HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

ĐINH THỊ THANH NGA

ỨNG DỤNG TRÍ TUỆ NHÂN TẠO TRONG HỆ THỐNG CẢNH BÁO CHÁY

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(*Theo định hướng ứng dụng*)

HÀ NỘI - NĂM 2025

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



ĐINH THỊ THANH NGA

ỨNG DỤNG TRÍ TUỆ NHÂN TẠO TRONG HỆ THỐNG CẢNH BÁO CHÁY

Chuyên ngành: KỸ THUẬT VIỄN THÔNG
Mã số: 8.52.02.08

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT
(*Theo định hướng ứng dụng*)

NGƯỜI HƯỚNG DẪN KHOA HỌC : PGS.TS NGUYỄN TIẾN BAN

HÀ NỘI - NĂM 2025

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi.

Các số liệu, kết quả nêu trong đề án tốt nghiệp là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tác giả đề án tốt nghiệp

Đinh Thị Thanh Nga

LỜI CẢM ƠN

Trước hết, em xin gửi lời biết ơn sâu sắc tới **PGS.TS Nguyễn Tiến Ban**, người hướng dẫn trực tiếp đề tài. Sự tận tâm, uyên thâm trong kiến thức cùng những góp ý tỉ mỉ, sâu sắc của thầy đã soi đường, dẫn lối, giúp em vượt qua những khó khăn trong nghiên cứu, đồng thời nuôi dưỡng tinh thần học hỏi, tư duy sáng tạo và khao khát khám phá khoa học. Những buổi trao đổi, góp ý của thầy không chỉ là kiến thức, mà còn là nguồn động lực tinh thần vô giá, giúp đề tài này dần hiện thực hóa.

Em xin gửi lời tri ân tới **Học viện Công nghệ Bưu chính Viễn thông**, nơi đã đặt nền móng kiến thức vững chắc và mở ra những chân trời học thuật đầy cảm hứng. Mỗi bài giảng, mỗi cơ hội tiếp cận công nghệ tiên tiến tại Học viện đều là hành trang quý báu, giúp em vững tin trên con đường nghiên cứu chuyên sâu.

Đồng thời, em xin chân thành cảm ơn **các thầy cô và đồng nghiệp tại Trường Cao đẳng nghề Công nghiệp Hà Nội**, nơi em đang công tác. Sự hỗ trợ tận tình về cơ sở vật chất, chia sẻ kinh nghiệm thực tiễn, cũng như những lời động viên khích lệ, đã tạo điều kiện để đề tài được triển khai thành công và gắn kết chặt chẽ với thực tiễn nghề nghiệp.

Cuối cùng, em xin gửi lời biết ơn sâu sắc tới gia đình và những người thân yêu, những bờ vai vững chắc, luôn tin tưởng và động viên em trên mỗi bước đường học tập và nghiên cứu. Em hy vọng rằng kết quả của đề tài sẽ là một đóng góp nhỏ nhưng ý nghĩa, đồng thời là bước khởi đầu cho những nghiên cứu tiếp theo, tiếp tục mở ra những chân trời tri thức mới.

