

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG



ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

Sinh viên : Vũ Tiến Tâm

Giảng viên hướng dẫn: TS. Hồ Thị Hương Thơm

Hải Phòng -2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

XÂY DỰNG HỆ THỐNG CẢNH BÁO CHÁY TỪ
HÌNH ẢNH BẰNG YOLOV7

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH: CÔNG NGHỆ THÔNG TIN

Sinh viên thực hiện : Vũ Tiến Tâm

Giảng viên hướng dẫn: TS. Hồ Thị Hương Thơm

Hải Phòng - 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên : Vũ Tiến Tâm - **MSV** : 2012111013

Lớp : CT2401C

Ngành : Công Nghệ Thông Tin

Tên đề tài : Xây dựng hệ thống cảnh báo cháy bằng YoloV7

NHIỆM VỤ ĐỀ TÀI

1 Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp (về lý luận, thực tiễn, các số liệu cần tính toán và các bản vẽ).

a. Mô tả tóm tắt đề tài

Tìm hiểu và xây dựng chương trình cảnh báo cháy với YOLOv7.

b. Nội dung hướng dẫn

- Tìm hiểu về hiện trạng cháy nổ xảy ra hiện nay và bài toán cảnh báo cháy nổ để hạn chế rủi ro.
- Tìm hiểu tổng quan về bài toán phát hiện đối tượng và mô hình phát hiện đối tượng bằng YOLOv7.
- Thu thập dữ liệu hình ảnh cháy để huấn luyện mô hình phát hiện cháy
- Cài đặt mô hình cảnh báo cháy bằng YOLOv7 và xây dựng chương trình ứng dụng mô hình phát hiện trên hình ảnh và videos.
- Nhận xét, đánh giá và kết luận.

c. Kết quả cần đạt được

- Đã xây dựng được chương trình cảnh báo cháy bằng YOLOv7 từ dữ liệu hình ảnh và videos
- Viết báo cáo đồ án tốt nghiệp

2. Các số liệu cần thiết để tính toán.

- Tài liệu tham khảo về ngôn ngữ python, các thư viện hỗ trợ
- Huấn luyện dữ liệu trên colab
- Dữ liệu về phát hiện đối tượng bằng phương pháp học sâu và YOLO
- Nhận xét, đánh giá và kết luận

3. Địa điểm thực tập tốt nghiệp.

CÁC CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Họ và tên : Hồ Thị Hương Thom

Học hàm, học vị : Tiến sĩ

Cơ quan công tác : Trường Đại học Hàng Hải Việt Nam

Nội dung hướng dẫn:

- Tìm hiểu về bài toán cảnh báo cháy từ hình ảnh bằng YoloV7
- Tìm hiểu phương pháp phát hiện, nhận diện đám cháy, ngôn ngữ lập trình python để xây dựng chương trình cảnh báo cháy từ hình ảnh bằng YoloV7
- Thử nghiệm, nhận xét, đánh giá và kết luận

Kết quả cần đạt được

- Tài liệu mô tả các kết quả đã thực hiện được
- Chương trình cảnh báo cháy từ hình ảnh bằng YoloV7

Đề tài tốt nghiệp được giao ngày 13 tháng 1 năm 2024

Yêu cầu phải hoàn thành xong trước ngày 20 tháng 05 năm 2024

Đã nhận nhiệm vụ ĐTTN

Sinh viên

Đã giao nhiệm vụ ĐTTN

Giảng viên hướng dẫn

Hải Phòng, ngày tháng năm 202.....

TRƯỞNG KHOA

Cộng Hòa Xã Hội Chủ Nghĩa Việt Nam
Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP

Họ và tên giảng viên : Hồ Thị Hương Thơm
Đơn vị công tác : Trường Đại học Hàng Hải Việt Nam
Họ và tên sinh viên : Vũ Tiến Tâm
Chuyên ngành : Công Nghệ Thông Tin
Nội dung hướng dẫn : Toàn bộ đề tài

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp

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

2. Đánh giá chất lượng của đồ án/khóa luận (so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T.T.N, trên các mặt lý luận, thực tiễn, tính toán số liệu...)

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

3. Ý kiến của giảng viên hướng dẫn tốt nghiệp

Được bảo vệ Không được bảo vệ Điểm hướng dẫn

Hải Phòng, ngày.....tháng.....năm 2023

Giảng viên hướng dẫn

(ký và ghi rõ họ tên)

Cộng hòa xã hội chủ nghĩa Việt Nam
Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN CHẤM PHẢN BIỆN

Họ và tên giảng viên

Đơn vị công tác:.....

Họ và tên sinh viên: Chuyên ngành:.....

Đề tài tốt nghiệp:

.....

1. Phần nhận xét của giảng viên chấm phản biện

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

2. Những mặt còn hạn chế

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

3. Ý kiến của giảng viên chấm phản biện

Được bảo vệ Không được bảo vệ Điểm phản biện

Hải Phòng, ngày.....tháng.....năm 2024

Giảng viên chấm phản biện
(ký và ghi rõ họ tên)

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI	5
1.1. Đặt vấn đề	5
1.2. Khái niệm.....	5
1.2.1. Đối tượng trong hình ảnh	5
1.2.2. Phát hiện đối tượng	6
1.2.3. Nhận dạng đối tượng.....	6
1.3. Các phương pháp phát hiện phổ biến hiện nay.....	6
1.3.1.YOLO (You Only Look Once).....	6
1.3.2. SSD (Single Shot MultiBox Detector)	9
1.4. Bài toán cảnh báo cháy từ hình ảnh	10
CHƯƠNG 2. MÔ HÌNH CẢNH BÁO CHÁY TỪ HÌNH ẢNH	12
2.1. Sơ đồ hoạt động của mô hình cảnh báo cháy.....	12
2.2. Phương pháp phát hiện đối tượng bằng YOLOv7	13
2.2.1. Phát hiện đối tượng.....	13
2.3. Giới thiệu về cảnh báo cháy bằng YOLOv7.....	15
2.4. Lựa chọn phiên bản YOLO trong phát hiện cháy và cảnh báo.....	16
CHƯƠNG 3. CÀI ĐẶT, THỬ NGHIỆM VÀ ĐÁNH GIÁ	19
3.1. Môi trường cài đặt.....	19
3.1.1. Giới thiệu môi trường Google Colab.....	19
3.1.2. Thực thi trên môi trường Google Colab	20
3.1.3. Phân chia tập dữ liệu huấn luyện.....	20
3.2. Môi trường cài đặt.....	21
3.2.1. Ngôn ngữ Python	21
3.2.2. Thư viện	22
3.3. Tạo bộ dữ liệu học để huấn luyện	23
3.3.1. Giới thiệu bộ dữ liệu phát hiện cháy(fire-detection)	23
3.3.2. Gắn nhãn cho bộ dữ liệu FireData.....	23
3.3. Đào tạo mô hình, huấn luyện cảnh báo cháy	25
3.3.1. Huấn luyện mô hình.....	25
3.3.2. Giá trị của độ chính xác của từng lớp (Precision)	27

3.3.3. Giá trị của độ nhớ (recall).....	27
3.3.4. Biểu đồ mAP_0.5.....	28
3.3.5. Biểu đồ Trung bình của các giá trị AP trên nhiều ngưỡng IoU từ 0.5 đến 0.95.	28
3.3.6. Huấn luyện mô hình.....	29
3.4. Đánh giá mô hình sau thử nghiệm	31
KẾT LUẬN	32
TÀI LIỆU THAM KHẢO	33

Danh mục hình ảnh

Hình 1.1: Sơ đồ kiến trúc mạng YOLO.....	7
Hình 1.2: Các bản đồ đặc trưng (feature maps) của mạng YOLO với hình dạng đầu vào (input shape) là 416x416, hình dạng đầu ra (output shape) có kích thước lần lượt là 13x13, 26x26 và 52x52.	8
khác nhau.	10
Hình 1.3: Cách thức phân chia feature map để nhận diện các hình ảnh với những kích thước	
Hình 1.4 Sơ đồ kiến trúc của mạng SSD.....	10
Hình 2.1. Sơ đồ hoạt động của mô hình cảnh báo cháy.....	12
Hình 2.2: Kiến trúc YOLOv7 và cơ chế hoạt động.....	14
Hình 2.3: Sơ đồ tổng hợp lớp của YOLO v7.....	17
Hình 3.1: Cấu trúc thư mục sau khi phân chia dữ liệu huấn luyện.....	21
Hình 3.2: Bộ dữ liệu Data-detection.....	23
Hình 3.3: Gắn nhãn cho bộ dữ liệu Fire-detection.....	24
Hình 3.4: Phần ảnh sau khi được phát hiện đối tượng lửa được gắn nhãn.....	24
Hình 3.5: Chi tiết một tập tin chứa tọa độ đối tượng lửa được gắn nhãn.....	25
Hình 3.6: Biểu đồ giá trị độ chính xác của từng lớp qua mỗi giai đoạn.....	27
Hình 3.7: Biểu đồ giá trị độ chính xác qua mỗi giai đoạn.....	27
Hình 3.8: Biểu đồ giá trị mAP_0.5 qua từng giai đoạn.....	28
Hình 3.9: Biểu đồ giá trị mAP_0.5:0.95 qua từng giai đoạn.....	28
Hình 3.10: Hình ảnh cháy được phát hiện.....	29
Hình 3.11: Hình ảnh cháy được phát hiện.....	29
Hình 3.12: Hình ảnh cháy được phát hiện.....	30
Hình 3.13: Hình ảnh cháy được phát hiện.....	30
Hình 3.14: Hình ảnh cháy được phát hiện.....	31

DANH MỤC CÁC TỪ VIẾT TẮT

STT	Ký hiệu chữ viết tắt	Chữ viết đầy đủ
1	YOLO	You Only Look Once
2	CNN	Neuron Convolutional
3	SSD	Single Shot MultiBox Detector
4	R-CNN	Region-based Convolutional Neural Network
5	RAM	Random Access Memory
6	NMS	Non-maximum Suppression
7	NAS	Network Attached Storage
8	VoVNet	Variety of View Network
9	GPU	Graphics Processing Unit
10	TPU	Tensor Processing Unit
11	URL	Uniform Resource Locator
12	MVC	Model-View-Controller

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

1.1. Đặt vấn đề

Trong thời đại công nghệ ngày nay, việc sử dụng trí tuệ nhân tạo và thị giác máy tính đang ngày càng trở nên phổ biến và có ứng dụng rộng rãi trong nhiều lĩnh vực, từ giám sát an ninh đến quản lý rủi ro và cứu hỏa. Trong bối cảnh đó, việc phát triển hệ thống cảnh báo cháy từ hình ảnh không chỉ mang lại lợi ích về tính nhanh chóng và hiệu quả mà còn đóng vai trò quan trọng trong việc bảo vệ người và tài sản.

Đề án này tập trung vào việc áp dụng một trong những công nghệ tiên tiến nhất trong lĩnh vực nhận diện đối tượng, đó là YOLOv7 (You Only Look Once version 7), để phát hiện và cảnh báo về nguy cơ cháy từ hình ảnh. YOLOv7 là một mô hình nhận diện vật thể mạnh mẽ, có khả năng phát hiện vật thể nhanh chóng và chính xác trên cả ảnh và video.

Trong đề án này, em đã tiến hành thu thập dữ liệu, huấn luyện mô hình và phát triển một hệ thống cảnh báo cháy từ hình ảnh sử dụng YOLOv7. Em hy vọng rằng công việc của em sẽ đóng góp vào việc tăng cường khả năng phát hiện và phản ứng sớm trước nguy cơ cháy, từ đó giảm thiểu thiệt hại và nguy cơ cho cộng đồng.

