

# NÂNG CAO HIỆU NĂNG PHÁT HIỆN ĐÁM CHÁY SỬ DỤNG THỊ GIÁC MÁY DỰA TRÊN MẠNG NƠ-RON YOLOV5

IMPROVEMENT THE FIRE DETECTION PERFORMANCE USING COMPUTER VISION BASED ON YOLOV5 NEURAL NETWORK

Đỗ Thị Ngọc Ánh<sup>1</sup>, Hoàng Mạnh Kha<sup>1</sup>,  
Lê Anh Tuấn<sup>1,\*</sup>, Nguyễn Ngọc Anh<sup>1</sup>

DOI: <https://doi.org/10.57001/huih5804.37>

## TÓM TẮT

Phát hiện đám cháy sớm và chính xác đang là yêu cầu cấp thiết đặt ra đối với các hệ thống cảnh báo cháy. Để góp phần giải quyết bài toán này, bài báo trình bày quá trình xây dựng ứng dụng phát hiện đám cháy sử dụng mạng nơ-ron tích chập YOLOv5. Ứng dụng sử dụng mô hình mạng nơ-ron YOLOv5 để phát hiện đám cháy đã cho kết quả thử nghiệm đạt độ chính xác cao và tốc độ xử lý nhanh có khả năng nhận dạng theo thời gian thực. Với thực trạng cháy nổ hiện tại đồng thời qua phân tích các hệ thống cảnh báo cháy đang có trên thị trường, hệ thống nhận dạng ngọn lửa sử dụng mạng nơ-ron khi kết hợp với các hệ thống camera an ninh hứa hẹn sẽ cải thiện được nhiều thiếu sót của các hệ thống báo cháy thông thường đang gặp phải, giúp giảm thiểu thiệt hại về người và cơ sở vật chất khi xảy ra hỏa hoạn.

**Từ khóa:** Phát hiện lửa, báo cháy, mạng nơ-ron tích chập, YOLOv5.

## ABSTRACT

Early and accurate in fire detection is an urgent requirement for fire alarm systems. To contribute to solving this problem, this paper presents a fire detection model based on YOLOv5 convolutional neural network (CNNs). Using YOLOv5 convolution neural network to fire detection has given high accuracy upper 90% and real time detection. The current fire and explosion situation combine analysis of the operation of fire alarm systems be used, a fire detection based on YOLOv5 convolutional neural network when combine a security camera system improve the performance of the fire alarm system.

**Keywords:** Fire detection, CNN-convolution neural network, YOLOv5.

<sup>1</sup>Trường Đại học Công nghiệp Hà Nội

\*Email: [anhthuandt04k2@gmail.com](mailto:anhthuandt04k2@gmail.com)

Ngày nhận bài: 02/6/2022

Ngày nhận bài sửa sau phản biện: 15/7/2022

Ngày chấp nhận đăng: 27/10/2022

## 1. GIỚI THIỆU

Xuất phát từ thực tế hiện nay, tình hình cháy nổ trong nước ngày một tăng cao kéo theo rất nhiều hệ lụy, thiệt hại nghiêm trọng về người và tài sản. Với tốc độ xây dựng cơ sở hạ tầng vô cùng mạnh mẽ, các tòa nhà cao tầng, trung tâm thương mại, trụ sở văn phòng luôn là những địa điểm tiềm ẩn những nguy cơ về hỏa hoạn. Chính vì vậy, việc phát hiện cháy sớm ngăn chặn hiệu quả các trường hợp gây cháy

diện rộng. Để phát hiện ngọn lửa các hệ thống báo cháy thông thường hiện nay đang sử dụng chính là cảm biến nhiệt độ, tuy nhiên phương pháp này có nhược điểm là không phát hiện được ngọn lửa nhỏ và chỉ phát hiện được khi ngọn lửa đã lan rộng làm nhiệt độ tăng cao. Phương pháp sử dụng kỹ thuật nhận dạng hình ảnh từ camera giúp khắc phục nhược điểm trên, với phương pháp này chỉ cần sử dụng một máy quay camera hoặc tận dụng hệ thống camera có sẵn là có thể giám sát không gian rộng.

Hệ thống phát hiện cháy sử dụng camera thông thường sử dụng trực tiếp dữ liệu hình ảnh, video từ các hệ thống camera giám sát, sau đó đưa qua các thuật toán xử lý hình ảnh, các thuật toán nhận dạng AI, từ đó đưa ra các cảnh báo khi phát hiện đám cháy. Ưu điểm của hệ thống là có thể tận dụng dữ liệu từ các hệ thống camera giám sát có sẵn, tuy nhiên độ chính xác hệ thống phụ thuộc phần lớn vào thuật toán xử lý, yêu cầu cấu hình phần cứng mạnh.