MỤC LỤC

LỜI CAM ĐOAN ..... **i**
LỜI CẢM ƠN ..... **ii**
MỤC LỤC ..... **iii**
DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT ..... **vi**
DANH MỤC CÁC BẢNG ..... **x**
MỞ ĐẦU **1**
CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG CẢNH BÁO CHÁY ỨNG DỤNG TRÍ TUỆ NHÂN TẠO **4**
1.1. Tổng quan về trí tuệ nhân tạo **4**
1.1.1. Khái niệm và vai trò của trí tuệ nhân tạo 4
1.1.2. Cấu trúc và thành phần cơ bản của trí tuệ nhân tạo 5
1.1.3. Các nhánh phát triển quan trọng của trí tuệ nhân tạo 6
1.1.4. Mối quan hệ giữa trí tuệ nhân tạo và kỹ thuật viễn thông 7
1.1.5. Ứng dụng trí tuệ nhân tạo trong hệ thống cảnh báo cháy thông minh 7
1.2. Nguyên lý hoạt động của hệ thống cảnh báo cháy **8**
1.2.1. Tổng quan 8
1.2.2. Cấu trúc và các thành phần chính 9
1.2.3. Nguyên lý hoạt động tổng quát 12
1.2.4. Hiệu quả và xu hướng ứng dụng 13
1.3. Thuật toán YOLO trong phát hiện đối tượng **14**
1.3.1. Nguyên lý hoạt động 14
1.3.2. Ưu điểm kỹ thuật của YOLO 16
1.3.3. Ứng dụng YOLO trong hệ thống cảnh báo cháy 17
1.4. Các yếu tố ảnh hưởng đến hiệu quả hệ thống cảnh báo cháy **18**
1.4.1. Chất lượng dữ liệu đầu vào 18
1.4.2. Cấu hình và hiệu suất mô hình trí tuệ nhân tạo 19
1.4.3. Năng lực xử lý của phần cứng và độ trễ toàn hệ thống 20
1.4.4. Ảnh hưởng của môi trường vận hành 20

1.4.5. Kết nối mạng và nền tảng IoT 20
1.5. Kết luận chương 1 21
CHƯƠNG 2: GIẢI PHÁP CẢNH BÁO CHÁY DỰA TRÊN MÔ HÌNH HỌC SÂU VÀ TRÍ TUỆ NHÂN TẠO 22
2.1. Cơ sở lựa chọn giải pháp 22
2.1.1. Bối cảnh khoa học và thực tiễn 22
2.1.2. Cơ sở công nghệ - sự phát triển của học sâu và thị giác máy tính 22
2.1.3. Cơ sở lựa chọn về mặt kỹ thuật và ứng dụng trong viễn thông 23
2.2. Mô hình học sâu áp dụng 23
2.2.1. Tổng quan về mô hình học sâu trong phát hiện cháy 23
2.3. Quy trình phát hiện cảnh báo cháy 25
2.3.1. Thu thập dữ liệu hình ảnh 27
2.3.2. Tiền xử lý ảnh 27
2.3.3. Suy luận bằng mô hình YOLO 27
2.3.4. Ra quyết định cảnh báo 27
2.3.5. Kích hoạt cảnh báo & truyền tín hiệu 28
2.3.6. Lưu trữ, phản hồi và cải tiến 28
2.4. Đánh giá giải pháp 29
2.4.1. Hiệu năng mô hình (COCO metrics) 30
2.4.2. Độ trễ hệ thống (End-to-End) 30
2.4.3. Một số tham số hiệu năng khác 30
2.5. Kết luận chương 2 31
CHƯƠNG 3: THỬ NGHIỆM VÀ ĐÁNH GIÁ HỆ THỐNG CẢNH BÁO CHÁY BẰNG YOLO 32
3.1. Xây dựng và huấn luyện mô hình YOLO 32
3.1.1. Thu nhận dữ liệu và cấu hình phần cứng 32
3.1.2. Chuẩn bị dữ liệu và tiền xử lý 32
3.1.3. Cấu hình huấn luyện mô hình YOLO 33
3.1.4. Tích hợp mô hình vào ứng dụng Python 33

3.1.5. Kết nối PLC FX3U và hệ thống cảnh báo.....34
3.2. Phân tích dữ liệu giả lập.....34
3.2.1. Mục tiêu mô phỏng.....34
3.2.2. Thu thập và xử lý dữ liệu giả lập.....34
3.2.3. Các kịch bản đánh giá 35
3.2.4. Các chỉ số đánh giá.....35
3.3. Kết quả thử nghiệm và đánh giá 35
3.3.1. Hiệu suất mô hình.....35
3.3.2. Tốc độ suy luận 36
3.3.3. Tích hợp với PLC FX3U 36
3.3.4. Độ tin cậy và khả năng chống nhiễu 36
3.4. Kết luận Chương 3.....36
KẾT LUẬN 38
DANH MỤC TÀI LIỆU THAM KHẢO.....41
PHỤ LỤC.....44

