt nạ bao gồm các hình ảnh “có mặt nà” và “không có mặt nạ”. Chúng tôi sử dụng tập dữ liệu để xây dụng máy dò mặt nạ COVID-19 với tầm nhìn máy tính và học sâu bằng Python, OpenCV và TensorFlow/Keras.

Tập dữ liệu này bao gồm 1.376 hình ảnh thuộc hai lớp:

* with\_mask: 690 hình ảnh
* without\_mask: 686 hình ảnh

Mục tiêu của chúng tôi là đào tạo mô hình học sâu tùy chỉnh để phát hiện xem một người *có* đeo mặt nạ hay *không*. 

*Hình 4. Tập dữ liệu phát hiện mặt nạ*

### 2.1.2 Khai phá dữ liệu

Để tạo tập dữ liệu này, Prajna đã có giải pháp khéo léo là:

1. Chụp *ảnh khuôn mặt bình thường*
2. Sau đó, tạo *tập lệnh Python tầm nhìn máy tính tùy chỉnh* để thêm mặt nạ vào chúng, do đó tạo tập dữ liệu *nhân tạo* (nhưng vẫn có thể áp dụng trong thế giới thực)

Phương pháp này thực sự dễ dàng hơn rất nhiều so với âm thanh một khi bạn [áp dụng các điểm mốc trên khuôn mặt cho vấn đề](https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/) .

Các mốc trên khuôn mặt cho phép chúng tôi tự động suy ra vị trí của các cấu trúc trên khuôn mặt, bao gồm:

* Mắt
* Lông mày
* Mũi
* Miệng
* Đường hàm

Để sử dụng các mốc trên khuôn mặt để xây dựng tập dữ liệu về các khuôn mặt đeo khẩu trang, trước tiên chúng ta cần bắt đầu với hình ảnh của một người *không* đeo khẩu trang:

**

*Hình 5. Bức ảnh của một người không đeo mặt.*

Từ đó, chúng tôi áp dụng tính năng phát hiện khuôn mặt để tính toán vị trí hộp giới hạn của khuôn mặt trong hình ảnh:

**

*Hình 6. Bước tiếp theo là áp dụng tính năng nhận diện khuôn mặt. Ở đây, chúng tôi đã sử dụng phương pháp học sâu để thực hiện nhận diện khuôn mặt với OpenCV*

Khi chúng tôi biết *vị trí* của khuôn mặt trong hình ảnh, chúng tôi có thể trích xuất Khu vực ưa thích (ROI) của khuôn mặt:



*Hình 7. Bước tiếp theo là trích xuất ROI khuôn mặt với OpenCV và NumPy cắt.*

Và từ đó, chúng tôi áp dụng các điểm mốc trên khuôn mặt, cho phép chúng tôi xác định vị trí của mắt, mũi, miệng, v.v.:

**

*Hình 8. Sau đó, chúng tôi phát hiện các điểm mốc trên khuôn mặt bằng cách sử dụng dlib để chúng tôi biết vị trí đặt mặt nạ trên khuôn mặt.*

Tiếp theo, chúng ta cần một hình ảnh của một mặt nạ (với nền trong suốt), chẳng hạn như hình dưới đây:

**

*Hình 9. Ví dụ về khẩu trang / lá chắn COVID-19.*

Mặt nạ này sẽ *tự động* được áp dụng cho khuôn mặt bằng cách sử dụng các điểm mốc trên khuôn mặt (cụ thể là các điểm dọc theo cằm và mũi) để tính toán *vị trí* mặt nạ sẽ được đặt.

Mặt nạ sau đó được thay đổi kích thước và xoay, đặt nó trên khuôn mặt:



*Hình 10. Trong hình này, mặt nạ được đặt trên khuôn mặt của người đó trong khung hình ban đầu.*

Sau đó, chúng tôi có thể lặp lại quá trình này cho tất cả các hình ảnh đầu vào của mình, từ đó tạo bộ dữ liệu mặt nạ nhân tạo của chúng tôi:



*Hình 11. Hình ảnh mặt nạ COVID-19 nhân tạo được hiển thị.*

## 2.2 Cấu trúc dự án

Các *dataset/* thư mục chứa dữ liệu được mô tả trong phần “Bộ dữ liệu phát hiện mặt nạ COVID-19 của chúng tôi” .