Trong các kỹ thuật nhận dạng hình ảnh, giải pháp áp dụng các mô hình mạng nơ-ron học sâu được đánh giá đạt hiệu năng tốt nhất. Kỹ thuật nhận dạng đối tượng (Object Detection) đã được sử dụng rộng rãi để phát hiện khuôn mặt, phát hiện xe, đếm số người đi bộ, hệ thống bảo mật và xe không người lái và các hệ thống phục vụ cho mục đích cảnh báo sớm từ xa. Các thuật toán nhận diện các đối tượng sử dụng các phương pháp cổ điển như phân vùng theo ngưỡng màu sắc, kỹ thuật luồng quang... hay sử dụng các thuật toán được hỗ trợ trong OpenCV đã không đạt được hiệu suất đủ tốt để làm việc trong nhiều điều kiện khác nhau. Việc áp dụng đột phá và nhanh chóng của kỹ thuật học sâu bắt đầu từ năm 2012 đã dẫn đến sự ra đời của các thuật toán và phương pháp phát hiện đối tượng hiện đại và chính xác cao như R-CNN [9], Fast-RCNN [10], Faster-RCNN [11] và nhanh hơn nhưng rất chính xác như SSD [12] và YOLO [1].

Nhận dạng đối tượng trong thị giác máy bao gồm 2 nhiệm vụ chính là:

- Phân loại hình ảnh (Image Classification): Dự đoán nhãn của các đối tượng trong một hình ảnh
- Định vị đối tượng (Object Localization): Xác định vị trí của đối tượng trong ảnh bằng khung chữ nhật bao quanh

đối tượng (Bounding Box). Trong một hình ảnh có thể có nhiều đối tượng, mỗi đối tượng có thể xuất hiện nhiều vị trí khác nhau với các kích thước khác nhau.

Trong bài báo trình bày cơ sở lý thuyết mạng YOLO và quy trình huấn luyện mô hình mạng YOLOv5 để nhận dạng ngọn lửa. Sử dụng mô hình đã được huấn luyện để xây dựng chương trình nhận dạng ngọn lửa, sau đó thực hiện các thực nghiệm nhận dạng từ đó đưa ra đánh giá hiệu năng của mô hình.

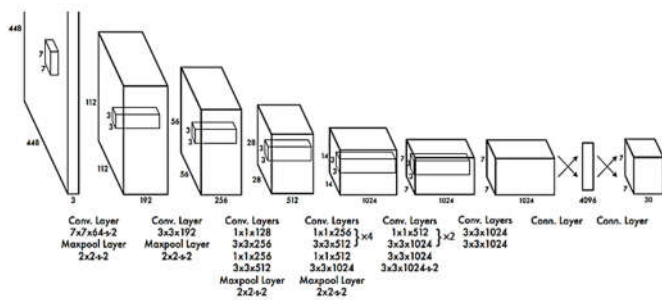
**2. CƠ SỞ LÝ THUYẾT MẠNG YOLO**

**2.1. Giới thiệu mạng YOLO**

YOLO là một kiến trúc mạng CNN được sử dụng trong phát hiện, nhận dạng và phân loại đối tượng. Đối bài toán phân loại (Classification) chỉ có khả năng phân loại đối tượng bằng các dự đoán nhãn thì YOLO giải quyết bài toán phát hiện đối tượng (Object Detection), không chỉ có thể phát hiện nhiều đối tượng với nhiều nhãn khác nhau mà còn có thể xác định vị trí cụ thể của các đối tượng trong cùng một hình ảnh bằng các khung bao quanh đối tượng hình chữ nhật (Bounding Box). YOLO là viết tắt của cụm từ "You only look once" nói nên khả năng về tốc độ nhận dạng của mô hình này, YOLO được đánh giá là mô hình cho tốc độ nhận dạng nhanh nhất có khả năng nhận dạng theo thời gian thực. Kiến trúc YOLO được xây dựng từ các lớp tích chập (Convolution layers) để trích xuất ra các đặc trưng của đối tượng và các lớp kết nối đầy đủ (full connected layer) để dự đoán nhãn và vị trí của đối tượng. Dữ liệu đầu vào là các hình ảnh, mô hình sẽ dự đoán vị trí, kích thước và nhãn của các Bounding Box.

