BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KĨ THUẬT THÀNH PHỐ HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN CHUYÊN NGÀNH

TÌM HIỂU VÀ ỨNG DỤNG NHẬN DIỆN KHUÔN MẶT

Sinh viên thực hiện :

Nguyễn Thị Xuân Thanh **19110285**

Lê Trần Minh Nhựt **19110257**

Giáo viên hướng dẫn : PGS.TS. Hoàng Văn Dũng

Thành phố Hồ Chí Minh, năm 2022

BỘ GIÁO DỤ BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KĨ THUẬT THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN CHUYÊN NGÀNH

TÌM HIỂU VÀ ỨNG DỤNG NHẬN DIỆN KHUÔN MẶT

Sinh viên thực hiện :

Nguyễn Thị Xuân Thanh **19110285**

Lê Trần Minh Nhựt **19110257**

Giáo viên hướng dẫn : PGS.TS. Hoàng Văn Dũng

Thành phố Hồ Chí Minh, năm 2022

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KĨ THUẬT THÀNH PHỐ HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN CHUYÊN NGÀNH

TÌM HIỂU VÀ ỨNG DỤNG NHẬN DIỆN KHUÔN MẶT

Sinh viên thực hiện :

Nguyễn Thị Xuân Thanh **19110285**

Lê Trần Minh Nhựt **19110257**

Giáo viên hướng dẫn : PGS.TS. Hoàng Văn Dũng

Thành phố Hồ Chí Minh, năm 2022

NHẬN XÉT CỦA GIẢNG VIÊN

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

STT	NỘI DUNG THỰC HIỆN	NHẬN XÉT

ĐIỂM (BẰNG SỐ): .....
BẰNG CHỮ: .....
CHỮ KÝ GV:.....

GIỚI THIỆU ĐỀ TÀI

MSSV	Họ và tên	Tỷ lệ thực hiện
19110285	Nguyễn Thị Xuân Thanh	100%
19110257	Lê Trần Minh Nhựt	100%

Thời gian thực hiện: **Từ: 16/9/2022 Đến: 30/11/2022**

Khoa**: Đào tạo Chất lượng cao**

Ngành: **Công nghệ thông tin**

Tên đề tài: **Thiết kế và xây dựng website bán thực phẩm**

GVHD: PGS. TS Hoàng Văn Dũng

Lý do chọn đề tài

Hiện nay việc áp dụng những kỹ thuật bảo mật an toàn thông tin đang rất cần thiết nhằm đối phó với sự xâm nhập trái phép đang ngày càng gia tăng. Trong đó có cả việc bảo mật bằng khuôn mặt. Kỹ thuật nhận diện khuôn mặt đã ra đời từ rất lâu nhưng đến hiện nay nó mới được phổ biến vì thế giới đang phát triển hiện đại hoá. Bắt kịp theo xu hướng đó, chúng em quyết định nghiên cứu và áp dụng kỹ thuật nhận diện khuôn mặt trong tiểu luận chuyên ngành này.

Mục tiêu đề tài

- Tìm hiểu khái niệm và nguyên lí hoạt động của kỹ thuật nhận diện khuôn mặt.
- Xây dựng ứng dụng nhận diện khuôn mặt

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG

LỜI CẢM ƠN

Lời nói đầu tiên, nhóm thực hiện xin được gửi đến thầy Hoàng Văn Dũng – giảng viên hướng dẫn tiểu luận chuyên ngành lời cảm ơn chân thành và sâu sắc nhất.

Sau hơn bốn tháng tìm hiểu và nghiên cứu, nhóm em đã tiến hành xây dựng tiểu luận chuyên ngành để có thể tổng hợp các kiến thức đã học cũng như vận dụng chúng vào thực tế với đề tài "Tìm hiểu và ứng dụng nhận diện khuôn mặt".

Nhóm thực hiện xin cảm ơn sự quan tâm và giúp đỡ tận tình của thầy trong quá trình giảng dạy. Cảm ơn thầy đã luôn giải đáp những thắc mắc cũng như đưa ra những nhận xét, góp ý giúp nhóm thực hiện cải thiện chất lượng công việc của nhóm.

Vì khả năng còn hạn chế nên trong quá trình thực hiện báo cáo không tránh khỏi sai sót, kính mong nhận được những ý kiến đóng góp từ thầy để nhóm có thể cải thiện hơn sau này.

Nhóm thực hiện xin chân thành cảm ơn.

PHẦN 1 : MỞ ĐẦU

CHƯƠNG I : TỔNG QUAN ĐỀ TÀI

1.1. Tên đề tài

Đề tài : Tìm hiểu và ứng dụng nhận diện khuôn mặt

1.2. Tính cấp thiết của đề tài

Hiện nay, việc các hacker tấn công vào dữ liệu người dùng đang trở thành một vấn đề nan giải. Chỉ cần biết được mật mã là hacker đã có thể ung dung truy cập trái phép, đánh cắp hay thậm chí là chỉnh sửa, xoá hết tất cả dữ liệu. Theo thông kê từ Kaspersky cho thấy :

- 53% doanh nghiệp ở Đông Nam Á khi bị tấn công mạng phải bồi thường cho khách hàng, 51% gặp khó khăn trong việc thu hút khách hàng mới, 41% phải chi trả cho các khoản phạt và 30% bị mất đối tác kinh doanh.
- Hầu hết sự cố đều bị rò rỉ những thông tin khách hàng như thông tin nhận dạng cá nhân (53%), thông tin xác thực (33%), thanh toán hoặc thông tin về thẻ tín dụng (32%), số tài khoản (27%), và các dữ liệu cá nhân khác (26%)
- Cũng theo báo cáo, 30% thông tin cá nhân của nhân viên bị rò rỉ, ngoài ra còn có dữ liệu bảo mật của công ty (23%) và thông tin về sở hữu trí tuệ (16%).

Không chỉ doanh nghiệp mà cả người dùng cũng đang rất đau đầu vì vấn đề hacker này. Khi các hacker tấn công vào máy tính, điện thoại với hình thức nhập mật mã và đánh cắp hết tất cả những hình ảnh cá nhân rồi đe doạ tống tiền của người dùng.

Từ vấn đề trên, chúng em quyết định tìm hiểu và ứng dụng nhận diện khuôn mặt nhằm tìm ra hướng giải quyết tốt nhất. Thay vì phải nhập mật khẩu, chỉ với một thiết bị thông minh tích hợp camera nhận diện khuôn mặt là người dùng có thể dễ dàng truy cập vào máy tính, điện thoại mà không lo ngại việc bị hacker biết được mật mã.

1.3. Ý nghĩa khoa học và thực tiễn

Việc phát triển công nghệ nhận diện khuôn mặt đang là xu hướng của thế giới hiện nay khi nhu cầu về đời sống cũng như bảo mật an toàn đòi hỏi nâng cao. Công nghệ mang lại hiệu quả vô cùng to lớn khi khả năng của nó được áp dụng ở rất nhiều lĩnh vực từ bảo mật cho đến cả việc chụp ảnh. Cùng với sự ra đời của Windows Hello, FaceID, gần như một nửa thế giới đã và đang sử dụng công nghệ này trong nhiều trường hợp như bảo vệ dữ liệu và thông tin, ngăn ngừa tội phạm, bảo vệ địa điểm lẫn

các sự kiện quan trọng, mở khoá điện thoại, quảng cáo thông minh, tìm người mất tích, hỗ trợ người mù, chuẩn đoán y khoa, theo dõi hoạt động…. Người sử dụng có thể dễ dàng hơn trong việc kiểm soát, hệ thống sẽ nhận diện khuôn mặt một cách kín đáo bằng cách chụp ảnh khuôn mặt của những ai bước vào một khu vực được xác định từ camera giám sát.

Hầu như lĩnh vực nào cũng đều áp dụng được công nghệ nhận diện khuôn mặt chỉ với hai thiết bị camera và smartphone, laptop, pc.

1.4. Mục đích nghiên cứu

Công nghệ nhận diện khuôn mặt là một trong những tính năng bảo mật tốt nhất đáng để nghiên cứu và phát triển. Xây dựng ứng dụng nhận diện khuôn mặt giúp người dùng an tâm hơn trong vấn đề bảo mật mà không lo bị người khác nhóm ngó.

1.5. Đối tượng nghiên cứu

Đề tài tìm hiểu và ứng dụng nhận diện khuôn mặt.

1.6. Phạm vi nghiên cứu

Đề tài được áp dụng cho những người muốn tìm hiểu về công nghệ nhận diện khuôn mặt.

1.7. Công nghệ sử dụng

Machine Learning : Python

Xử lí ảnh : YOLOv5

CHƯƠNG II : KHẢO SÁT HIỆN TRẠNG

Trước khi bắt tay vào việc thiết kế một ứng dụng nhận diện khuôn mặt, chúng em đã khảo sát rất nhiều phần mềm nhận diện khuôn mặt có chung ý tưởng với nhóm chúng em. Sau đây là một số phần mềm, trang web được nhóm em tham khảo và lấy ý tưởng cho đề tài của chúng em.

2.1. Tính năng nhận diện khuôn mặt trên điện thoại Iphone

Iphone là dòng điện thoại nổi tiếng được phát triển bởi hang Apple. Iphone hiện đang được đánh giá là một trong những dòng điện thoại mà người tiêu dùng sử dụng selfie, chụp ảnh nhiều nhất thế giới. Iphone tích hợp tính năng nhận diện khuôn mặt trong ứng dụng chụp ảnh để tự động lấy nét và cân bằng khuôn mặt phù hợp hơn. Từ Iphone X, Apple lại tiếp tục phát triển tính năng này trong chức năng bảo mật có tên gọi là FaceID. FaceID là tính năng mở khoá điện thoại thông qua khuôn mặt.

