

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ ĐIỆN TỬ



PHẠM VĂN DUY
NGUYỄN HỮU TRIỀN

THIẾT BỊ NHẬN DIỆN BIỂN BÁO TỐC ĐỘ
CHO NGƯỜI LÁI XE HƠI

Chuyên ngành: Hệ thống thông minh

Mã chuyên ngành: 52510312

KHOÁ LUẬN TỐT NGHIỆP

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

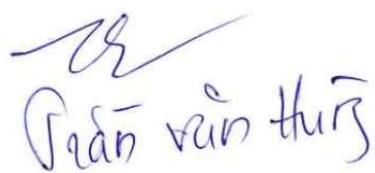
THÔNG TIN CHUNG

Họ và tên sinh viên: Phạm Văn Duy MSHV: 18084071
Lớp : DHDTM14A Khóa: 14
Chuyên ngành : Hệ thống thông minh Mã chuyên ngành: 52510312
SĐT : 0382094198 Email: sensork77@gmail.com
Địa chỉ liên hệ : 162/4 Lê Văn Khương, phường Hiệp Thành, Quận 12
Tên đề tài : Thiết bị nhận diện biển báo tốc độ cho người lái xe hơi
Họ và tên sinh viên: Nguyễn Hữu Triển MSHV: 18057871
Lớp : DHDTM14B Khóa: 14
Chuyên ngành : Hệ thống thông minh Mã chuyên ngành: 52510312
SĐT : 0823789757 Email: nguyentrien0800@gmail.com
Địa chỉ liên hệ : 64 TL 04 phường Thành Lộc, Quận 12
Tên đề tài : Thiết bị nhận diện biển báo tốc độ cho người lái xe hơi
Người hướng dẫn : ThS. Trần Văn Hùng
SĐT : 0918093262
Email : tranvanhung@iu.edu.vn
Cơ quan công tác : Trường Đại Học Công Nghiệp TP.HCM

Tp. Hồ Chí Minh, ngày 01 tháng 06 năm 2022

Người hướng dẫn

(Ký và ghi rõ họ tên)


Trần Văn Hùng

Sinh viên

(Ký và ghi rõ họ tên)



Phạm Văn Duy



Nguyễn Hữu Triển

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

Tp. Hồ Chí Minh, ngày 11 tháng 10 năm 2021

PHIẾU ĐĂNG KÝ KHÓA LUẬN TỐT NGHIỆP

(Đồ án tốt nghiệp/Báo cáo tốt nghiệp)

Bộ môn : Hệ thống thông minh

Tên đề tài : Thiết bị nhận dạng biển báo tốc độ cho người lái xe hơi

Chuyên Ngành: Hệ thống thông minh

Sinh viên :

1. Họ và tên: Phạm Văn Duy

Mã sinh viên: 18084071

Email: phamduy25012000@gmail.com

Số điện thoại: 0382094198

2. Họ và tên: Nguyễn Hữu Triển

Mã sinh viên: 18057871

Email: nguyentrien0800@gmail.com

Số điện thoại: 0823789757

Người hướng dẫn:

1. Họ và tên: Trần Văn Hùng

Học vị/Học hàm: : Thạc Sỹ

Đơn vị công tác: Trường ĐH Công Nghiệp Tp.HCM

Ngày nhận đề tài: 11/10/2021 Ngày hoàn thành (dự kiến):

BỘ MÔN CHỦ QUẢN

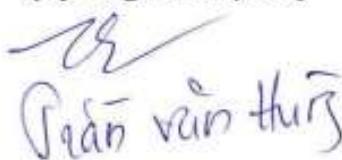
(Ký và ghi rõ họ tên)

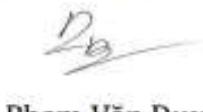
GIÁNG VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

SINH VIÊN ĐĂNG KÝ

(Ký và ghi rõ họ tên)


Trần Văn Hùng


Phạm Văn Duy


Nguyễn Hữu Triển

LỜI CẢM ƠN

Lời đầu tiên cho chúng em xin gửi lời cảm ơn sâu sắc đến toàn thể giảng viên trường Đại học Công Nghiệp Thành phố Hồ Chí Minh nói chung và giảng viên khoa Công nghệ Điện tử nói riêng đã đồng hành và giúp đỡ chúng em từ khi bước vào giảng đường đại học. Khoảng thời gian 4 năm được học tập và nghiên cứu tại trường không dài nhưng đó là khoảng thời gian thật sự đáng nhớ và ý nghĩa đối với chúng em. Chúng em xin gửi lời cảm ơn đặc biệt đến thầy ThS. Trần Văn Hùng đã chỉ dạy, hướng dẫn và giúp đỡ tận tâm giúp chúng em cải thiện những sai sót trong quá trình thực hiện khóa luận tốt nghiệp. Mặc dù tình hình dịch bệnh có khó khăn nhưng thầy luôn tạo những điều kiện tốt nhất cho chúng em. Đề tài được nhóm thực hiện nghiêm túc và tâm quyết cao nhưng không thể tránh được những sai sót rất mong được sự góp ý và đánh giá từ quý thầy cô. Em xin chân thành gửi lời chúc sức khỏe đến tất cả quý thầy cô giảng viên trường Đại học Công Nghiệp Thành phố Hồ Chí Minh để tiếp tục sự nghiệp trồng người.

Một lần nữa chúng em Phạm Văn Duy và Nguyễn Hữu Triển xin gửi lời cảm ơn chân thành đến quý thầy cô.

MỤC LỤC

LỜI CẢM ƠN	3
MỤC LỤC.....	4
DANH MỤC HÌNH ẢNH	6
DANH MỤC BẢNG BIÊU	8
DANH MỤC TỪ VIẾT TẮT.....	9
CHƯƠNG 1 TỔNG QUAN VỀ ĐỀ TÀI.....	10
1.1 Giới thiệu đề tài	10
1.2 Mục tiêu đề tài	10
1.3 Giới thiệu đề tài	11
1.4 Phương pháp nghiên cứu	12
1.5 Đối tượng và phạm vi nghiên cứu	12
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT.....	13
2.1 Tổng quan về thị giác máy tính và xử lý ảnh	13
2.1.1 Thị giác máy tính	13
2.1.2 Xử lý ảnh.....	13
2.1.3 Lý thuyết về hình ảnh trong lập trình	14
2.1.4 Khái niệm Pixel	14
2.2 Tổng quan về YOLOv5	15
2.2.1 Phân loại	15
2.2.2 Nguyên lý hoạt động của yolov5	17
2.2.3 Kiến trúc của YOLOv5.....	18
2.2.4 Cấu trúc cấu YOLOv5	20
2.2.5 Thuật toán sử dụng của YOLOv5.....	20
2.2.6 Hàm kích hoạt, hàm suy hao.....	26
2.2.7 Những cải tiến của YOLOv5	29
2.3 Giới thiệu về Google API.....	29
CHƯƠNG 3 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	31
3.1 Phân tích	31

3.2	Các bước xây dựng tập huấn luyện	32
3.3	Ngôn ngữ lập trình và thư viện sử dụng	35
3.4	Các thành phần phần cứng sử dụng.....	36
3.5	Nguyên lý hoạt động nhận diện biển báo giao thông	40
CHƯƠNG 4 THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ.....		44
4.1	Đánh giá kết quả sau quá trình huấn luyện.....	44
4.2	Sản phẩm	45
4.3	Kết quả thực nghiệm.....	48
CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		58
5.1	Kết luận.....	58
5.2	Hướng phát triển.....	58
TÀI LIỆU THAM KHẢO.....		59

DANH MỤC HÌNH ẢNH

Hình 1. 1 Biển báo tốc độ giao thông	11
Hình 2. 1 Miêu tả thị giác máy tính	13
Hình 2. 2 Quá trình xử lý ảnh	14
Hình 2. 3 Pixel của ảnh	15
Hình 2. 4 Phân loại YOLOv5.....	16
Hình 2. 5 So sánh từng loại YOLO đã có	16
Hình 2. 6 Nguyên lý nhận diện của YOLOv5.....	17
Hình 2. 7 Sơ đồ kiến trúc Yolo	18
Hình 2. 8 Lớp ẩn của mô hình YOLOv5	19
Hình 2. 9 Cấu trúc của YOLOv5	20
Hình 2. 10 Sơ đồ khối thuật toán sử dụng YOLOv5	20
Hình 2. 11 Giới thiệu Residual Blocks	21
Hình 2. 12 Bounding box	22
Hình 2. 13 Chỉ số IoU	22
Hình 2. 14 Kết quả sau qua trình dự đoán.....	23
Hình 2. 15 Chuyển đổi lớp fully connected layer cuối thành lớp tích chập	24
Hình 2. 16 Cách xác định anchor box	25
Hình 2. 17 Feature map	25
Hình 2. 18 Hình ảnh minh họa của thuật toán nox-max suppression	26
Hình 2. 19 Đồ thị hàm kích hoạt SiLu	26
Hình 2. 20 Tọa độ của Localization loss.....	28
Hình 2. 21 Giới thiệu Google Drive API	30
Hình 3. 1 Sơ đồ khói hệ thống thiết bị nhận diện	31
Hình 3. 2 Tập dataset huấn luyện biển báo nhận diện	32
Hình 3. 3 Tập dataset huấn luyện biển báo nhận diện	33
Hình 3. 4 Liên kết Google colab với drive.....	33
Hình 3. 5 Sử dụng tài nguyên Tesla T4 hoặc Tesla K80 huấn luyện	34
Hình 3. 6 Giải nén tập dữ liệu huấn luyện bên Google drive	34
Hình 3. 7 Điều chỉnh đường dẫn theo class biển báo.....	35
Hình 3. 8 Tiến hành huấn luyện	35
Hình 3. 9 Logo thư viện OpenCV	36
Hình 3. 10 Jetson Nano Developer Kit Carrier Board A02	37
Hình 3. 11 Màn hình LCD 4.3 inch HDMI(B)	38
Hình 3. 12 Webcam HD Logitech C271	39
Hình 3. 13 Loa máy tính JVJ JS20.....	40
Hình 3. 14 Lưu đồ giải thuật quá trình nhận diện	41

Hình 3. 15 Lưu đồ giải thuật GDA	42
Hình 4. 1 Kết quả sau quá trình huấn luyện.....	44
Hình 4. 2 Mặt ngang của thiết bị.....	45
Hình 4. 3 Mặt trước của sản phẩm thiết bị.....	46
Hình 4. 4 Phần cứng bên trong của thiết bị.....	47
Hình 4. 5 Mặt sau của thiết bị nhận dạng.....	47
Hình 4. 6 Detect biển báo 20 km/h	48
Hình 4. 7 Detect biển báo 30 km/h	49
Hình 4. 8 Detect biển báo 40 km/h	50
Hình 4. 9 Detect biển báo 50 km/h	51
Hình 4. 10 Detect biển báo 60 km/h	51
Hình 4. 11 Detect biển báo 70 km/h	52
Hình 4. 12 Detect biển báo 80 km/h	53
Hình 4. 13 Detect biển báo 90 km/h	54
Hình 4. 14 Detect biển báo 100 km/h	55
Hình 4. 15 Detect biển báo 120 km/h	55
Hình 4. 16 Detect trong trường hợp còn nhiều hơn 1 biển báo	56
Hình 4. 17 Hình ảnh lưu được sẽ gửi lên Google Drive Api	57

DANH MỤC BẢNG BIỂU

Bảng 3. 1 Thông số kỹ thuật	37
Bảng 3. 2 Webcam HD Logitech C271 với thông số kỹ thuật	39
Bảng 3. 3 Thông số kỹ thuật loa JvJ JS20	40
Bảng 4. 1 Thống kê độ chính xác từng Class.....	45

DANH MỤC TỪ VIẾT TẮT

AI	Artificial Intelligence
API	Appliaction Programming Interface
CNN	Convolutional Neural Nework
CPU	Central Processing Unit
GDA	Google Drive Api
GPU	Graphics Procssing Unit
HDMI	High-Definition Multimedia Interface
USB	Universal Serial Bus
UART	Unversal Asynchronous
IOU	Intersection Over Union
IOT	Internet of thing
SPI	Serial Peripheral Receiver/Transmitter
RAM	Random Access Memory
RGB	Red green blue
YOLO	You Only Look Once
PAN	Path Aggregation Network
PC	Personal Computer

CHƯƠNG 1 TỔNG QUAN VỀ ĐỀ TÀI

1.1 Giới thiệu đề tài

Theo vài năm trở về đây; công nghệ trí tuệ nhân tạo được kế thừa và phát triển theo những năm thập niên 1950; khi cuộc cách mạng 4.0 đang bùng nổ và phát triển mạnh mẽ trên thế giới nói chung và Việt Nam nói riêng. Mọi ngành nghề, công việc đều có thể áp dụng công nghệ AI (trí tuệ nhân tạo). Chính vì thế trí tuệ nhân tạo và xử lý ảnh hay còn được gọi là nhận dạng hình ảnh, video được đưa vào trong giao thông giúp điều khiển tín hiệu làn đường giao thông dễ dàng phát hiện và xử lý khi cần phạt nguội một phương tiện xe ô tô phạm lỗi khi tham gia giao thông.

1.2 Mục tiêu đề tài

Trong đề tài này, mục tiêu được đề ra cần nhận diện biển báo tốc độ cho người lái xe hơi và thông báo cho tài xế biết tốc độ khi đi tới bảng thông báo giới hạn tốc độ. Qua đó cần xây dựng mô hình cũng như thuật toán phù hợp để thiết bị nhận dạng chính xác và tương đối khi đưa ra thực nghiệm.

Về nội dung , mục tiêu cần thực hiện trong đề tài:

- Xây dựng model, giải thuật để nhận dạng cho thích hợp và tổng quát với đề tài. Convolutional Neural Network.
- Sử dụng thuật toán nhận dạng và chuẩn xác bao gồm: CNN (Convolutional Neural Network), Deep Learning.
- Tìm hiểu các thư viện và phần mềm phục vụ đề tài.
- Lập trình thực hiện nhận dạng biển báo trên máy tính những Jetson nano.
- Nhận dạng được biển báo giới hạn tốc độ khi xe di chuyển trên đường.
- Đưa hình ảnh nhận diện lên Google API.
- Tạo tập dữ liệu huấn luyện mẫu (các biển báo thực tế).
- Phân tích đánh giá nhận xét số lượng epoch sau khi train.
- Điều chỉnh và đưa ra thực nghiệm mô hình để đạt được kết quả tốt nhất.

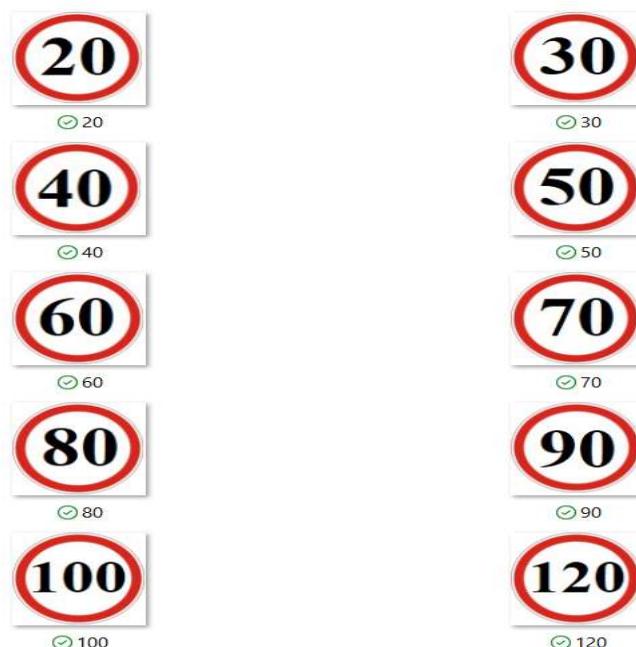
Về kỹ năng, mục tiêu cần đạt :

- Giải quyết vấn đề.
- Đọc và tìm hiểu tài liệu liên quan đến đề tài.
- Kỹ năng lập trình.

1.3 Giới thiệu đề tài

Đề tài khóa luận của nhóm là nhận diện biển báo tốc độ giao thông và thông báo cho tài xế giới hạn với 10 biển bao gồm:

- Biển báo 20 km/h.
- Biển báo 30 km/h.
- Biển báo 40 km/h.
- Biển báo 50 km/h.
- Biển báo 60 km/h.
- Biển báo 70 km/h.
- Biển báo 80 km/h.
- Biển báo 90 km/h.
- Biển báo 100 km/h.
- Biển báo 120 km/h.