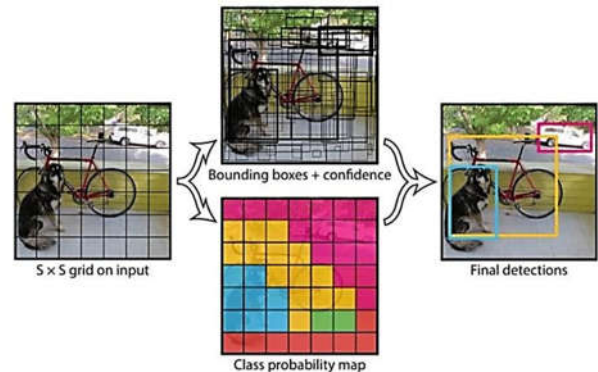
**2.2. Kiến trúc mạng YOLO**

YOLO có kiến trúc bao gồm phần trích xuất đặc trưng (Feature Extractor) và phát hiện đối tượng (Extra Layers). Phần trích xuất đặc trưng là các lớp convolution sẽ cho đầu ra là các bản đồ đặc trưng (Feature map). Phần phát hiện vật thể bao gồm các lớp kết nối đầy đủ (Fully connected layers) dùng để phát hiện, dự đoán nhãn và tọa độ Bounding Box của đối tượng trên các bản đồ đặc trưng (Feature map) của phần Feature Extractor. Hình ảnh cần nhận dạng sẽ được đi qua khâu trích xuất đặc trưng để có đầu ra ra là bản đồ đặc trưng (Feature map) với các kích thước khác nhau. Các bản đồ đặc trưng sau đó được đưa qua khâu phát hiện đối tượng (Extra Layers) để dự đoán ra các thông tin về tên đối tượng, vị trí và kích thước của Bounding Box bao quanh đối tượng.



Hình 1. Kiến trúc mạng YOLO [1]

Dữ liệu đầu vào là hình ảnh hoặc một frame được trích xuất từ Video. Ảnh đầu vào sau đó được chia thành mạng lưới có S x S ô (hay còn gọi là grid size). Thông thường grid size sẽ là 3 x 3, 7 x 7, 9 x 9, ...



Hình 2. Mô hình hoạt động của mạng YOLO [1]

**2.3. Nguyên lý hoạt động mạng YOLO**

YOLO sẽ phân chia hình ảnh thành các ô vuông dạng lưới (grid) và xác định xem trong mỗi ô vuông liệu có tầm của vật thể cần xác định hay không. Nếu có, mô hình YOLO sẽ khoanh vùng đối tượng bằng hộp mồi (Anchor Box), sau khi có kết quả lọc chính xác thì sẽ xuất ra bounding box. Thuật toán sử dụng một mạng nơ ron đơn với đặc trưng có được từ các feature map của các lớp tích chập để dự đoán bounding box ở mỗi ô và xác suất loại đối tượng nào đang được chứa bên trong. Sau cùng, ta sẽ có rất nhiều bounding box được thuật toán đưa ra với kích thước khác nhau. Sử dụng thuật toán Non-Maxima Suppresstion (NMS) ta có thể loại được các hầu hết các bounding box là miễn bao của cùng một đối tượng, có tỉ lệ thấp và giữ lại các bounding box có tỉ lệ khớp cao.

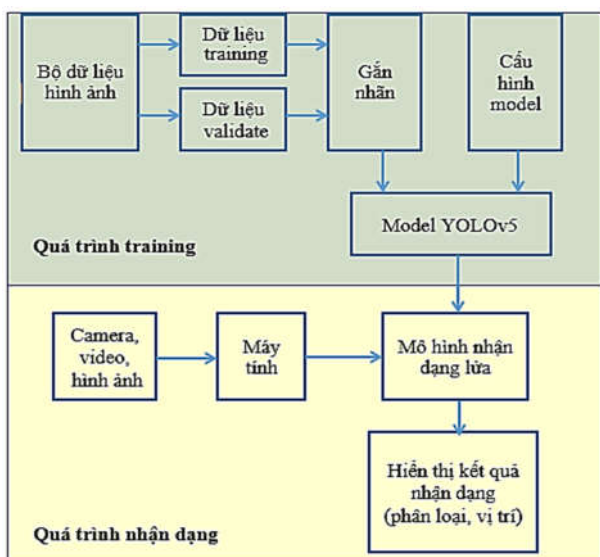
Đầu ra của YOLO khi dự đoán 1 Bounding Box là một vector gồm các thành phần gồm: xác suất dự báo có vật thể (P0) được định nghĩa  $Pr(Object) * IOU(pred, truth)$ , tọa độ tâm kích thước chiều rộng, cao của Bouding Box (x, y, w, h), các vector phân phối xác suất dự báo của các classes. Kích thước vector đầu ra được tính bằng (5 + số classes) ví dụ khi ta huấn luyện nhận dạng 50 đối tượng(classes) thì kích thước vector đầu ra dự đoán cho mỗi Bounding Box sẽ có kích thước là 55. Như vậy đầu ra của cả mô hình là một ma trận 3 chiều có kích thước  $S \times S \times B \times (5+C)$  trong đó SxS là kích thước của các Feature map ở phần trích xuất đặc trưng, B là số lượng Bounding Box tương ứng là số Anchor Box và tại mỗi ô (cell) áp dụng 3 Anchor Box như vậy  $B = 3$ , C là số classes.