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT		
Viết tắt	**Tiếng Anh**	**Tiếng Việt**
AI	Artificial Intelligence	Trí tuệ nhân tạo
Active Learning	Active Learning	Học chủ động
Anchor-free Detection Head	Anchor-free Detection Head	Cơ chế phát hiện không dùng hộp neo
Alarm and Actuation Unit	Alarm and Actuation Unit	Khối cảnh báo và phản ứng
Batch Size	Batch Size	Kích thước lô dữ liệu
Backbone	Backbone	Mạng trích xuất đặc trưng
BMS	Building Management System	Hệ thống quản lý tòa nhà thông minh
Bounding Box	Bounding Box	Hộp giới hạn đối tượng
CLAHE	Contrast Limited Adaptive Histogram Equalization	Cân bằng histogram thích nghi giới hạn
Classification & NMS	Classification & Non-Maximum Suppression	Phân loại & loại bỏ trùng lặp
Class Probabilities	Class Probabilities	Xác suất phân lớp
CMOS	Complementary Metal-Oxide-Semiconductor	Công nghệ bán dẫn CMOS
CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
Control Logic Unit	Control Logic Unit	Khối điều khiển logic
CRC	Cyclic Redundancy Check	Mã dư vòng
CV	Computer Vision	Thị giác máy tính
DAU	Data Acquisition Unit	Khối thu thập dữ liệu
Data	Data	Dữ liệu
Decision-Making	Decision-Making	Dự đoán mức độ cháy

Debounce	Debounce	Khử nhiễu bật/tắt
Deep Learning (DL)	Deep Learning	Học sâu
DNN	Deep Neural Network	Mạng nơ-ron nhiều tầng
Edge Computing	Edge Computing	Điện toán biên
Edge Processing Unit	Edge Processing Unit	Khối xử lý biên
End-to-End	End-to-End	Độ trễ toàn hệ thống
Epochs	Epochs	Vòng huấn luyện
Event Logging	Event Logging	Ghi nhật ký hệ thống
Faster R-CNN	Faster R-CNN	Phát hiện đối tượng dựa vùng
Feature Extraction	Feature Extraction	Trích xuất đặc trưng
Firebase	Firebase	Nền tảng đồng bộ đám mây
Fire Pattern Recognition	Fire Pattern Recognition	Nhận dạng mẫu cháy
FAS	Fire Alarm System	Hệ thống báo cháy
FPS	Frames Per Second	Số khung hình/giây
Function Code	Function Code	Mã chức năng
GIoU Loss	Generalized IoU Loss	Hàm tổn thất GIoU
Global Context	Global Context	Ngữ cảnh toàn cục
GPU	Graphics Processing Unit	Bộ xử lý đồ họa
HOG	Histogram of Oriented Gradients	Biểu đồ hướng gradient
HTTP	HyperText Transfer Protocol	Giao thức siêu văn bản
IBM	International Business Machines	Tập đoàn IBM
ID	Identifier	Địa chỉ thiết bị
IoT	Internet of Things	Internet vạn vật
IoU	Intersection over Union	Tỉ lệ chồng lấp
IR	Infrared	Hồng ngoại
Jetson Nano	Jetson Nano	Thiết bị tính toán biên
KD	Knowledge Distillation	Chưng cất tri thức
Learning Rate	Learning Rate	Tốc độ học