Hình 1. 1 Biển báo tốc độ giao thông

1.4 Phương pháp nghiên cứu

- Áp dụng các môn học như xử lý ảnh, thị giác máy tính, trí tuệ nhân tạo để thực hiện và phát triển đề tài.
- Tham khảo những dataset liên quan đến đề tài, sau đó xây dựng mô hình huấn luyện cho đề tài.
- Xin ý kiến chỉ dẫn từ giảng viên hướng dẫn để đề tài được hoàn thiện và phát triển hơn.

1.5 Đối tượng và phạm vi nghiên cứu

Xây dựng thiết bị nhận dạng biển báo cần có:

- Tìm hiểu một số loại máy tính nhúng mini đủ đáp ứng cho đề tài.
- Tìm hiểu một số loại mô hình train như yolo, tensorflow...
- Tiến hành xây dựng sơ đồ khái hệ thống.
- Xây dựng lưu đồ giải thuật sau đó lập trình đưa ra kết quả nhận diện.
- Tiến hành lắp ráp thi công mô hình đưa ra thử nghiệm.
- Kiểm tra kết quả và đánh giá sửa lỗi trong quá trình hệ thống làm việc còn chưa có hiệu quả cao.

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

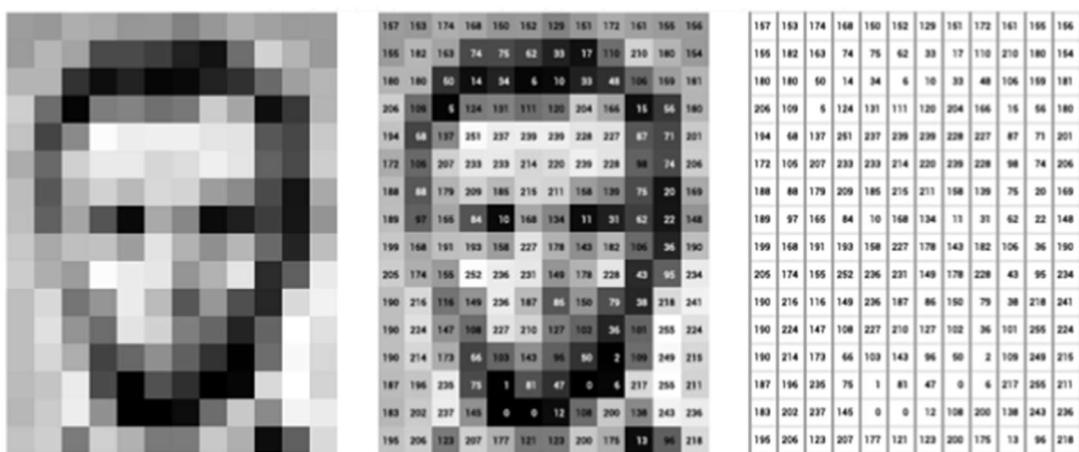
2.1 Tổng quan về thị giác máy tính và xử lý ảnh

2.1.1 Thị giác máy tính

Thị giác máy tính là một lĩnh vực được dùng khá phổ biến hiện nay bao gồm thu thập, xử lý, phân tích và nhận dạng đối tượng.[6]

Quá trình thị giác máy tính được chia thành 3 giai đoạn:

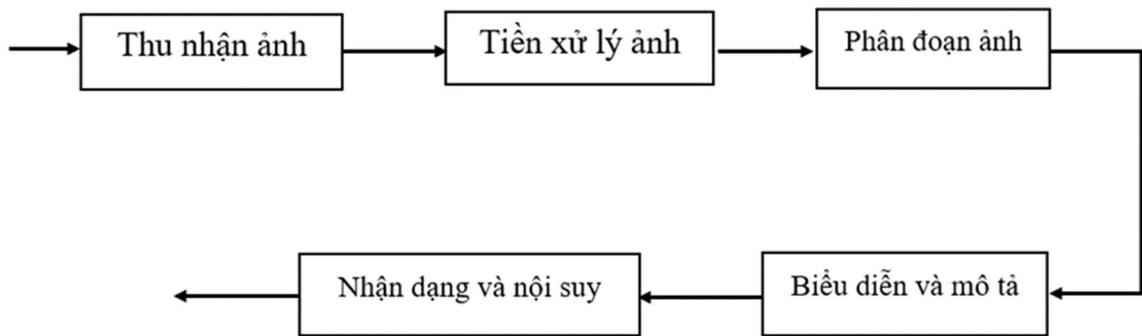
- Thu thập: có chức năng thu thập hình ảnh từ máy ảnh, camera, điện thoại thông minh, những thiết bị có ghi hình...
- Xử lý: việc này được thực hiện bằng các máy tính mini, laptop hoặc PC; giúp phát hiện những đặc trưng, đường viền,.. của đối tượng.
- Thấu hiểu: máy tính sẽ phân biệt những hình ảnh từ quá trình xử lý nhờ vào các đặc trưng. Kết quả sau quá trình phân loại sẽ được dùng trong các hệ thống thông minh.



Hình 2. 1 Miêu tả thị giác máy tính

2.1.2 Xử lý ảnh

Xử lý ảnh là quá trình thay đổi ảnh đầu vào để tạo ra những hình ảnh có kết quả có chất lượng tốt hơn, có các đặc trưng mong muốn. Ảnh đầu vào được thu thập từ camera, máy ảnh hoặc điện thoại thông minh, những thiết bị có ghi hình, sẽ được đánh số để thực hiện các bước xử lý ảnh.[6]



Hình 2. 2 Quá trình xử lý ảnh

Quá trình xử lý ảnh gồm có các giai đoạn:

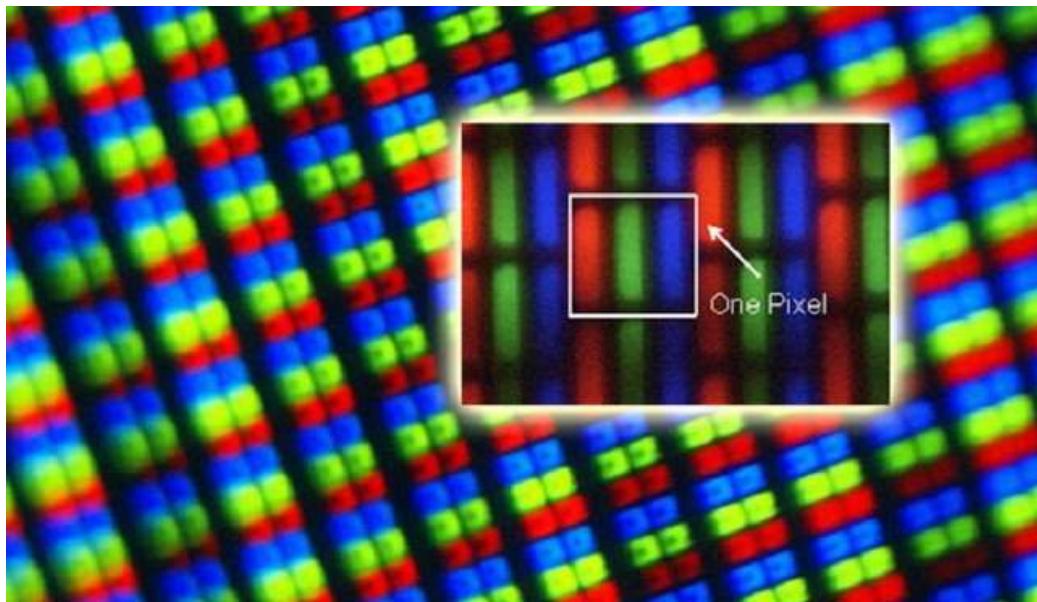
- Thu nhận: là quá trình thu nhận hình ảnh theo thời gian thực, ảnh nhận được có thể là ảnh trắng đen hoặc là ảnh tương tự. Kết quả sẽ được lưu lại để thực hiện cho quá trình tiếp theo.
- Tiền xử lý: là quá trình này có chức năng là lọc nhiễu, kiểm tra độ tương phản để nâng cao chất lượng hình ảnh để cho máy tính thực hiện quá trình phân loại một cách tốt hơn nhờ vào các đặc điểm của đối tượng.
- Phân đoạn ảnh: sử dụng để tách ảnh sau khi được xử lý ở quá trình tiền xử lý thành các vùng dựa vào các đặc trưng của ảnh như màu sắc, độ sáng,...
- Biểu diễn và mô tả: là kết quả sau khi phân đoạn chia các điểm ảnh kết hợp với mã liên kết của các vùng lân cận. Quá trình này cần thiết cho quá trình xử lý tiếp theo của máy tính hoặc là cơ sở đến phân biệt đối tượng trong một ảnh.
- Nhận dạng và nội suy: quá trình này thực hiện được nhờ các mẫu đã học được để xác định đối tượng. Nội suy là phán đoán trên cơ sở nhận dạng.

2.1.3 Lý thuyết về hình ảnh trong lập trình

Ảnh khi được thu thập bằng camera, máy ảnh,... là tập hợp nhiều điểm màu với màu sắc, độ sáng khác nhau được chia đều dưới dạng ma trận thường được gọi là ảnh kỹ thuật số. Các điểm màu đó được gọi là pixel và được đánh số theo một thứ tự nhất định.

2.1.4 Khái niệm Pixel

Khi phóng to một bức ảnh thu thập được từ camera, máy ảnh,... chúng ta sẽ thấy các ô vuông với mắt thường và sẽ không thể nhìn thấy được và 3 màu cơ bản là màu đỏ, xanh lá, xanh biển được gọi là giá trị RGB.[10]



Hình 2. 3 Pixel của ảnh

2.2 Tổng quan về YOLOv5

YOLO là một mô hình mạng CNN có nhiệm vụ phát hiện, nhận dạng và phân loại đối tượng được tạo ra từ các fully connected layer và convolutional layers.

YOLO không phải là thuật toán tối ưu nhất nhưng YOLO là thuật toán có tốc độ xử lý và độ chính xác cao trong các mô hình object detection. Tốc độ xử lý của YOLO hầu như là real time với độ chính xác cao.

Ưu điểm của YOLO là sử dụng toàn bộ thông tin bức hình và dự đoán các đối tượng có trong đó theo kiểu end to end.

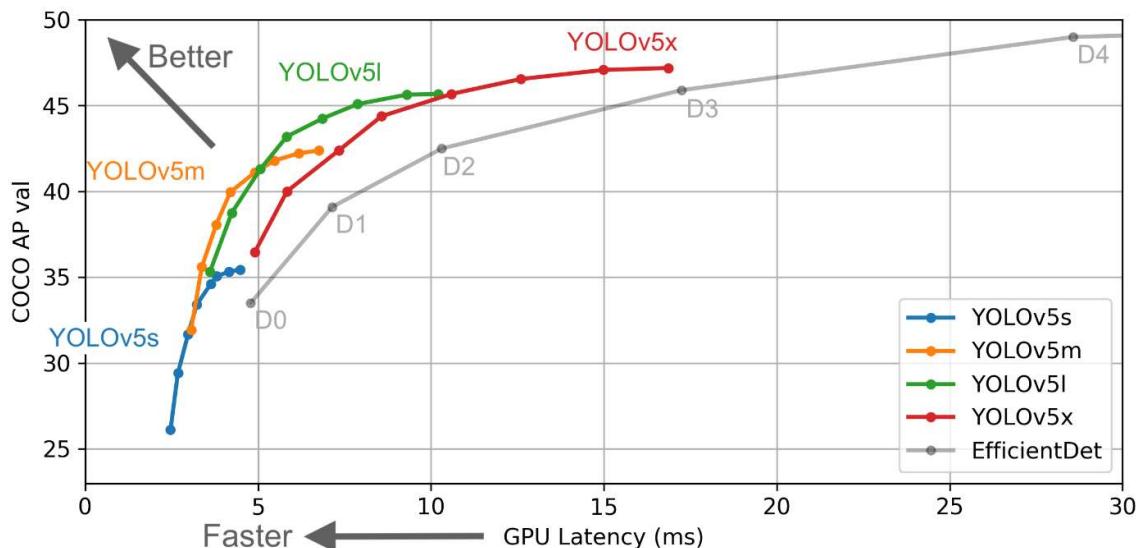
YOLO dễ dàng phát hiện và phân loại nhiều vật thể từ nhiều nhãn khác nhau trong một khung hình thay vì nhận diện và phân loại được một dạng nhãn nhất định.

YOLO đã và đang ngày một hoàn thiện, đã có 5 phiên bản YOLO ra đời là YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv5. Trong đó phiên bản mới nhất là YOLOv5 đã khắc phục được những nhược điểm của các phiên bản YOLO trước đó.

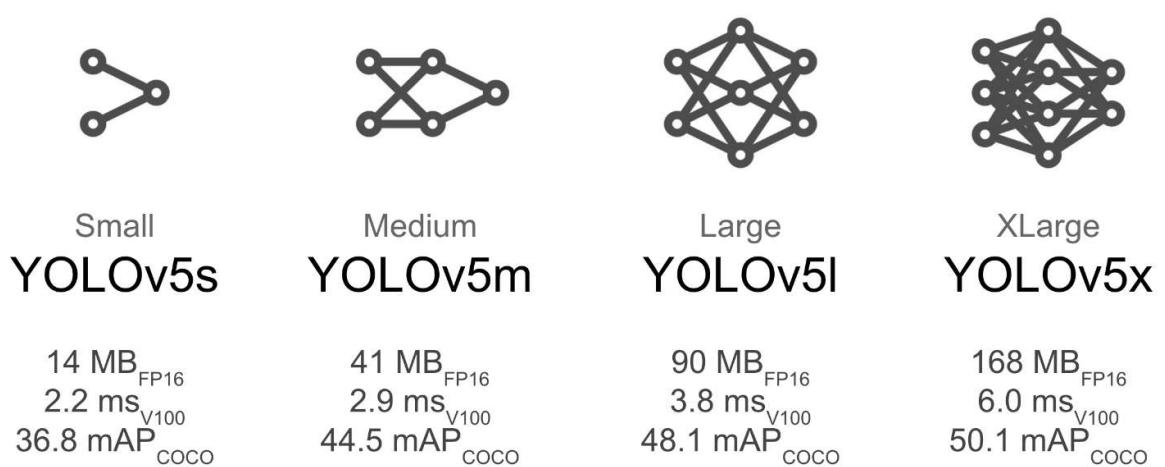
2.2.1 Phân loại

YOLOv5 được chỉnh thành nhiều loại khác nhau đồng nghĩa sẽ tạo ra các trọng số khác nhau gồm các thông số về độ chính xác và loss khác nhau, các mô hình được sắp xếp theo mức độ thấp tới cao bao gồm: YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x thì mức độ nhỏ thì kém chính xác và mức độ cao thì tính chính xác sẽ cao và tương đối hơn so với mức độ thấp.[9]

YOLOv5 nhỏ hơn chắc chắn sẽ dễ sử dụng trong sản xuất và nhiều dự án vì YOLOv5 được phát triển nguyên bản từ Pythorch chứ không phải từ Darknet. Từ việc điều chỉnh sửa đổi kiến trúc đơn giản trong quá trình chuyển sang nhiều môi trường khác nhau.



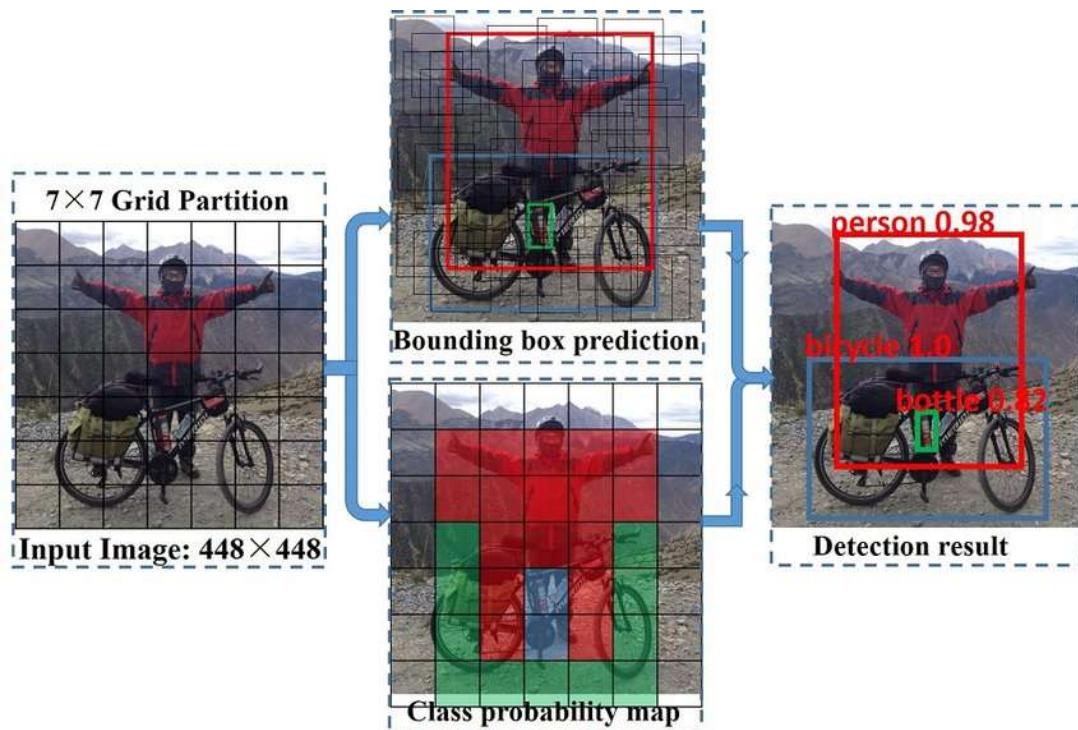
Hình 2. 4 Phân loại YOLOv5



Hình 2. 5 So sánh từng loại YOLO đã có

2.2.2 Nguyên lý hoạt động của yolov5

Khi dữ liệu đầu vào là ảnh hoặc video,... mô hình sẽ phân tích và nhận dạng xem đối tượng vào hay không. Tiếp theo xác định tọa độ của những đối tượng cần nhận dạng dựa trên khung hình có được từ ảnh đầu vào. Dữ liệu vào sẽ được chia theo dạng LxL. Quá trình chia tỉ lệ này rất quan trọng vì sẽ ảnh hưởng đến việc phát hiện đối tượng có trong một khung hình.