Mô hình YOLO hiện tại có 5 phiên bản YOLOv1 [1], YOLOv2 [2], YOLOv3 [3] do Joseph Redmon phát triển từ 2016, YOLOv4 [4] do Alexey Bochkovskiy phát triển năm 2020 và YOLOv5 do Glenn Jocher phát hành trong cùng năm 2020 [5]. Khác với những phiên bản trước, YOLOv5 được phát triển dựa trên PyTorch thay vì DarkNet [5]. Đây là một ưu điểm lớn YOLOv5 vì PyTorch phổ biến hơn do đó sẽ có nhiều tài liệu và hướng dẫn để tham khảo về mô hình này. YOLOv5 đề xuất 5 phiên bản như sau:

YOLOv5n - Nano version, YOLOv5s - Small version, YOLOv5m - Medium version, YOLOv5l - Large version, YOLOv5x - Extra-large version. YOLOv5 được công bố gần đây với những so sánh ban đầu cho thấy độ chính xác tương đương YOLOv4 và có tốc độ nhanh hơn khi thực hiện dự đoán. Trong ứng dụng này tác giả sử dụng mô hình YOLOv5 với phiên bản YOLOv5s.

### 3. ỨNG DỤNG PHÁT HIỆN LỬA

#### 3.1. Mô hình hệ thống



Hình 3. Mô hình hệ thống

Bộ dữ liệu dataset được chia thành 2 tập dữ liệu bao gồm tập huấn luyện(training) và tập xác minh (validate), các hình ảnh sau đó được tiến hành gắn nhãn xác định vị trí các ngọn lửa. Tiến hành cấu hình các thông số cho Model, sử dụng tập dữ liệu huấn luyện và tập validate để huấn luyện cho mô hình. Mô hình sau khi được huấn luyện thành công bộ trọng số được lưu dưới định dạng "model.pt". Mô hình huấn luyện được sử dụng cho chương trình thực thi nhận dạng ngọn lửa với dữ liệu đầu vào lấy từ các hình ảnh, videos, camera. Kết quả quá trình nhận dạng phần mềm sẽ hiển thị trực tiếp kết quả phân loại "fire" và vị trí của ngọn lửa trên màn hình đồng thời lưu lại hình ảnh hoặc video nhận dạng.

#### 3.2. Chuẩn bị bộ dữ liệu

Tập dữ liệu được sử dụng để huấn luyện mô hình bao gồm 1403 ảnh có kích thước 224x224, được xây dựng bằng cách chọn lọc từ bộ dữ liệu hơn 10000 ảnh do Durham University [6]. Hình ảnh trong bộ dữ liệu có nhiều khung cảnh và kích thước ngọn lửa khác nhau. Trong 1403 ảnh gồm có 1194 ảnh có lửa, 209 ảnh không có ngọn lửa. Mỗi ảnh xuất hiện có thể xuất hiện một hoặc nhiều ngọn lửa với các kích thước khác nhau, tất cả các ngọn lửa đều phải được gắn nhãn do vậy mỗi ảnh có thể có nhiều hơn 1 nhãn. Tổng số có 3129 ngọn lửa được gắn nhãn, quá trình gắn nhãn cho nhóm tác giả thực hiện. Bộ dữ liệu được phân chia 80% cho tập huấn luyện và 20% cho tập xác minh (validate).

Cấu trúc tập dữ liệu YOLOv5 yêu cầu bao gồm:

- Một thư mục images chứa ảnh: trong đó có 2 thư mục train và val để chứa ảnh train và ảnh validate.

- Một thư mục labels chứa nhãn (các file txt đó) và cũng có 2 thư mục tương tự như images.