Loss Function	Loss Function	Hàm mất mát
LTE	Long Term Evolution	Công nghệ 4G
mAP	mean Average Precision	Độ chính xác trung bình
ML	Machine Learning	Học máy
Modbus	Modbus	Giao thức Modbus
Monitoring & Data Logging	Monitoring & Data Logging	Giám sát & lưu trữ
MQTT	Message Queuing Telemetry Transport	Giao thức truyền tin nhẹ
Multi-Spectral Sensors	Multi-Spectral Sensors	Cảm biến đa phổ
Mosaic	Mosaic	Gộp ảnh
Near Real-Time	Near Real-Time	Gần thời gian thực
NMS	Non-Maximum Suppression	Loại bỏ trùng lặp
NPU	Neural Processing Unit	Bộ xử lý AI
Object Detection	Object Detection	Phát hiện đối tượng
Objectness	Objectness	Độ tin cậy đối tượng
Object Confidence Score	Object Confidence Score	Điểm tin cậy đối tượng
ONNX	Open Neural Network Exchange	Định dạng mô hình tối ưu
Optimizer	Optimizer	Bộ tối ưu hóa
Optical Flow	Optical Flow	Dòng quang học
Optical Smoke Sensor	Optical Smoke Sensor	Cảm biến khói quang
PAN-FPN	Path Aggregation Network + FPN	Mạng trộn đặc trưng
PLC	Programmable Logic Controller	Bộ điều khiển lập trình
Precision	Precision	Độ chính xác
Predictive Fire Alarm	Predictive Fire Alarm	Cảnh báo cháy dự đoán

System	System	
Pre-processing	Pre-processing	Tiền xử lý hình ảnh
Raspberry Pi 5	Raspberry Pi 5	Máy tính nhúng
Real-Time	Real-Time	Thời gian thực
Real-Time Monitoring	Real-Time Monitoring	Giám sát thời gian thực
Real-time Analytics	Real-time Analytics	Phân tích thời gian thực
Recall	Recall	Độ nhạy
Retraining	Retraining	Huấn luyện lại
ROI	Region of Interest	Vùng quan tâm
RTSP	Real-Time Streaming Protocol	Truyền phát thời gian thực
SCADA	Supervisory Control And Data Acquisition	Giám sát và thu thập dữ liệu
SIFT	Scale-Invariant Feature Transform	Đặc trưng bất biến tỉ lệ
Smart Control System	Smart Control System	Điều khiển thông minh
Smart Safety Ecosystem	Smart Safety Ecosystem	Hệ sinh thái an toàn
SSD	Single Shot MultiBox Detector	Phát hiện 1 bước
SURF	Speeded-Up Robust Features	Đặc trưng bền vững
TensorRT	TensorRT	Tối ưu mô hình sâu
TensorRT Optimization	TensorRT Optimization	Tối ưu TensorRT
Texture	Texture	Kết cấu
Thermistor	Thermistor	Cảm biến nhiệt
ThingsBoard	ThingsBoard	Nền tảng IoT
UI	User Interface	Giao diện người dùng
Unsupervised Learning	Unsupervised Learning	Học không giám sát
YOLO	You Only Look Once	Phát hiện đối tượng

DANH MỤC CÁC BẢNG

Bảng 3. 1. Kết quả thử nghiệm sau huấn luyện 35
Bảng 3. 2. Kết quả khi triển khai trên các thiết bị biên.....36

DANH MỤC HÌNH ẢNH

Hình 2. 1 Lưu đồ thuật toán hệ thống cảnh báo báo cháy.....25
Hình 2. 2 Sơ đồ khối chức năng của hệ thống báo cháy dựa trên trí tuệ nhân tạo....26
Hình 2. 3 Mô hình báo cháy.....29
.....29
Hình 2. 4 Mô hình báo cháy dựa trên AI 29