Ba hình ảnh *examples*/ được cung cấp để có thể kiểm tra trình phát hiện mặt nạ hình ảnh tĩnh.

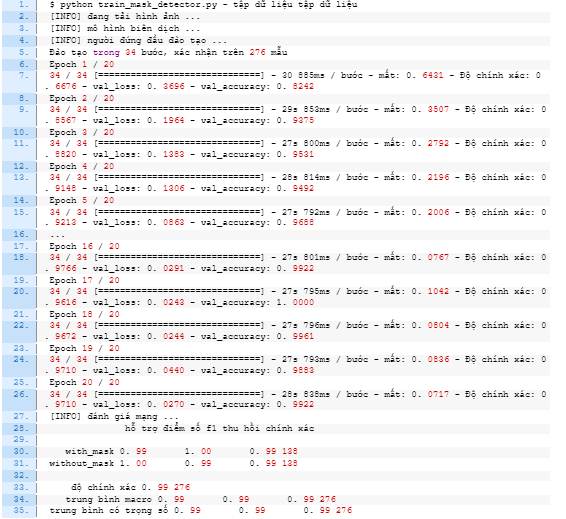
Cấu trúc bao gồm 3 phần như sau :

* *train\_mask\_detector.py*: Chấp nhận tập dữ liệu đầu vào của chúng tôi và tinh chỉnh MobileNetV2 dựa trên đó để tạo mask\_detector.model. Lịch sử đào tạoplot.png chứa các đường cong độ chính xác / mất mát cũng được tạo ra
* *detect\_mask\_image.py*: Thực hiện phát hiện mặt nạ trong hình ảnh tĩnh
* *detect\_mask\_video.py*: Sử dụng webcam của bạn, tập lệnh này áp dụng tính năng phát hiện mặt nạ cho mọi khung hình trong luồng

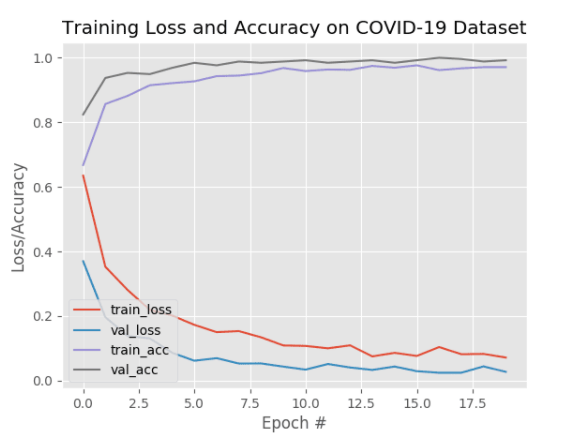
## 2.3 Huấn luyện mô hình

Hiện chúng tôi đã sẵn sàng đào tạo máy dò mặt nạ của mình bằng Keras, TensorFlow và Deep Learning.

Từ đó, mở một thiết bị đầu cuối và thực hiện lệnh sau:

**

*Hình 12. Một số câu lệnh để khởi chạy mô hình*



*Hình 13. Các đường cong đào tạo về độ chính xác / tổn thất của máy dò mặt nạ COVID-19*

Có thể thấy, mức độ chính xác của mô hình đạt được độ chính xác ~99% trên bộ thử nghiệm.

## 3.4 Triển khai dự án

Các bước thực hiện như sau :

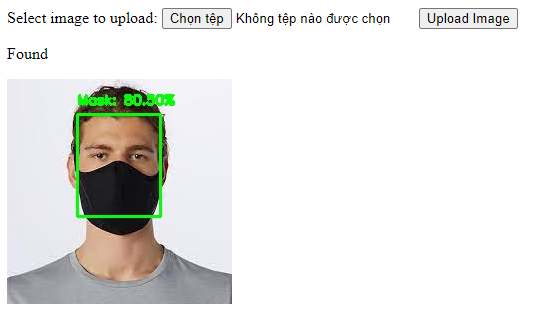
1. Tải hình ảnh đầu vào từ đĩa
2. Phát hiện khuôn mặt trong hình ảnh
3. Áp dụng máy dò mặt nạ để phân loại khuôn mặt *with\_mask* hoặc *without\_mask*.