Dữ liệu đầu vào ảnh của YOLOv5 theo format darknet với mỗi 1 file .txt sẽ cho 1 ảnh có chứa đối tượng là label, còn với những ảnh không có đối tượng thì bỏ qua. File .txt có định dạng như sau :

- Mỗi hàng sẽ là một đối tượng

- Mỗi hàng sẽ có format như sau: class x\_center y\_center width height trong đó class x\_center y\_center là tọa độ x, y của tâm khung bao vật thể, width, height là chiều rộng và cao của khung bao vật thể

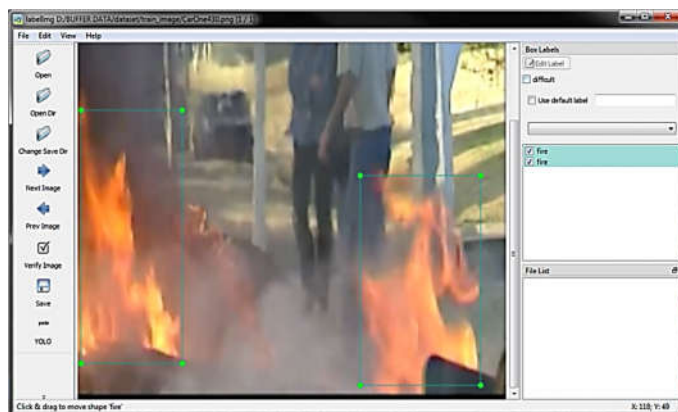
- Tọa độ của các box sẽ được chuẩn hóa (từ 0-1) theo format xywh

- Class sẽ bắt đầu từ 0

```
0 0.524554 0.517857 0.200893 0.6339290
1 0.513393 0.064732 0.017857 0.031250
2 0.444196 0.604911 0.183036 0.4598210
3 0.252232 0.531250 0.495536 0.9017860
4 0.252232 0.531250 0.495536 0.901786
```

Hình 4. Định dạng File.txt gắn nhãn của YOLOv5

Tất cả hình ảnh sử dụng cho training đều phải được gắn nhãn, tác giả sử dụng phần mềm LabelIMG để gắn nhãn cho hình ảnh.



Hình 5. Gắn nhãn cho ảnh sử dụng phần mềm LabelIMG

#### 3.3. Huấn luyện mô hình

Trong quá trình huấn luyện mạng nơ ron sẽ tính toán lần lượt tất cả các ảnh của tập dữ liệu đầu vào và sử dụng lặp lại các ảnh này nhiều lần, mục đích để tối ưu hàm mất mát. Quá trình tối ưu này sẽ giúp cho mạng nơ-ron tìm được bộ trọng số tốt nhất, giúp cho quá trình nhận dạng được chính xác nhất. Mô hình sau khi huấn luyện thành công bộ trọng số sẽ được lưu trữ trong file kết quả có định dạng ".pt". Trong YOLOv5 ngoài việc bắt đầu quá trình huấn luyện mới từ đầu YOLOv5 còn cho phép người dùng sử dụng bộ trọng số có sẵn để tiếp tục quá trình huấn

luyện để cho quá trình huấn luyện nhanh hơn thay vì phải huấn luyện từ đầu.

Quá trình huấn luyện mô hình sử dụng một lượng lớn tài nguyên của máy tính, nếu máy tính có cấu hình không đủ mạnh thời gian huấn luyện diễn ra rất dài. Ta có thể huấn luyện trên Google Colab để giảm thời gian huấn luyện đồng thời đảm bảo hiệu quả và chất lượng của mô hình.

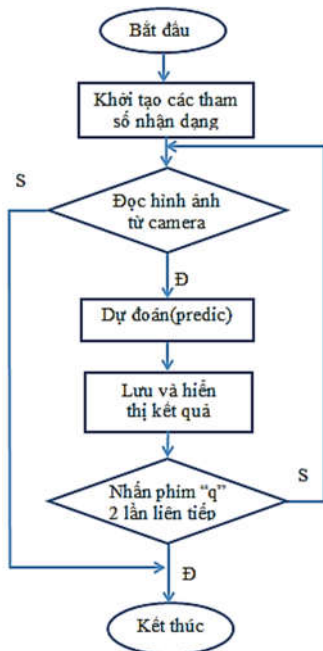
Bảng 1. Danh mục các tham số cấu hình cho mạng

Tên tham số	Giá trị tham số	Ý nghĩa tham số
img	224	Kích thước ảnh đầu vào
Batch size	64	Số lượng ảnh sử dụng mỗi vòng huấn luyện
epochs	1000	Số lượt huấn luyện trên toàn bộ dữ liệu
data	fire_dataset	File chứa đường dẫn tập dữ liệu
weight	yolov5s	Lựa chọn phiên bản model yolov5s
device	0	Lựa chọn GPU training, chọn 'cpu' nếu không có GPU
name	defaut	Đường dẫn thư mục lưu model

Số lượng ảnh tối đa của mỗi batch phụ thuộc vào cấu hình phần cứng của máy tính, nhóm tác giả sử dụng máy chủ google colab mỗi batch sử dụng 64 hình ảnh.