1.2. Khái niệm

1.2.1. Đối tượng trong hình ảnh

Đối tượng trong hình ảnh là các yếu tố, vật thể, hoặc phần tử cụ thể mà máy ảnh hoặc người xem hình ảnh có thể nhận diện được. Khái niệm này không chỉ áp dụng cho nhiếp ảnh mà còn cho các lĩnh vực khác như xử lý ảnh, trí tuệ nhân tạo, và thị giác máy tính.

Một đối tượng trong hình ảnh có thể là bất kỳ thứ gì từ con người, động vật, đến các đối tượng vật lý như xe hơi, cây cỏ, đồ đạc trong phòng, v.v. Trong phân tích hình ảnh, việc nhận diện và phân loại đối tượng là một trong những thách thức quan trọng.

1.2.2. Phát hiện đối tượng

Đây là quá trình tìm kiếm và xác định vị trí của đối tượng trong một hình ảnh. Phát hiện đối tượng thường bao gồm việc xác định các hộp giới hạn (bounding boxes) xung quanh các đối tượng trong hình ảnh cũng như xác định loại đối tượng. Các phương pháp phổ biến cho việc phát hiện đối tượng bao gồm sử dụng các mô hình học sâu như các mạng nơ-ron convolutional (CNN) và các kỹ thuật như R-CNN và các phiên bản cải tiến, YOLO (You Only Look Once), SSD, và RetinaNet.

1.2.3. Nhận dạng đối tượng

Sau khi đối tượng được phát hiện, bước tiếp theo là nhận dạng đối tượng, tức là xác định chính xác loại đối tượng đó là gì. Điều này thường liên quan đến việc so sánh các đặc trưng của đối tượng đã được rút trích từ hình ảnh với các đặc trưng của các đối tượng đã biết trước trong cơ sở dữ liệu. Các phương pháp cho việc nhận dạng đối tượng có thể dựa trên học máy, học sâu, hoặc các kỹ thuật xử lý ảnh truyền thống.

Cả phát hiện và nhận dạng đối tượng đều là những thách thức quan trọng trong lĩnh vực trí tuệ nhân tạo và có ứng dụng rộng rãi trong nhiều lĩnh vực như nhận dạng khuôn mặt, giám sát an ninh, tự động hóa công nghiệp, xe tự lái, và nhiều ứng dụng khác.

1.3. Các phương pháp phát hiện phổ biến hiện nay

1.3.1. YOLO (You Only Look Once)

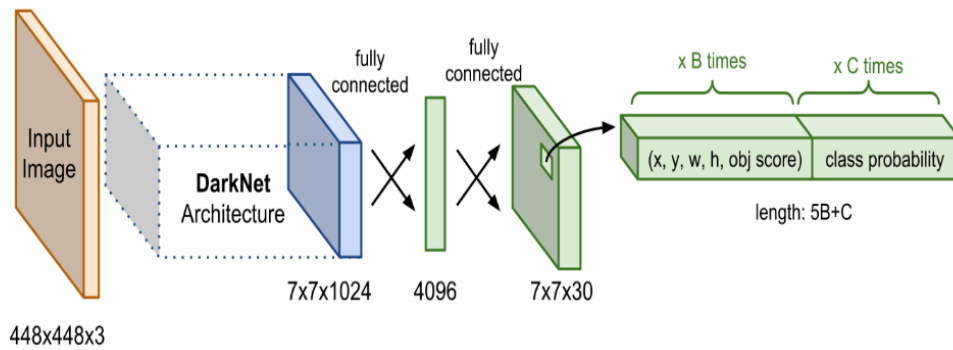
YOLO là phương pháp phát hiện đối tượng nhanh chóng và chính xác bằng cách sử dụng một mạng nơ-ron duy nhất để dự đoán các hộp giới hạn và xác suất của các lớp đối tượng trong một lần chạy.

YOLO được biết đến với khả năng thời gian thực và hiệu suất cao. Thật vậy, về độ chính xác thì YOLO có thể không phải là thuật toán tốt nhất nhưng nó là thuật toán nhanh nhất trong các lớp mô hình phát hiện đối tượng (object detection). Nó có thể đạt được tốc độ gần như real time mà độ chính xác không quá giảm so với cái model thuộc top đầu.

Kiến trúc mạng YOLO bao gồm base network là các mạng phép tích chập (convolution) làm nhiệm vụ trích xuất đặc trưng. Phần phía sau là những lớp bổ sung(

Extra Laves) được áp dụng để phát hiện vật thể trên bản đồ đặc trưng (feature map) của mạng cơ sở (base network).

- Sơ đồ kiến trúc mạng của YOLO

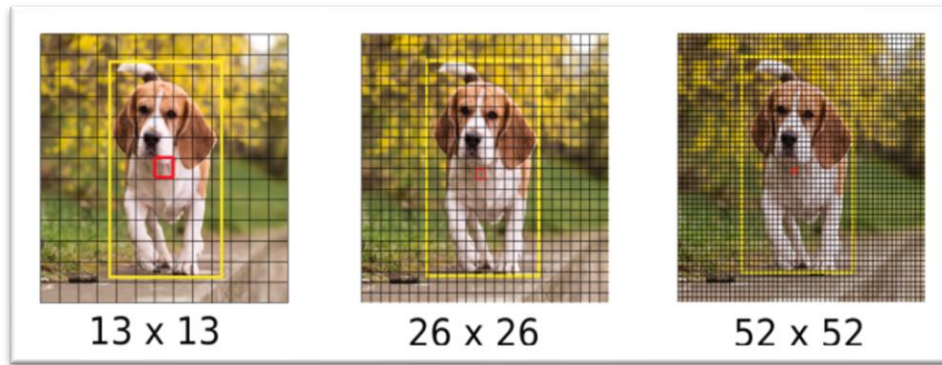


Hình 1.1: Sơ đồ kiến trúc mạng YOLO

Thành phần kiến trúc darknet (Darknet Architecture) được gọi là mạng cơ sở có tác dụng trích xuất đặc trưng. Đầu ra của mạng cơ sở là một feature map có kích thước 7x7x1024 sẽ được sử dụng làm đầu vào cho các lớp bổ sung (Extra layers) có tác dụng dự đoán nhãn và tọa độ hộp giới hạn (bounding box) của vật thể.

Trong YOLO version 3 tác giả áp dụng một mạng trích xuất đặc trưng là darknet-53. Mạng này gồm 53 lớp tích chập kết nối liên tiếp, mỗi layer được theo sau bởi một chuẩn hóa lô (batch normalization) và một đơn vị tuyến tính rò rỉ (activation Leaky Relu). Để giảm kích thước của output sau mỗi lớp tích chập, tác giả lấy mẫu gi bằng các filter với kích thước là 2. Mẹo này có tác dụng giảm thiểu số lượng tham số cho mô hình.

YOLOv3 dự báo trên nhiều bản đồ đặc trưng (feature map). Những feature map ban đầu có kích thước nhỏ giúp dự báo được các vật thể kích thước lớn. Những feature map sau có kích thước lớn hơn trong khi hộp mẫu (anchor box) được giữ cố định kích thước nên sẽ giúp dự báo các vật thể kích thước nhỏ.



Hình 1.2: Các bản đồ đặc trưng (*feature maps*) của mạng YOLO với hình dạng đầu vào (*input shape*) là 416x416, hình dạng đầu ra (*output shape*) có kích thước lần lượt là 13x13, 26x26 và 52x52.

Trên mỗi một ô của các feature map chúng ta sẽ áp dụng 3 khung neo (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 feature map x 3 anchor box).

Đồng thời trên một feature map hình vuông $S \times S$, mô hình YOLOv3 sinh ra một số lượng anchor box là: $S \times S \times 3$. Như vậy số lượng khung neo (anchor boxes) trên một bức ảnh sẽ là:

$$(13 \times 13 + 26 \times 26 + 52 \times 52) \times 3 = 10647 (\text{anchor boxes})$$

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 hộp giới hạn.

Một số lưu ý khi huấn luyện YOLO:

- Khi huấn luyện YOLO sẽ cần phải có RAM dung lượng lớn hơn để lưu được 10647 hộp giới hạn như trong kiến trúc này.
- Không thể thiết lập các `batch_size` quá lớn như trong các mô hình phân loại vì rất dễ hết bộ nhớ. Gói darknet của YOLO đã chia nhỏ một batch thành các phân nhóm cho vừa với RAM.

Thời gian xử lý của một bước trên YOLO lâu hơn rất rất nhiều lần so với các mô hình phân loại. Do đó nên thiết lập bước giới hạn huấn luyện cho YOLO nhỏ. Đối với các tác vụ nhận diện dưới 5 lớp, dưới 5000 bước là có thể thu được nghiệm tạm chấp nhận được. Các mô hình có nhiều lớp hơn có thể tăng số lượng bước theo cấp số nhân tùy bạn.

1.3.2. SSD (*Single Shot MultiBox Detector*)

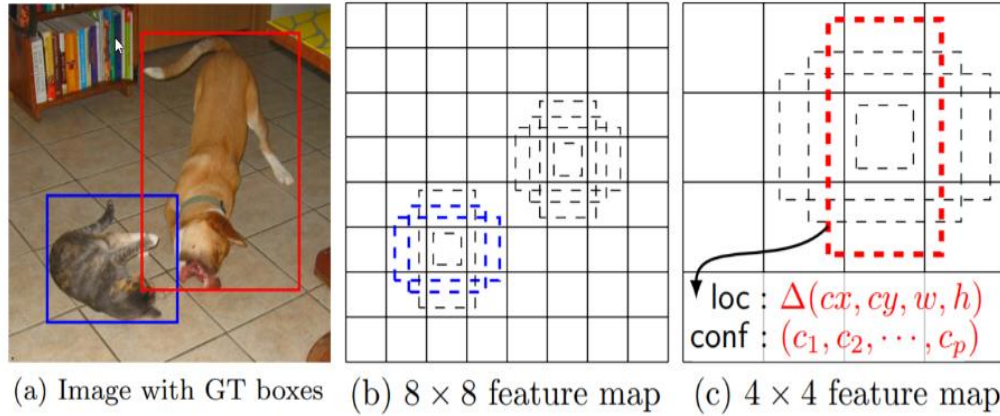
SSD cũng là một phương pháp phát hiện đối tượng nhanh chóng và chính xác bằng cách sử dụng một mạng nơ-ron duy nhất để dự đoán các hộp giới hạn và các lớp đối tượng cùng một lúc.

SSD có thể đạt được hiệu suất cao và làm việc trực tiếp trên toàn bức ảnh một cách hiệu quả.

Cách thức phân chia bản đồ đặc trưng (feature map) để nhận diện các hình ảnh với những kích thước khác nhau.