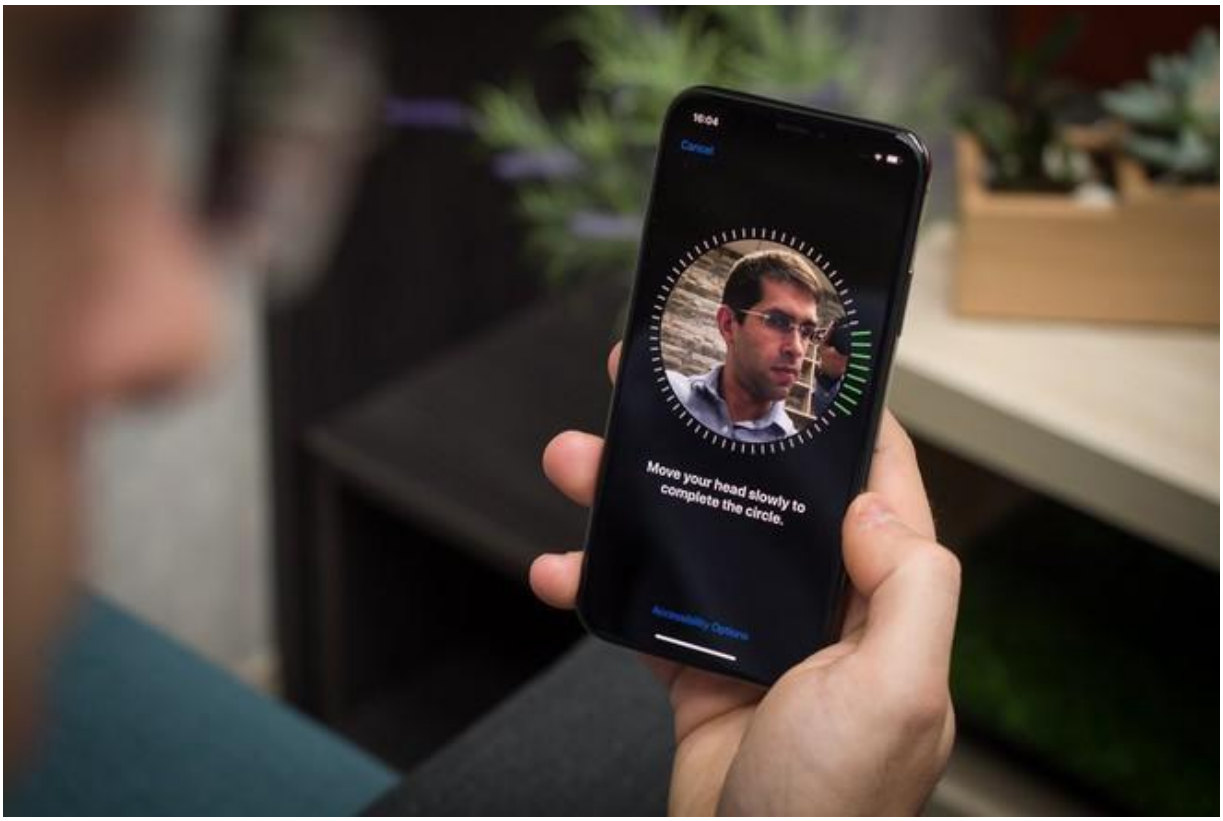
Nguyên lý hoạt động của tính năng nhận diện khuôn mặt này được phát triển dựa trên các thuật toán về hình học và trắc quang. Các bước tiến hành nhận diện khuôn mặt như sau : quét toàn bộ vật thể trong khung hình, thực hiện cơ chế nhận diện thông qua màu da, màu nhiệt, mắt mũi miệng để tìm ra khuôn mặt.

Hình 1.1: Tính năng FaceID trên dòng điện thoại Iphone

2.2. Tính năng nhận diện khuôn mặt trên điện thoại Android

Cũng tương tự như Iphone, những chiếc điện thoại sự dụng hệ điều hành Android cũng ứng dụng nhận diện khuôn mặt vào tính năng chụp ảnh và bảo mật từ phiên bản Android 10 và Android Q. Khác với Iphone, Andriod sử dụng thêm nhận diện khuôn mặt vào tính năng thanh toán trực tuyến. Nhưng nhận diện khuôn mặt trên Android lại có nhược điểm khá lớn là chỉ dùng hình ảnh 2D để nhận diện khiến cho Android dễ bị đánh lừa khi đưa một bức ảnh 2D vào thay vì khuôn mặt gốc.

Hiện nay, Android đã khắc phục được nhược điểm trên bằng cách áp dụng camera TrueDepth giống của Apple.



Hình 1.2: Nhận dạng khuôn mặt trên hệ điều hành Android

2.3. Phần mềm BioID

Đây là ứng dụng được phát triển trên nền tảng IOS và Android. Ứng dụng sử dụng sinh trắc học để xác minh danh tính và cho phép người dùng đăng nhập vào ứng dụng hay trình duyệt được thiết lập bảo mật trên điện thoại thông qua khuôn mặt.

Các tính năng của BioID là :

- phân biệt được ảnh video với mặt thật thông qua việc phát hiện độ sống động trên từng nét ảnh.
- Phân tích các chuyển động và độ dài để đảm bảo sự diện của người dùng.

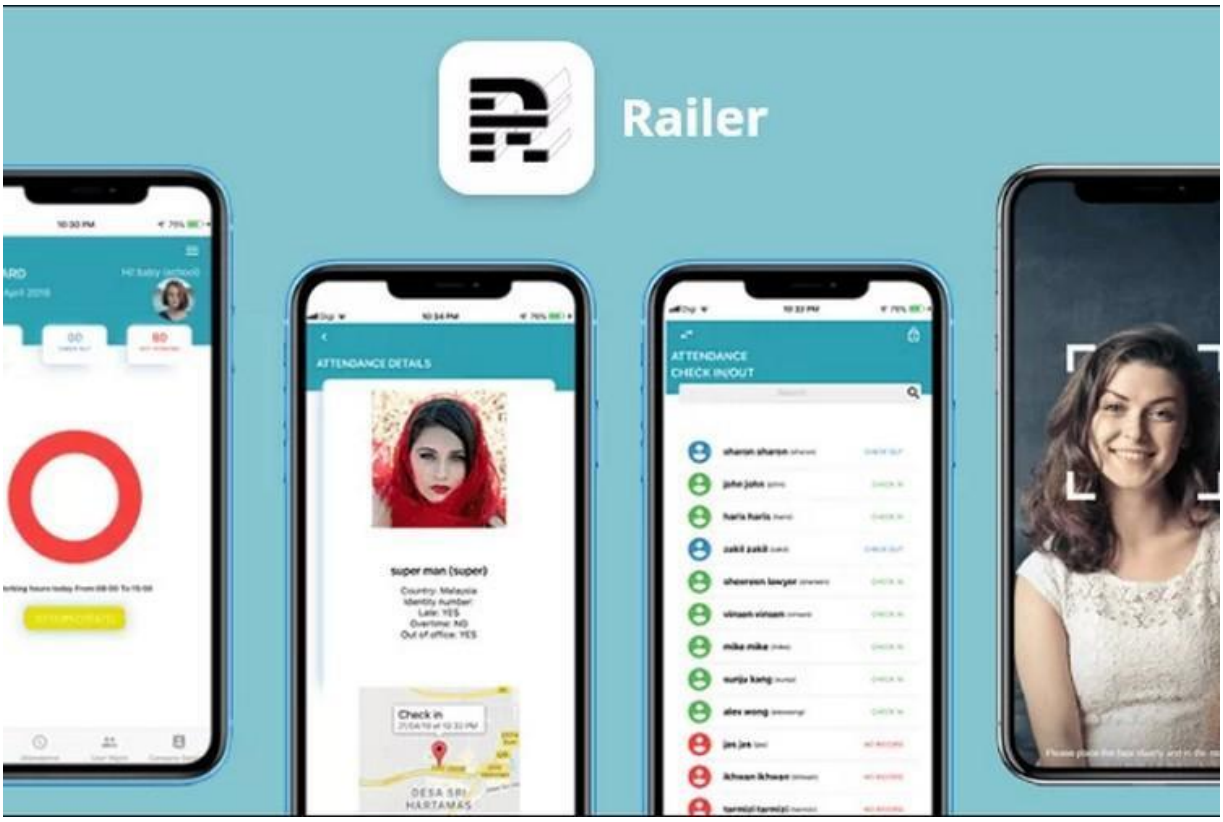
Hình 1.3: Ứng dụng BioID

2.4. Phần mềm Railer

Railer là ứng dụng chấm công áp dụng kỹ thuật nhận diện khuôn mặt được phát triển trên nền tảng di động. Đây được coi là một trong những ứng dụng nhận dạng hay được sử dụng để theo dõi hoạt động của nhân viên.

Các tính năng của Railer :

- Nhân viên sử dụng khuôn mặt để đăng nhập và đăng xuất.
- Tự động check-in và check-out khi nhận diện chính xác khuôn mặt nhân viên.
- Cung cấp phân tích và báo cáo cho người sử dụng.

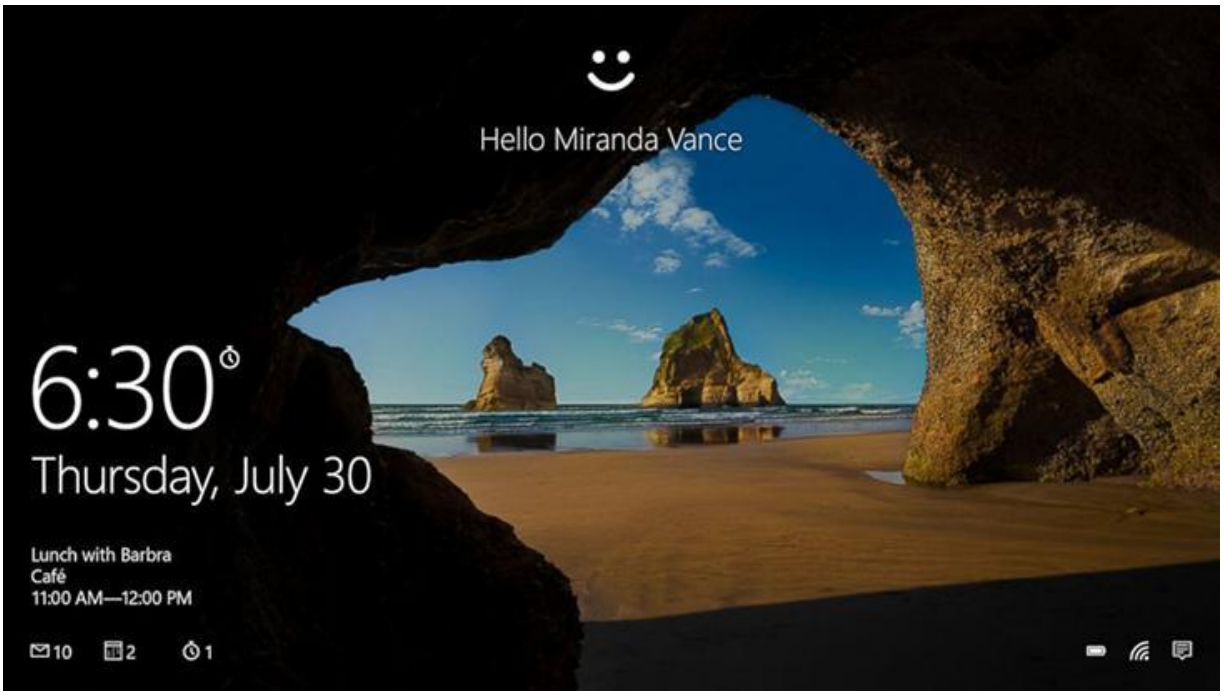


Hình 1.4: Ứng dụng chấm công Railer

2.5. Windows Hello của Microsoft

Windows Hello là tính năng bảo mật áp dụng kỹ thuật nhận diện khuôn mặt của hang Microsoft. Hiện nay, tính năng này đang được hỗ trợ trên window 10 và các dòng diện thoại Lumia.

Windows Hello được ra mắt trước cả FaceID của Apple nên được xem là người tiên phong trong việc áp dụng kỹ thuật nhận diện khuôn mặt. Windows Hello được xem là một đối thủ đáng gờm của FaceID khi vượt trội cả về độ nhạy, độ chính xác và tốc độ xác minh người dùng. Nhưng Windows Hello không được phổ biến như FaceID vì nó chỉ áp dụng trên một vài thiết bị laptop.



Hình 1.5: Windows Hello của Microsoft

PHẦN 2 : NỘI DUNG CHÍNH

CHƯƠNG III : CƠ SỞ LÝ THUYẾT

3.1. Nhận diện khuôn mặt (Face Recognition)

Công nghệ nhận diện khuôn mặt là một ứng dụng tự động xác định nhận dạng người từ một bức ảnh kỹ thuật số hoặc từ một khung hình video. Một trong những cách nhận diện của nó là so sánh các đặc điểm khuôn mặt từ hình ảnh và một cơ sở dữ liệu về khuôn mặt. Đa phần công nghệ này được sử dụng trong các hệ thống an ninh và sử dụng làm một dạng sinh trắc học nhận dạng trên các thiết bị thông minh hiện nay. Nó được ứng dụng với nhiều công việc như tìm người, xác định danh tính, bảo mật điện thoại.