## 3.5 Xây dựng API

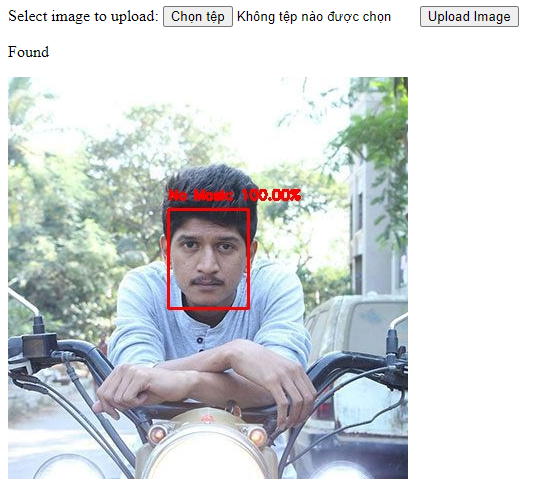
API của đề tài này được xây dựng bằng framework Flask. Đây là một API khá đơn giản, nó nhận ảnh của người dùng upload lên, trả về hệ thống, đưa vào mô hình đã huấn luyện trước đó, và cuối cùng trả lại kết quả chẩn đoán cho người dùng.

# Chương 3. THIẾT KẾ PHÁT TRIỂN ỨNG DỤNG

* + - 1. **Kết quả**

**

*Hình 14. Kết quả khi nhận diện được khẩu trang*

**

*Hình 15. Kết quả khi nhận diện được khẩu trang*

# KẾT LUẬN

## Đánh giá kết quả

Đây là đồ án cơ sở tổng hợp của cả một kỳ học tập tại trường, giúp cho sinh viên chúng em có thêm những kỹ năng như làm việc nhóm, thuyết trình, làm báo cáo, thiết kế bố cục, kỹ năng sáng tạo. Bên cạnh đó là củng cố và phát triển kỹ năng, kiến thức môn Công nghệ Web, Trí tuệ nhân tạo.

Nhóm chúng em đã phân công công việc sao cho hợp lí, thành viên cũng dốc hết tâm huyết làm bài này. Tuy nhiên, do thời gian cũng như khả năng còn hạn chế, nên vẫn chưa thực sự hoàn chỉnh, còn nhiều sai sót.

Với tất cả kiến thức và kỹ năng đã được học, ở những đồ án tiếp theo, chúng em sẽ cố gắng hoàn thiện hơn sản phẩm hiện tại.

Nhưng còn một số điểm hạn chế như sai:

* Cơ sở dữ liệu còn sơ sài
* Giao diện còn đơn giản
* Còn một số chức năng chưa hoàn thành

## Hướng phát triển

Sau khi hoàn thành đề tài, nhóm em sẽ tiếp tục nghiên cứu và phát triển ứng dụng như tối ưu hóa tốc độ xử lý, thêm các tính năng khác, phát triển để đưa ứng dụng vào thực tế cũng như phát triển thêm các chức năng để phù hợp với nhu cầu thực tế.

# TÀI LIỆU THAM KHẢO

[1].https://lagi.wiki/python#:~:text=Python%20l%C3%A0%20ng%C3%B4n%20ng%E1%BB%AF%20l%E1%BA%ADp,ph%C3%A1t%20b%E1%BB%99%20nh%E1%BB%9B%20t%E1%BB%B1%20%C4%91%E1%BB%99ng.

[2].COVID-19: Máy dò mặt nạ với OpenCV, Keras / TensorFlow và Deep Learning - PyImageSearch

[3]. https://phamdinhkhanh.github.io/2019/09/29/OverviewObjectDetection.html

[4].

[5].