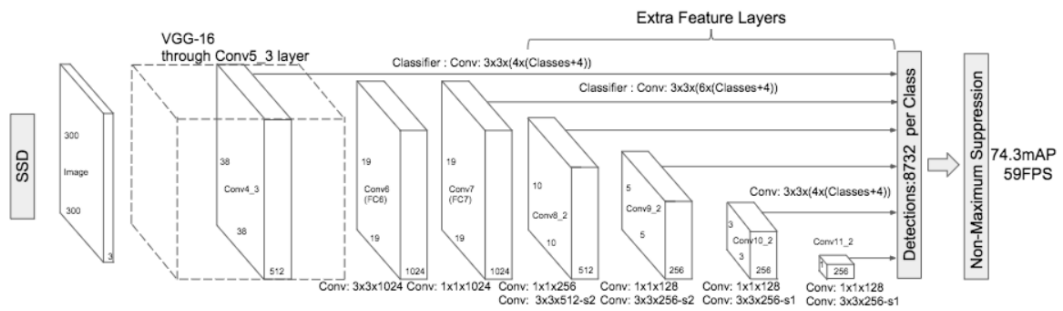
SSD chỉ cần duy nhất đầu vào là 1 bức ảnh và các hộp thực tế (ground truth boxes) ám chỉ vị trí hộp giới hạn (bounding box) các vật thể trong suốt quá trình huấn luyện. Trong quá trình phát hiện vật thể, trên mỗi một bản đồ đặc trưng (feature map), chúng ta đánh giá các một tập hợp nhỏ gồm những hộp mẫu tương ứng với các tỷ lệ cạnh khác nhau (aspect ratio) lên các bản đồ đặc trưng (features map) có kích thước (scales) khác nhau (chẳng hạn kích thước 8x8 và 4x4 trong hình (b) và (c)). Đối với mỗi hộp mẫu (các boxes nét đứt trong hình) ta cần dự báo một phân phối xác suất $c = (c_1, c_2, \dots, c_n)$ tương ứng với các class $C = C_1, C_2, \dots, C_n$. Tại thời điểm huấn luyện, đầu tiên chúng ta cần phù hợp với các hộp mặc định (match default boxes) với hộp thực tế (ground truth boxes) sao cho mức độ sai số được đo lường qua mất mát vị trí (localization loss) là nhỏ nhất. Sau đó ta sẽ tìm cách tối thiểu hóa sai số của nhãn dự báo tương ứng với mỗi vật thể được phát hiện trong hộp mẫu thông qua mất mát về độ tin cậy (confidence loss).

Như vậy hàm mất mát (loss function) của nhận dạng đối tượng (object detection) sẽ khác với loss function của các tác vụ phân loại ảnh (image classification) ở chỗ có thêm mất mát về vị trí (localization loss) về sai số vị trí của hộp dự đoán (predicted boxes) so với các hộp dự đoán (ground truth boxes).



Hình 1.3: Cách thức phân chia feature map để nhận diện các hình ảnh với những kích thước khác nhau.

Kiến trúc của mô hình SSD



Hình 1.4 Sơ đồ kiến trúc của mạng SSD

SSD dựa trên một tiến trình lan truyền thuận của một kiến trúc chuẩn (chẳng hạn VGG16) để tạo ra một khối feature map output gồm 3 chiều ở giai đoạn sớm. Chúng ta gọi kiến trúc mạng này là base network (tính từ input Image đến Conv7 trong hình 3.2.2). Sau đó chúng ta sẽ thêm những kiến trúc phía sau base network để tiến hành nhận diện vật thể như phần Extra Feature Layers trong sơ đồ.

1.4. Bài toán cảnh báo cháy từ hình ảnh

Phát hiện sớm và phản ứng nhanh chóng: Bằng cách sử dụng YOLOv7 để nhận diện đám cháy từ hình ảnh, hệ thống có thể phát hiện và cảnh báo về sự cố cháy một cách nhanh chóng, giúp giảm thiểu thời gian phản ứng và giảm thiệt hại.

Bảo vệ an toàn con người: Hệ thống cảnh báo cháy từ hình ảnh có thể giúp bảo vệ tính mạng và an toàn của con người bằng cách cung cấp cảnh báo kịp thời để sơ tán hoặc triển khai các biện pháp cứu hỏa.

Bảo vệ tài sản và môi trường: Phát hiện cháy sớm cũng giúp giảm thiểu thiệt hại về tài sản và môi trường, từ việc ngăn chặn sự lan rộng của đám cháy đến việc triển khai các biện pháp khắc phục sớm.

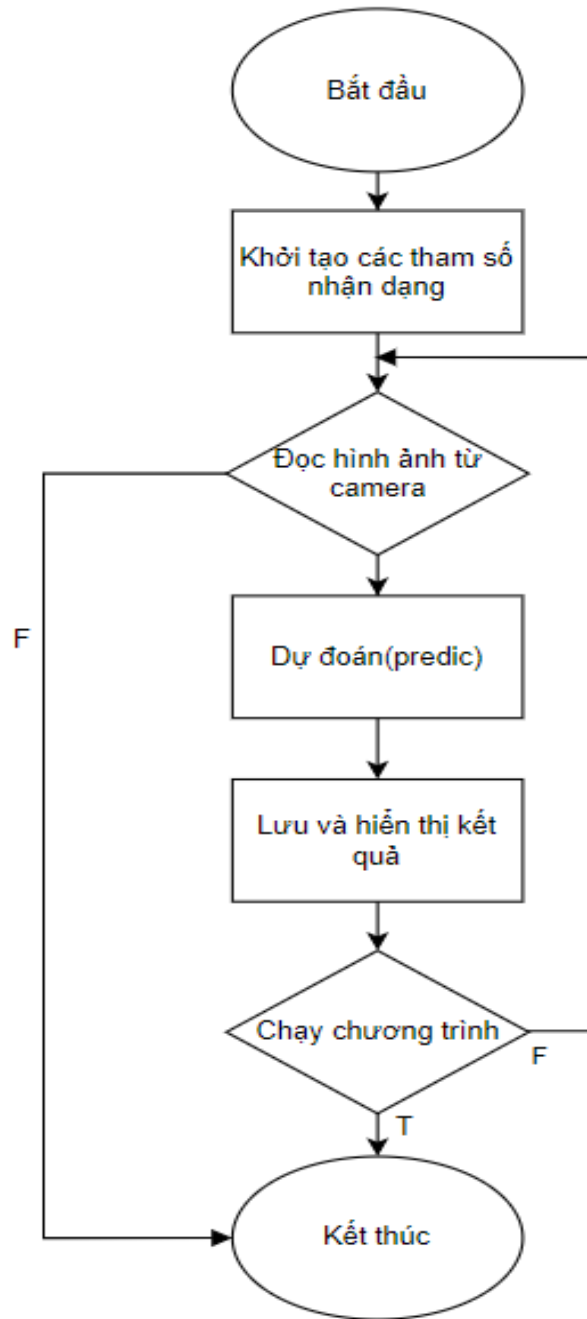
Tăng cường hiệu suất cứu hỏa: Bằng việc cung cấp thông tin chính xác về vị trí và quy mô của đám cháy, hệ thống có thể hỗ trợ quản lý cứu hỏa trong viện phân bổ nguồn lực và điều phối các hoạt động cứu hỏa hiệu quả hơn.

Giảm thiểu rủi ro và nâng cao chất lượng sống: Sự hiện diện của hệ thống cảnh báo cháy từ hình ảnh giúp giảm thiểu rủi ro cháy và tạo ra một môi trường sống và làm việc an toàn hơn cho cộng đồng.

CHƯƠNG 2. MÔ HÌNH CẢNH BÁO CHÁY TỪ HÌNH ẢNH

2.1. Sơ đồ hoạt động của mô hình cảnh báo cháy

Sơ đồ hoạt động của mô hình cảnh báo cháy



Hình 2.1.Sơ đồ hoạt động của mô hình cảnh báo cháy

Mô hình cho phép thực hiện nhận dạng với tùy chọn dữ liệu đầu vào bao gồm file ảnh, file video hoặc hình ảnh trực tiếp từ camera. Với dữ liệu đầu vào là hình ảnh cần cung cấp đường dẫn tuyệt đối của file ảnh, kết quả nhận dạng là hình ảnh được lưu lại trên đó chỉ ra vị trí ngọn lửa và độ tin cậy của dự đoán.

2.2. Phương pháp phát hiện đối tượng bằng YOLOv7

Phát hiện đối tượng là một bài toán thị giác máy tính liên quan đến việc xác định và định vị các đối tượng trong hình ảnh hoặc video. Nó là một phần quan trọng của nhiều ứng dụng, chẳng hạn như camera giám sát thông minh, ô tô tự lái hoặc người máy. Các thuật toán phát hiện đối tượng có thể được chia thành hai loại chính: phát hiện một giai đoạn (Single-shot object detection) và phát hiện hai giai đoạn (Two-shot object detection).[1]

2.2.1. Phát hiện đối tượng

Single-shot object detection

Thuật toán phát hiện đối tượng một lần duy nhất (Single-shot object detection) sử dụng một lần truyền hình ảnh đầu vào để đưa ra dự đoán về sự hiện diện và vị trí của các đối tượng trong ảnh. Nó xử lý toàn bộ hình ảnh trong một lần chạy, khiến chúng trở nên hiệu quả về mặt tính toán.

Tuy nhiên, single-shot object detection thường kém chính xác hơn so với các phương pháp khác và kém hiệu quả hơn trong việc phát hiện các đối tượng nhỏ. Các thuật toán như vậy có thể được sử dụng để phát hiện các đối tượng theo thời gian thực trong các môi trường hạn chế về tài nguyên.

YOLO là thuật toán phát hiện đối tượng một lần duy nhất (single-shot object detection) sử dụng mạng thần kinh tích chập (CNN) để xử lý hình ảnh.

Two-shot object detection

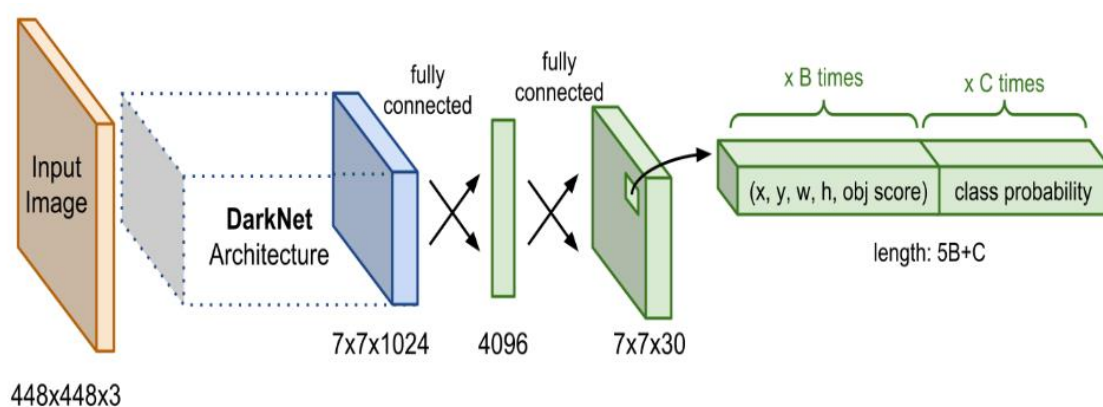
Phát hiện đối tượng hai lần (two-shot object detection) sử dụng hai lần truyền hình ảnh đầu vào để đưa ra dự đoán về sự hiện diện và vị trí của đối tượng. Lướt đầu tiên được sử dụng để tạo một tập hợp các đề xuất hoặc vị trí tiềm năng của đối tượng. Lướt thứ hai được sử dụng để tinh chỉnh các đề xuất này và đưa ra dự đoán cuối cùng. Cách tiếp cận này chính xác hơn thuật toán phát hiện đối tượng một lần duy nhất (Single-shot object detection) nhưng cũng tốn kém hơn về mặt tính toán.

Nhìn chung, sự lựa chọn giữa phát hiện đối tượng một lần và hai lần tùy thuộc vào các yêu cầu và ràng buộc cụ thể của ứng dụng.

Nói chung, Single-shot object detection phù hợp hơn cho các ứng dụng thời gian thực, trong khi two-shot object detection tốt hơn cho các ứng dụng đề cao độ chính xác.