Số lượt huấn luyện toàn bộ dữ liệu (epochs) trong quá trình huấn luyện tác giả Glenn Jocher khuyến cáo sử dụng tối thiểu 300 epochs [7], thực nghiệm cho thấy các tham số mô hình bão hòa tại epochs đạt 1000 lần.

### 3.4. Chương trình phát hiện đám cháy sử dụng hình ảnh từ camera



Hình 6. Giải thuật của hệ thống

Giải thuật của hệ thống phát hiện đám cháy sử dụng hình ảnh từ camera như hình 6. Thiết lập các tham số cho quá trình nhận dạng như bảng 2.

Bảng 2. Danh mục cấu hình các tham số nhận dạng

Tên tham số	Giá trị tham số	Ý nghĩa tham số
img	640	Kích thước ảnh đầu vào theo pixel
source	0	Nguồn dữ liệu đầu vào chọn '0' với dữ liệu nhận dạng từ camera, hoặc đường dẫn tới hình ảnh, video có sẵn
conf-thres	0.25	Thiết lập ngưỡng độ tin cậy dự đoán
iou-thres	0.45	Thiết lập ngưỡng IoU
weight	best.pt	Lựa chọn bộ trọng số tốt nhất của kết quả training model yolov5s
device	0	Lựa chọn GPU training, chọn 'cpu' nếu không có GPU

Phần mềm 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.

## 4. KẾT QUẢ THỰC NGHIỆM

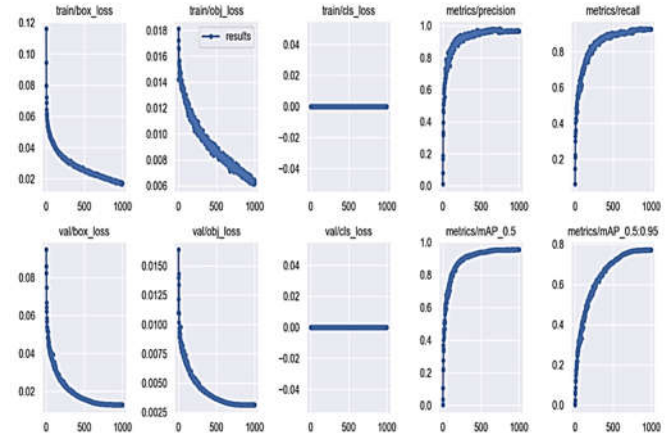
### 4.1. Kết quả

Kết thúc quá trình huấn luyện có 2 bộ trọng số được mô hình lưu lại bao gồm:

- Bộ trọng số tốt nhất: best.pt
- Bộ trọng số cuối cùng của quá trình huấn luyện: last.pt

Trong đó bộ trọng số best.pt được sử dụng cho chương trình nhận dạng.

Thông số chi tiết kết quả training thể hiện trong hình 7.



Hình 7. Biểu đồ kết quả quá trình training của model

Biểu đồ hình 7 cho thấy, sau quá trình training 1000 epoch độ chính xác trung bình dự đoán của training (precision) đạt 95,39% khi, độ mất mát (loss) 2,3%, kích thước mô hình 13,6Mb. Đối với quá trình validate độ chính xác trung bình của phát hiện tại IoU = 0,5 (mAP\_0.5) đạt 95,4%

Ứng dụng nhận dạng 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 chính xác của dự đoán.



Hình 8. Kết quả nhận dạng trực tiếp qua camera

### 4.2. ThỰC NGHIỆM VÀ ĐÁNH GIÁ

Để đánh giá hiệu năng của mô hình, nhóm tác giả đánh giá hai thông số bao gồm độ chính xác trong nhận dạng và tốc độ nhận dạng. Độ chính xác được đánh giá trong 3 điều kiện: Trong nhà thiếu sáng, trong nhà đủ ánh sáng, ngoài trời.

Đánh giá độ chính xác trong nhận dạng nhóm tác giả sử dụng hai thông số gồm:

**Precision:** Biểu thị độ chính xác trong dự đoán tên và vị trí đối tượng;

**Recall:** Biểu thị khả năng phát hiện đối tượng trong dữ liệu đầu vào.

Trong hệ thống yêu cầu phát hiện ngọn lửa là nhiệm vụ quan trọng nhất do đó tham số Recall được quan tâm hơn Precision. Recall càng cao khả năng phát hiện lửa càng cao đồng nghĩa là khả năng bỏ sót lửa càng thấp.