Hình 2. 6 Nguyên lý nhận diện của YOLOv5

Ví dụ như hình 2.6 Khi có ảnh đầu vào:

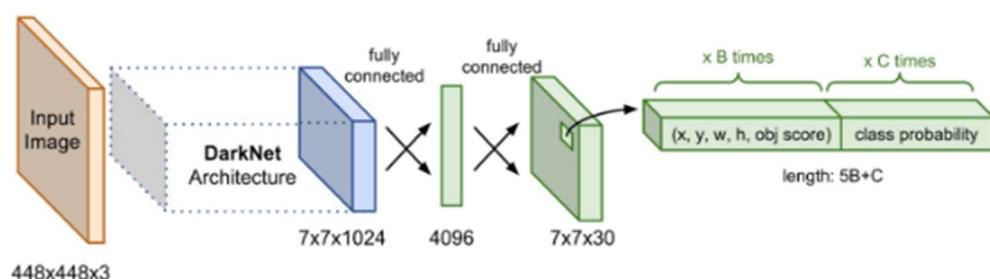
- Mô hình sẽ tiến hành nhận dạng đối tượng.
- Sau đó sẽ định vị vùng chứa vật thể.
- Cuối cùng là tiến hành phân loại từng thuộc vào mong muốn người sử dụng.

2.2.3 Kiến trúc của YOLOv5

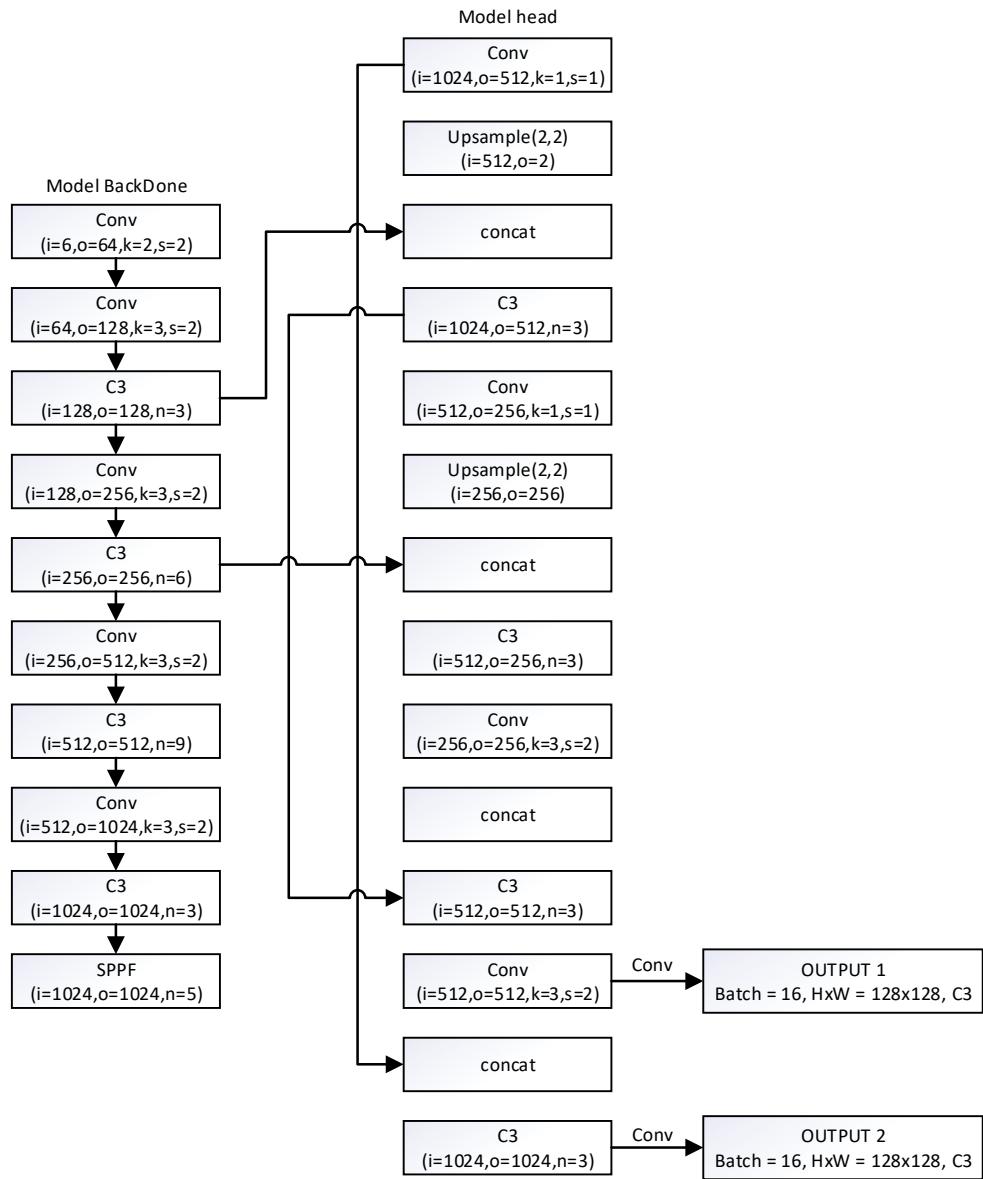
Kiến trúc mạng YOLOv5 gồm 2 phần:[1]

- Base Network được sử dụng chủ yếu là các convolution layer có chức năng đưa ra những trích xuất đặc trưng.
- Extra Layers sử dụng nhận diện vật thể trên feature map của base net.

Base Network của mạng YOLO sử dụng đa số là các convolution và fully connected. Darknet Architecture gọi là base network có nhiệm vụ đưa ra trích xuất đặc trưng. Đầu ra của Base network là feature map được sử dụng để input cho những Extra layers nhằm có tác dụng dự đoán nhãn vật thể và tâm Bounding Box của đối tượng.



Hình 2. 7 Sơ đồ kiến trúc Yolo



Hình 2. 8 Lớp ẩn của mô hình YOLOv5

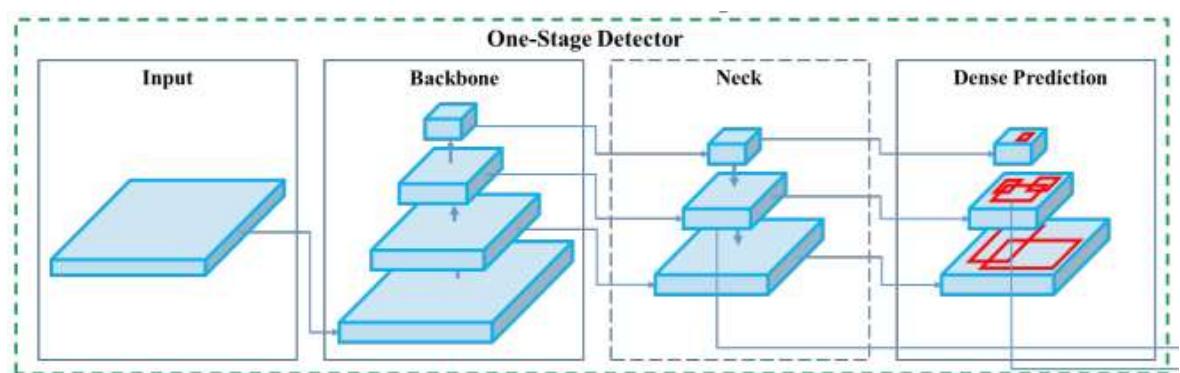
Trong đó:

- Model Backbone chứa các layer có chức năng trích xuất đặc trưng của input.
- Model Head chứa các layer concat có chức năng kết hợp các đặc trưng của ảnh đó và đưa kết quả chính xác nhất khi qua các hàm kích hoạt từ đó tính toán đầu ra của hệ thống. Lớp này thường sử dụng để thực hiện phân phát hiện cuối cùng.
- Kết quả của hệ thống được tính toán thông qua hàm kích hoạt và được tối ưu bởi hàm tối ưu trong hệ thống.

2.2.4 Cấu trúc của YOLOv5

Cấu trúc mạng YOLOv5 có 3 phần gồm:[1][4]

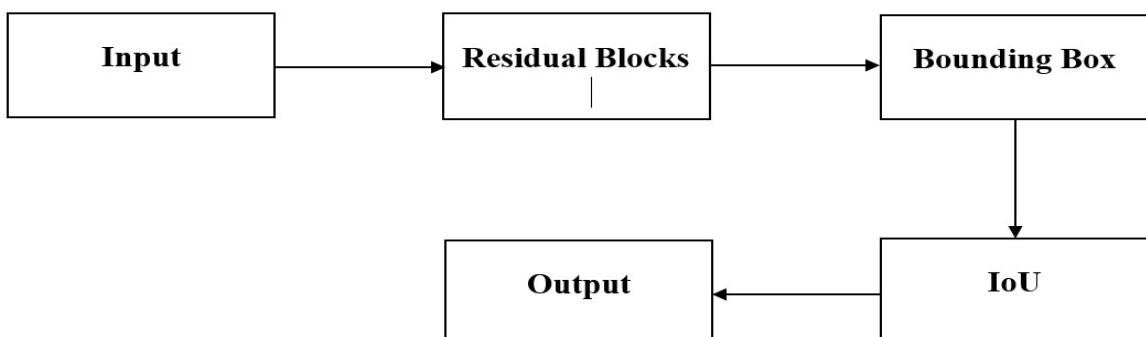
- Backbone là 1 mô hình pre-train để quá trình huấn luyện học các đặc trưng và vị trí của vật thể.
- Head có chức năng tăng độ chính xác của quá trình chuẩn đoán class và các bounding box. Head có 2 tầng là:
 - o Tầng 1: Dense Prediction chuẩn đoán toàn bộ dataset đầu vào với các mô hình.
 - o Tầng 2: Sparse Prediction dự đoán với từng mảng vào cần được dự đoán.
 - o Neck: Phần ở giữa Backbone và Head và phần Neck.



Hình 2. 9 Cấu trúc của YOLOv5

Cấu trúc của YOLOv5 cũng gần như tương tự với các phiên bản YOLO trước đó với phần Neck sử dụng PAN(giải quyết tình trạng mạng Deep Learning bị mất mát thông tin) tuy nhiên ngõ vào Backbone của YOLOv5 sử dụng khối focus thay cho 3 lớp đầu trong thuật toán Yolov3. Vì vậy cấu trúc Backbone của YOLOv5 cũng được gọi là Focus-CSPDarknet53 và nhiệm vụ của khối này là tính toán xử lý nhưng sử dụng bộ nhớ CUDA nhỏ.

2.2.5 Thuật toán sử dụng của YOLOv5



Hình 2. 10 Sơ đồ khái thuât toán sử dụng YOLOv5

2.2.5.1 Input

Đầu vào của mô hình có thể là hình ảnh, video,...

2.2.5.2 Residual Blocks

Khối này có nhiệm vụ chia hình ảnh đầu vào thành dạng lưới bao gồm các ô vuông. Mỗi ô vuông có kích thước là WxH.



Hình 2. 11 Giới thiệu Residual Blocks

Ví dụ ảnh trên, ảnh đầu vào được chia thành nhiều ô vuông có kích thước bằng nhau, nhiệm vụ của nó là phát hiện xem có đối tượng cần nhận dạng không (phụ thuộc vào dataset). Nếu 1 ô vuông có đối tượng tương thích với dataset thì ô vuông đó sẽ chịu trách nhiệm phát hiện đối tượng (ô vuông màu xanh).[1]

2.2.5.3 Bounding box

Bounding box là điều cần thiết nhất để YOLO có thể nhận dạng và phân loại đối tượng. Để dự đoán bounding box cho một đối tượng cần phải dựa đến phép biến đổi anchor box và cell. Trong thuật toán YOLO bounding box phải chuẩn đoán sao cho không lệch khỏi tâm quá nhiều.[2]

Khi phân loại và nhận diện đối tượng trong môi trường real-time thì các bounding box regressin sẽ được sử dụng thay vì sử dụng các bounding box thông thường vì các giá trị của bounding box trong môi trường real-time sẽ được thay đổi liên tục.

Một bounding box gồm:

- Chiều cao h
- Chiều rộng w
- Đối tượng c
- Tâm của box (bx,by)

Đường bao xung quanh đối tượng có dạng : $y = (p_c, b_x, b_y, b_h, b_w, c)$



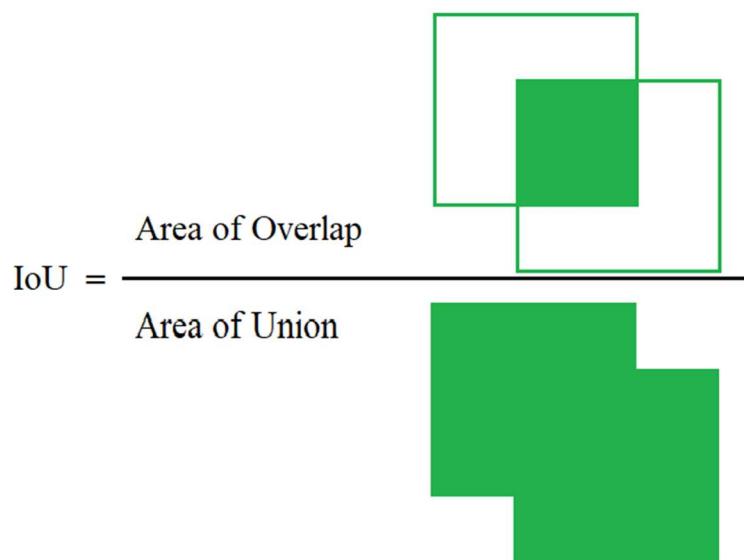
Hình 2. 12 Bounding box

2.2.5.4 Chỉ số IoU

Chỉ số IoU được tính bằng:

$$IoU = \frac{boxA \cap boxB}{boxA \cup boxB} \quad (2-1)$$

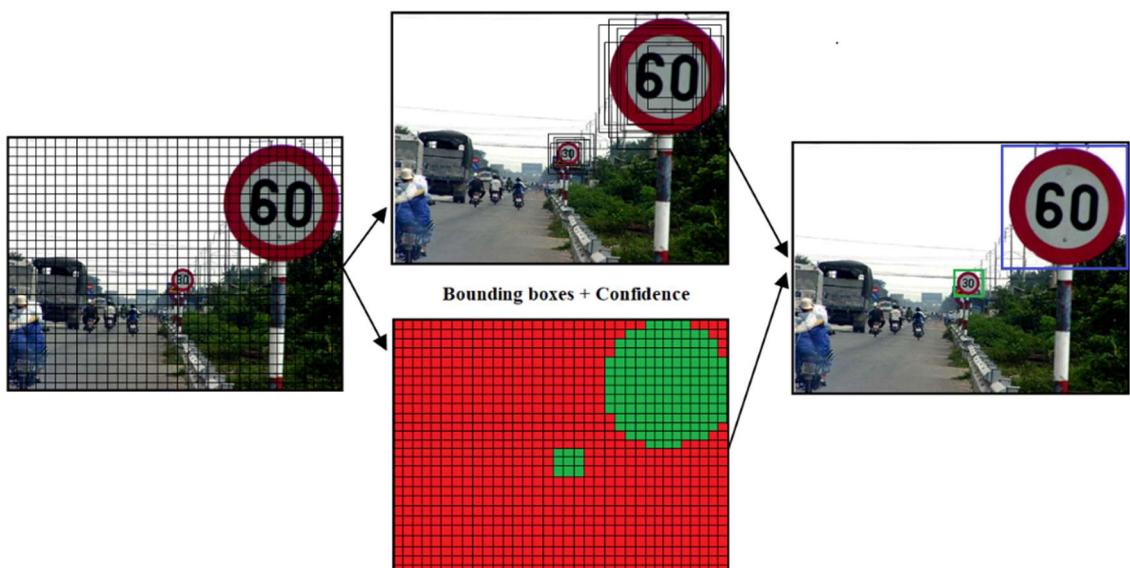
Trong đó box A thường là anchor box (hay còn gọi là groundtruth bounding box) được xuất hiện trong quá trình gán nhãn khi huấn luyện, box B thường là bounding box của hệ thống xác định xuất hiện trong quá trình kiểm tra độ chính xác. Việc tính chỉ số IoU để đánh giá mô hình đã nhận diện đối tượng đúng hay chưa và giúp khả năng phát hiện nhận diện đối tượng một cách nhanh và chính xác hơn.[3]



Hình 2. 13 Chỉ số IoU

2.2.5.5 Output

Output là kết quả đạt được sau khi kết thúc quá trình chuẩn đoán bằng việc kết hợp Residual Blocks, Bounding Box và chỉ số IoU.