Cách thức hoạt động của YOLOv7. Kiến trúc YOLOv7

Mô hình YOLOv7 được đào tạo trước với ImageNet, một tập dữ liệu chứa nhiều hình ảnh khác nhau. Sau đó, nó được chuyển đổi để thực hiện phát hiện đối tượng. Lớp kết nối đầy đủ cuối cùng của YOLOv7 thực hiện dự đoán về xác suất của lớp và tọa độ của hộp giới hạn.



Hình 2.2: Kiến trúc YOLOv7 và cơ chế hoạt động

Thành phần kiến trúc Darknet (Darknet Architecture) được gọi là base network có tác dụng trích xuất đặc trưng. Đầu ra của mạng cơ sở (base network) là một bản đồ đặc trưng (feature map) có kích thước **7x7x1024** sẽ được sử dụng làm đầu vào cho các lớp bổ sung (Extra layers) có tác dụng dự đoán nhãn và tọa độ hộp giới hạn (bounding box) của vật thể.

YOLOv7 chia hình ảnh đầu vào thành lưới $S \times S$. Nếu tâm của một đối tượng rơi vào một ô lưới thì ô lưới đó có nhiệm vụ phát hiện đối tượng đó. Mỗi ô lưới dự đoán các hộp giới hạn B và điểm tin cậy cho các hộp đó. Các điểm tin cậy này phản ánh mức độ tin cậy của mô hình rằng hộp chứa một đối tượng và mức độ chính xác mà mô hình cho rằng hộp được dự đoán.

YOLOv7 dự đoán nhiều hộp giới hạn trên mỗi ô lưới. Tại thời điểm đào tạo, ta chỉ muốn một bộ dự đoán hộp giới hạn thể hiện cho từng đối tượng. YOLOv7 chỉ định bộ dự đoán dựa trên chỉ số IOU hiện tại cao nhất với thực tế. Điều này dẫn đến sự chuyên môn hóa giữa các bộ dự đoán hộp giới hạn. Mỗi công cụ dự đoán trở nên tốt hơn trong việc dự báo các kích thước, tỷ lệ khung hình hoặc loại đối tượng nhất định.

Một kỹ thuật quan trọng được sử dụng trong các mô hình YOLOv7 là NMS (non-maximum suppression). NMS là một bước hậu xử lý được sử dụng để cải thiện độ chính xác và hiệu quả của việc phát hiện đối tượng. Trong phát hiện đối tượng, thông thường có nhiều hộp giới hạn được tạo cho một đối tượng trong một hình ảnh. Các hộp giới hạn này có thể chồng lên nhau hoặc nằm ở các vị trí khác nhau, nhưng tất cả chúng đều đại diện cho cùng một đối tượng. NMS được sử dụng để xác định và loại bỏ các hộp giới hạn dư thừa hoặc không chính xác và đề xuất một hộp giới hạn duy nhất cho từng đối tượng trong ảnh.

2.3. Giới thiệu về cảnh báo cháy bằng YOLOv7

Cảnh báo cháy bằng YOLOv7 là một ứng dụng của công nghệ phát hiện đối tượng trong lĩnh vực an ninh và an toàn. Bằng cách sử dụng mô hình YOLOv7, người ta có thể xây dựng các hệ thống cảnh báo cháy hiệu quả để phát hiện sớm và phản ứng nhanh chóng đối với các nguy cơ cháy nổ. Dưới đây là những lợi ích khi sử dụng cảnh báo cháy bằng YOLOv7:

1. Phát hiện cháy và khói: YOLOv7 có khả năng phát hiện các đối tượng cháy và khói trong hình ảnh hoặc video. Bằng cách này, nó có thể giúp hệ thống cảnh báo cháy xác định vị trí và phạm vi của nguy cơ cháy nổ.

2. Xử lý thời gian thực: YOLOv7 được thiết kế để hoạt động ở tốc độ cao, cho phép xử lý thời gian thực của dữ liệu video đầu vào. Điều này là rất quan trọng đối với việc cảnh báo cháy, vì nó cần phản ứng nhanh chóng để đảm bảo an toàn cho con người và tài sản.

3. Tích hợp hệ thống cảnh báo: Khi YOLOv7 phát hiện cháy hoặc khói, thông tin này có thể được tích hợp vào các hệ thống cảnh báo cháy tự động. Các hệ thống này có thể kích hoạt các cảnh báo, báo động và hệ thống chữa cháy để ứng phó với tình huống nguy hiểm.

4. Dự án môi trường và công nghiệp: Cảnh báo cháy bằng YOLOv7 có thể được triển khai trong nhiều lĩnh vực như các nhà máy sản xuất, tòa nhà cao tầng, nhà ga, và các khu vực môi trường nhạy cảm với nguy cơ cháy nổ.

Tích hợp với hệ thống giám sát: YOLOv7 có thể tích hợp với các hệ thống giám sát video để cung cấp một cách tiếp cận toàn diện đối với an ninh và an toàn trong các môi trường phức tạp.

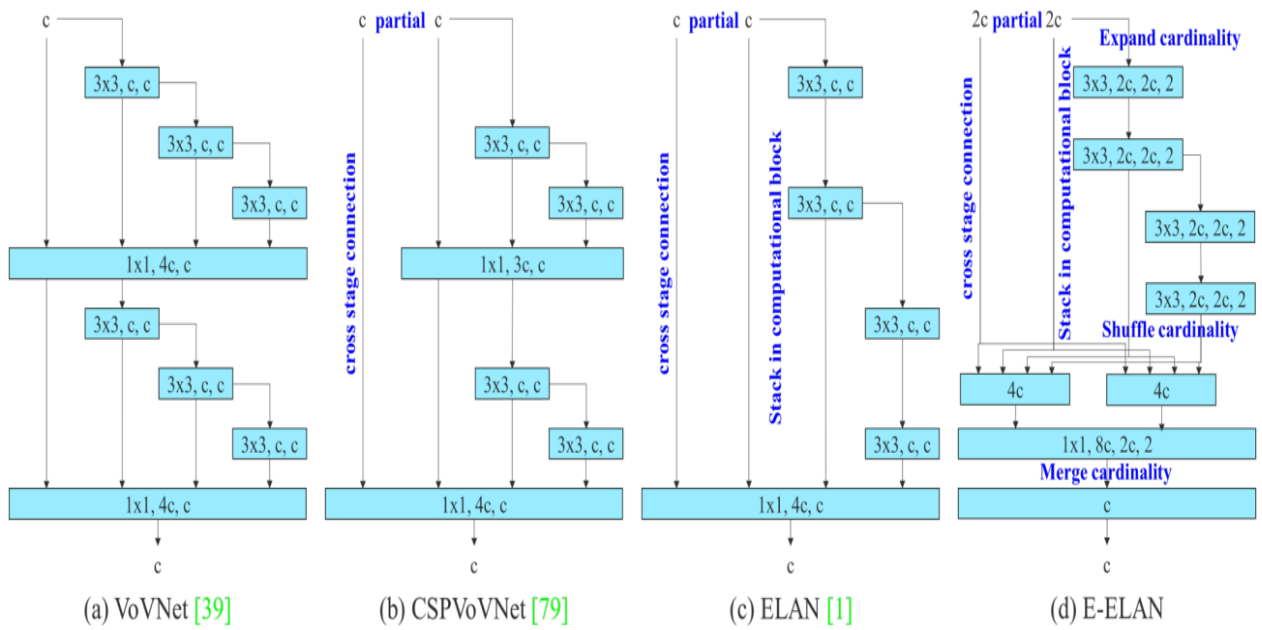
2.4. Lựa chọn phiên bản YOLOv7 trong phát hiện cháy và cảnh báo.

Thay đổi quy mô mô hình là quá trình tăng hoặc giảm tỷ lệ một mô hình hiện có để mô hình này phù hợp với các thiết bị điện toán khác nhau. Tỷ lệ mô hình thường sử dụng nhiều yếu tố khác nhau như số lớp (chiều sâu), kích thước của hình ảnh đầu vào (độ phân giải), số lượng kim tự tháp đặc trưng (giai đoạn) và số lượng kênh (chiều rộng). Các yếu tố này đóng một vai trò quan trọng trong việc đảm bảo cân bằng giữa các tham số mạng, tốc độ giao thoa, tính toán và độ chính xác của mô hình.

Một trong những phương pháp chia tỷ lệ được sử dụng phổ biến nhất là NAS hoặc “Kiến trúc mạng Tìm kiếm tự động” để tìm kiếm các hệ số tỷ lệ phù hợp từ các công cụ tìm kiếm mà không cần bất kỳ quy tắc phức tạp nào. Nhược điểm chính của việc sử dụng NAS là nó là một cách tiếp cận tốn kém để tìm kiếm các hệ số tỷ lệ phù hợp.

Hầu hết mọi mô hình tái tham số hóa mô hình đều phân tích các yếu tố tỷ lệ riêng lẻ và duy nhất một cách độc lập và hơn nữa, thậm chí còn tối ưu hóa các yếu tố này một cách độc lập. Đó là do kiến trúc NAS hoạt động với các hệ số tỷ lệ không tương quan.

Điều đáng chú ý là các mô hình dựa trên nôi như VoVNet or Mạng lưới dày đặc thay đổi chiều rộng đầu vào của một vài lớp khi độ sâu của mô hình được thu nhỏ. YOLOv7 hoạt động trên kiến trúc dựa trên phép nôi được đề xuất và do đó sử dụng phương pháp chia tỷ lệ hỗn hợp.



Hình 2.3: Sơ đồ tổng hợp lớp của YOLOv7

Thay đổi quy mô mô hình là quá trình tăng hoặc giảm tỷ lệ một mô hình hiện có để mô hình này phù hợp với các thiết bị điện toán khác nhau. Tỷ lệ mô hình thường sử dụng nhiều yếu tố khác nhau như số lớp (chiều sâu), kích thước của hình ảnh đầu vào (độ phân giải), số lượng kim tự tháp đặc trưng (giai đoạn) và số lượng kênh (chiều rộng). Các yếu tố này đóng một vai trò quan trọng trong việc đảm bảo cân bằng giữa các tham số mạng, tốc độ giao thoa, tính toán và độ chính xác của mô hình.

Một trong những phương pháp chia tỷ lệ được sử dụng phổ biến nhất là NAS hoặc kiến trúc mạng Tìm kiếm tự động tìm kiếm các hệ số tỷ lệ phù hợp từ các công cụ tìm kiếm mà không cần bất kỳ quy tắc phức tạp nào. Nhược điểm chính của việc sử dụng NAS là nó là một cách tiếp cận tốn kém để tìm kiếm các hệ số tỷ lệ phù hợp.

Hầu hết mọi mô hình tái tham số hóa mô hình đều phân tích các yếu tố tỷ lệ riêng lẻ và duy nhất một cách độc lập và hơn nữa, thậm chí còn tối ưu hóa các yếu tố này một cách độc lập. Đó là do kiến trúc NAS hoạt động với các hệ số tỷ lệ không tương quan.

Điều đáng chú ý là các mô hình dựa trên nổi như VoVNet or Mạng lưới dày đặc thay đổi chiều rộng đầu vào của một vài lớp khi độ sâu của mô hình được thu nhỏ.

YOLOv7 hoạt động trên kiến trúc dựa trên phép nối được đề xuất và do đó sử dụng phương pháp chia tỷ lệ hỗn hợp.

Với đề tài cảnh báo cháy từ hình ảnh, em lựa chọn phiên bản YOLOv7 để thực hiện vì hệ thống có được những sự cải tiến từ những phiên bản cũ và kèm theo đó là những tính năng mới chắc chắn nó sẽ là một sự lựa chọn hợp lý cho đề tài của mình.