Cách thức thực hiện để nhận diện một khuôn mặt :

Bước 1 : Thực hiện Face Detection với ảnh cần nhận dạng

Bước 2 : Training khuôn mặt được nhận dạng thành model

Bước 3 : So sánh ảnh cần nhận dạng với khuôn mặt đã được training trong một dataset

Bước 4 : Nhận dạng khuôn mặt thông qua việc so sánh ở bước 3 bằng cách trả về tên người và độ chính xác của khuôn mặt đó.

3.2. Phát hiện khuôn mặt (Face Detection)

Công nghệ phát hiện khuôn mặt được xem là bước đầu tiên của nhận diện khuôn mặt. Nếu nhận diện khuôn mặt là để xác định danh tính người thì phát hiện khuôn mặt dùng để phát hiện bất kì đối tượng nào là người trong một khung ảnh.

3.3. Tổng quan về YOLO

YOLO (You Only Look Once) là một trong những thư viện mở dùng để nhận dạng đối tượng trong ảnh và video có tốc độ xử lí rất nhanh với mức độ chính xác trên 80% ở thời điểm hiện tại. YOLO được tạo ra từ việc kết hợp giữa các convolutional layers và connected layers. Trong đóp các convolutional layers sẽ trích xuất ra các feature của ảnh, còn full-connected layers sẽ dự đoán ra xác suất đó và tọa độ của đối tượng.

YOLO là thuật toán object detection nên mục tiêu của mô hình không chỉ là dự báo nhãn cho vật thể như các bài toán classification mà nó còn xác định location của vật

thể. Do đó YOLO có thể phát hiện được nhiều vật thể có nhãn khác nhau trong một bức ảnh thay vì chỉ phân loại duy nhất một nhãn cho một bức ảnh

Tính đến thời điểm hiện tại YOLO đã có tổng cộng 5 phiên bản(v1,v2,v3,v4,v5). Trong đó bản v5 là bản mới nhất, khác phục được các nhược điểm của các phiên bản trước như: lỗi về việc xác định vị trí của vật thể, các ràng buộc về không gian trên những bounding box, mỗi grid cell chỉ có thể predict rất ít bounding box,...

YOLOv1 được ra mắt vào năm 2016 với tựa đề "You Only Look Once : Unified, Real-Time Object Detection" nhằm giải quyết bài toán Real Time Object Detection. Cách thức hoạt động của YOLOv1 thường có 2 bước : sử dụng sliding window lấy vùng khác nhau của ảnh và phân loại vị trí của đối tượng để sắp xếp đối tượng thuộc lớp nào.

Vào năm 2017 thì "YOLO9000: Better, Faster, Stronger" và YOLOv2 được giới thiệu với độ chính xác đạt 76,8%. YOLO9000 được áp dụng thêm kỹ thuật Batch Normalization nhằm giảm thời gian training và tăng tính phổ quát cho mạng. Kỹ thuật này cũng giúp YOLOv2 tăng mAP (giá trị trung bình của độ tin cậy của những class khác nhau) lên xấp xỉ 2%

Và một năm sau đó, YOLOv3 được xem là một phiên bản được kì vọng nhất với nhiều cải tiến mới. Những cải tiến đó bao gồm : Logistic Regression, Logistic Classifier rời rạc, Darknet 53, Multi-scale Prediction, Skip-layer Concatenation.

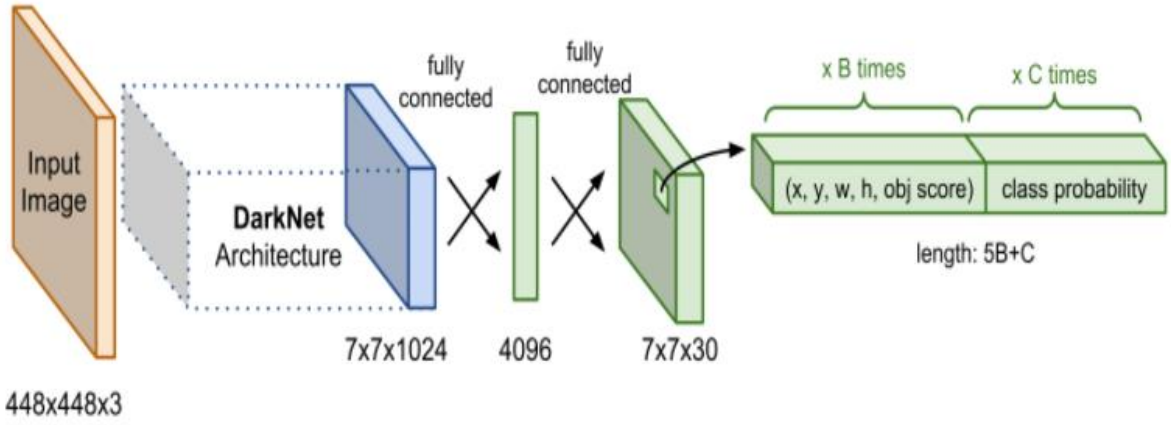
Ở YOLOv4, độ chính xác và tốc độ được tăng cao hơn so với YOLOv3 trên cùng một tập dataset và CPU. Cấu trúc của YOLOv4 bao gồm : Backbone, Neck, Heads. Backbone là một kiến trúc Deep Learning hoạt động như một Feature Exactor. Tất cả các backbone cơ bản chính là Classification Models. Kiến trúc của backbone có 3 thành phần chính là BoF (Bag of Freebies), BoS (Bag of Specials), và CSPDarknet53.

Đến hiện tại YOLOv5 ra đời và không có quá nhiều thay đổi so với YOLOv4. Bởi vì YOLOv5 tập trung vào tốc độ và tính tiện dụng.

3.1.1. Kiến trúc mạng

Kiến trúc YOLO bao gồm: Base network là các mạng convolution làm nhiệm vụ trích xuất đặc trưng. Phần phía sau là những Extra Layers được áp dụng để phát hiện vật thể trên feature map của base network.

Base network của YOLO sử dụng chủ yếu là các convolutional layer và các fully conntected layer. Các kiến trúc YOLO cũng khá đa dạng và có thể tùy biến thành các version cho nhiều input shape khác nhau.



Hình 2.1: Kiến trúc mạng YOLO

Thành phần Darknet Architechture được gọi là base network có tác dụng trích suất đặc trưng. Output của base network là một feature map có kích thước `7x7x1024` sẽ được sử dụng làm input cho các Extra layers có tác dụng dự đoán nhãn và tọa độ bounding box của vật thể.

Ở phiên bản thứ 3 của YOLO tức là YOLOv3 tác giả áp dụng một mạng feature extractor là darknet-53. Mạng này gồm 53 convolutional layers kết nối liên tiếp, mỗi layer được theo sau bởi một batch normalization và một activation Leaky Relu. Để giảm kích thước của output sau mỗi convolution layer, tác giả down sample bằng các filter với kích thước là 2. Mẹo này có tác dụng giảm thiểu số lượng tham số cho mô hình.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Hình 2.2: Các layer trong mạng darknet-53

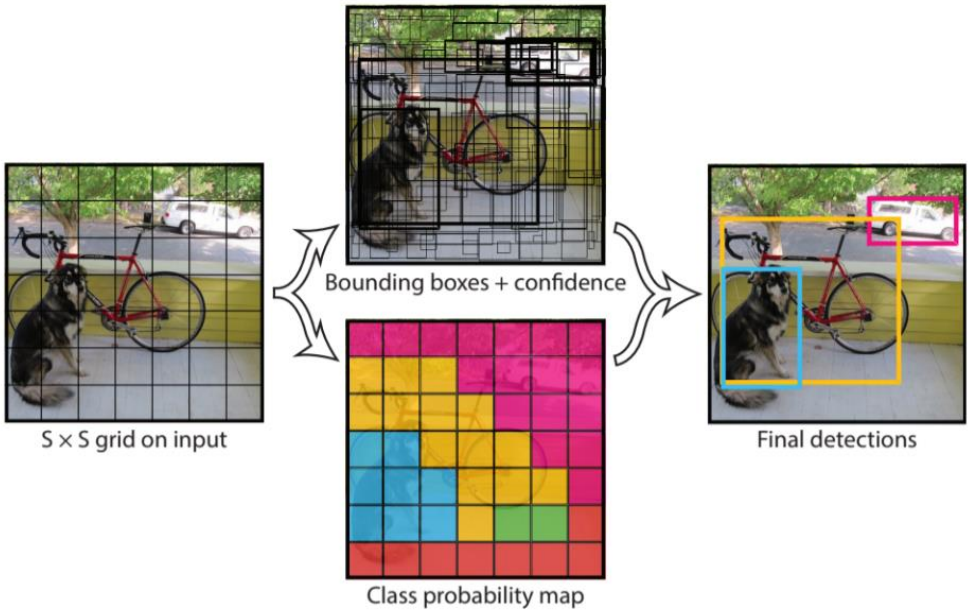
Các bức ảnh khi được đưa vào mô hình sẽ được scale để về chung một kích thước phù hợp với input shape của mô hình và sau đó được gom lại thành batch đưa vào huấn luyện.

Hiện tại YOLO đang hỗ trợ 2 đầu vào chính là 416x416 và 608x608. Mỗi một đầu vào sẽ có một thiết kế các layers riêng phù hợp với shape của input. Sau khi đi qua các layer convolutional thì shape giảm dần theo cấp số nhân là 2. Cuối cùng ta thu được một feature map có kích thước tương đối nhỏ để dự báo vật thể trên từng ô của feature map.

Kích thước của feature map sẽ phụ thuộc vào đầu vào. Đối với input 416x416 thì feature map có các kích thước là 13x13, 26x26 và 52x52. Và khi input là 608x608 sẽ tạo ra feature map 19x19, 38x38, 72x72.

3.1.2. Nguyên lí hoạt động

Đầu vào của mô hình là một ảnh, mô hình sẽ nhận dạng ảnh đó có đối tượng nào hay không, sau đó sẽ xác định tọa độ của đối tượng trong bức ảnh. Ảnh đầu vào được chia thành SxS ô thường thì sẽ là 3x3,7x7,9x9,…. Việc chia ô có ảnh hưởng đến việc phát hiện đối tượng của mô hình.



Hình 2.3: Cách hoạt động của mạng YOLO

Với Input là 1 ảnh, đầu ra mô hình là một ma trận 3 chiều có kích thước SxSx(5 x N + M) với số lượng tham số mỗi ô là (5 x N + M) với N và M lần lượt là số lượng Box và Class mà mỗi ô cần dự đoán. Xét ví dụ ở hình trên chia thành 7x7 ô, mỗi ô cần dự đoán 2 bounding box và 3 object: con chó, ô tô, xe đạp thì output sẽ là 7x7x13, mỗi ô sẽ có 13 tham số, cho kết quả trả về (7x7x2 =98) bounding box.