Hình 2. 14 Kết quả sau qua trình dự đoán

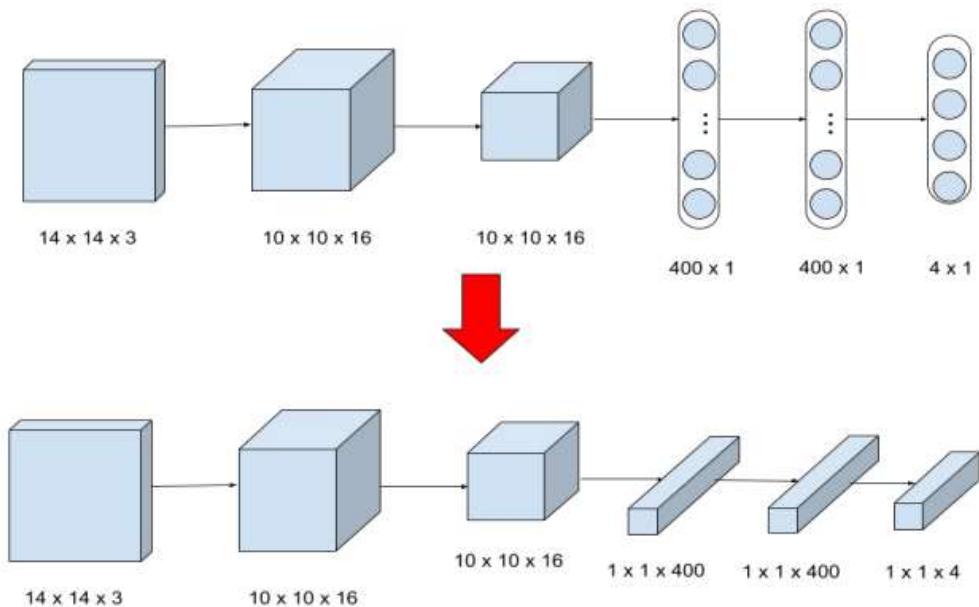
Quá trình hoạt động:

- Đầu tiên ảnh đầu vào được chia thành nhiều ô nhỏ với kích thước các ô đều bằng nhau, các ô này có chức năng phát hiện đối tượng và lưu kết quả vị trí của ô đó.
- Sau khi đối tượng được bao bởi các box nhưng khi có nhiều hơn 1 box sẽ sử dụng chỉ số IoU để chọn ra box có chỉ số lớn nhất. Mục đích của việc này nhằm loại bỏ các box chưa có độ chuẩn xác cao giúp quá trình nhận dạng và phân loại tốt hơn.

Ví dụ hình trên ảnh đầu vào được chia thành các ô vuông có kích thước bằng nhau, sau đó sử dụng mạng nơ ron tích chập để dự đoán và làm nổi bật lên các ô vuông chứa đối tượng đã được gán nhãn.

2.2.5.6 Grid System

Grid system được thực hiện ở cuối mô hình có chức năng chuyển thành lớp tích chập thay thế cho các lớp fully connected layer. Việc thay thế giúp phân loại đối tượng một cách tối ưu hơn bằng cách phép tích chập mà không làm thay đổi kết quả vì vẫn bám được tọa độ của đối tượng.[4][5]



Hình 2. 15 Chuyển đổi lớp fully connected layer cuối thành lớp tích chập

2.2.5.7 Output của YOLO

Output của YOLO là một vecto:

$$[p_0, (t_x t_y t_z t_u), (p_1, p_2, \dots, p_n)]$$

Trong đó:

- p_0 : xác suất chuẩn đoán đối tượng.
- (t_x, t_y, t_z, t_u) : giúp xác định các tạo độ trong bounding box.
- (p_1, p_2, \dots, p_n) : vecto xác suất chuẩn đoán của các classes.

+ Số lượng output sẽ được tính dựa trên số lượng classes theo công thức : classes 5.

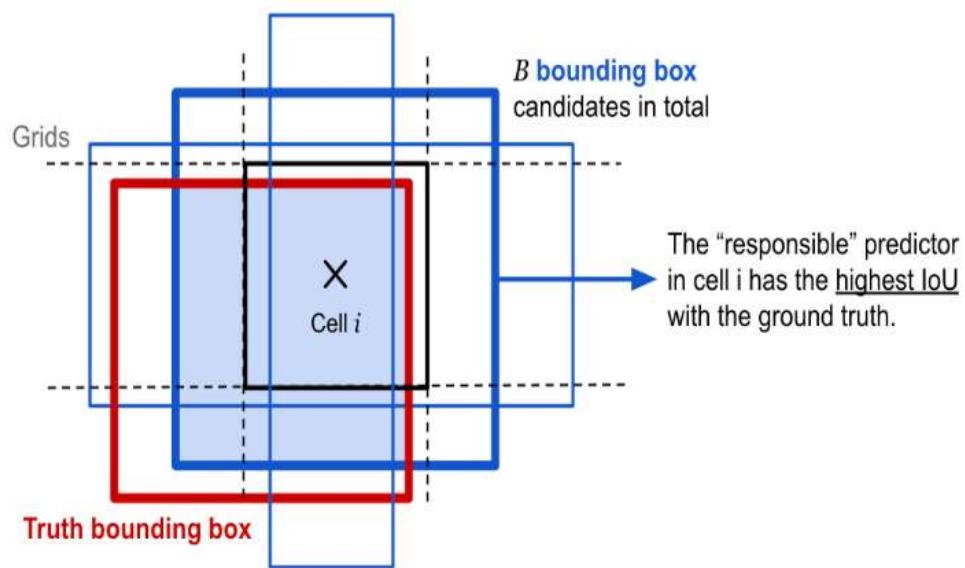
+ Khi đó số lượng tham số output sẽ là: (classes + 5) x A

Với A là anchors/cell (A có thể là 3)

2.2.5.8 Anchor box

Muốn có được các Bounding box cho đối tượng thì mô hình sử dụng YOLO cần có các Anchor box để xác định và bao xung quanh đối tượng. Cuối cùng sẽ sử dụng thuật toán regression bounding box để lấy giá trị anchor box tạo bounding box cho đối tượng.[2]

Một đối tượng cần nhận dạng chỉ được quyền có 1 anchor box. Trong trường hợp có nhiều hơn 1 anchor box ta sẽ dùng đến chỉ số IoU để chọn ra anchor box có chỉ số IoU lớn nhất.

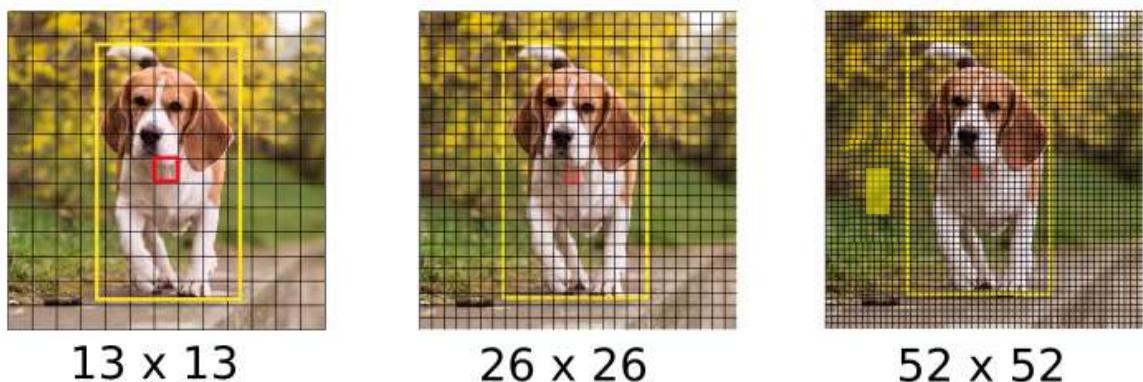


Hình 2. 16 Cách xác định anchor box

2.2.5.9 Feature map

Những feature map có chức năng dự đoán trước object của đối tượng. Tùy vào kích thước đối tượng sẽ được sử dụng các feature map cho phù hợp.[3]

Khi đối tượng có kích thước lớn sẽ sử dụng các feature map nhỏ để chuẩn đoán. Khi đối tượng có kích thước nhỏ sẽ sử dụng các feature map kết hợp với anchor box chuẩn đoán.



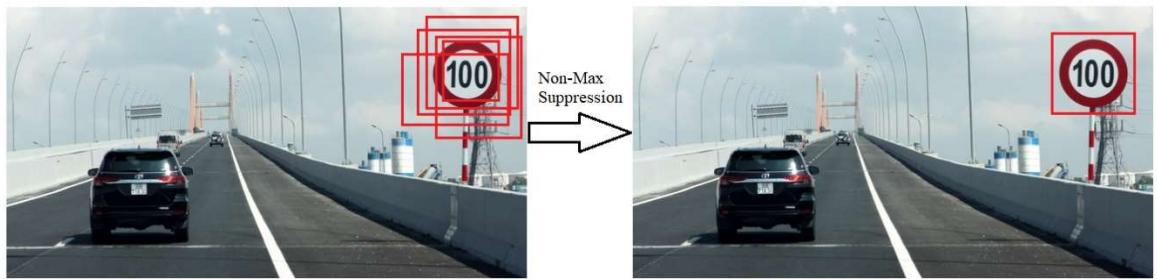
Hình 2. 17 Feature map

2.2.5.10 Thuật toán Non-max

Trong quá trình kiểm tra sẽ xuất hiện nhiều bounding box đi cùng đó là các xác suất chuẩn đoán với độ chính xác khác nhau và chỉ số IoU khác nhau. Thuật toán Non-max suppression lựa chọn duy nhất bounding box nào có chỉ số IoU lớn nhất.

Quá trình loại bỏ thuật toán non-max suppression diễn ra ở 3 bước:

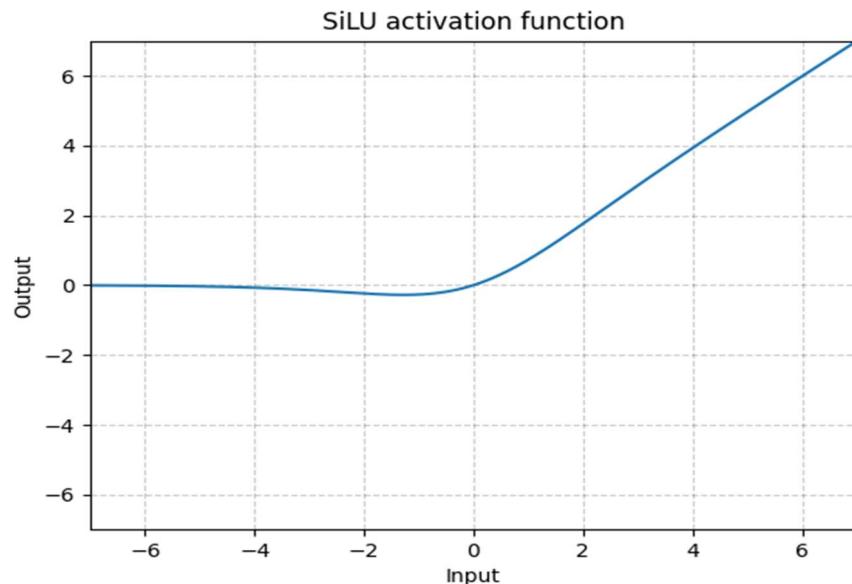
- Bước 1: Loại bỏ các bounding box có tỷ lệ xuất hiện vật thể thấp.
- Bước 2: Chọn các bounding box có tỷ lệ xuất hiện vật thể cao nhất.
- Bước 3: Nếu có nhiều hơn 1 bounding box có cùng tỷ lệ xuất hiện, vật thể lúc này sẽ dựa vào chỉ số IoU để loại bỏ (lựa chọn bounding box có chỉ số IoU cao nhất). Kết thúc quá trình sẽ giữ các bounding box có tỷ lệ nhận dạng đối tượng tốt nhất.



Hình 2. 18 Hình ảnh minh họa của thuật toán nox-max suppression

2.2.6 *Hàm kích hoạt, hàm suy hao*

2.2.6.1 *Hàm kích hoạt SiLu*



Hình 2. 19 Đồ thị hàm kích hoạt SiLu

Việc kích hoạt hàm SiLu thay cho các hàm tương tự như Relu, tanh, sin và Sigmoid. Vì SiLu không tăng theo kiểu tuyến tính và có tính ổn định hơn các hàm kích hoạt khác.

Hàm kích hoạt SiLu được tính bằng:

$$a_k(s) = \sigma(z_k) \left(1 + (1 - \sigma(z_k)) \right) \quad (2-2)$$

$$zk = -\log((zk - 2)/(zk + 2)) \quad (2-3)$$

Hàm kích hoạt SiLu được dùng để cập nhật trọng số:

$$\nabla_{Wik} a_k(s) = \sigma(z_k)(1 + z_k(1 - \sigma(z_k)))S_i \quad (2-3)$$

2.2.6.2 Hàm suy giảm (Loss Function)

Loss function được sử dụng có chức năng để tính độ lỗi bằng tổng độ lỗi của 3 độ lỗi sau:

- Độ lỗi dự đoán nhãn của Object classification loss.
- Độ lỗi dựa trên chuẩn đoán tọa độ, kích thước của bounding box – localization loss.
- Độ lỗi để xem bounding box có chứa đối tượng cần nhận dạng hay không (Confidence loss).

Việc sử dụng độ lỗi trong quá trình huấn luyện có chức năng:

- Tăng classification score lớp đúng khi box bounding chứa đối tượng.
- Dựa vào các bounding boxes và chọn bounding box tốt nhất.

2.2.6.3 Classification Loss

Việc tính các classification loss được thực hiện với các đối tượng có gán nhãn. Giá trị classification loss được tính bằng:

$$L_{classification} = \sum_{i=0}^{S^2} I_i^{obj} \sum_{c \in class} (p_i(c) - \hat{p}_i(c))^2 \quad (2-4)$$

Trong đó:

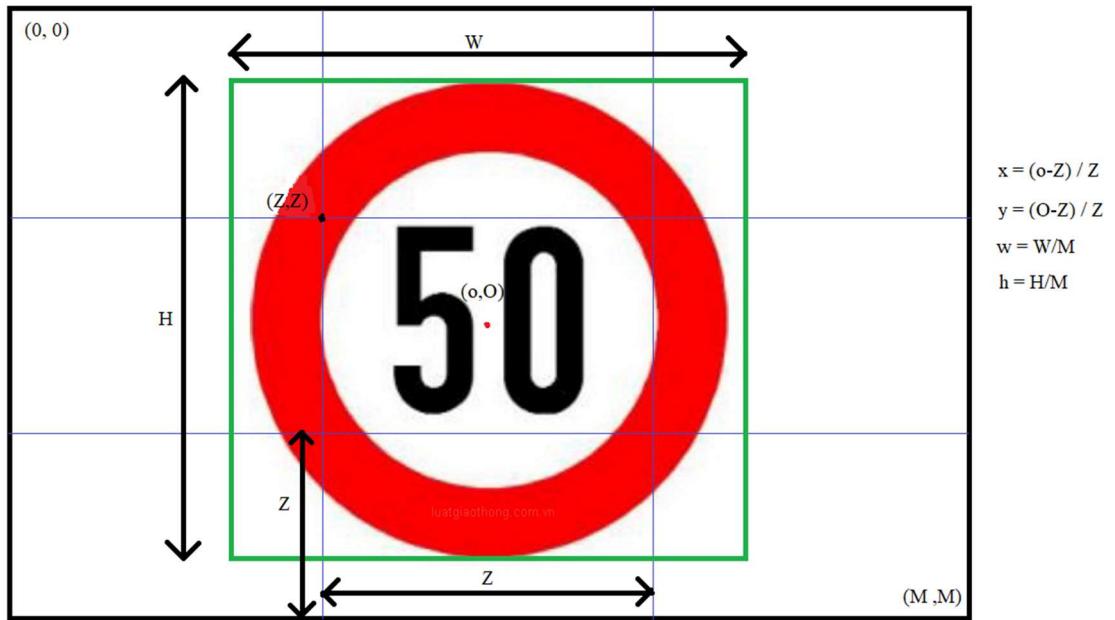
$\hat{p}_i(c)$: Xác suất ở lớp c của các ô vuông tương ứng với mô hình.