CHƯƠNG 3. CÀI ĐẶT, THỬ NGHIỆM VÀ ĐÁNH GIÁ

3.1. Môi trường cài đặt

3.1.1. Giới thiệu môi trường Google Colab

Google Colab là một môi trường phát triển và chia sẻ mã nguồn mở trực tuyến, được cung cấp miễn phí bởi Google. Dựa trên nền tảng của Jupyter Notebook, Google Colab cung cấp một cách tiếp cận linh hoạt và tiện lợi cho việc viết và chia sẻ mã nguồn, đặc biệt là trong lĩnh vực học máy và khoa học dữ liệu. Dưới đây là một số điểm nổi bật của môi trường Google Colab:

1. **Miễn Phí và Trực Tuyến:** Google Colab hoàn toàn miễn phí và hoạt động trực tuyến thông qua trình duyệt web, không cần cài đặt bất kỳ phần mềm nào. Điều này cho phép người dùng truy cập và làm việc trên môi trường Colab từ bất kỳ thiết bị nào có kết nối internet.
2. **Tích Hợp với Google Drive:** Colab tích hợp tốt với Google Drive, cho phép người dùng lưu trữ và truy cập các tập tin dữ liệu và notebook trực tiếp từ tài khoản Google Drive của họ. Điều này giúp dễ dàng quản lý và chia sẻ dữ liệu và notebook giữa nhiều người dùng.
3. **Hỗ Trợ GPU và TPU:** Colab cung cấp khả năng sử dụng miễn phí các tài nguyên GPU (Graphics Processing Unit) và TPU (Tensor Processing Unit) của Google. Điều này giúp tăng tốc độ huấn luyện cho các mô hình học sâu và tính toán dữ liệu lớn.
4. **Thư Viện và Cài Đặt Linh Hoạt:** Colab cung cấp một môi trường làm việc linh hoạt cho việc cài đặt và sử dụng các thư viện và công cụ phổ biến trong lĩnh vực học máy và khoa học dữ liệu như TensorFlow, PyTorch, scikit-learn, và nhiều thư viện khác.
5. **Chia Sẻ Dễ Dàng:** Colab cho phép người dùng chia sẻ notebook của mình một cách dễ dàng thông qua liên kết URL. Điều này giúp các nhà nghiên cứu và nhà phát triển có thể dễ dàng chia sẻ mã nguồn, kết quả và ý tưởng với người khác.

Tóm lại, Google Colab là một công cụ mạnh mẽ và linh hoạt cho các nhà nghiên cứu, nhà phát triển và sinh viên trong việc phát triển và chia sẻ mã nguồn trong lĩnh vực học máy và khoa học dữ liệu.

3.1.2. Thực thi trên môi trường Google Colab

Google Colab là một dịch vụ thực hiện việc cung cấp các máy ảo để thực hiện các tác vụ trên internet. Khi sử dụng Google Colab để huấn luyện mô hình, ta cần tải lên tập dữ liệu đã được chuẩn bị trong phần trước với máy ảo trên Google Colab.

Trên Google Drive ta tạo một thư mục có tên là YOLOv7_train. Tại máy tính của chúng ta cần phải nén thư mục data mà ta đã chuẩn bị và phân chia theo cấu trúc thành file firedata.zip và đưa nó lên thư mục YOLOv7_train ở trên Google Drive.[3]

3.1.3. Phân chia tập dữ liệu huấn luyện

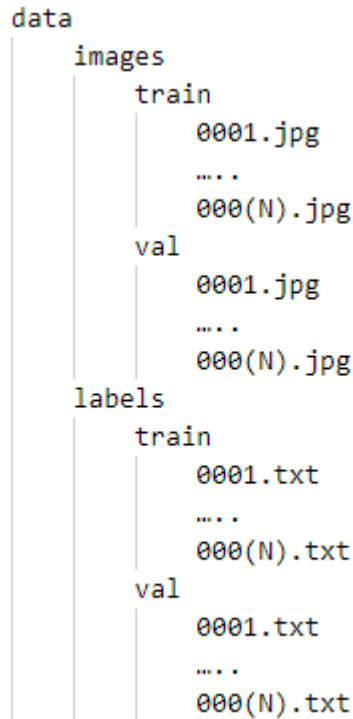
Để huấn luyện một mô hình YOLOv7, chúng ta cần có một tập dữ liệu đủ lớn và đa dạng về đối tượng và góc nhìn. Tập dữ liệu này cần có các ảnh chứa đối tượng đã được gắn nhãn cần nhận diện cùng với thông tin về bounding box của các đối tượng đó.

Chúng ta thu thập được dữ liệu cần huấn luyện bao gồm các ảnh về các đám cháy với các đối tượng cần nhận dạng là lửa. Và các nhãn “fire” đã được gắn nhãn. Tuy nhiên, trước khi có thể thực hiện việc huấn luyện được mô hình YOLOv7 ta cần phải phân chia tập dữ liệu trên thành 2 phần chính.

- Phần huấn luyện (train) sẽ bao gồm 80% dữ liệu trong bộ dữ liệu ban đầu đã thu thập và gắn nhãn.
- Phần xác thực (valid) sẽ bao gồm 20% dữ liệu trong bộ dữ liệu ban đầu đã thu thập và gắn nhãn.

Lưu ý rằng, các file chứa nhãn và tọa độ của đối tượng (định dạng file .txt) cũng phải được phân chia ra theo như tỉ lệ 80% cho huấn luyện (train) và 20% cho xác thực (val).

Khi huấn luyện dữ liệu với mô hình YOLOv7, ta cần phải thực hiện tạo một cấu trúc thư mục chứa cả ảnh và nhãn theo quy tắc của mô hình này đã đưa ra.



Hình 3.1: Cấu trúc thư mục sau khi phân chia dữ liệu huấn luyện

Các file chứa nhãn và tọa độ của đối tượng đã gắn nhãn (các file định dạng .txt) này cũng phải có tên giống với tên ảnh mà ta thu thập. Ví dụ: ảnh có tên 0001.jpg thì file sẽ có tên 0001.txt – việc đặt tên giống nhau như này giúp việc huấn luyện mô hình biết được các nhãn và tọa độ tương ứng cho từng ảnh.

3.2. Môi trường cài đặt

3.2.1. Ngôn ngữ Python

Là một ngôn ngữ lập trình phổ biến được thiết kế bởi Guido van Rossum và được phát hành lần đầu vào năm 1991. Python là ngôn ngữ lập trình thông dịch, có nghĩa là mã nguồn Python được thực thi theo từng dòng từ trên xuống dưới. Python hỗ trợ cả lập trình hướng đối tượng và hướng thủ tục nội tại, giúp lập trình viên có thể sử dụng mô hình phù hợp với yêu cầu của dự án.

Python nổi tiếng với cú pháp rõ ràng, dễ đọc, giúp giảm thiểu đáng kể thời gian cần thiết để viết và kiểm tra mã nguồn. Python được ứng dụng rộng rãi trong nhiều lĩnh vực như phân tích dữ liệu, học máy, phát triển web, tự động hóa, và nhiều hơn nữa.

Python cũng có một cộng đồng lập trình viên lớn mạnh, nơi mà bạn có thể tìm thấy hàng ngàn thư viện và framework hỗ trợ. Một số thư viện nổi tiếng bao gồm NumPy và Pandas dành cho tính toán khoa học và phân tích dữ liệu, TensorFlow và PyTorch dành cho học máy, và Django và Flask dành cho phát triển web.

Python là một công cụ mạnh mẽ và linh hoạt, phù hợp cho cả người mới học lập trình và những lập trình viên kỹ năng cao và được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau. [2]

3.2.2. Thư viện

Trong bài toán phân loại, cần phải sử dụng rất nhiều thư viện và framework để hỗ trợ cho việc xây dựng mô hình và huấn luyện. Dưới đây là một số thư viện và framework hỗ trợ được sử dụng:

1. Django là 1 web framework khá nổi tiếng được viết hoàn toàn bằng ngôn ngữ Python. Nó là 1 framework với đầy đủ các thư viện, module hỗ trợ các web-developer. Django sử dụng mô hình MVC và được phát triển bởi Django Software Foundation (một tổ chức phi lợi nhuận độc lập) Django tập trung vào tính năng “có thể tái sử dụng” và “có thể tự chạy” của các phần tử phần mềm độc lập (component), tính năng phát triển nhanh, không làm lại những gì đã làm. Một số website phổ biến được xây dựng từ Django là Pinterest, Instagram, Mozilla, và Bitbucket.

2. Pytorch là framework được phát triển bởi Facebook. Đây là một ông lớn về công nghệ đầu tư rất nhiều nguồn lực cho việc phát triển Trí tuệ nhân tạo. Pytorch được phát triển với giấy phép mã nguồn mở do đó nó tạo được cho mình một cộng đồng rất lớn. Một cộng đồng lớn đồng nghĩa với nhiều tài nguyên để học và các vấn đề của bạn có thể đã có ai đó giải quyết và chia sẻ với cộng đồng.

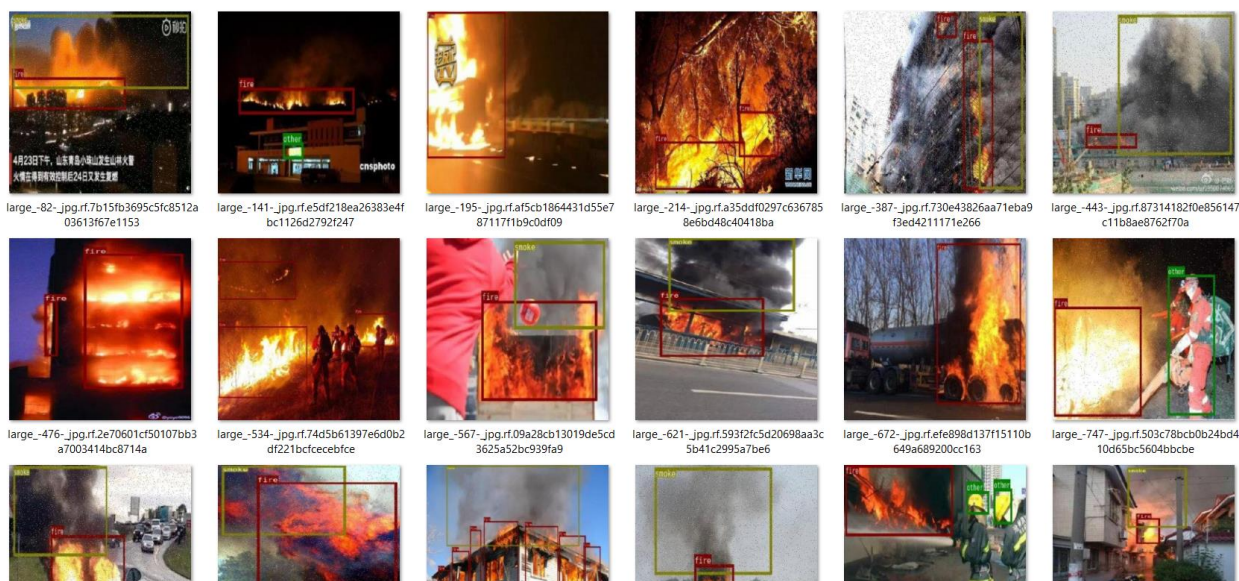
3. Numpy (Numeric Python): là một thư viện toán học rất phổ biến và mạnh mẽ của Python. NumPy được trang bị các hàm số đã được tối ưu, cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần khi chỉ sử dụng Python đơn thuần.