MỞ ĐẦU

1. Lý do chọn đề tài

Trong bối cảnh đô thị hóa nhanh, mật độ xây dựng cao và việc sử dụng thiết bị điện ngày càng phổ biến, nguy cơ cháy nổ trở thành vấn đề nghiêm trọng, đòi hỏi các giải pháp cảnh báo hiệu quả và kịp thời. Tại Việt Nam, số lượng các vụ cháy trong khu dân cư, nhà máy, xưởng sản xuất có xu hướng gia tăng, trong khi hệ thống cảnh báo cháy hiện nay phần lớn vẫn dựa trên cảm biến truyền thống như cảm biến khói, nhiệt hoặc khí gas. Những phương pháp này thường có độ trễ lớn, vùng phát hiện giới hạn, thiếu khả năng phân tích ngữ cảnh và khó thích ứng với môi trường phức tạp, dễ xảy ra cảnh báo giả và hoạt động kém hiệu quả trong các không gian mở hoặc môi trường bụi bẩn.

Trong khi đó, sự phát triển nhanh chóng của trí tuệ nhân tạo (AI), đặc biệt là các kỹ thuật thị giác máy tính (Computer Vision) và học sâu (Deep Learning), đã mở ra khả năng ứng dụng vào việc phát hiện sớm cháy thông qua phân tích hình ảnh từ camera giám sát. Trong số các mô hình hiện đại, YOLO (You Only Look Once) nổi bật với khả năng nhận dạng đối tượng theo thời gian thực với độ chính xác và tốc độ cao, có thể áp dụng hiệu quả trong phát hiện khói/lửa.

Vì vậy, việc nghiên cứu và phát triển hệ thống cảnh báo cháy thông minh ứng dụng AI, là cần thiết và cấp thiết. Đề tài có ý nghĩa khoa học trong việc khai thác các công nghệ AI tiên tiến phục vụ mục tiêu an toàn, và có tính khả thi cao nhờ khả năng tích hợp với các hệ thống camera hiện hữu.

2. Tổng quan về vấn đề nghiên cứu

Hiện nay, các hệ thống cảnh báo cháy truyền thống được triển khai rộng rãi với cơ chế chủ yếu dựa vào các cảm biến khói hoặc nhiệt độ. Tuy nhiên, những phương pháp này bị giới hạn về phạm vi phát hiện, độ tin cậy và khả năng hoạt động trong môi trường khắc nghiệt.

Các nghiên cứu gần đây trên thế giới đã đề xuất ứng dụng thị giác máy tính để phát hiện khói và lửa thông qua phân tích video thời gian thực. Một số mô hình như CNN, Faster R-CNN, SSD, YOLO đã được sử dụng cho mục đích này. Đặc biệt,

YOLO đã chứng minh khả năng vượt trội về tốc độ xử lý và độ chính xác trong các bài toán nhận diện đối tượng, bao gồm cả khói và ngọn lửa.

Tại Việt Nam, một số nghiên cứu bước đầu triển khai AI trong cảnh báo cháy, tuy nhiên vẫn còn thiếu các hệ thống hoàn chỉnh, chưa tích hợp hiệu quả giữa phần mềm và phần cứng, cũng như chưa có đánh giá toàn diện trong điều kiện thực tế.

Đề án này tập trung giải quyết các hạn chế hiện nay trong hệ thống cảnh báo cháy, bao gồm: nâng cao độ chính xác trong nhận diện khói và lửa, giảm độ trễ trong quá trình cảnh báo, đồng tích hợp mô hình trên nền tảng chi phí thấp, dễ triển khai và vận hành. Đây cũng chính là trọng tâm mà luận văn hướng tới.

3. Mục tiêu nghiên cứu

Mục tiêu của đề tài là nghiên cứu và phân tích nguyên lý hoạt động cũng như các hạn chế của những hệ thống cảnh báo cháy hiện nay; từ đó, trên cơ sở lý thuyết về trí tuệ nhân tạo và đặc biệt là thuật toán YOLO trong xử lý ảnh thời gian thực, xây dựng mô hình phát hiện cháy dựa trên dữ liệu hình ảnh thu từ camera kết hợp thông tin cảm biến môi trường nhằm nâng cao độ chính xác và giảm thiểu độ trễ cảnh báo. Hệ thống sẽ được đánh giá thông qua các tiêu chí định lượng như độ chính xác, thời gian phản hồi và độ tin cậy vận hành, đồng thời đề xuất các hướng ứng dụng và mở rộng trong lĩnh vực phòng cháy chữa cháy thông minh cũng như các hệ thống nhúng phục vụ giám sát an toàn và điều khiển công nghiệp.

4. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu của đề tài là hệ thống cảnh báo cháy thông minh tích hợp giải pháp học máy và trí tuệ nhân tạo, trong đó trọng tâm là mô hình phát hiện cháy dựa trên xử lý ảnh thời gian thực từ camera sử dụng thuật toán YOLO kết hợp phân tích dữ liệu đa cảm biến (khói, nhiệt độ, khí gas) nhằm nâng cao độ chính xác và giảm thiểu cảnh báo giả. Hệ thống được thiết kế theo hướng kết hợp giữa khả năng nhận thức của AI và cơ chế tự học, tự hiệu chỉnh của các thuật toán học máy để thích ứng với điều kiện môi trường khác nhau.

Phạm vi nghiên cứu:

Nghiên cứu tập trung vào việc ứng dụng mô hình học sâu YOLO trong phát hiện khói và lửa nhằm phục vụ hệ thống cảnh báo cháy và việc ứng dụng trí tuệ nhân tạo (AI) trong phát hiện cháy. Phạm vi thử nghiệm giới hạn trong mô hình thu nhỏ tại phòng thí nghiệm.

Các nội dung thực hiện bao gồm nghiên cứu lý thuyết, phát triển mô hình AI (thuật toán YOLO), tích hợp với phần cứng và cảm biến, kiểm thử trong môi trường mô phỏng, và đánh giá hiệu quả hệ thống.

5. Phương pháp nghiên cứu

Phương pháp phân tích – tổng hợp tài liệu:

Nghiên cứu lý thuyết về cảnh báo cháy, các phương pháp nhận diện khói/lửa, mô hình YOLO và các công nghệ liên quan.

Phương pháp thực nghiệm:

Thu thập các dữ liệu giả lập từ môi trường thí nghiệm nhằm tăng tính đa dạng và độ tin cậy. Gán nhãn dữ liệu và huấn luyện mô hình YOLO.

CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG CẢNH BÁO CHÁY ỨNG DỤNG TRÍ TUỆ NHÂN TẠO

1.1. Tổng quan về trí tuệ nhân tạo

1.1.1. Khái niệm và vai trò của trí tuệ nhân tạo

Trí tuệ nhân tạo (AI) là lĩnh vực khoa học nghiên cứu những cơ chế máy móc giúp có thể xử lý thông tin theo hướng thông minh, tức là không chỉ tính toán đơn thuần mà còn có khả năng học hỏi, nhận biết và đưa ra lựa chọn dựa trên dữ liệu. Ở góc nhìn hiện đại, AI được mô tả như một tập hợp các cách tiếp cận nhằm giải quyết bài toán một cách tự động cho phép hệ thống kỹ thuật tiếp nhận tín hiêu, phân tích đặc trưng, đưa ra các quy luật và thực hiện các tác vụ theo định hướng mục tiêu.

Nhiều tổ chức khoa học đã mô tả AI dưới dạng hệ thống có năng lực tiếp thu tri thức từ kinh nghiệm, diễn giải thông tin phức tạp và lựa chọn phương án thích hợp trong môi trường thay đổi. dù các diễn đạt khác nhau, các định nghĩa đều nhấn mạnh ba đặc điểm cốt lõi:

- Khả năng lọc từ dữ liệu hoặc tình huống thực tế.
- Khả năng suy luận, xây dựng mô hình biểu diễn tri thức.
- Khả năng ra quyết định, tối ưu hành vi theo mục tiêu.