I_i^{obj} : Bằng 1 nếu có đối tượng và ngược lại bằng 0.

2.2.6.4 Localization Loss

Tính các giá trị lỗi cho Bounding box được chuẩn đoán với tọa độ tâm và tọa độ chiều dài và chiều rộng so với dữ liệu mẫu.[1][5]

Quá trình này không được tính toán trực tiếp mà phải trả về vùng [0:1] so với tọa độ tâm và không được chuẩn đoán trực tiếp mà bị chuẩn đoán lệch tâm so với Bounding box. Thay đổi kích thước ảnh và tọa độ tâm giúp quá trình hội tụ nhanh hơn so với giá trị mặc định.



Hình 2. 20 Tọa độ của Localization loss

Tính giá trị tọa độ tâm, tọa độ chiều rộng và chiều dài của Bounding box. Giá trị của độ lỗi Localization loss được tính bởi tổng giá trị chuẩn đoán tọa độ tâm (x, y) và chiều dài, chiều rộng (w, h) của ô vuông chứa đối tượng. Mỗi bounding box sẽ có một giá trị IoU tốt nhất.

Giá trị hàm lỗi dự đoán tọa độ tâm (x, y) của predicted bounding box và (\hat{x}, \hat{y}) là tọa độ tâm của truth bounding box được tính :

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B l_i^{obj} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \quad (2-5)$$

Giá trị hàm lỗi chuẩn đoán tọa độ chiều rộng (w, h) của predicted bounding box so với truth bounding box được tính như sau :

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B l_i^{obj} \left(\sqrt{W_i} - \sqrt{\hat{W}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \quad (2-6)$$

Localization loss là độ lỗi quan trọng nhất trong 3 loại độ lỗi trên vì thế giá trị của localization loss lớn hơn so với các giá trị độ lỗi còn lại.

2.2.6.5 Confidence Loss

Confidence loss đã thể hiện các giá trị điểm lỗi giữa các Bounding box đã chuẩn hóa và các Bounding box chưa chuẩn hóa. Điểm lỗi này gồm tất cả các ô vuông chứa và không chứa đối tượng.

$$L_{confidence} = \sum_{i=0}^{S^2} \sum_{j=0}^B l_i^{obj} (x_i - \hat{x}_i)^2 + \lambda_{coord} \sum_{j=0}^B I_{ij}^{noobj} (C_i - \hat{C}_i)^2 \quad (2-7)$$

2.2.6.6 Total loss

Tổng các hàm lỗi là tổng của 3 hàm lỗi trên :

$$L_{total} = L_{classification} + L_{localization} + L_{confidence} \quad (2-8)$$

2.2.7 Những cải tiến của YOLOv5

YOLOv5 được cải tiến trong PyTorch nên được hưởng những tính năng từ Pytorch vì thế dễ triển khai hơn. Được dùng trong các lĩnh vực học tập và nghiên cứu khả năng tự học nhanh, tốc độ chính xác cao. YOLOv5 với dung lượng nhỏ hơn so với nhiều mô hình CNN khác (YOLOv4 là 244 MB). Chính vì thế YOLOv5 được dùng cho nhiều thiết bị IoT vì dễ dàng và thuận tiện.

2.3 Giới thiệu về Google API

Google Drive là dịch vụ lưu trữ dữ liệu trên đám mây của google, cho phép người dùng lưu trữ dữ liệu với dung lượng miễn phí là 15GB và sử dụng quá giới hạn 15GB cần trả phí theo mức độ hàng tháng hàng năm để sử dụng Google Driver. Lý do sử dụng Google Drive là người dùng dễ dàng truy cập mọi lúc mọi nơi, miễn phí phù hợp cho quá trình học tập và nghiên cứu với dung lượng thấp và dễ dàng truy cập bằng mọi thiết bị thông minh, laptop.

Nhóm sử dụng Google Drive để khi xe đi qua khu vực có biển báo giao thông sẽ được lưu trữ lại và gửi thẳng lên Drive.

Để thuận tiện khi mỗi lần truy cập thì nhóm thực hiện tạo Credentials cho các ứng dụng API để tránh mỗi lần truy cập chương trình sẽ yêu cầu đăng nhập khi tạo ra file JSON mới.[8]



Hình 2. 21 Giới thiệu Google Drive API

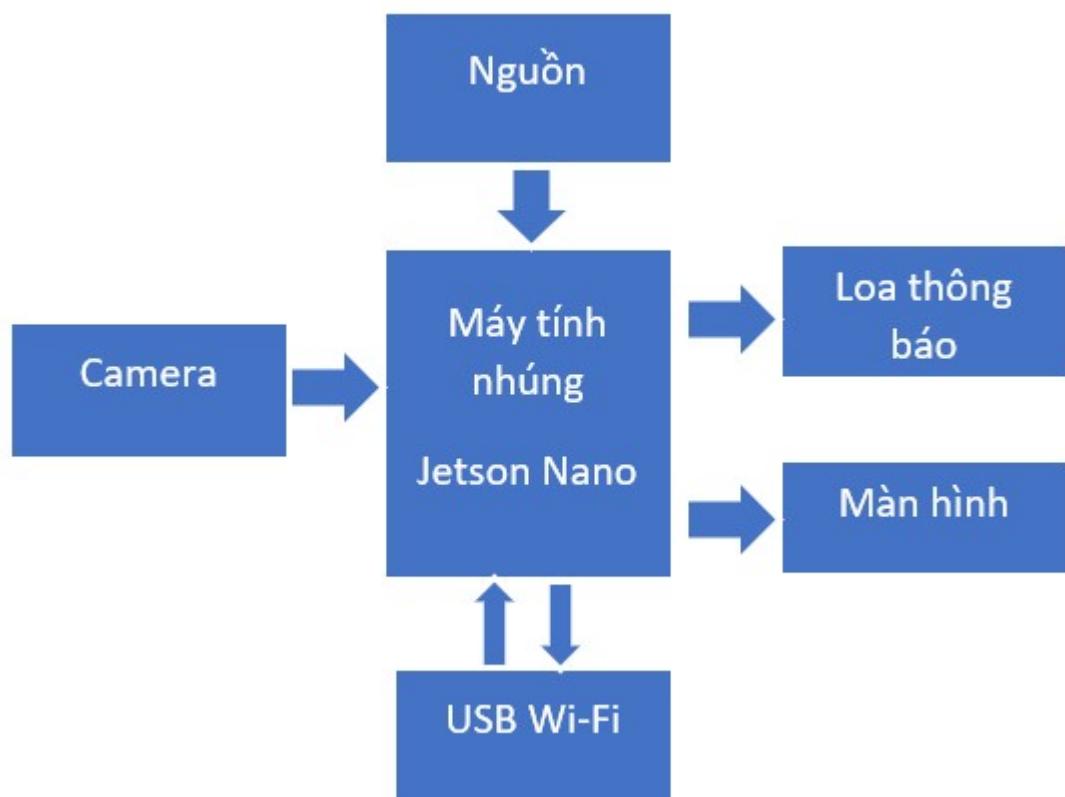
CHƯƠNG 3 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1 Phân tích

Thiết bị chia thành 2 phần: phần một là nhận diện và phân loại biển báo tốc độ, phần hai là thông báo cho người lái xe biết khi vào khu vực có biển báo tốc độ.

Để có thể nhận dạng và phân loại biển báo tốc độ cần trang bị camera để thu thập hình ảnh theo thời gian thực và gửi dữ liệu về bộ xử lý trung tâm. Để xử lý được dữ liệu thu thập được từ camera cần có bộ xử lý trung tâm có khả năng xử lý với tốc độ và độ chính xác cao.

Thiết kế sơ đồ khái niệm máy nhận dạng tốc độ:



Hình 3. 1 Sơ đồ khái niệm hệ thống thiết bị nhận diện

Chức năng của từng khái như sau:

- Nguồn được dùng để cung cấp nguồn cho toàn bộ hệ thống thiết bị nhận dạng bao gồm máy tính nhúng Jetson Nano, camera, màn hình, loa thông báo và một số thiết bị ngoại vi như usb wifi cập nhật dữ liệu lên Google API.

- Camera: thu thập nhận dữ liệu từ bên ngoài với những biển báo tốc độ giao thông và truyền dữ liệu vừa thu thập về máy tính nhúng Jetson Nano.
- Khối USB Wi-Fi sử dụng để truyền tải dữ liệu chụp được lên Google API và ngược lại cấp tín hiệu mạng cho máy tính nhúng.
- Màn hình: sử dụng để quan sát tốc độ nhận diện và hiển thị khoảng cách tương đối từ xe tới biển báo tốc độ.
- Loa thông báo: được truyền thông tin âm thanh từ bộ xử lý trung tâm ra loa giúp tài xế nghe và điều chỉnh tốc độ cho phù hợp.

3.2 Các bước xây dựng tập huấn luyện

Bước 1 Chuẩn bị tập dataset biển báo cho quá trình huấn luyện:

- Chuẩn bị những loại biển báo tốc độ 20, 30, 40, 50, 60, 70, 80, 90, 100, 120 sử dụng web makesense.ai vẽ label cho từng loại biển đưa ra thông số cho từng tấm hình.



Hình 3. 2 Tập dataset huấn luyện biển báo nhận diện



Hình 3. 3 Tập dataset huấn luyện biến báo nhận diện

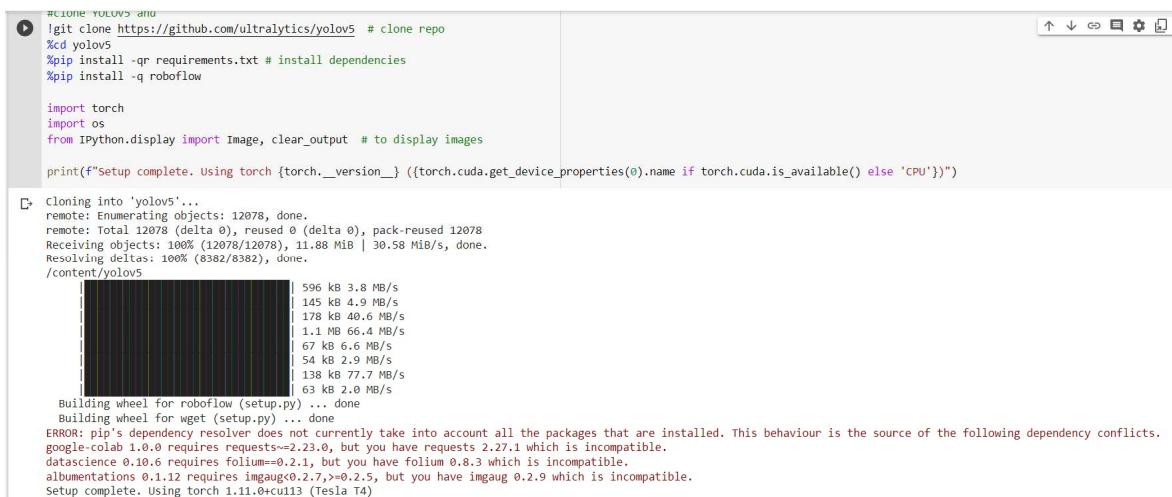
Bước 2 Quá trình huấn luyện:

- Google Colab là nền tảng được sử dụng miễn phí trong thời gian nhất định để huấn luyện mô hình giảm thiểu việc hao mòn nóng PC (máy tính cá nhân) khi train trực tiếp bằng CPU và GPU của máy.
- Trước khi huấn luyện mô hình cần liên kết với tài khoản Google.

```
[ ] from google.colab import drive
drive.mount('/content/drive')
```

Hình 3. 4 Liên kết Google colab với drive

- Nhóm sử dụng mô hình YOLOv5 để tiến hành huấn luyện trên Google Colab.



```
#clone yolov5 and
!git clone https://github.com/ultralytics/yolov5 # clone repo
%cd yolov5
%pip install -qr requirements.txt # install dependencies
%pip install -q roboflow

import torch
import os
from IPython.display import Image, clear_output # to display images

print(f"Setup complete. Using torch {torch.__version__} ({torch.cuda.get_device_properties(0).name if torch.cuda.is_available() else 'CPU'})")

Cloning into 'yolov5'...
remote: Enumerating objects: 12078, done.
remote: Total 12078 (delta 0), reused 0 (delta 0), pack-reused 12078
Receiving objects: 100% (12078/12078), 11.88 MiB | 30.58 MiB/s, done.
Resolving deltas: 100% (8382/8382), done.
/content/yolov5
Building wheel for roboflow (setup.py) ... done
Building wheel for wget (setup.py) ... done
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
google-colab 1.0.0 requires requests<~2.23.0, but you have requests 2.27.1 which is incompatible.
datscience 0.10.6 requires folium==0.2.1, but you have folium 0.8.3 which is incompatible.
albumentations 0.1.12 requires imgaug<0.2.7,>=0.2.5, but you have imgaug 0.2.9 which is incompatible.
Setup complete. Using torch 1.11.0+cu113 (Tesla T4)
```

Hình 3. 5 Sử dụng tài nguyên Tesla T4 hoặc Tesla K80 huấn luyện

- Khi liên kết ta được lựa chọn tài nguyên là GPU hoặc CPU thường thì train bằng GPU sẽ rất nhanh so với CPU; và GPU khi train thường có hai dạng là Tesla K80 và Tesla T4; khi huấn luyện gấp may mắn sẽ lựa chọn được Tesla T4 với tốc độ huấn luyện nhanh hơn K80 và nhanh gấp nhiều lần khi huấn luyện với CPU. Bởi do sài nguồn tài nguyên miễn phí nên chỉ sử dụng train model trong thời gian cho phép muốn sử dụng lâu hơn thì cần tính phí để tiếp tục.
- Giải nén file dataset coco128 có dữ liệu ảnh và thông số khi vẽ label.



```
#!unrar x /content/coco128.rar
!unrar x /content/drive/MyDrive/coco128.rar

UNRAR 5.50 freeware Copyright (c) 1993-2017 Alexander Roshal

Extracting from /content/drive/MyDrive/coco128.rar

Creating coco128 OK
Extracting coco128/_DS_Store OK
Creating coco128/images OK
Extracting coco128/images/_DS_Store OK
Creating coco128/images/train2017 OK
Extracting coco128/images/train2017/0.png OK
Extracting coco128/images/train2017/1.png OK
Extracting coco128/images/train2017/10.png OK
Extracting coco128/images/train2017/100.png OK
Extracting coco128/images/train2017/1000.png OK
Extracting coco128/images/train2017/1001.png OK
Extracting coco128/images/train2017/1002.png OK
Extracting coco128/images/train2017/1003.png OK
Extracting coco128/images/train2017/1004.png OK
Extracting coco128/images/train2017/1005.png OK
Extracting coco128/images/train2017/1006.png OK
Extracting coco128/images/train2017/1007.png OK
Extracting coco128/images/train2017/1008.png OK
```

Hình 3. 6 Giải nén tập dữ liệu huấn luyện bên Google drive

- Chính sửa file coco128.yaml trong Google Colab để lựa chọn đúng class theo đề tài và đường dẫn của ảnh đã được giải nén.

```

Resources coco128.yaml X
1 # YOLOv5 by Ultralytics, GPL-3.0 license
2 # COCO128 dataset https://www.kaggle.com/ultralytics/coco128 (first 128 images from COCO train2017) by Ultralytics
3 # Example usage: python train.py --data coco128.yaml
4 # parent
5 #   yolov5
6 #     datasets
7 #       coco128 + downloads here (7 MB)
8
9
10 # Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/to/imgs.txt, or 3) list: [path/to/imgs1, path/to/imgs2, ...]
11 path: ..content/yolov5/coco128 # dataset root dir
12 train: /content/yolov5/coco128/images/train2017 # train images (relative to 'path') 128 images
13 val: /content/yolov5/coco128/images/val2017 # val images (relative to 'path') 128 images
14 test: # test images (optional)
15
16 # Classes
17 nc: 10 # number of classes
18 names: ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'] # class names
19
20
21 # Download script/URL (optional)
22 download: https://ultralytics.com/assets/coco128.zip
23