Độ chính xác dự đoán đối tượng:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Khả năng phát hiện đối tượng:

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

Trong đó:

TP - True Positive: Thực tế có đối tượng, dự đoán có đối tượng;

FN - False Negative: Thực tế có đối tượng, dự đoán không có đối tượng;

TN - True Negative: Thực tế không có đối tượng, dự đoán không có đối tượng;

FP - False Positive: Thực tế không có đối tượng, dự đoán có đối tượng.

Bên cạnh đó tác giả đã thực hiện so sánh kết quả với giải pháp đã được công bố [8] sử dụng YOLOv3. Kết quả so sánh được thực hiện với các điều kiện giống nhau cho cả hai giải pháp như: sử dụng cùng 1 bộ dữ liệu huấn luyện, cài đặt cùng 1 bộ tham số (IoU, image size), thực thi trên cùng một nền tảng phần cứng.

#### 4.2.1. ThỰC NGHIỆM VỚI ĐIỀU KIỆN TRONG NHÀ, THIẾU ÁNH SÁNG

Thực nghiệm được thực hiện lấy dữ liệu từ camera trong nhà với điều kiện phòng thiếu ánh sáng. Nhận dạng đám cháy trong khoảng thời gian 30s, trích xuất hình ảnh từ video kết quả thu được 947 hình ảnh kích thước 480x854. Hình ảnh sau khi nhận dạng như trong hình 9.



Hình 9. Ảnh nhận dạng lửa trong nhà thiếu sáng

Kết quả nhận dạng sau khi trích xuất frame ảnh như trong bảng 3.

Bảng 3. Bảng kết quả so sánh dự đoán với điều kiện trong nhà, thiếu ánh sáng

Giải pháp	Ảnh có lửa		Ảnh không có lửa		Precision	Recall
	Phát hiện (TP)	Không phát hiện (FN)	Phát hiện (FP)	Không phát hiện (TN)		
Đề xuất-YOLOv5s	524	7	11	405	97,94%	98,68%
YOLOv3 [8]	401	130	0	416	100%	75,52%

Kết quả trên cho thấy hiệu năng của giải pháp đề xuất là tốt hơn so với giải pháp sử dụng mô hình YOLOv3 trong khả năng phát hiện lửa thể hiện qua tham số Recall. Lưu ý rằng trong ứng dụng phát hiện và cảnh báo cháy, tham số Recall cần được ưu tiên cao hơn.

#### 4.2.2. ThỰC NGHIỆM VỚI ĐIỀU KIỆN TRONG NHÀ ĐỦ ÁNH SÁNG

Thực nghiệm được thực hiện lấy dữ liệu từ camera trong nhà với điều kiện phòng đủ ánh sáng. Nhận dạng đám cháy trong khoảng thời gian 28s. Trích xuất hình ảnh từ video kết quả thu được 844 hình ảnh 480x854 pixel. Ảnh nhận dạng lửa trong nhà đủ ánh sáng như trong hình 10.



Hình 10. Ảnh nhận dạng lửa trong nhà đủ ánh sáng

Kết quả nhận dạng sau khi trích xuất frame ảnh như trong bảng 4.

Kết quả nhận dạng sử dụng giải pháp YOLOv5 vẫn đạt độ chính xác trên 95%, giải pháp sử dụng YOLOv3 đạt độ

chính xác thấp hơn rất nhiều. Thực nghiệm cho thấy giải pháp sử dụng YOLOv3 phát hiện ngọn lửa kích thước nhỏ kém hơn, nhạy cảm với ánh sáng môi trường hơn nhiều so với YOLOv5.

Bảng 4. Bảng kết quả so sánh dự đoán với điều kiện trong nhà đủ sáng

Giải pháp	Ảnh có lửa		Ảnh không có lửa		Precision	Recall
	Phát hiện (TP)	Không phát hiện (FN)	Phát hiện (FP)	Không phát hiện (TN)		
Đề xuất-YOLOv5s	750	34	0	60	100%	95,66%
YOLOv3[8]	243	541	0	60	100%	30,99%

**4.2.3. Thực nghiệm với điều kiện ngoài trời**

Thực nghiệm được thực hiện bằng lấy dữ liệu từ camera ngoài trời. Nhận dạng đám cháy trong khoảng thời gian 34s. Trích xuất hình ảnh từ video kết quả thu được 1010 hình ảnh kích thước 480x854. Ảnh nhận dạng đúng ngọn lửa ngoài trời như trong hình 11.



Hình 11. Ảnh nhận dạng đúng ngọn lửa ngoài trời

Kết quả nhận dạng sau khi trích xuất frame ảnh như trong bảng 5.