4. OpenCV là một thư viện mã nguồn mở hàng đầu cho thị giác máy tính (computer vision), xử lý ảnh và máy học, và các tính năng tăng tốc GPU trong hoạt động thời gian thực.

3.3. Tạo bộ dữ liệu học để huấn luyện

3.3.1. Giới thiệu bộ dữ liệu phát hiện cháy (fire-detection)

Data fire-detection là một bộ dữ liệu ảnh lớn với 916 hình ảnh về các vụ cháy trên khắp thế giới. Từ các hỏa hoạn nhỏ tại các gia đình đến các vụ cháy lớn ảnh hưởng đến các công ty và khu dân cư, bộ data bao gồm có 679 train set, 154 valid set, 86 test set[4]



Hình 3.2: Bộ dữ liệu Data-detection

Vì thời gian chuẩn bị ngắn và do còn thiếu kiến thức dẫn đến còn nhiều sai sót trong phần chuẩn bị, em xin phép sử dụng bộ firedata gồm một phần thu thập trên mạng và một phần tự tạo.

3.3.2. Gắn nhãn cho bộ dữ liệu FireData

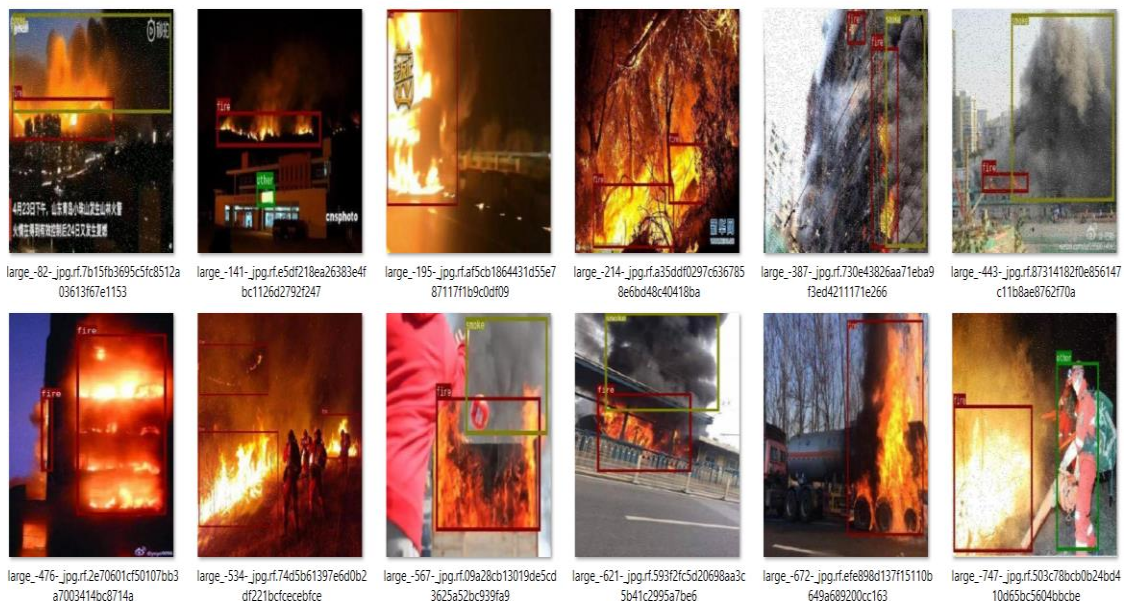
Sử dụng các công cụ gắn nhãn hình ảnh <https://www.makesense.ai/> để xác định và đánh dấu vị trí của đối tượng lửa trên mỗi hình ảnh. Mỗi hình ảnh được gắn nhãn với các bounding box (hộp giới hạn) bao quanh đối tượng lửa và được gắn nhãn "fire".



Hình 3.3: Gắn nhãn cho bộ dữ liệu Fire-detection

Trong bộ dữ liệu Fire-detection cung cấp cho chúng ta 916 bức ảnh về các vụ hỏa hoạn hoặc các đám cháy. Vì thế ta cần thực hiện tìm trong 916 bức ảnh đó, đám cháy xuất hiện ở ảnh nào thì ta thực hiện gắn nhãn vào khu vực có lửa (được đánh dấu bằng hình chữ nhật)

Việc gắn nhãn dữ liệu, nhằm mục đích chỉ ra rằng tọa độ của đối tượng “fire” mà ta cần đánh dấu. Tọa độ này sẽ giúp mô hình YOLOv7 tập chung học chính xác đối tượng và có thể phát hiện vị trí đối tượng cháy trong 1 ảnh.



Hình 3.4: Phần ảnh sau khi được phát hiện đối tượng lửa được gắn nhãn

Sau khi gắn nhãn, ta thu được các tệp tin định dạng *.txt, các tệp tin này có tên trùng với tên ảnh của bộ dữ liệu Fire-detection ban đầu. Trong mỗi tệp tin sẽ chứa các tọa độ tương ứng của đối tượng “fire” mà ta đã gắn nhãn.



Hình 3.5: Chi tiết một tệp tin chứa tọa độ đối tượng lửa được gắn nhãn

Bên trong tệp tin chứa tọa độ được gắn nhãn, sẽ gồm nhiều dòng và mỗi dòng đại diện 1 đối tượng “fire” bên trong ảnh. Trong mỗi dòng sẽ có vị trí (x1,y1) và (x2,y2) để tạo ra 1 khung hình chữ nhật bao quanh đối tượng “fire” đó.

3.3. Đào tạo mô hình, huấn luyện cảnh báo cháy

3.3.1. Huấn luyện mô hình

Để đào tạo mô hình và huấn luyện cảnh báo cháy dựa trên bộ datafire ta sử dụng YOLOv7 trên Google Colab. Đây là dịch vụ máy tính đám mây của Google, cho phép người dùng tạo ra các tệp notebook Jupyter để thực thi mã Python. Nó cung cấp một môi trường tính toán đám mây miễn phí, với các tính năng như GPU miễn phí, RAM miễn phí đến 12GB có thể mở rộng đến 25.5GB nếu có trả phí, lưu trữ đám mây và nhiều tính năng khác. Ngoài ra, Google Colab còn cung cấp cho người dùng miễn phí 100GB dung lượng lưu trữ trên Google Drive. Google Colab được thiết kế để hỗ trợ việc phát triển và huấn luyện các mô hình học máy và các ứng dụng AI. Nó cung

cấp một môi trường tính toán đám mây miễn phí, mạnh mẽ và linh hoạt cho các nhu cầu học tập và nghiên cứu.

Sử dụng dòng code dưới đây để train model

```
%cd /content/drive/MyDrive/fire_detect/yolov7
!python train.py --batch 16 --epochs 30 --data /content/drive/MyDrive/fire_detect/yolov7/Fire-Detection-1/data.yaml --weights 'pretrain/yolov7.pt'
```

Chương trình phát hiện đám cháy bằng YOLOv7 được đào tạo 5 epoch batch bằng 16 kích thước ảnh 640x640. Thời gian đào tạo 5 epoch kéo dài khoảng 7 phút

```
Epoch      gpu_mem    box      obj      cls      total    labels  img_size
0/4         1.46G     0.08007  0.0145   0.017   0.1116   28      640: 100% 43/43 [01:28<00:00, 2.06s/it]
Class      Images    Labels   P        R        mAP@.5  mAP@.5:.95: 0% 0/5 [00:00<?, ?it/s]/usr/local/lib/py
return_VF.meshgrid(tensors, **kwargs) # type: ignore[attr-defined]
Class      Images    Labels   P        R        mAP@.5  mAP@.5:.95: 100% 5/5 [00:14<00:00, 2.89s/it]
all        154      256     0.00912  0.0742   0.00234  0.000395

Epoch      gpu_mem    box      obj      cls      total    labels  img_size
1/4         12.6G    0.07045  0.01133  0.01348  0.09526   26      640: 100% 43/43 [01:02<00:00, 1.45s/it]
Class      Images    Labels   P        R        mAP@.5  mAP@.5:.95: 100% 5/5 [00:05<00:00, 1.13s/it]
all        154      256     0.0844   0.0977   0.0241   0.0057

Epoch      gpu_mem    box      obj      cls      total    labels  img_size
2/4         10.9G    0.06651  0.01088  0.01172  0.08912   22      640: 100% 43/43 [01:01<00:00, 1.44s/it]
Class      Images    Labels   P        R        mAP@.5  mAP@.5:.95: 100% 5/5 [00:06<00:00, 1.24s/it]
all        154      256     0.182    0.137    0.0544   0.0107

Epoch      gpu_mem    box      obj      cls      total    labels  img_size
3/4         10.9G    0.06255  0.01076  0.01011  0.08342   11      640: 100% 43/43 [01:06<00:00, 1.54s/it]
Class      Images    Labels   P        R        mAP@.5  mAP@.5:.95: 100% 5/5 [00:04<00:00, 1.03it/s]
all        154      256     0.108    0.141    0.0396   0.00745

Epoch      gpu_mem    box      obj      cls      total    labels  img_size
4/4         10.9G    0.05985  0.01122  0.01047  0.08154   16      640: 100% 43/43 [00:58<00:00, 1.37s/it]
Class      Images    Labels   P        R        mAP@.5  mAP@.5:.95: 100% 5/5 [00:10<00:00, 2.17s/it]
all        154      256     0.106    0.227    0.0723   0.013
None       154      256     0.106    0.227    0.0723   0.013

5 epochs completed in 0.120 hours.

Optimizer stripped from runs/train/exp6/weights/last.pt, 74.8MB
Optimizer stripped from runs/train/exp6/weights/best.pt, 74.8MB
```

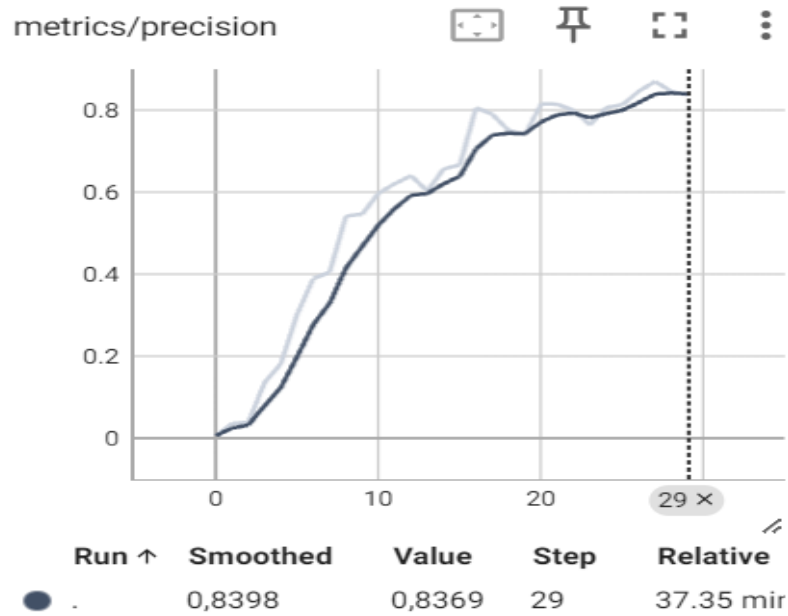
Sau đây là bảng kết quả thử nghiệm mô hình cảnh báo cháy bằng hình ảnh từ YOLOv7, batch= 16 kích thước ảnh 640x640 với các số epoch khác nhau.