```

Hình 3. 7 Điều chỉnh đường dẫn theo class biển báo

Bước 3 Tiến trình huấn luyện model

- Huấn luyện bằng YOLOv5 với kích thước đầu vào là 128 với 16, 32 và 64 lớp và 500 epoch.

```

#!python train.py --img 128 --batch 16 --epochs 100 --data /content/yolov5/data/coco128.yaml --weights yolov5l.pt --cache
#!python train.py --img 128 --batch 32 --epochs 500 --data /content/yolov5/data/coco128.yaml --weights yolov5x.pt --cache
#!python train.py --img 128 --batch 32 --epochs 600 --data /content/yolov5/data/coco128.yaml --weights /content/yolov5/runs/train/exp/weights/last.pt --cache

100% 755k [00:00<00:00, 17.0MB/s]
Downloading https://github.com/ultralytics/yolov5/releases/download/v6.1/yolov5x.pt to yolov5x.pt...
100% 166M/166M [00:02<00:00, 83.2MB/s]

Overriding model.yaml nc=80 with nc=10

      from    n    params   module           arguments
0      -1    1      8800  models.common.Conv  [3, 88, 6, 2, 2]
1      -1    1     115520  models.common.Conv  [80, 160, 3, 2]
2      -1    4     309120  models.common.C3  [160, 160, 4]
3      -1    1     461440  models.common.Conv  [160, 320, 3, 2]
4      -1    8     2259200  models.common.C3  [320, 320, 8]
5      -1    1     1844480  models.common.Conv  [320, 640, 3, 2]
6      -1   12     13125120  models.common.C3  [640, 640, 12]
7      -1    1     73753600  models.common.Conv  [640, 1280, 3, 2]
8      -1    4     19676160  models.common.C3  [1280, 1280, 4]
9      -1    1     4099840  models.common.SPPF  [1280, 1280, 5]
10     -1    1     820480  models.common.Conv  [1280, 640, 1, 1]
11     -1    1      0  torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
12     [-1, 6]    1      0  models.common.Concat  [1]
13     -1    4     5332480  models.common.C3  [1280, 640, 4, False]
14     -1    1     205440  models.common.Conv  [640, 320, 1, 1]
15     -1    1      0  torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
16     [-1, 4]    1      0  models.common.Concat  [1]
17     -1    4     1335840  models.common.C3  [640, 320, 4, False]
18     -1    1     922240  models.common.Conv  [320, 320, 3, 2]
19     [-1, 14]   1      0  models.common.Concat  [1]
20     -1    4     4922880  models.common.C3  [640, 640, 4, False]
21     -1    1     3687680  models.common.Conv  [640, 640, 3, 2]
22     [-1, 10]   1      0  models.common.Concat  [1]

Executing (14m 45s) Cell > system() > _system_compat() > _run_command() > _monitor_process() > _poll_process()

```

Hình 3. 8 Tiến hành huấn luyện

3.3 Ngôn ngữ lập trình và thư viện sử dụng

Python là một ngôn ngữ dễ dàng lập trình thao tác, không ròm rà về mặt hình thức giống những ngôn ngữ lập trình khác. Hiện tại mang trí tuệ nhân tạo, thị giác máy tính và xử lý ảnh đang được các chuyên gia lựa chọn Python để lập trình. Chính vì thế nhóm đã lựa chọn Python để lập trình cho đề tài. Ngoài việc lựa chọn ngôn ngữ lập trình Python thì cần có thư viện hỗ trợ trong quá trình lập trình điều khiển, nhận

dạng nhất là trong xử lý ảnh cũng như thị giác máy tính là Opencv. Thư viện Opencv là thư viện có nguồn mở được tối ưu tốt do đó được rất nhiều người lập trình và sử dụng thư viện mở này.



Hình 3. 9 Logo thư viện OpenCV

3.4 Các thành phần phần cứng sử dụng

3.3.1 Máy tính nhúng Jetson Nano

Jetson Nano là một máy tính nhúng mini tuy nhỏ gọn và nhẹ nhưng vô cùng mạnh mẽ và nó có thể chạy song song với mạng thần kinh cho các chương trình xử lý ảnh real-time. Jetson nano là một máy tính được tích hợp có 2 hình dạng: một module với kích thước 70x45mm để sử dụng trong các thiết kế sản phẩm và được sở hữu bộ nhớ trong 16GB. Dạng hai sẽ là một bộ kit phát triển giải pháp hoàn chỉnh cho nhà phát triển và sử dụng thẻ nhớ microSD như hình bên dưới.

Bảng 3. 1 Thông số kỹ thuật

Mode	Jetson Nano Developer Kit Carrier Board A02
GPU	128-core Maxwell
CPU	Quad-core ARM A57@ 1.43 GHz
Memory	4GB 64-bit LPDDR4 25.6 GB/s
Storage	microSD
Camera	1x MIPI CSI-2 DPHY lanes
Connectivity	Gigabit Ethernet, M.2 Key E
Display	HDMI 2.0 and eDP 1.4
USB	4x USB 3.0, USB 2.0 Micro-B
Others	GPIO, I2C, I2S, SPI, UART



Hình 3. 10 Jetson Nano Developer Kit Carrier Board A02

3.3.2 Màn hình hiển thị

Màn hình là thiết bị dùng để hiển thị hình ảnh, video theo dạng Pixel. Hầu hết màn hình có giải màu sắc để mắt con người có thể nhìn thấy sắc nét và chính xác.

Màn hình được thiết kế qua nhiều công nghệ, chất liệu kiểu dáng khác nhau:

- Màn hình dùng công nghệ điện tử tương tự (Analog).
- Màn hình dùng công nghệ điện tử số (giấy điện tử, màn hình LCD, màn hình tinh thể lỏng, màn hình OLED, màn hình Plasma).

- Màn hình chiếu (Rạp phim).

Nhóm lựa chọn màn hình LCD 4.3 inch HDMI(B) cảm ứng điện dung:

- Màn hình IPS.
- Độ phân giải màn hình 800x480, cấu hình bằng phần mềm nên tối 1920x1080.
- Hỗ trợ các loại máy tính nhúng nhỏ mini có cổng HDMI hoặc micro HDMI.
- Hỗ trợ jack 3.5mm đưa tín hiệu âm thanh từ máy tính nhúng ra loa.
- Jack micro-B cấp nguồn cho màn hình.



4.3inch HDMI LCD (B)

800x480, IPS

Hình 3. 11 Màn hình LCD 4.3 inch HDMI(B)

3.3.3 Webcam

Webcam là thiết bị ghi hình kỹ thuật số kết nối với PC, laptop qua cổng USB. Webcam cũng giống như mắt con người cũng thu thập hình ảnh nhận diện được đưa hình ảnh về cho máy tính cũng như mắt đưa về bộ não để xử lý. Nó truyền tải dữ liệu đó về máy tính nhúng, laptop, PC để xử lý. Ví dụ học online cần đến Webcam để giảng viên có thể quan sát sinh viên trên lớp qua một số trang mạng xã hội, nó phục vụ cho mọi hoạt động như công việc, học tập và giải trí. Hiện nay nhiều dòng webcam đã tích hợp sẵn mic thoại do đó webcam rất tiện lợi cho mọi lĩnh vực khi thiết bị PC, laptop thiết camera có sẵn.

Bảng 3. 2 Webcam HD Logitech C271 với thông số kỹ thuật

Mode	Logitech C270
Độ phân giải	720p/30fps
Loại tiêu cự	lấy nét cố định
Công nghệ thấu kính	tiêu chuẩn
Micro tích hợp	đơn âm
Phạm vi quan sát	60
Độ dài dây cáp	1.5m



Hình 3. 12 Webcam HD Logitech C271

3.3.4 Loa thông báo

Loa có chức năng thu nhập âm thanh từ các thiết bị xử lý phía trước hoặc từ đầu ra âm thanh tín hiệu từ laptop, điện thoại, tivi ... sau đó truyền về vi mạch âm thanh và tái tạo âm thanh tăng hoạt động công suất làm loa phóng đại những âm thanh cần phát ra.

Loa được dùng kết nối với máy tính nhúng jetson để đưa tín hiệu âm thanh ra loa và là thiết bị đầu ra phổ biến. Một vài loại loa được thiết kế để kết nối với tất cả những loại phát âm thanh như điện thoại, laptop.

Cấu tạo của loa có 6 bộ phận cấu tạo lên loa gồm khung sườn, viền nhún, màng nhện, nam châm, cuộn dây đồng, màng loa.

Bảng 3. 3 Thông số kỹ thuật loa JVJ JS20

Loa máy tính JVJ JS20	
Công suất	6W
Loa siêu trầm	3Wx2
Jack 3.5	1
Hệ thống yêu cầu	cổng USB để cấp nguồn giắc cắm 3.5
Cổng tín hiệu vào	3.5mm stereo
Tần số	20Hz - 20KHz

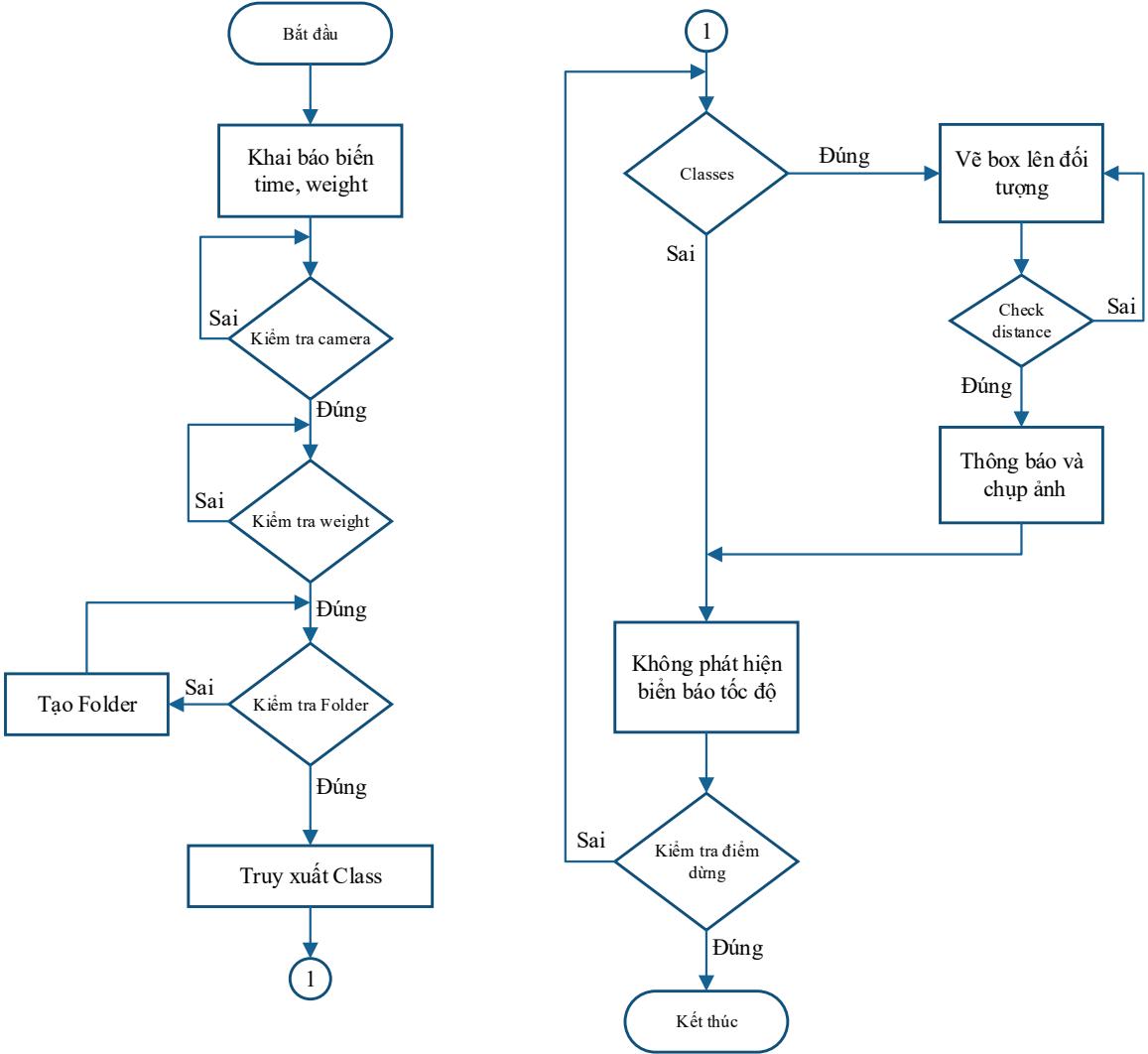


Hình 3. 13 Loa máy tính JVJ JS20

3.5 Nguyên lí hoạt động nhận diện biển báo giao thông

Thiết bị nhận dạng biển báo tốc độ được chia thành 3 phần:

- Phần 1: Nhận dạng, phân loại biển báo (gồm có các biển 20-30-40-50-60-70-80-90-100-120) và được hiển thị lên màn hình.
- Phần 2: Thông báo cho người lái xe biết khi đi vào có khu vực có biển báo tốc độ (thiết bị chỉ thông báo khi đúng khoảng cách đã được thiết lập trước đó).
- Phần 3: Biển báo khi đúng khoảng cách sẽ được chụp lại và gửi lên Google Drive Api.

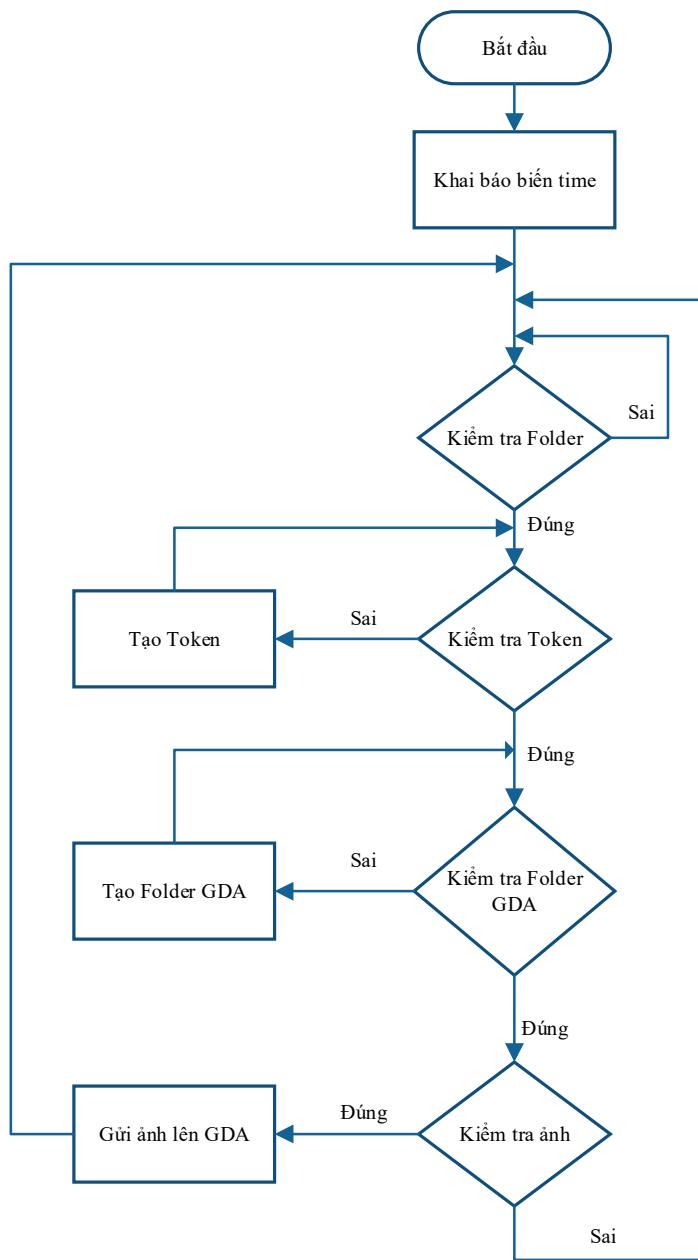


Hình 3. 14 Lưu đồ giải thuật quá trình nhận diện

Quá trình hoạt động của mô hình được thực hiện như sau:

- Khi thiết bị được bật, chương trình sẽ bắt đầu khai báo biến thời gian để sử dụng cho việc tạo folder và hiển thị thời gian lên màn hình. Khai báo file weight có được qua quá trình huấn luyện.
- Tiếp theo sẽ kiểm tra xem đã lắp camera và đã có file weight chưa, nếu chưa sẽ báo lỗi và dừng chương trình.
- Sau đó sẽ tạo folder với tên theo ngày để lưu các hình đã nhận diện được khi đúng khoảng cách. Truy xuất các class đã huấn luyện để tiến hành nhận dạng và phân loại các đối tượng. Khi phát hiện đối tượng mô hình sẽ tiến hành phân loại các class và vẽ bounding box cho đối tượng.