3.1.3. Kết quả output

Output của mô hình YOLO là một véc tơ sẽ bao gồm các thành phần:

$$y^T = [\rho_0, \underbrace{\langle t_x, t_y, t_w, t_y \rangle}_{boundingbox}, \langle \underbrace{p_1, p_2, \dots . p_c}_{score\ of\ c\ classes} \rangle]$$

Trong đó:

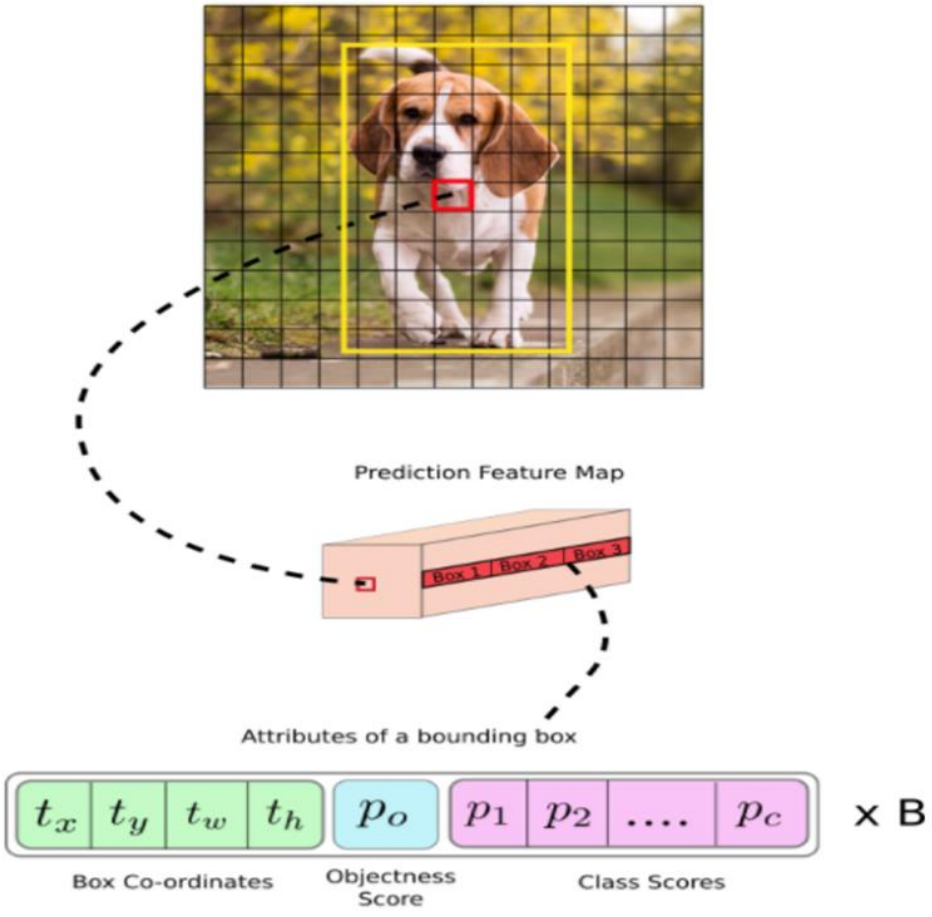
p_0 : là xác suất dự báo vật thể xuất hiện trong bounding box.

$\underbrace{\langle t_x, t_y, t_w, t_y \rangle}_{boundingbox}$: giúp xác định bounding box. Trong đó t_x,t_y là tọa độ tâm và t_w,t_h là kích thước rộng, dài của bounding box.

$\underbrace{p_1, p_2, \ldots . p_c}_{score\ of\ c\ classes}$: là véc tơ phân phối xác suất dự báo của các classes.

Việc hiểu output khá là quan trọng để chúng ta cấu hình tham số chuẩn xác khi huấn luyện model qua các open source như darknet. Như vậy output sẽ được xác định theo số lượng classes theo công thức (n_class+5). Nếu huấn luyện 80 classes thì bạn sẽ có output là 85. Trường hợp bạn áp dụng 3 anchors/cell thì số lượng tham số output sẽ là:

(n_class + 5) x3= 85x3=255

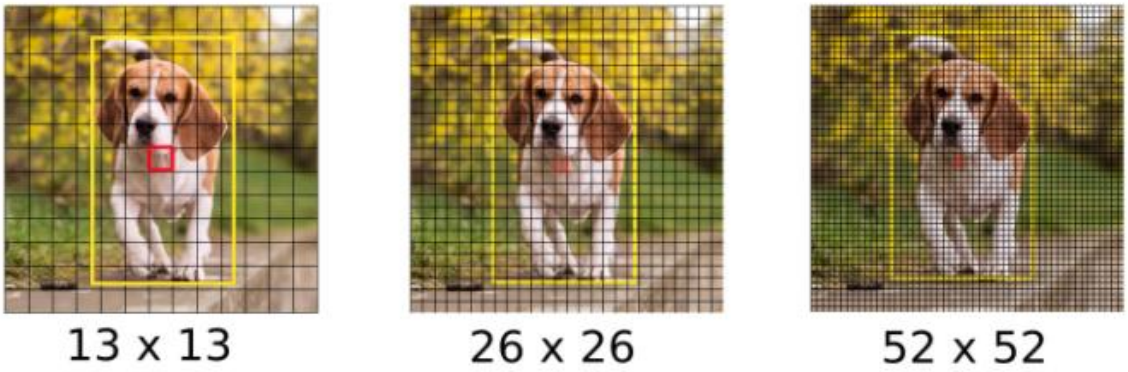


Hình 2.3: Kiến trúc một output của model YOLO

Hình ảnh gốc là một feature map kích thước 13x13. Trên mỗi một cell của feature map chúng ta lựa chọn ra 3 anchor boxes với kích thước khác nhau lần lượt là Box 1, Box 2, Box 3 sao cho tâm của các anchor boxes trùng với cell. Khi đó output của YOLO là một véc tơ concatenate của 3 bounding boxes. Các attributes của một bounding box được mô tả như dòng cuối cùng trong hình.

a) Dự báo trên nhiều feature map

Cũng tương tự như SSD, YOLO (cụ thể hơn là YOLOv3) dự báo trên nhiều feature map. Những feature map ban đầu có kích thước nhỏ giúp dự báo được các object kích thước lớn. Những feature map sau có kích thước lớn hơn trong khi anchor box được giữ cố định kích thước nên sẽ giúp dự báo các vật thể kích thước nhỏ.



Hình 2.5: Các feature maps của mạng YOLOv3 với input shape là 416x416, output là 3 feature maps có kích thước lần lượt là 13x13, 26x26 và 52x52.

Trên mỗi một cell của các feature map chúng ta sẽ áp dụng 3 anchor box để dự đoán vật thể. Như vậy số lượng các anchor box khác nhau trong một mô hình YOLO sẽ là 9 (3 featue map x 3 anchor box).

Đồng thời trên một feature map hình vuông SxS, mô hình YOLOv3 sinh ra một số lượng anchor box là: SxSx3. Như vậy số lượng anchor boxes trên một bức ảnh sẽ là:

$$(13x13+26+52x52)x3=10647(\text{ anchor box})$$

Đây là một số lượng rất lớn và là nguyên nhân khiến quá trình huấn luyện mô hình YOLO vô cùng chậm bởi chúng ta cần dự báo đồng thời nhãn và bounding box trên đồng thời 10647 bounding boxes.

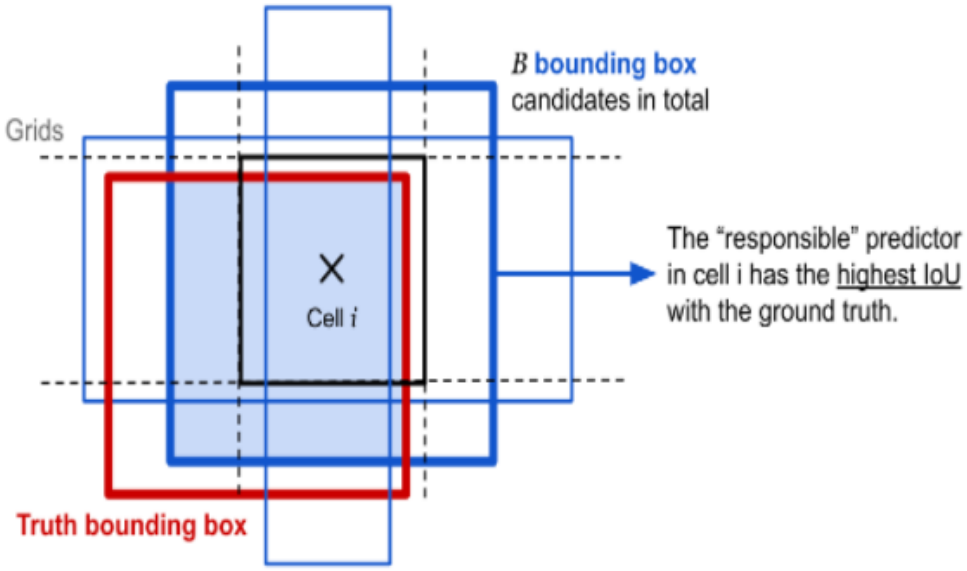
Một số lưu ý khi huấn luyện YOLO:

- Khi huấn luyện YOLO sẽ cần phải có RAM dung lượng lớn hơn để save được 10647 bounding boxes như trong kiến trúc này.
- Không thể thiết lập các batch_size quá lớn như trong các mô hình classification vì rất dễ Out of memory. Package darknet của YOLO đã chia nhỏ một batch thành các subdivisions cho vừa với RAM.
- Thời gian xử lý của một step trên YOLO lâu hơn rất rất nhiều lần so với các mô hình classification. Do đó nên thiết lập steps giới hạn huấn luyện cho YOLO nhỏ. Đối với các tác vụ nhận diện dưới 5 classes, dưới 5000 steps là có thể thu được nghiệm tạm chấp nhận được. Các mô hình có nhiều classes hơn có thể tăng số lượng steps theo cấp số nhân tùy người dùng.

b) Anchor box

Để tìm được bounding box cho vật thể, YOLO sẽ cần các anchor box làm cơ sở ước lượng. Những anchor box này sẽ được xác định trước và sẽ bao quanh vật thể một cách tương đối chính xác. Sau này thuật toán regression bounding box sẽ tinh chỉnh lại anchor box để tạo ra bounding box dự đoán cho vật thể. Trong một mô hình YOLO:

- Mỗi một vật thể trong hình ảnh huấn luyện được phân bố về một anchor box. Trong trường hợp có từ 2 anchor boxes trở lên cùng bao quanh vật thể thì ta sẽ xác định anchor box mà có IoU với ground truth bounding box là cao nhất.