STT	Số ảnh train	Số ảnh test/Valid	Epoch	Thời gian	P	R	mAP	mAP 5-95
1	679	154	5	0.120	0.109	0.172	0.0557	0.013
2	679	154	10	0.222	0.571	0.621	0.554	0.224
3	679	154	15	0.335	0.758	0.672	0.71	0.313
4	679	154	30	0.664	0.821	0.797	0.473	0.473

Bảng 3.5: Bảng kết quả thử nghiệm

3.3.2. Giá trị của độ chính xác của từng lớp (Precision)

Giá trị độ chính xác của từng lớp sau khi huấn luyện mô hình với epoch bằng 30

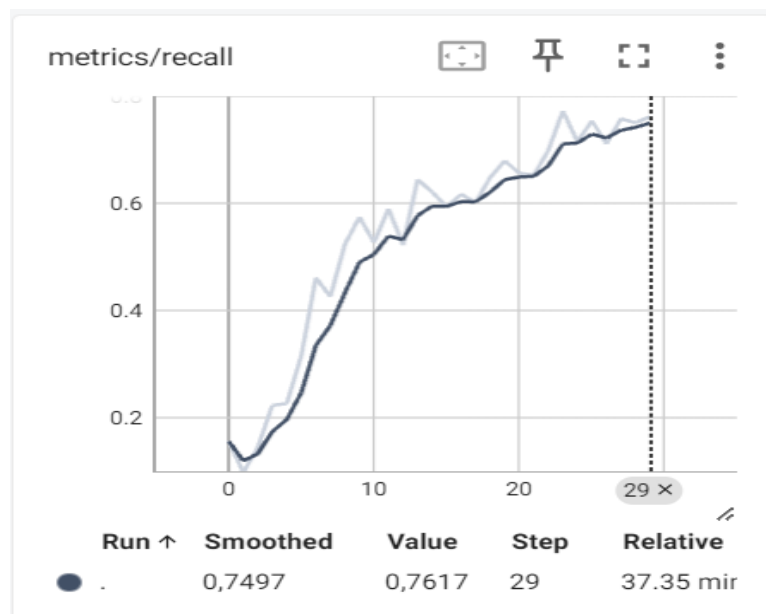


Hình 3.6: Biểu đồ giá trị độ chính xác của từng lớp qua mỗi giai đoạn

Qua hình 3.6 ta thấy giá trị độ chính xác từng lớp tăng dần qua mỗi lần huấn luyện.

3.3.3. Giá trị của độ nhớ (recall)

Giá trị độ nhớ(recall) của từng lớp sau khi huấn luyện mô hình với epoch bằng 30

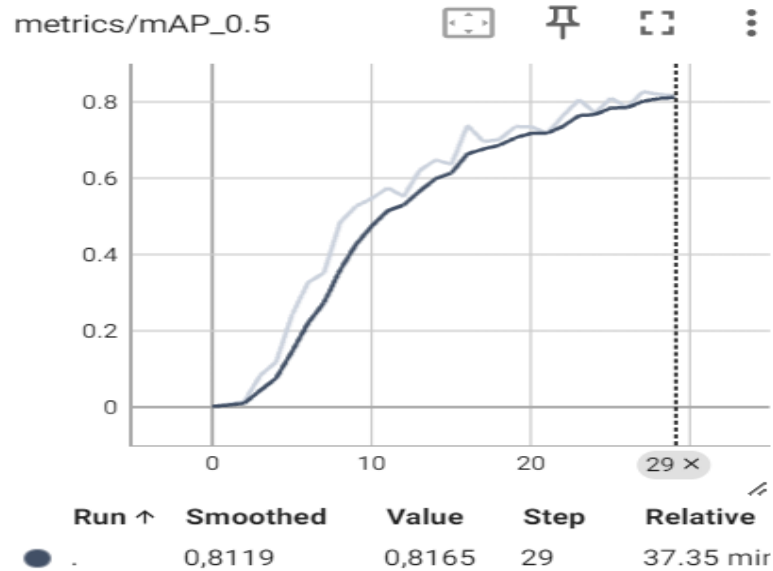


Hình 3.7: Biểu đồ giá trị độ chính nhớ qua mỗi giai đoạn

Qua hình 3.7 ta thấy giá trị độ chính xác từng lớp tăng dần qua mỗi lần huấn luyện

3.3.4. Biểu đồ mAP_0.5

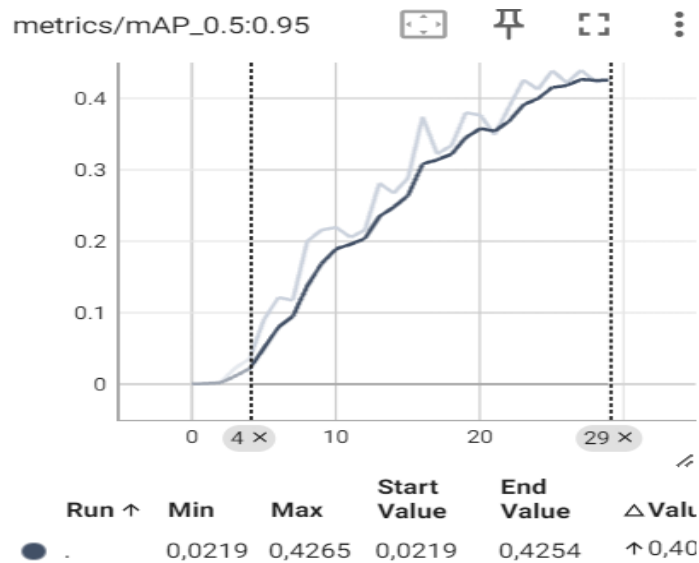
Giá trị mAP_0.5 của từng lớp sau khi huấn luyện mô hình với epoch bằng 30



Hình 3.8: Biểu đồ giá trị mAP_0.5 qua từng giai đoạn

Qua biểu đồ ta thấy giá trị mAP_0.5 qua từng giai đoạn tăng dần

3.3.5. Biểu đồ trung bình của các giá trị AP trên nhiều ngưỡng IoU từ 0.5 đến 0.95.

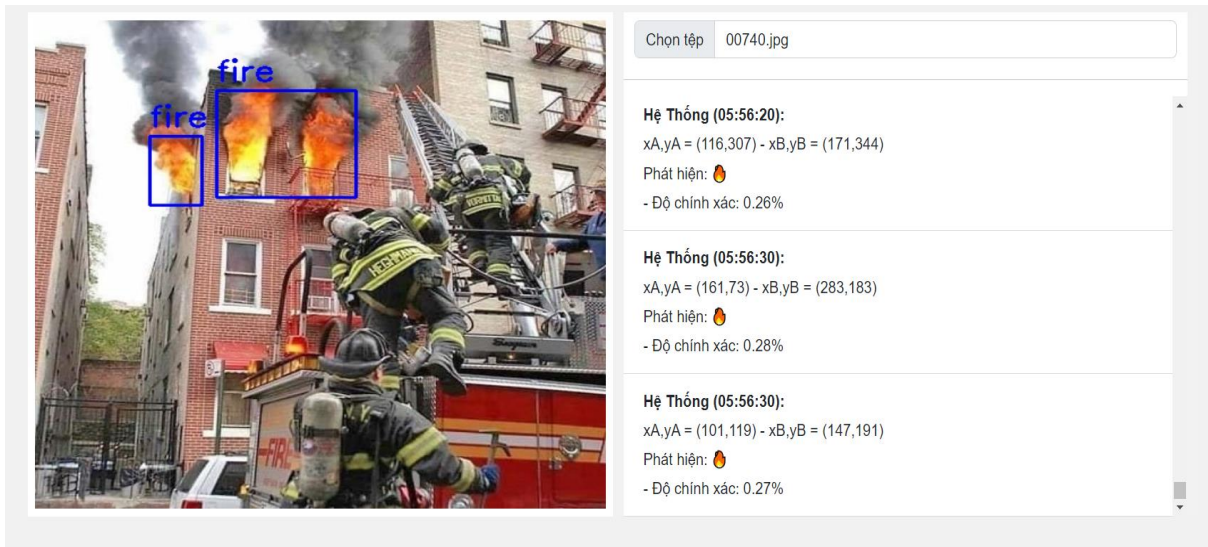


Hình 3.9: Biểu đồ giá trị mAP_0.5:0.95 qua từng giai đoạn

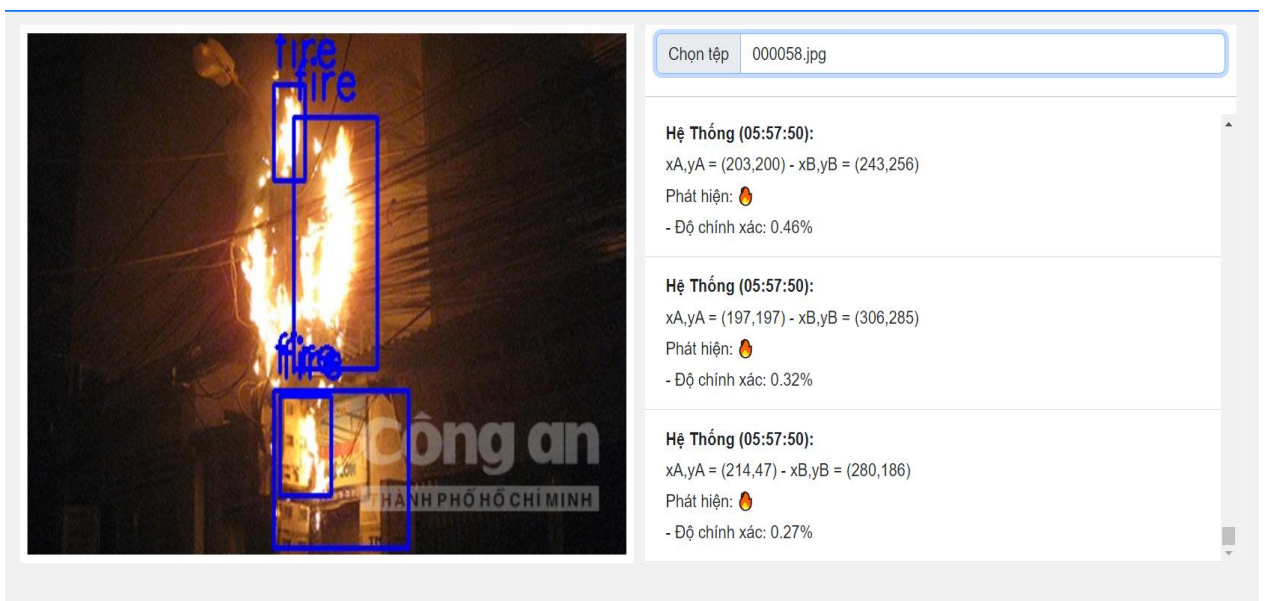
Qua biểu đồ ta thấy giá trị mAP_0.5:0.95 qua từng giai đoạn tăng dần