- Sau đó sẽ kiểm tra khoảng cách từ camera đến đối tượng. Nếu đối tượng nằm trong vùng đã được thiết lập thiết bị sẽ thông báo đồng thời chụp lại ảnh và lưu vào folder đúng ngày.
- Cuối cùng dữ liệu chụp được sẽ được gửi lên Google Drive.



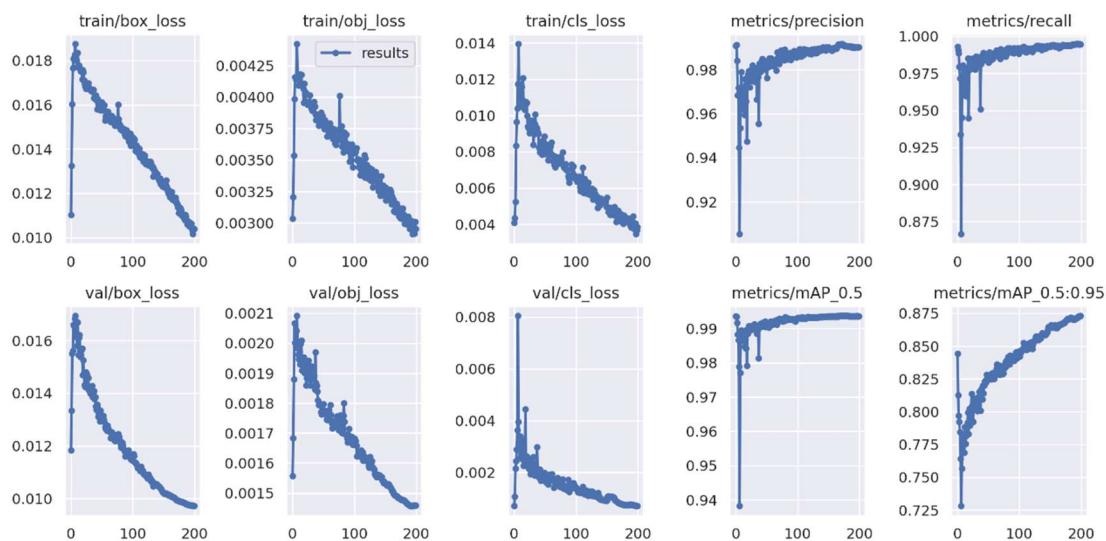
Hình 3. 15 Lưu đồ giải thuật GDA

Quá trình thực hiện gửi hình ảnh lên Google Drive được thực hiện như sau:

- Khi bắt đầu sẽ khai báo biến thời gian sau đó sẽ kiểm tra xem hệ thống đã tạo folder để chứa ảnh khi nhận dạng được chưa, nếu chưa sẽ đợi đến sẽ folder được tạo đúng ngày và tiến hành bước tiếp theo.
- Tiếp theo sẽ kiểm tra xem đã có file token chưa, nếu chưa sẽ tạo file token mới, nếu đã có thì sẽ tiến hành bước tiếp theo.
- Kiểm tra xem đã có folder đúng ngày tháng trên drive chưa, nếu chưa sẽ tạo folder mới đúng ngày tháng, nếu đã có sẽ tiếp hành gửi ảnh lên google drive.
- Bước tiến hành để gửi ảnh lên drive chương trình sẽ kiểm tra xem ảnh đó đã có trên google drive chưa, nếu chưa sẽ thì ảnh đó sẽ được gửi và ngược lại để tránh sự trùng lặp.

CHƯƠNG 4 THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

4.1 Đánh giá kết quả sau quá trình huấn luyện



Hình 4. 1 Kết quả sau quá trình huấn luyện

Thực hiện train nhỏ giọt khoảng 200 epoch (vì số lượng dataset khá lớn và mô hình xử dụng là yolov5x nên quá trình train cần kha nhiều thời gian, trung bình để train được 200 epoch cần 5 giờ mà thời gian sử dụng tối đa của google colab chỉ cho khoảng 6 giờ nên sử dụng phương pháp train nhỏ giọt này là hợp lý) để kiểm tra chất lượng sau mỗi lần train. Kết thúc quá trình huấn luyện 1600 epoch với batch_size = 16 và lưu file có kết quả tốt nhất (best.pt)

Đánh giá kết quả giá trị val/clc_loss (giá trị mất mát) và metrics/mAP (độ chính xác):

- Giá trị val/clc_loss được tính bằng tổng các loss của các class. Giá trị loss càng giảm cho thấy độ mất mát của quá trình huấn luyện đang tốt, giảm thiểu được các tình trạng nhiễu và nhận dạng sai
- Giá trị metrics/mAP (độ chính xác) dần về bằng 1 cho thấy quá trình huấn luyện đang diễn ra tốt.

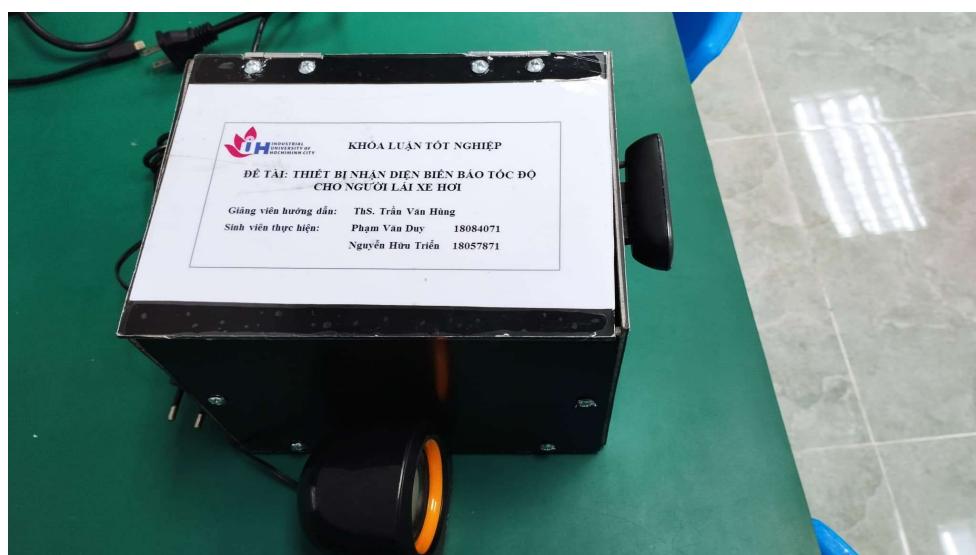
Bảng 4. 1 Thống kê độ chính xác từng Class

Class	Labels	Độ chính xác
ALL	4624	0.873
20	479	0.872
30	493	0.876
40	507	0.875
50	528	0.893
60	553	0.874
70	449	0.872
80	469	0.891
90	293	0.86
100	527	0.87
120	336	0.85

- Độ chính xác của mô hình được tính bằng trung bình cộng của tất cả các class.
- Độ chính xác của các class còn phụ thuộc vào nhiều yếu tố như chất lượng ảnh dataset, số lượng dataset cho từng class và số lượng labels. Đa số các class có số lượng dataset và label lớn sẽ có độ chính xác cao.

4.2 Sản phẩm

Chất liệu chính làm sản phẩm là tấm Alu.



Hình 4. 2 Mặt ngang của thiết bị



Hình 4. 3 Mặt trước của sản phẩm thiết bị



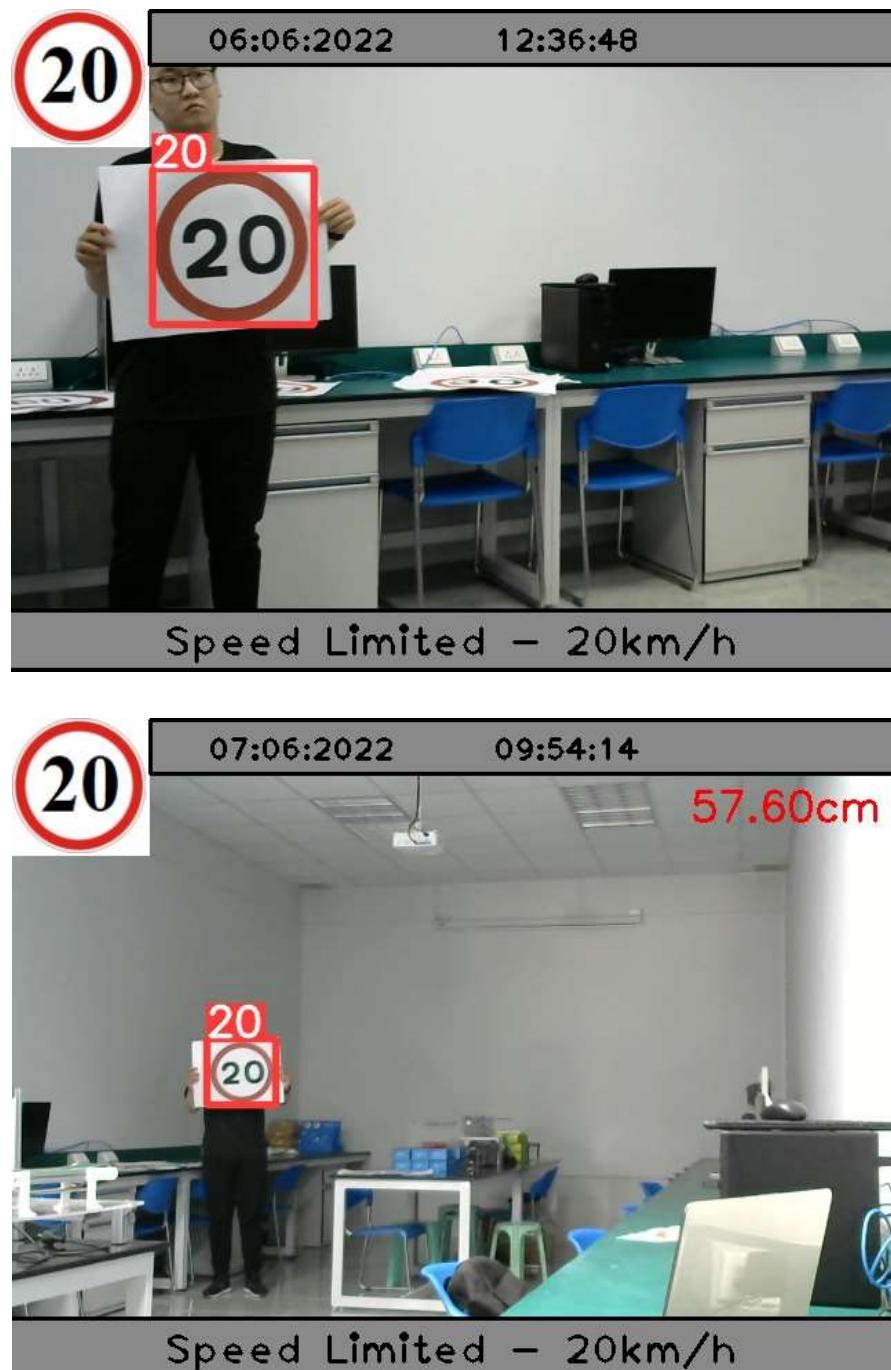
Hình 4. 4 Phần cứng bên trong của thiết bị



Hình 4. 5 Mặt sau của thiết bị nhận dạng

4.3 Kết quả thực nghiệm

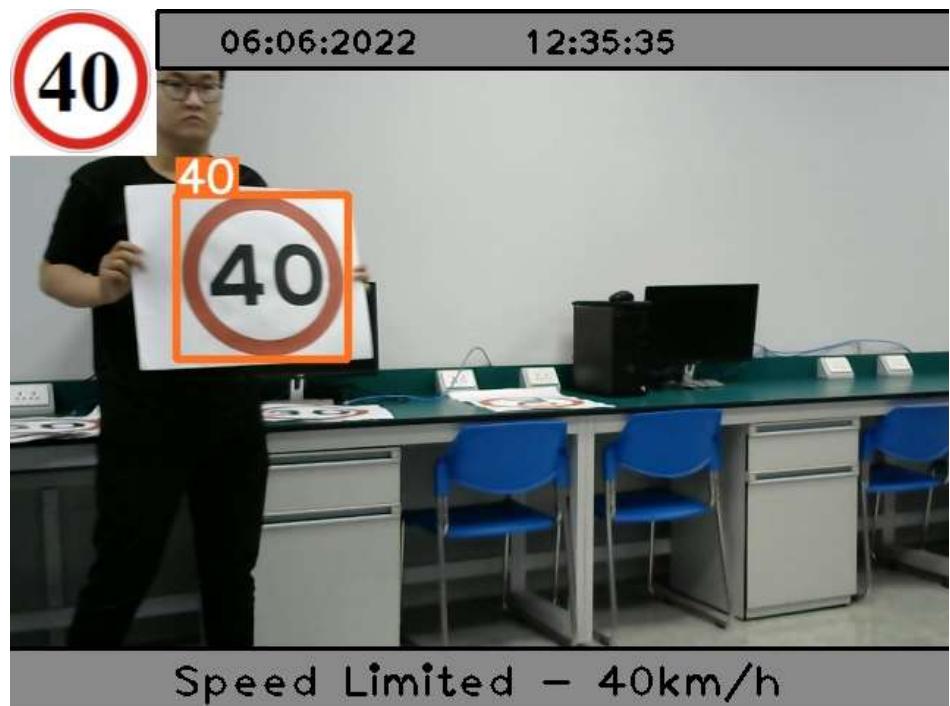
4.3.1 Hình ảnh nhận diện được các biển báo tốc độ