Bảng 5. Bảng kết quả so sánh dự đoán với điều kiện ngoài trời

Giải pháp	Ảnh có lửa		Ảnh không có lửa		Precision	Recall
	Phát hiện (TP)	Không phát hiện (FN)	Phát hiện (FP)	Không phát hiện (TN)		
Đề xuất-YOLOv5s	560	101	0	349	100%	84,72%
YOLOv3 [8]	262	399	0	349	100%	39,64%

Ở thực nghiệm này độ chính xác nhận dạng giảm mạnh ở cả 2 giải pháp do ảnh hưởng điều kiện ánh sáng môi trường, tuy nhiên giải pháp sử dụng YOLOv5 vẫn đạt độ chính xác trên 80%, giải pháp sử dụng YOLOv3 chỉ đạt dưới 40%. Điều này cho thấy giải pháp sử dụng YOLOv3 [8] kém ổn định với điều kiện ánh sáng môi trường so với YOLOv5.

**4.2.4. Thực nghiệm đánh giá tốc độ nhận dạng**

Thực nghiệm đánh giá tốc độ nhận dạng với đầu vào là video được đánh giá trên cùng nền tảng phần cứng là PC đạt kết quả như trong bảng 6.

Bảng 6. So sánh tốc độ nhận dạng trên PC

Phần cứng	CPU	RAM	GPU	FPS
FPS của giải pháp đề xuất-YOLOv5s	Intel Xeon(R) Bronze 3104-1.7GHz	16GB	NVIDIA Quadro P2200	51,8
FPS của giải pháp sử dụng YOLOv3 [8]				31,6

Bảng 6 cho thấy giải pháp sử dụng đề xuất sử dụng YOLOv5 cho tốc độ xử lý nhanh hơn nhiều so với giải pháp sử dụng YOLOv3 [8]. Thời gian nhận dạng này cho thấy mô hình sử dụng YOLOv5s hoàn toàn có thể ứng dụng trong các hệ thống báo cháy dân dụng sử dụng phần cứng có cấu hình trung bình.

**5. KẾT LUẬN**

Từ kết quả thực nghiệm cho thấy giải pháp đề xuất sử dụng YOLOv5 cho hiệu năng tốt hơn nhiều so với giải pháp sử dụng YOLOv3 về độ chính xác và tốc độ nhận dạng trong các điều kiện khác nhau. Hệ thống có khả năng phát hiện, nhận dạng tốt với điều kiện trong nhà và ngoài trời, đặc biệt với kiện trong nhà hệ thống cho kết quả nhận dạng với độ chính xác lên tới Precision ≥ 95% và Recall ≥ 95%. Tốc độ nhận dạng nhanh 19ms/frame ảnh cho phép triển khai nhận dạng theo thời gian thực. Bên cạnh đó kết quả chính xác của precision và recall phụ thuộc vào các yếu tố như điều kiện ánh sáng nền và kích thước của đối tượng trong ảnh, hệ thống hoạt động hiệu quả hơn với các đối tượng có kích thước lớn.

**TÀI LIỆU THAM KHẢO**

- [1]. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, 2016. *You Only Look Once: Unified, Real-Time Object Detection*. arXiv:1506.02640v5 [cs.CV].
- [2]. Joseph Redmon, Ali Farhadi, 2016. *YOLO9000: Better, Faster, Stronger*. arXiv:1612.08242v1 [cs.CV].
- [3]. Joseph Redmon, Ali Farhadi, 2018. *YOLOv3: An Incremental Improvement*. arXiv:1804.02767v1.
- [4]. Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, 2020. *YOLOv4: Optimal Speed and Accuracy of Object Detection*. arXiv:2004.10934v1, 2020
- [5]. Glenn Joche, 2020. <https://github.com/ultralytics/yolov5>.
- [6]. <https://collections.durham.ac.uk/files/r2d217qp536#.YjRZzUza01>
- [7]. Glenn Joche. <https://github.com/ultralytics/yolov5/wiki/Tips-for-Best-Training-Results>
- [8]. Pu Li, Wangda Zhao, 2020. *Image fire detection algorithms based on convolutional neural networks*. Case Studies in Thermal Engineering - journals elsevier.
- [9]. Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, 2014. *Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5)*. arXiv:1311.2524v5.
- [10]. Ross Girshick, 2015. *Fast R-CNN*. arXiv:1504.08083v2.
- [11]. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, 2016. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. arXiv:1506.01497v3.
- [12]. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, 2016. *SSD: Single Shot MultiBox Detector*. arXiv:1512.02325v5.

**AUTHORS INFORMATION**

**Do Thi Ngoc Anh, Hoang Manh Kha, Le Anh Tuan, Nguyen Ngoc Anh**  
Hanoi University of Industry