3.3.6. Huấn luyện mô hình

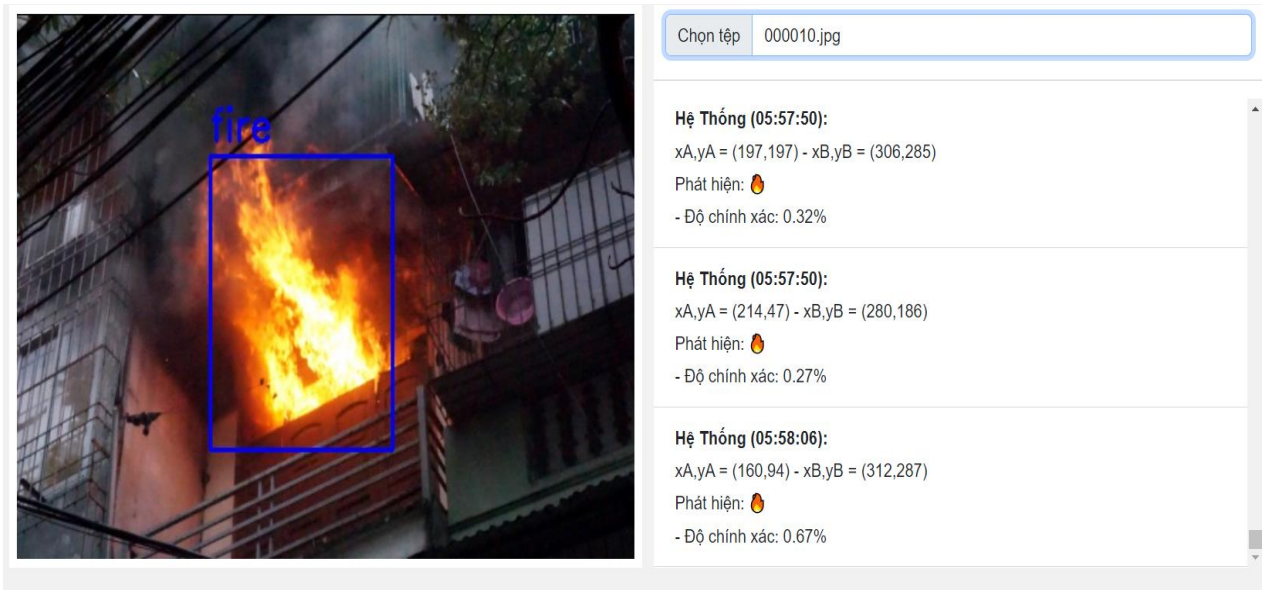
Sau khi đã hoàn thành các bước liên quan đến khai báo dữ liệu và thư mục chứa dữ liệu huấn luyện, ta có thể tiến hành huấn luyện mô hình dựa trên dữ liệu mà ta đã chuẩn bị trước đó. Dưới đây là một số hình ảnh về cháy được chương trình nhận diện và cảnh báo được



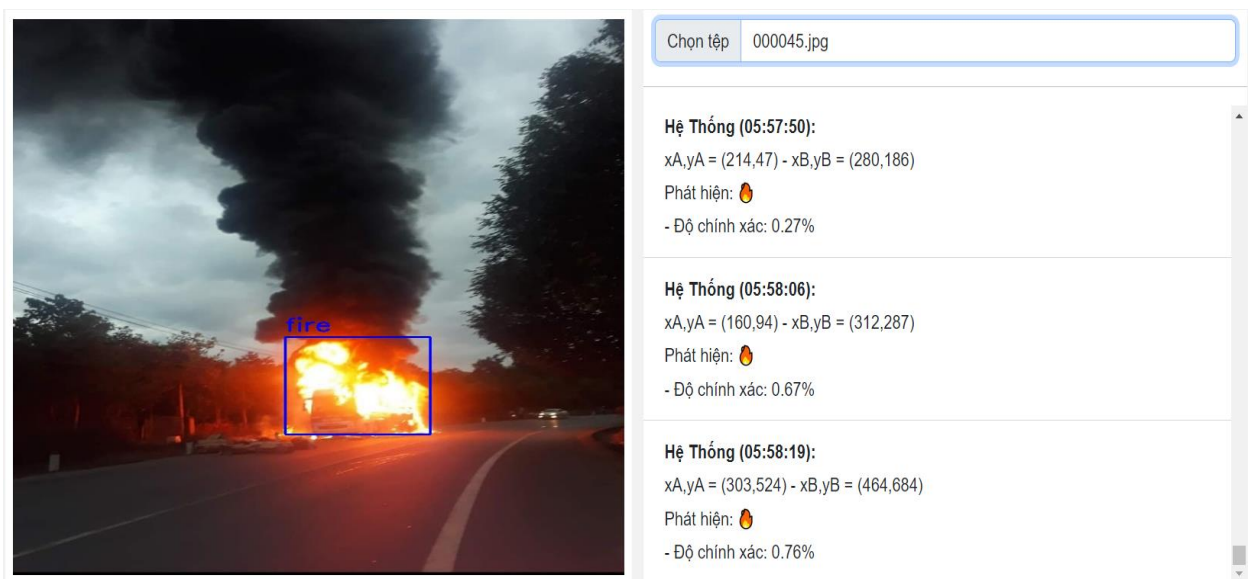
Hình 3.10: Hình ảnh cháy được phát hiện



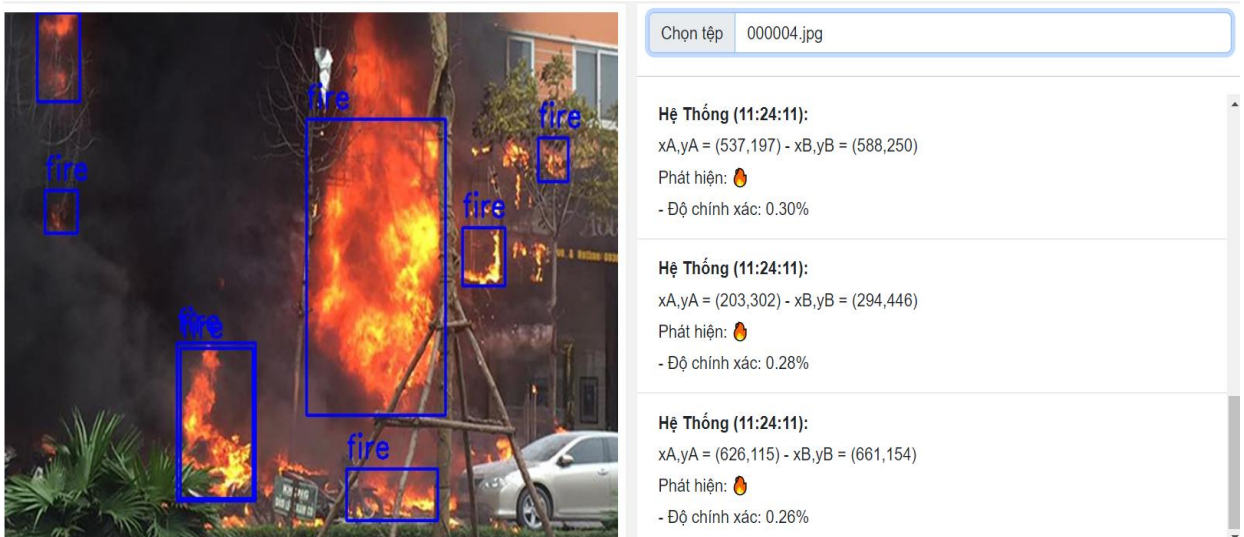
Hình 3.11: Hình ảnh cháy được phát hiện



Hình 3.12: Hình ảnh cháy được phát hiện



Hình 3.13: Hình ảnh cháy được phát hiện



Hình 3.14: Hình ảnh cháy được phát hiện

3.4. Đánh giá mô hình sau thử nghiệm

Các kết quả thử nghiệm thực tế cho thấy mô hình có khả năng nhận dạng và xác định được hình ảnh lửa từ 2-3 giây với độ chính xác khá cao. Tuy nhiên, mô hình vẫn chưa phân biệt được rõ ràng các đám cháy có ngọn lửa mang kích thước nhỏ và ngọn lửa trong các khung ảnh có độ sáng cao.

Về phần mô hình, kích thước của mô hình sau khi được huấn luyện khá lớn, vì thế việc sử dụng mô hình để dự đoán tốn khá nhiều thời gian vì phải thông qua nhiều số liệu thì mô hình mới đưa ra được 1 dự đoán.

Để đảm bảo tính ứng dụng cao trong thực tế và tận dụng tối đa các ưu điểm của Deep Learning và xử lý ảnh thị giác máy tính, mô hình cần phải được cập nhật liên tục và cần được huấn luyện thêm nữa để tăng độ chính xác khi phải gặp các trường hợp trong không gian mới với các dữ liệu mới.

Tổng thể, mô hình nhận dạng hỏa hoạn và lửa đã thực hiện được nhiệm vụ đó là nhận dạng và phát hiện ra lửa hoặc đám cháy. Tuy nhiên, cần cải thiện về tốc độ nhận dạng và xử lý của mô hình cũng như cải thiện hơn về độ chính xác để mô hình có thể sử dụng cho các hệ thống thực tế.

KẾT LUẬN

Sau thời gian 05 tháng nghiên cứu và thực hiện đề án tốt nghiệp, được sự chỉ bảo hướng dẫn tận tâm từ cô TS. Hồ Thị Hương Thơm – Khoa Công nghệ thông tin trường Đại học Hàng Hải Việt Nam để sinh viên có thể hoàn thiện được đề án với đề tài “ Cảnh báo cháy từ hình ảnh bằng YOLOv7” .

Một số mục tiêu em đã đạt được trong đề án này:

- Hoàn thành đề tài: “ Cảnh báo cháy từ hình ảnh bằng YOLOv7”;
- Đào tạo thành công mô hình nhận diện và cảnh báo cháy bằng mô hình YOLOv7
- Mô hình phát hiện cháy cho ra độ chính xác tương đối cao với epochs=30

Chúng ta có thể tích hợp, kế thừa mô đun xử lí hình ảnh cháy vào các mắt đọc của camera an ninh được đặt trong các tòa nhà, nhà riêng, phòng học, chung cư, khu công nghiệp, khu vui chơi, góc khuất để phát hiện và xử lí kịp thời những đám cháy tránh bùng phát gây nguy hiểm cho tính mạng con người hay tài sản. Cảnh báo cháy rất quan trọng, việc phát hiện sớm và phản ứng nhanh chóng là điều đầu tiên để kiểm soát hỏa hoạn lan rộng và có những phương pháp đối phó, xử lí với ngọn lửa tránh những tổn thất, mất mát đáng tiếc xảy ra. Nhất là khi trong những báo cáo và nghiên cứu, tỉ lệ cháy ở Việt Nam cao đặc biệt là các khu công nghiệp, khu có mật độ dân số cao và những cơ sở hạ tầng kém tỉ lệ cháy khá cao

Trong quá trình học tập và nghiên cứu, việc tìm hiểu và phát triển một lĩnh vực mà bản thân không có nhiều kinh nghiệm là một thử thách đối với bản thân của em. Dù đã rất cố gắng nghiên cứu, học hỏi nhưng do thời gian và hiểu biết của bản thân em còn hạn chế nên đề án chỉ dừng lại ở mức ứng dụng đơn giản chưa thể triển khai thực tế bởi việc này cần những kiến thức cùng với những công nghệ khác nhau . Vì vậy đề án chắc chắn cũng không tránh khỏi những thiếu sót, nên em rất mong nhận được sự đóng góp từ thầy cô.

Em xin chân thành cảm ơn!

TÀI LIỆU THAM KHẢO

- [1].Phương pháp phát hiện đối tượng bằng YOLOv7
(n.d.). Retrieved from <https://mpbpo.com.vn/yolov7/>
BPO. (n.d.). *BPO.MP*. Retrieved from BPO.MP: <https://mpbpo.com.vn/yolov7/>
- [2].Colaboratony
<https://colab.research.google.com/?hl=vi&pli=1&authuser=0>
- [3].Ngôn ngữ Python (glints, n.d.)
<https://glints.com/vn/blog/ngon-ngu-lap-trinh-python-la-gi/>
- [4].Dữ liệu thực nghiệm
<https://universe.roboflow.com/my-vd4ll/fire-detection-p8jid/dataset/1#>