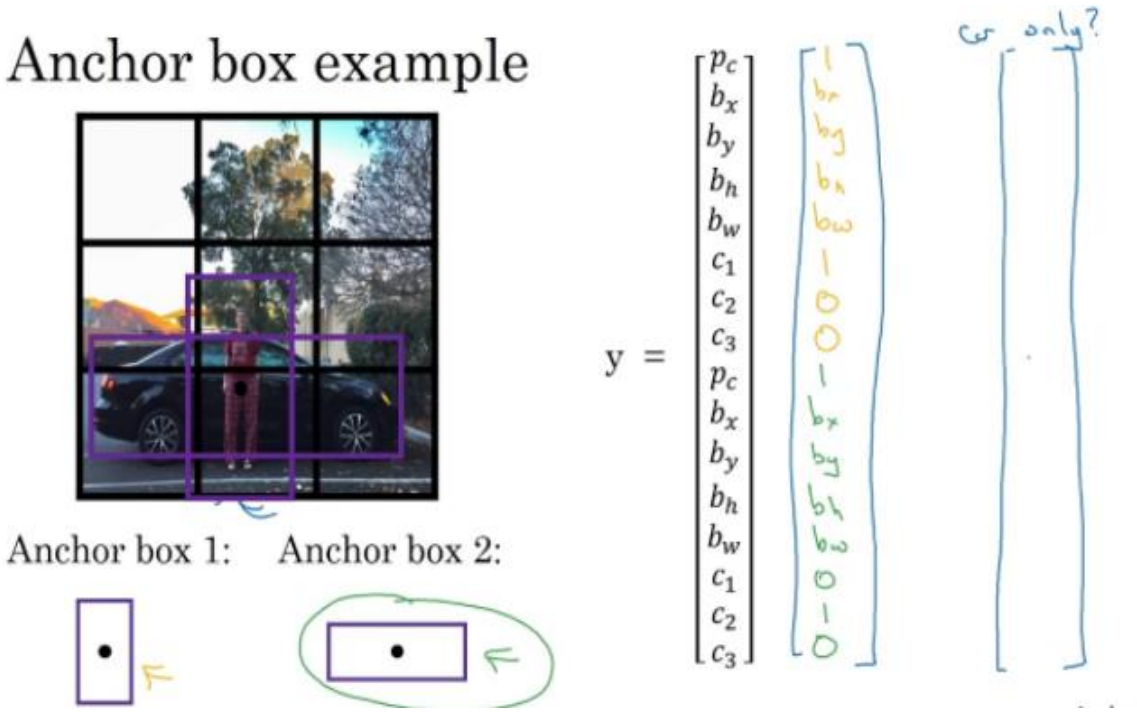
Hình 2.4: Xác định anchor box cho một vật thể

Từ Cell i ta xác định được 3 anchor boxes viền xanh như trong hình. Cả 3 anchor boxes này đều giao nhau với bounding box của vật thể. Tuy nhiên chỉ anchor box có đường viền dày nhất màu xanh được lựa chọn làm anchor box cho vật thể bởi nó có IoU so với ground truth bounding box là cao nhất.

- Mỗi một vật thể trong hình ảnh huấn luyện được phân bố về một cell trên feature map mà chứa điểm mid point của vật thể. Chẳng hạn như hình chú chó trong hình 3 sẽ được phân về cho cell màu đỏ vì điểm mid point của ảnh chú chó rơi vào đúng cell này. Từ cell ta sẽ xác định các anchor boxes bao quanh hình ảnh chú chó.

Như vậy khi xác định một vật thể ta sẽ cần xác định 2 thành phần gắn liền với nó là (cell, anchor box). Không chỉ riêng mình cell hoặc chỉ mình anchor box.

Một số trường hợp 2 vật thể bị trùng mid point, mặc dù rất hiếm khi xảy ra, thuật toán sẽ rất khó xác định được class cho chúng.



Hình 2.5: Khi 2 vật thể người và xe trùng mid point và cùng thuộc một cell. Thuật toán sẽ cần thêm những lượt tiebreak để quyết định đâu là class cho cell.

c) Hàm mất mát (Loss Function)

Sau khi đã định nghĩa được những thông tin mà mô hình cần phải dự đoán, và kiến trúc của mô hình CNN. Bây giờ là lúc mà chúng ta sẽ định nghĩa hàm lỗi.

YOLO sử dụng hàm độ lỗi bình phương giữ dự đoán và nhãn để tính độ lỗi cho mô hình. Cụ thể, độ lỗi tổng của chúng ta sẽ là tổng của 3 độ lỗi con sau:

- Độ lỗi của việc dữ đoán loại nhãn của Object-Classification loss
- Độ lỗi của dự đoán tạo độ cũng như chiều dài, rộng của boundary box - Localization loss
- Độ lỗi của ô vuông có chứa object nào hay không - Confidence loss

Chúng ta mong muốn hàm lỗi có chức năng sau:

- Trong quá trình huấn luyện, mô hình sẽ nhìn vào những ô vuông có chứa object. Tăng classification score lớp đúng của object đó lên.

- Sau đó, cũng nhìn vào ô vuông đó, tìm boundary box tốt nhất trong 2 boxes được dự đoán.

- Tăng localization score của boundary box đó lên, thay đổi thông tin boundary box để gần đúng với nhãn. Đối với những ô vuông không chứa object, giảm confidence score và chúng ta sẽ không quan tâm đến classification score và localization score của những ô vuông này.

d) Classification Loss

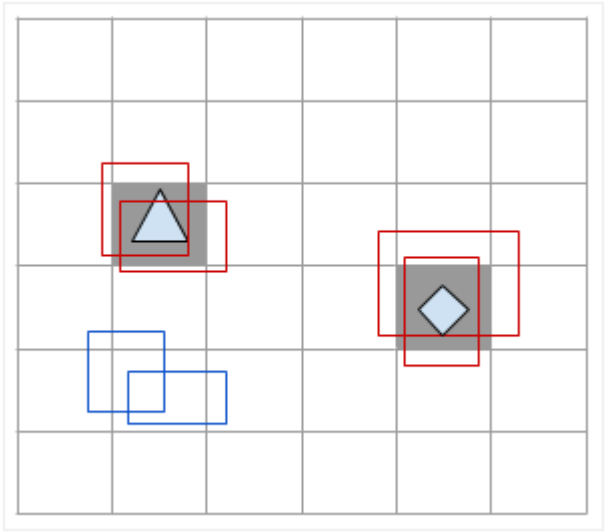
Chúng ta chỉ tính classification loss cho những ô vuông được đánh nhãn là có object. Classification loss tại những ô vuông đó được tính bằng đỗ lỗi bình phương giữa nhãn được dự đoán và nhãn đúng của nó.

$$L_{classification} = \sum_{i=0}^{s^2} II_i^{obj} \sum_{c \in class} (p_i(c) - \hat{p}_i(c))^2$$

Với:

II_i^{obj}: bằng 1 nếu ô vuông đang xét có object, ngược lại bằng 0

$\hat{p}_i(c)$: là xác xuất có điều của lớp c tại ô vuông tương ứng mà mô hình dự đoán



Hình 2.6: Tính toán Loss Function cho 2 object: tam giác và hình thoi.

Ví dụ, trong hình minh họa ở trên, chúng ta có 2 object tại ô vuông (dòng,cột) là (2,1) và (3,4), chứa object là hình tam giác và hình tức giác đều. Độ lỗi classification loss chỉ tính cho 2 object này mà ko quan tâm đến những ô vuông

khác. Lúc cài đặt chúng ta cần lưu ý phải nhân với một mask để triệt tiêu giá trị lỗi tại những ô vuông ko quan tâm.

e) Localization Loss

Localization loss dùng để tính giá trị lỗi cho boundary box được dự đoán bao gồm offset x,y và chiều dài, rộng so với nhãn chính xác của chúng ta. Các bạn nên lưu ý rằng, chúng ta không tính toán trực tiếp giá trị lỗi này trên kích thước của ảnh mà cần chuẩn dưới kính thước ảnh về đoạn [0-1] đối với tọa độ điểm tâm, và không dữ đoán trực tiếp điểm tâm mà phải dự đoán giá trị lệch offset x,y so với ô vuông tương ứng. Việc chuẩn hóa kích thước ảnh và dự đoán offset làm cho mô hình nhanh hội tụ hơn so với việc dự đoán giá trị mặc định.

$$L_{localization} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \prod_{ij}^{obj} [(offsetx_i - \hat{offsetx_i})^2 + (offsety_i - \hat{offsety_i})^2 + (width_i - \hat{width_i})^2 + (height_i - \hat{height_i})^2$$

Độ lỗi localization loss được tính bằng tổng đỗ lỗi bình phương của offsetx, offsety và chiều dài, rộng trên tất cả các ô vuông có chứa object. Tại mỗi ô vuông đúng, ta chọn 1 boundary box có IoU (Intersect over union) tốt nhất, rồi sau đó tính độ lỗi theo các boundary box này. Theo hình mình họa trên chúng ta có 4 boundary box tại ô vuông đúng có viền màu đỏ, chúng ta chọn 1 box tại mỗi ô vuông để tính độ lỗi. Còn box xanh được bỏ qua.

Localization loss là độ lỗi quan trọng nhất trong 3 loại độ lỗi trên. Do đó, ta cần đặt trọng số cao hơn cho độ lỗi này.

f) Confidence Loss

Confidence loss thể hiện độ lỗi giữa dự đoán boundary box đó chứa object so với nhãn thực tế tại ô vuông đó. Độ lỗi này tính nên cả những ô vuông chứa object và không chứa object.

$$L_{confidence} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \prod_{ij}^{obj} (C_i - \hat{C_i})^2 + \lambda_{noobject} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \prod_{ij}^{noobj} (C_i - \hat{C_i})^2$$

Độ lỗi này là độ lỗi bình phường của dự đoán boundary đó chứa object với nhãn thực tế của ô vuông tại vị trí tương ứng, chúng ta lưu ý rằng, độ lỗi tại ô

vuông mà nhãn chứa object quan trọng hơn là độ lỗi tại ô vuông không chứa object, do đó chúng ta cần sử dụng hệ số lambda để cân bằng điều này.

Tổng kết lại, tổng lỗi của chúng ta sẽ bằng tổng của 3 loại độ lỗi trên

$$L_{total} = L_{classification} + L_{localization} + L_{confidence}$$

3.1.4. Dự báo bounding box

Để dự báo bounding box cho một vật thể chúng ta dựa trên một phép biến đổi từ anchor box và cell.

YOLOv2 và YOLOv3 dự đoán bounding box sao cho nó sẽ không lệch khỏi vị trí trung tâm quá nhiều. Nếu bounding box dự đoán có thể đặt vào bất kỳ phần nào của hình ảnh, như trong mạng regional proposal network, việc huấn luyện mô hình có thể trở nên không ổn định.

Cho một anchor box có kích thước (p_w ,p_h) tại cell nằm trên feature map với góc trên cùng bên trái của nó là (C_x , C_y) mô hình dự đoán 4 tham số (t_x,t_y,t_w,t_h) trong đó 2 tham số đầu là độ lệch (offset) so với góc trên cùng bên trái của cell và 2 tham số sau là tỷ lệ so với anchor box. Và các tham số này sẽ giúp xác định bounding box dự đoán b có tâm (b_x,b_y) à kích thước (b_w,b_h) thông qua hàm sigmoid và hàm exponential như các công thức bên dưới:

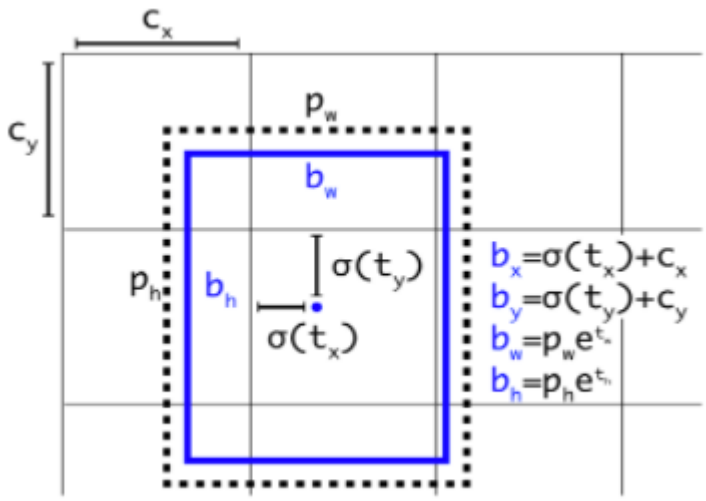
$$b_x=\sigma(t_x) + c_x$$

$$b_y=\sigma(t_y) + c_y$$

$$b_w= \rho_w e^{t_w}$$

$$b_h= \rho_h e^{t_h}$$

Ngoài ra do các tọa độ đã được hiệu chỉnh theo width và height của bức ảnh nên luôn có giá trị nằm trong ngưỡng [0, 1]. Do đó khi áp dụng hàm sigmoid giúp ta giới hạn được tọa độ không vượt quá xa các ngưỡng này.