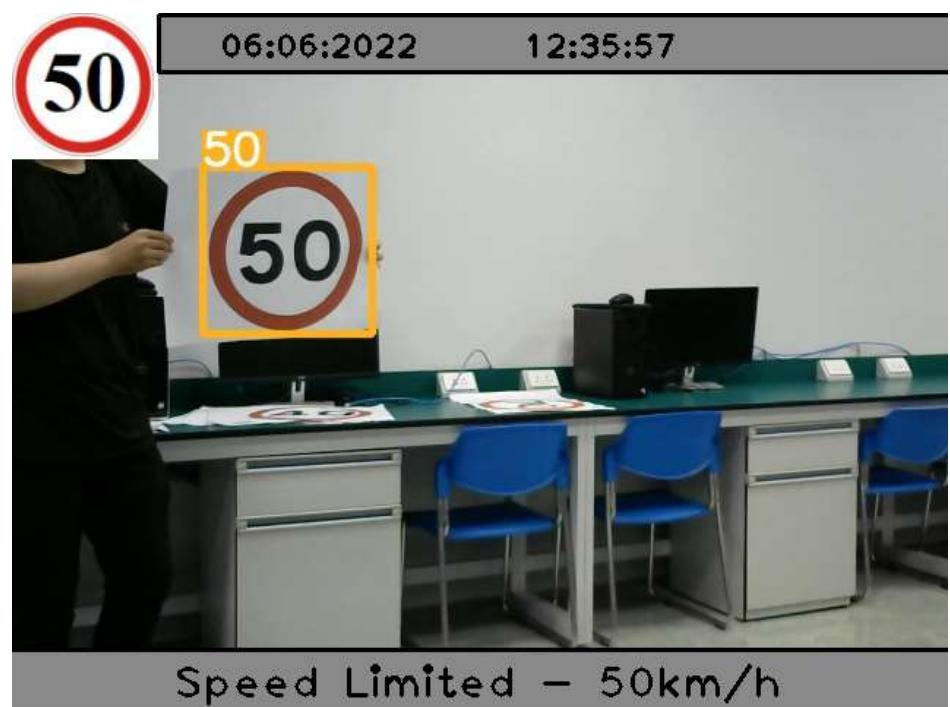
Hình 4. 6 Detect biển báo 20 km/h



Hình 4. 7 Detect biển báo 30 km/h

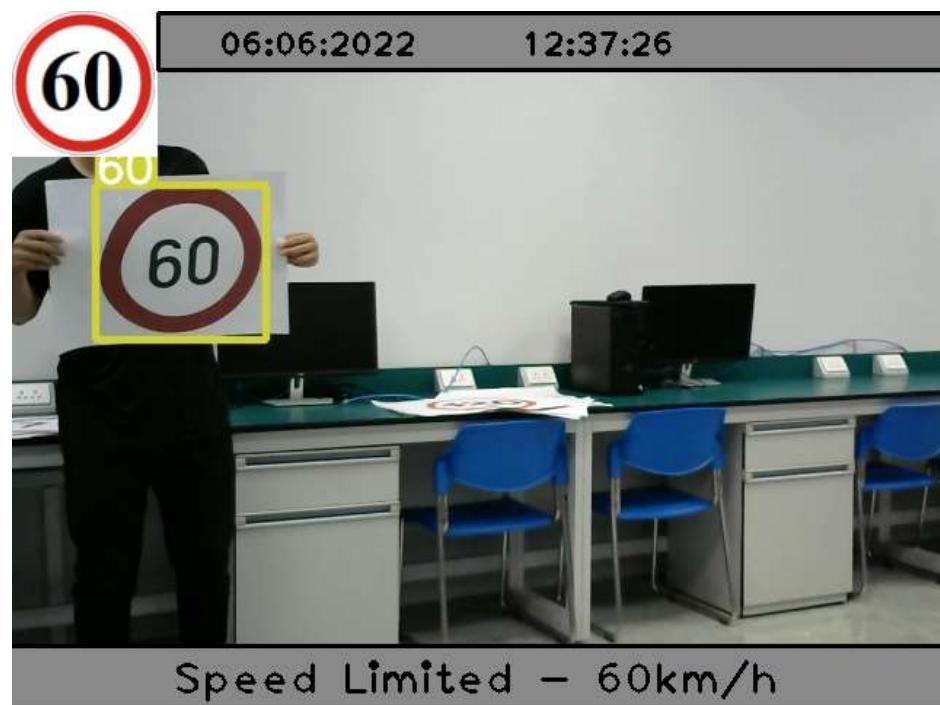


Hình 4. 8 Detect biển báo 40 km/h

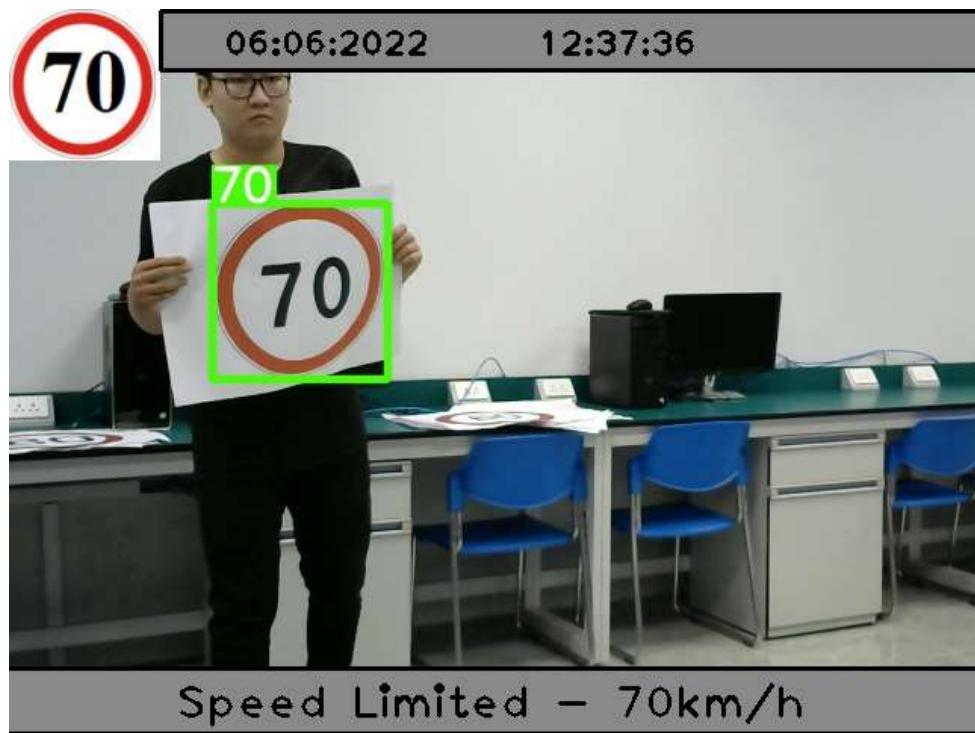




Hình 4. 9 Detect biển báo 50 km/h



Hình 4. 10 Detect biển báo 60 km/h



Hình 4. 11 Detect biển báo 70 km/h



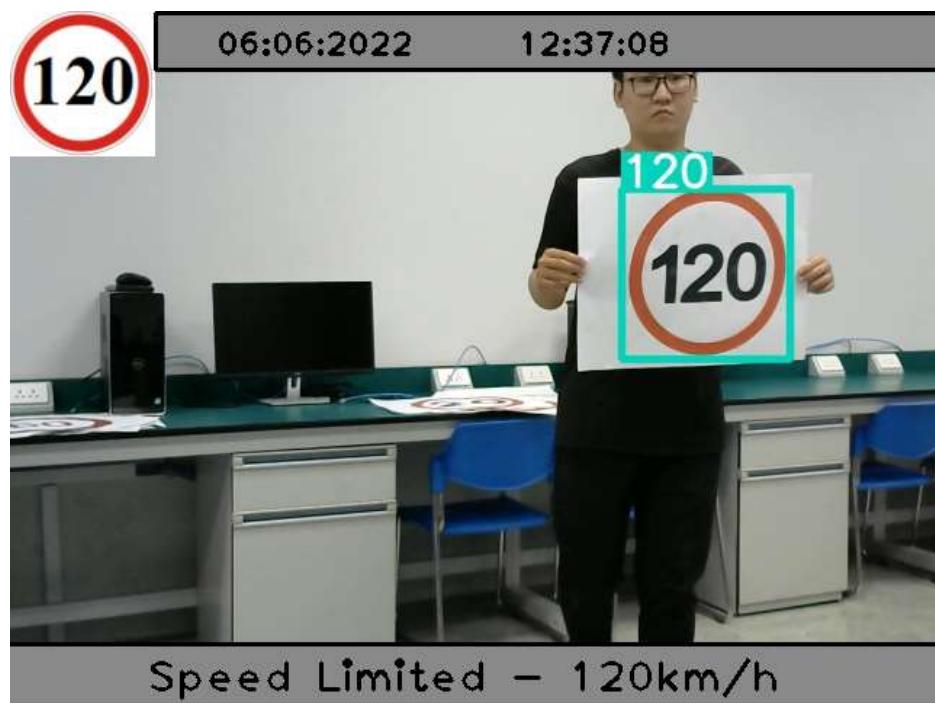
Hình 4. 12 Detect biển báo 80 km/h



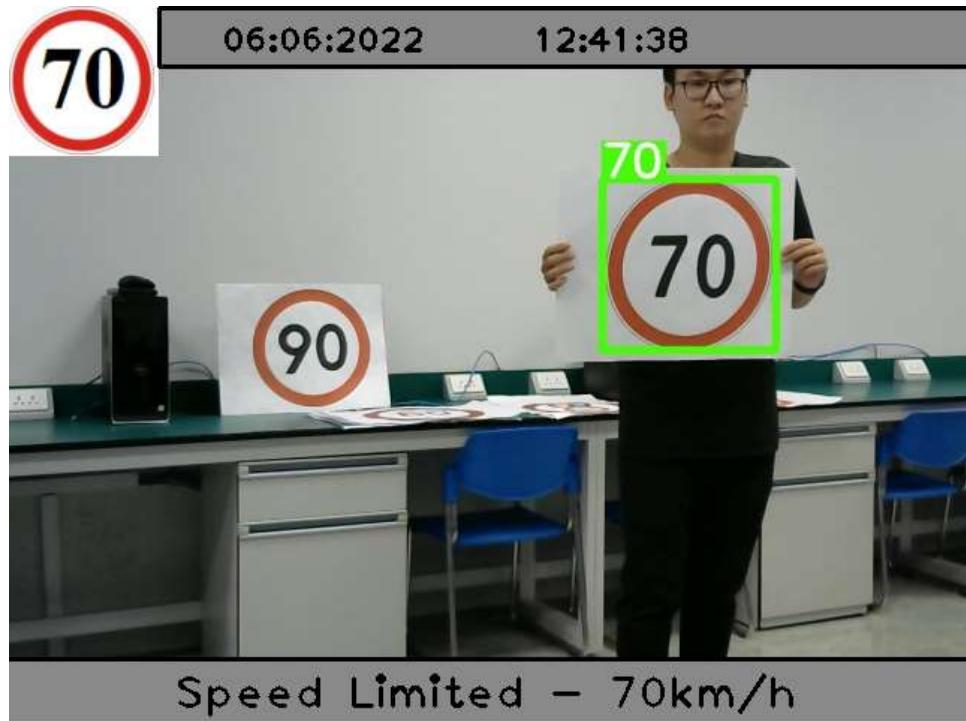
Hình 4. 13 Detect biển báo 90 km/h



Hình 4. 14 Detect biển báo 100 km/h



Hình 4. 15 Detect biển báo 120 km/h

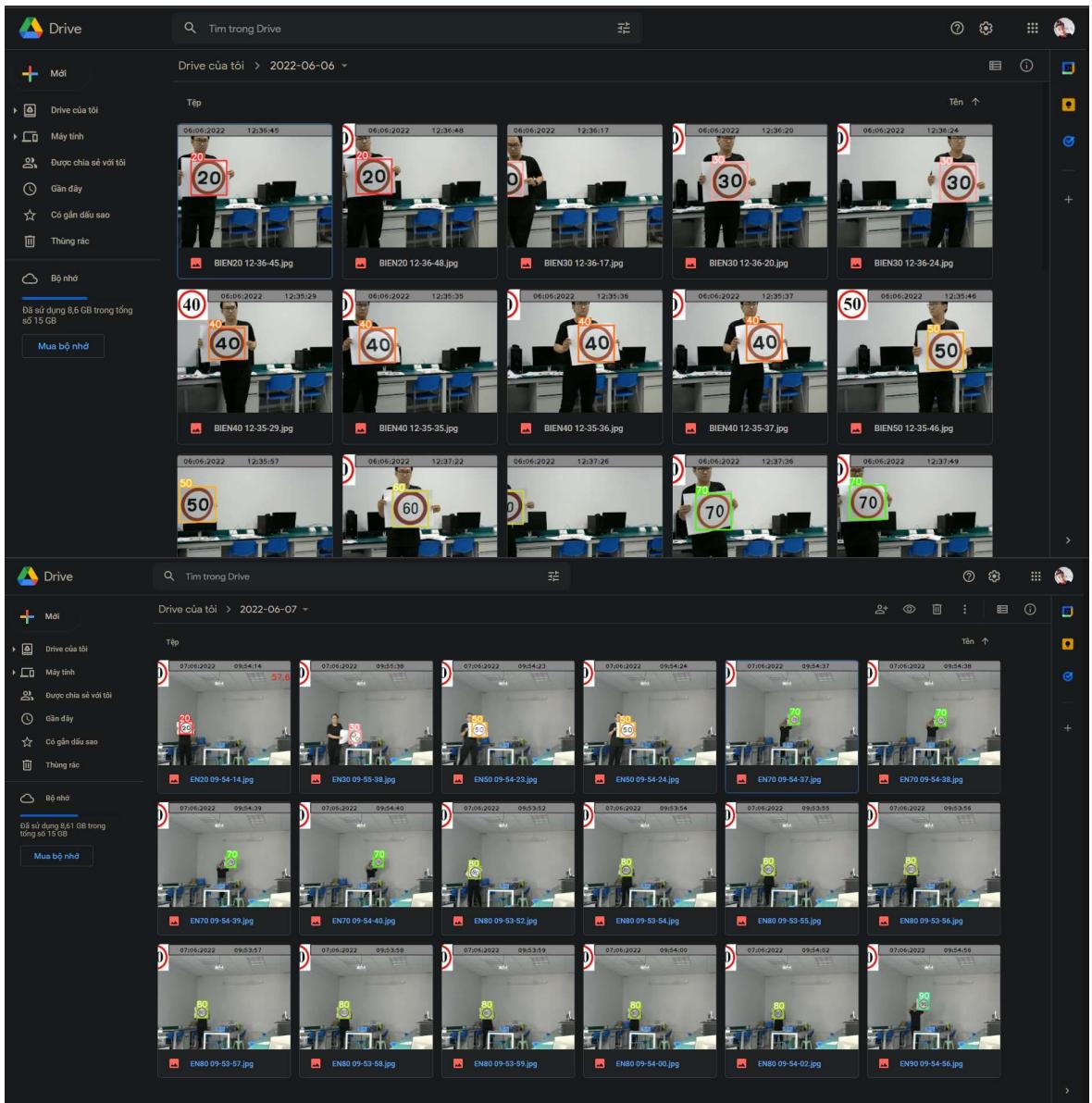


Hình 4. 16 Detect trong trường hợp còn nhiều hơn 1 biển báo

Nhận xét kết quả:

- Nhận dạng được tất cả các biển báo tốc độ đã được huấn luyện gồm các biển 20-30-40-50-60-70-80-90-100-120.
- Nhận dạng tốt các biển báo ở gần.
- Các biển báo ở xa chưa có độ chính xác tốt và còn tương đối nhiều.
- Các biển có độ chính xác cao là: 20-30-40-50-70-80.
- Trong trường hợp còn nhiều hơn 1 biển báo tốc độ sẽ ưu tiên nhận diện biển báo gần thiết bị hơn.

4.3.2 *Ảnh nhận diện được đúng khoảng cách được gửi lên Google Drive*



Hình 4. 17 Hình ảnh lưu được sẽ gửi lên Google Drive API

CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Mô hình xây dựng cơ bản đã nhận biết được những biển báo tốc độ giao thông mà nhóm đã thực hiện dựa trên tập dataset huấn luyện bằng mạng YOLOv5.

Độ chính xác một vài biển còn chưa cao do tập dataset huấn luyện còn ít, độ đa dạng về màu sắc cũng như ánh sáng biển báo lúc ngày và lúc đêm cũng chưa cụ thể rõ ràng.

5.1.1 Ưu điểm

Nhận diện những biển báo tốc độ cơ bản trong khoảng cách gần.

Thông báo chính xác khi đúng khoảng cách nhận diện biển báo.

Gửi dữ liệu ảnh chụp đúng biển báo tốc độ lên Google API.

5.1.2 Nhược điểm

Nhận diện còn hạn chế trong cự li xa.

Nhận diện vẫn còn sai và hơi nhiều khi ở cự ly xa, trời mưa và lúc trời tối thiếu ánh sáng.

Khả năng nhận diện real-time còn chậm do sử dụng máy tính nhúng chưa xử lý tốt và Webcam nhận diện còn hạn chế.

Camera chưa có khả năng chống rung khi xe đi đường xấu.

5.2 Hướng phát triển

Cải tiến những giải thuật xử lý ảnh, thu thập thêm nhiều dataset chất lượng tốt hơn.

Thay đổi thông số, thuật toán khi huấn luyện nhận diện biển báo.

Cập nhật đa dạng số tập mẫu để nhận dạng chi tiết rõ ràng hơn, chính xác khi đưa ra lắp đặt thiết bị thực tiễn.

Cải tiến camera chất lượng hơn khi lắp đặt thiết bị sử dụng ngoài thực tiễn.

TÀI LIỆU THAM KHẢO

- [1] P. J. Denning, "Editorial: Plagiarism in the Web," *Communications of the ACM*, vol. 98, no. 12, p. 29, Dec. 1995.
- [2] B. Martin, "Plagiarism: a misplaced emphasis," *Journal of Information Ethics*, vol. 3, no. 2, pp. 36-47, 1994.
- [3] B. Belkhouche et al, "Plagiarism detection in software designs," in *Proc. of the 42nd Ann. Southeast Regional Conf.*, 2004.
- [4] T. Doe, Dec. 2011. [Online]. Available:
<http://grad.uark.edu/dean/thesisguide.php>.
- [5] Cornell University Library PSEC Documentation Committee, Feb. 2010. [Online]. Available: <http://www.library.cornell.edu/resrch/citmanage/apa>.
- [6] N. D. Tập, "Lập trình bằng hợp ngữ," Nhà xuất bản Khoa học kỹ thuật, Hà Nội, 1998.
- [7] <https://machinelearningcoban.com/2017/03/04/overfitting/#-regularization>, "Machinelearningcoban," [Online]. [Accessed 05 2022].
- [8] <https://developers.google.com/drive/api/guides/about-sdk>, "Google Drive API", [Online]. [Accessed 04 2022].
- [9] <https://aicurious.io/posts/tim-hieu-yolo-cho-phat-hien-vat-tu-v1-den-v3/>, "YOLOv5", [Online]
- [10] <https://vi.wikipedia.org/wiki/Pixel>, "what is Pixel", [Online].