Hình 2.7: Công thức ước lượng bounding box từ anchor box

Hình chữ nhật nét đứt bên ngoài là anchor box có kích thước là (p_w,p_h). Tọa độ của một bounding box sẽ được xác định dựa trên đồng thời cả anchor box và cell mà nó thuộc về. Điều này giúp kiểm soát vị trí của bounding box dự đoán đâu đó quanh vị trí của cell và bounding box mà không vượt quá xa ra bên ngoài giới hạn này. Do đó quá trình huấn luyện sẽ ổn định hơn rất nhiều so với YOLOv1.

a) Non-max suppression

Do thuật toán YOLO dự báo ra rất nhiều bounding box trên một bức ảnh nên đối với những cell có vị trí gần nhau, khả năng các khung hình bị overlap là rất cao. Trong trường hợp đó YOLO sẽ cần đến non-max suppression để giảm bớt số lượng các khung hình được sinh ra một cách đáng kể.

Hình 2.8: Non-max suppression. Từ 3 bounding box ban đầu cùng bao quanh chiếc xe đã giảm xuống còn một bounding box cuối cùng.

Các bước của non-max suppression:

- Bước 1: Đầu tiên chúng ta sẽ tìm cách giảm bớt số lượng các bounding box bằng cách lọc bỏ toàn bộ những bounding box có xác suất chứa vật thể nhỏ hơn một ngưỡng threshold nào đó, thường là 0.5.

- Bước 2: Đối với các bouding box giao nhau, non-max suppression sẽ lựa chọn ra một bounding box có xác xuất chứa vật thể là lớn nhất. Sau đó tính toán chỉ số giao thoa IoU với các bounding box còn lại.

Nếu chỉ số này lớn hơn ngưỡng threshold thì điều đó chứng tỏ 2 bounding boxes đang overlap nhau rất cao. Ta sẽ xóa các bounding có có xác xuất thấp hơn và giữ lại bouding box có xác xuất cao nhất. Cuối cùng, ta thu được một bounding box duy nhất cho một vật thể.

3.1.5. Thuật toán sử dụng trong YOLOv5

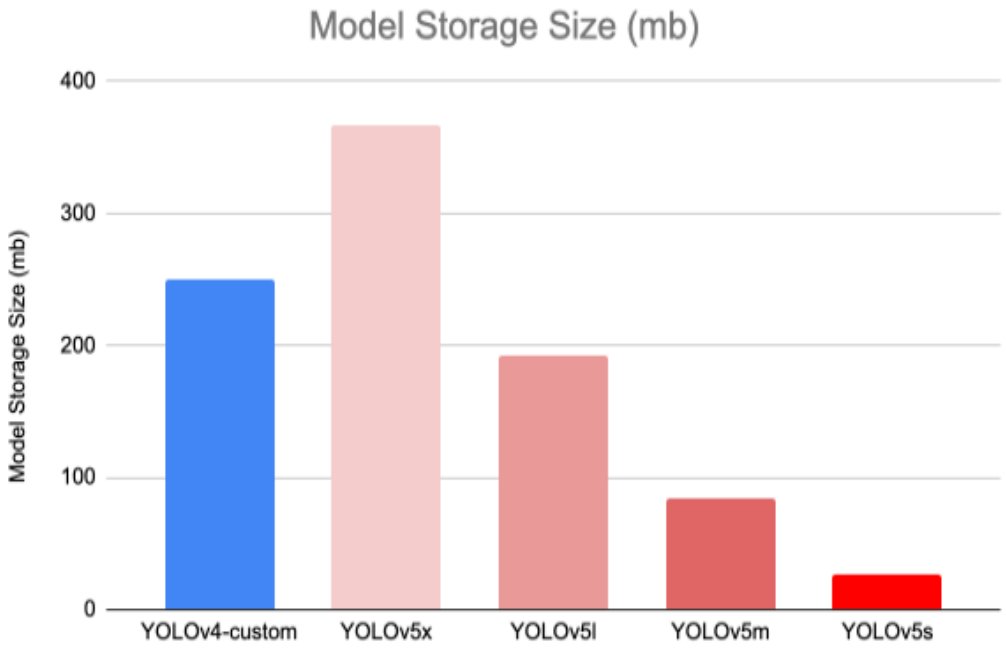
YOLOv5 là một bản cải tiến mang tính mở rộng theo một cách tự nhiên của YOLOv3 PyTorch bởi tác giả Glenn Jocher. Kho lưu trữ YOLOv3 PyTorch là điểm đến phổ biến cho các nhà phát triển để chuyển các trọng số YOLOv3 Darknet sang PyTorch và sau đó chuyển sang sản xuất. Những cải tiến này ban đầu được gọi là YOLOv4 nhưng do việc phát hành gần đây của YOLOv4 trong khuôn khổ Darknet, để tránh xung đột phiên bản, nó đã được đổi tên thành YOLOv5.

Thuật toán YOLOv5 về cơ bản cũng thừa kế các phương pháp cơ bản của các YOLO, tuy nhiên YOLOv5 áp dụng một số thuật toán phát hiện vật thể nhanh, tối ưu hóa các phép toán thực hiện song song giúp tăng tốc độ nhận diện và giảm thời gian huấn luyện một cách tối ưu.

a) Phân loại

Có 4 mô hình khác nhau: YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x. Đầu tiên là nhỏ nhất và kém chính xác nhất, cuối cùng là lớn nhất với độ chính xác lớn nhất. Tất cả các mô hình đều chạy trên PyTorch.

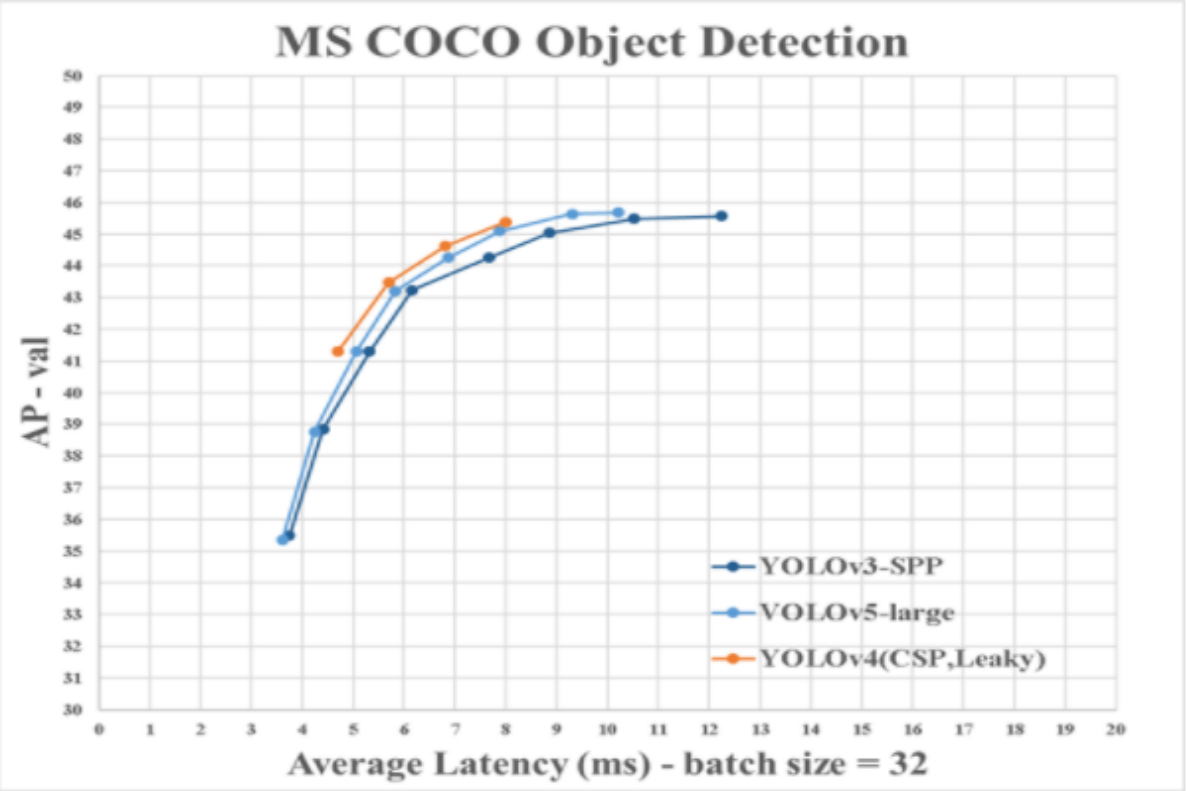
Về hiệu năng: YOLOv5 nhỏ hơn và thường dễ sử dụng hơn trong sản xuất. Do nó được triển khai nguyên bản trong PyTorch (chứ không phải Darknet), việc sửa đổi kiến trúc và rất đơn giản trong việc xuất sang nhiều môi trường triển khai khác nhau.



Hình 2.9: So sánh kích thước lưu trữ Model của các mẫu mã YOLOv5

Về tốc độ : YOLOv5 thực hiện suy luận hàng loạt ở khoảng 140 FPS theo mặc định.

Về độ chính xác: YOLOv5 gần như chính xác như YOLOv4 trong các tác vụ nhỏ (0,895 mAP so với 0,892 mAP trên BCCD). Trên các tác vụ lớn hơn như COCO, YOLOv4 hoạt động hiệu quả hơn.



Hình 2.10: So sánh độ trễ trung bình giữa các phiên bản YOLO(v3,v4,v5)

3.1.6. Cấu trúc của YOLOv5 trong việc nhận diện vật thể (Object Detection)

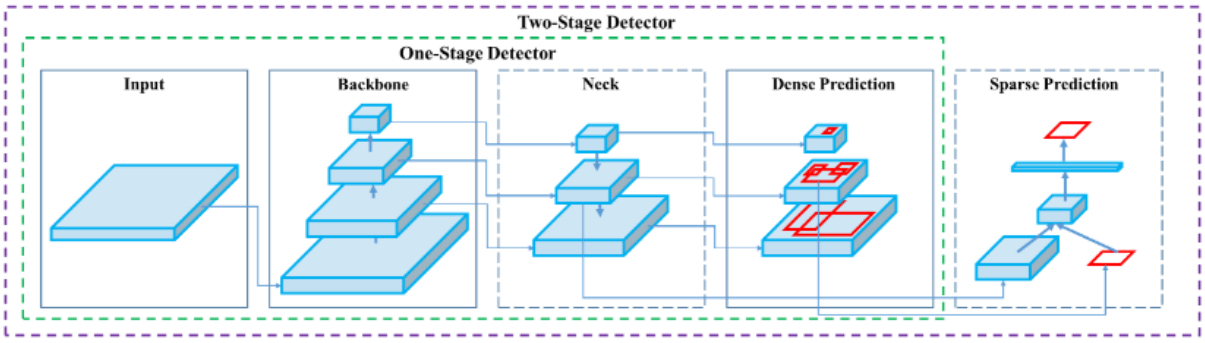
Bao gồm 3 phần chính:

- Backbone: Backbone là 1 mô hình pre-train của 1 mô hình học chuyển (transfer learning) khác để học các đặc trưng và vị trí của vật thể. Các mô hình học chuyển thường là VGG16, ResNet-50,...
- Head: Phần head được sử dụng để tăng khả năng phân biệt đặc trưng để dự đoán class và bounding-box. Ở phần head có thể áp dụng 1 tầng hoặc 2 tầng:

◆ Tầng 1: Dense Prediction, dự đoán trên toàn bộ hình với các mô hình RPN, YOLO, SSD,...

◆ Tầng 2: Sparse Prediction dự đoán với từng mảng được dự đoán có vật thể với các mô hình R-CNN series,..

- Neck: Ở phần giữa Backbone và Head, thường có thêm một phần Neck. Neck thường được dùng để làm giàu thông tin bằng cách kết hợp thông tin giữa quá trình bottom-up và quá trình top-down (do có một số thông tin quá nhỏ khi đi qua quá trình bottom-up bị mất mát nên quá trình top-down không tái tạo lại được).



Hình 2.11: Cấu trúc nhận diện vật thể của YOLOv5

3.1.7. Những cải tiến của YOLOv5 so với các phiên bản trước

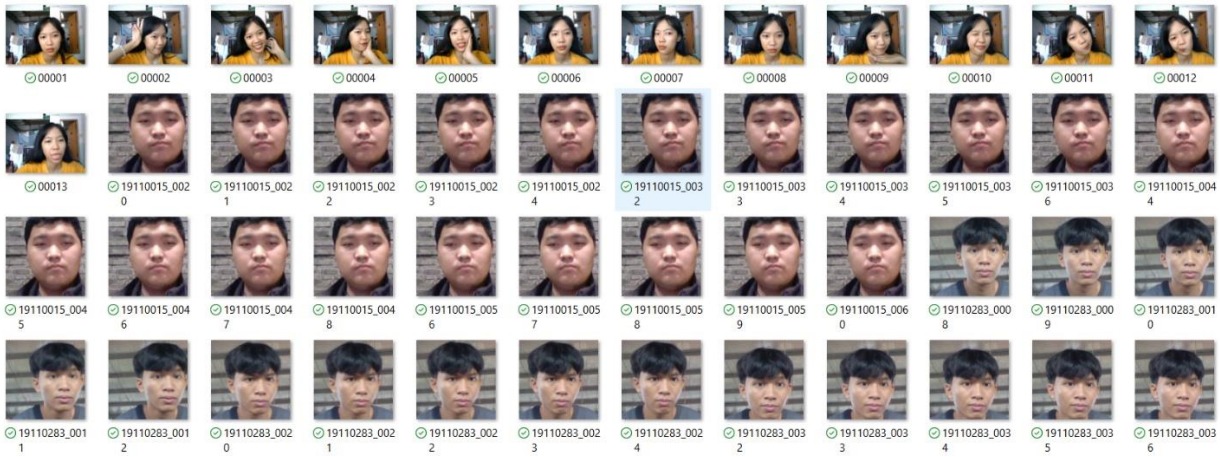
YOLOv5 được triển khai trong PyTorch ban đầu nên nó được hưởng lợi từ hệ sinh thái PyTorch đã được thiết lập: hỗ trợ đơn giản hơn và triển khai dễ dàng hơn. Hơn nữa, là một khung nghiên cứu được biết đến rộng rãi hơn, việc lặp lại trên YOLOv5 có thể dễ dàng hơn cho cộng đồng nghiên cứu rộng lớn hơn. Điều này cũng làm cho việc triển khai đến các thiết bị di động đơn giản hơn vì mô hình có thể được biên dịch sang ONNX và CoreML một cách dễ dàng.

Khả năng đào tạo cũng như khả năng suy luận rất là nhanh, độ chính xác cao. Cuối cùng YOLOv5 có dung lượng nhỏ. YOLOv5 rất nhỏ. Cụ thể, một tệp trọng số cho YOLOv5 là 27 megabyte. Trong khi đó một tệp trọng số của cho YOLOv4 (với kiến trúc Darknet) là 244 megabyte. YOLOv5 nhỏ hơn gần 90% so với YOLOv4. Điều này có nghĩa là YOLOv5 có thể được triển khai cho các thiết bị nhúng dễ dàng hơn nhiều.

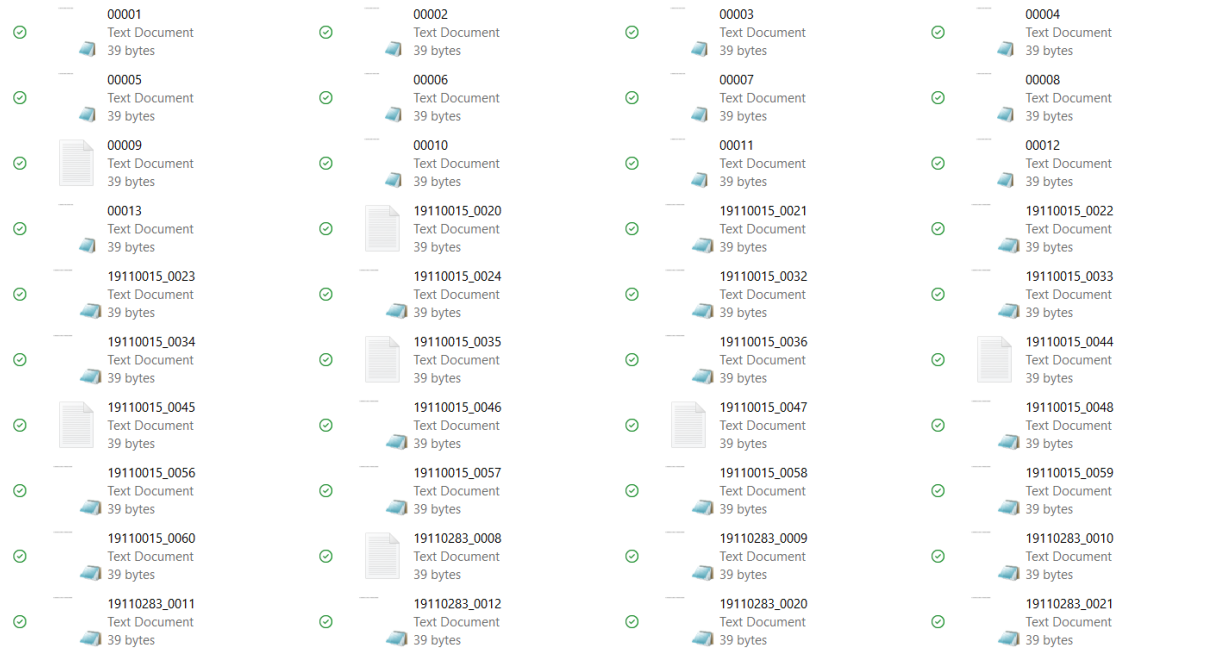
CHƯƠNG IV : THIẾT KẾ HỆ THỐNG

4.1. Tập dữ liệu chuẩn bị cho quá trình huấn luyện

Để thực hiện quá trình gán nhãn và huấn luyện ta cần chuẩn bị tập dữ liệu khoản 100 ảnh về 5 khuôn mặt với mỗi hình sẽ có 1 file gán nhãn đi kèm. Ta sẽ có thư mục data bao gồm images (ảnh được huấn luyện), label (file gắn nhãn), validation.



Hình 2.14: Folder images chứa ảnh được huấn luyện



Hình 2.15: Folder label chứa file gắn nhẫn tương ứng với từng bức ảnh trong folder images

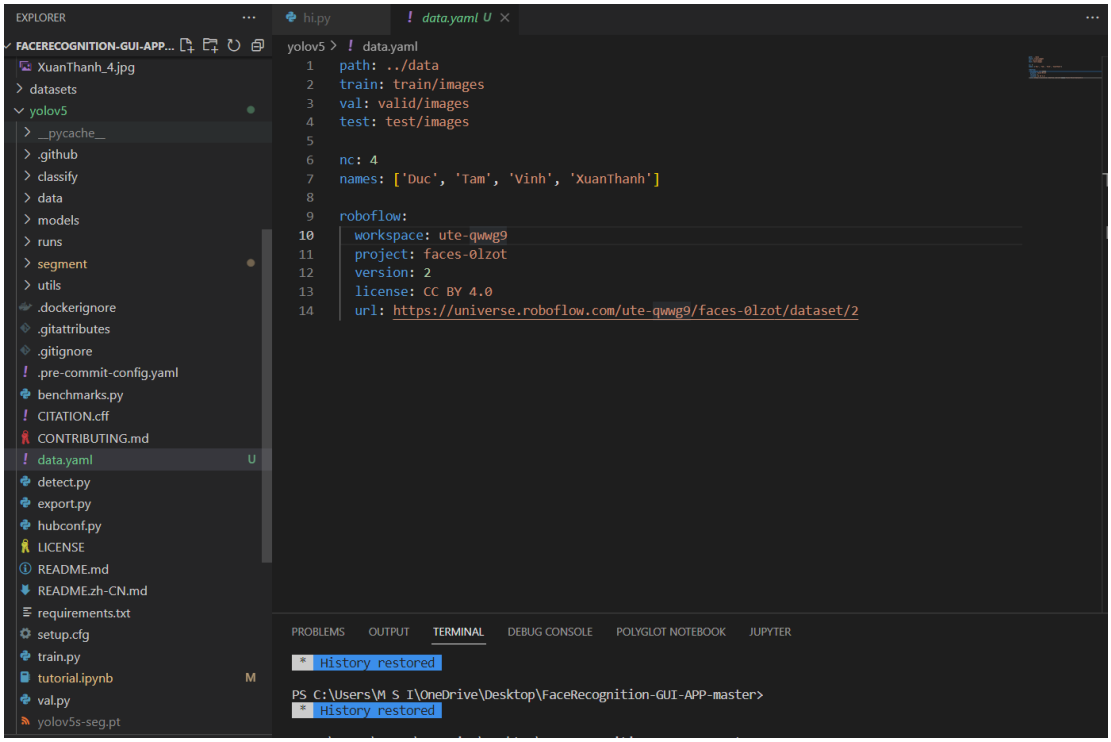
4.2. Quá trình huấn luyện

Bước đầu tiên, chúng ta cần phải cài đặt YOLOv5 và các thư viện cần thiết :

```
# clone YOLOv5 repository
!git clone https://github.com/ultralytics/yolov5  # clone repo
%cd yolov5
!git reset --hard 886f1c03d839575afecb059accf74296fad395b6
# install dependencies as necessary
!pip install -qr requirements.txt  # install dependencies (ignore errors)
```

Hình 2.16: Clone YOLOv5 từ github và cài đặt các dependencies cần thiết

Sau khi clone YOLOv5 về máy tính, chúng ta sẽ có được những folder như hình 2.16. Tiếp theo chúng ta sử dụng models YOLOv5 để tiến hành huấn luyện từ file train.py của YOLO kết hợp với data.yaml



Hình 2.17: File data.yaml khai báo class và thiết lập đường dẫn đến data

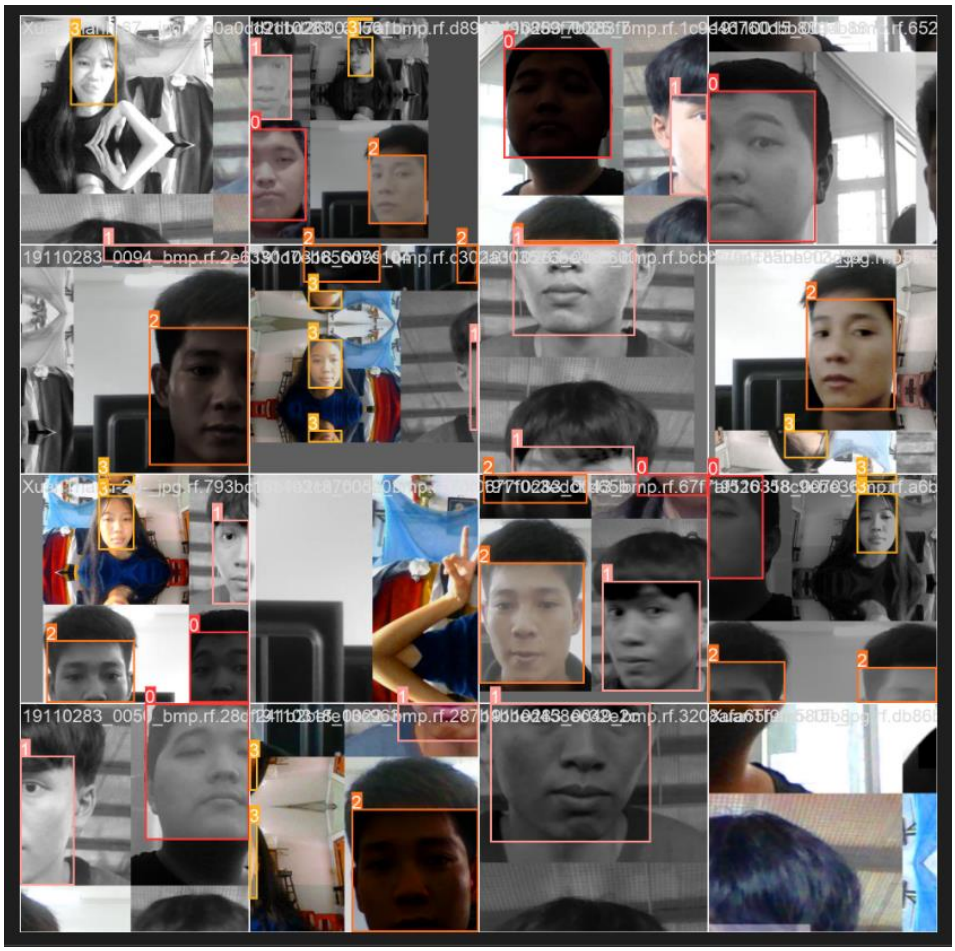
4.3. Tiến hành huấn luyện

Sau khi đã có data và label cùng với file data.yaml, chúng ta tiến hành thực hiện câu lệnh huấn luyện :

```
!cd yolov5 && python train.py --img 640 --batch 16 --epochs  100 --data data.yaml
--weights yolov5s.pt
```

Hình 2.18: Câu lệnh thực hiện huấn luyện với 16 lớp và 100 lần

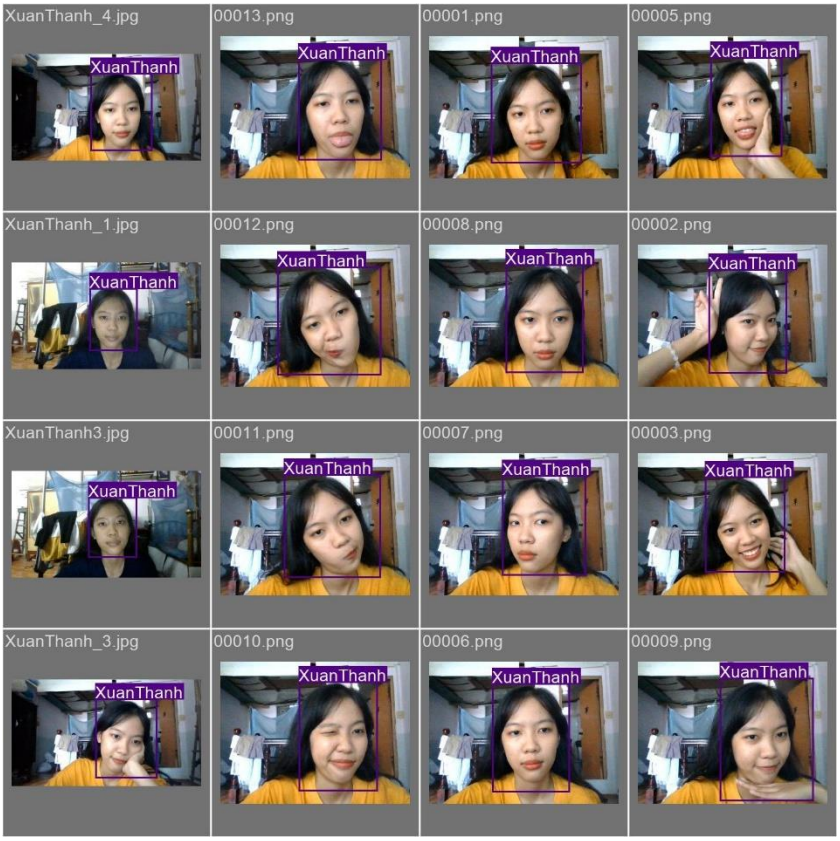
Sau khi huấn luyện xong, dữ liệu đã được huấn luyện sẽ được lưu ở mục yolov5/runs/train.



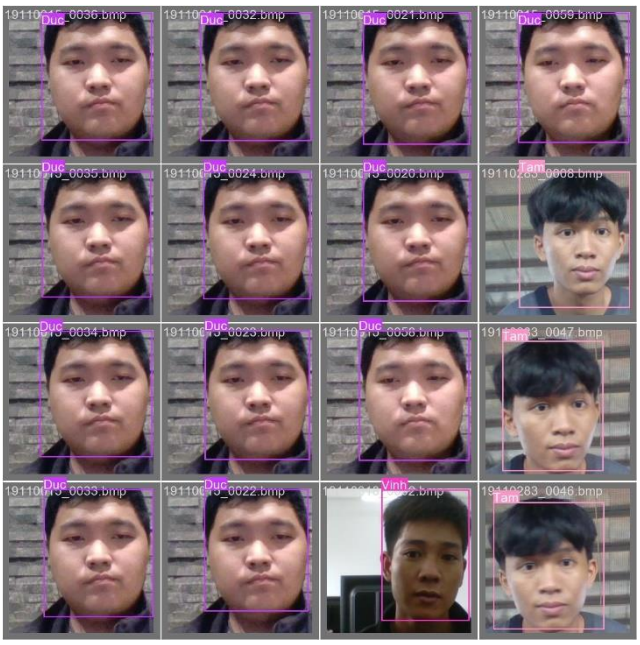
Hình 2.19: Ảnh mẫu các hình sau khi được train

CHƯƠNG V : KẾT QUẢ

5.1. Kết quả quá trình huấn luyện

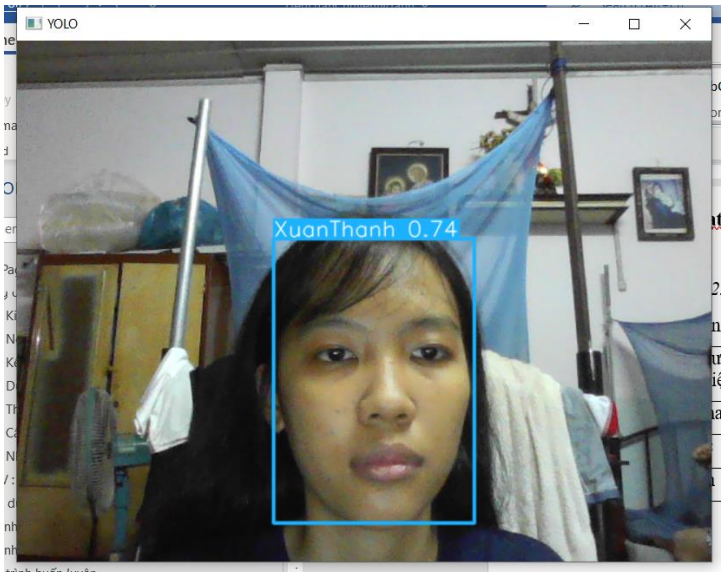


Hình 2.20: Kết quả hiển thị khuôn mặt của đối tượng "Xuân Thanh" sau khi được huấn luyện



Hình 2.21: Kết quả hiển thị khuôn mặt của đối tượng "Duc", "Vinh", "Tam" sau khi được huấn luyện

5.2. Hoạt động của hệ thống



Hình 2.22: Kết quả detect nhận diện khuôn mặt của đối tượng "Xuan Thanh"

Sau khi tiến hành kiểm tra, nhóm em đưa ra được bảng đánh giá mức độ chính xác:

Tên người nhận diện	Tổng số lần kiểm tra	Nhận diện đúng	Nhận diện sai	Độ chính xác
XuanThanh	5	5	0	100%
Duc	5	4	1	80%
Tam	5	3	2	60%

Bảng 1: Đánh giá mức độ chính xác của quá trình nhận diện

PHẦN 3 : KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. Kết luận

- Ứng dụng đã cơ bản thực hiện được việc nhận dạng khuôn mặt người dữa vào tập dữ liệu được huấn luyện

- Độ chính xác khá cao với tập dữ liệu có mỗi đối tượng trên 80 ảnh.

6.2. Hướng phát triển

- Phát triển thêm tính năng thêm khuôn mặt nhận dạng cho ứng dụng.

- Phát triển ứng dụng thành ứng dụng quản lí chấm công hoặc smartdoor.

- Đa dạng hoá tập dữ liệu có thể nhận dạng được đối tượng khác như động vật, đồ vật.

TÀI LIỆU THAM KHẢO

[1]. https://ais.gov.vn/doanh-nghiep-tai-dong-nam-a-ton-that-nang-ne-do-su-co-an-ninh-mang.htm

[2]. https://arxiv.org/pdf/1506.02640.pdf

[3]. https://aicurious.io/posts/tim-hieu-yolo-cho-phat-hien-vat-tu-v1-den-v3/

[4] Phạm Đình Khanh, "YOLO-You only look once", Khoa học dữ liệu-Khanh's blog

[5] Phạm Việt Bình, Đỗ Năng Toàn, "Giáo trình môn học Xử lý ảnh", Khoa Công nghệ Thông tin – Đại học Thái Nguyên, 2007.

[6] Nguyễn Quang Hoan, "Giáo trình Xử lý ảnh", Học viện Công nghệ Bưu chính Viễn thông, 2006.

[7] Nguyễn Đình Thúc, "Trí tuệ nhân tạo, mạng NơRon phương pháp và ứng dụng", NXB Giáo Dục 2000.

[8] Quốc Phạm,"Tìm hiểu mô hình YOLO cho bài toán Object Detection"

[9] Joseph Nelson, Jacob Solawetz, "YOLOv5 is Here:State-of-the-Art Object Detection at 140 FPS"