Trong lĩnh vực kỹ thuật viễn thông và các hệ thống giám sát hiện đại, AI ngày càng giữ vai trò trọng yếu. Các kỹ thuật học máy và học sâu được tích hợp vào nhiều tác vụ, chẳng hạn:

- Dự báo và điều chỉnh tài nguyên mạng biến động lưu lượng
- Lọc nhiễu, phân tích tín hiệu trong điều kiện truyền phức tạp.
- Nhận biết đối tượng trong hình ảnh hoặc dữ liệu cảm biến để phát hiện sớm bất thường.

Khi được áp dụng trong báo cháy, AI hỗ trợ hệ thống đánh giá bối cảnh từ dữ liệu hình ảnh thay vì chỉ phản hồi theo tín hiệu vượt ngưỡng. Công nghệ này cho phép nhận diện sớm các dấu hiệu bất thường như khói hoặc lửa, từ đó phát cảnh báo với

độ tin cậy cao hơn. Việc này góp phần giảm thiểu báo sai và nâng cao hiệu quả quan sát tại những khu vực mà cảm biến thông thường hoạt động kém.

1.1.2. Cấu trúc và thành phần cơ bản của trí tuệ nhân tạo

Một hệ thống AI có thể được hình dung như một chu trình liên tục, trong đó ba thành phần chính vận hành đồng bộ, bổ sung lẫn nhau để tạo ra khả năng nhận thức, dự đoán và phản ứng hiệu quả. Mỗi thành phần đều đóng vai trò thiết yếu, từ việc cung cấp dữ liệu thô đến ra quyết định và hành động.

Dữ liệu là yếu tố quyết định mọi khả năng cho mô hình AI. Chúng xuất phát từ nhiều nguồn khác nhau, bao gồm hình ảnh, video, âm thanh hoặc tín hiệu cảm biến, phản ánh các trạng thái môi trường cũng như hành vi con người. Sự phong phú, chính xác và khả năng đại diện của dữ liệu quyết định trực tiếp năng lực nhận diện và phân tích của hệ thống. Trong bối cảnh cảnh báo cháy, dữ liệu chủ yếu bao gồm hình ảnh camera giám sát, tín hiệu khói, mức nhiệt và nồng độ khí CO, cung cấp toàn cảnh về nguy cơ tiềm ẩn.

Lớp xử lý thông minh là trung tâm trí tuệ của hệ thống nơi các thuật toán học máy và học sâu thực hiện việc trích xuất đặc trưng phát hiện quy luật và tạo ra các dự đoán. Các chiến lược phổ biến gồm:

- Học có giám sát: Hệ thống được huấn luyện trên dữ liệu đã được gán nhãn để phân loại hoặc dự đoán hiện tượng, ví dụ Xác định hình ảnh có chứa lửa hay không.
- Học không giám sát: Hệ thống tự phân tích để phát hiện các mối quan hệ và mẫu ẩn trong dữ liệu chưa phân loại.
- Học tăng cường (Reinforcement Learning): Mô hình cải thiện hành vi thông qua phản hồi từ môi trường, cải thiện quyết định thông qua việc phân tích các dữ liệu thực tế. Học sâu triển khai mạng nơ-ron nhiều tầng, cho phép nhận dạng các đặc trưng tinh vi và phức tạp mà các phương pháp truyền thống khó có đạt được.

Ra quyết định và thực thi hành động (Decision & Actuation): Khi dự đoán đã được hình thành, hệ thống thực hiện ra quyết định và triển khai hành động. Trong cảnh báo cháy, hành động này có thể bao gồm kích hoạt báo động, gửi cảnh báo tức thời tới các thiết bị di động, hoặc điều khiển các thiết bị chữa cháy tự động. Nhờ vậy,

AI không chỉ nhận diện nguy cơ mà còn phản ứng kịp thời, tối ưu hóa an toàn và hiệu quả vận hành.

1.1.3. Các nhánh phát triển quan trọng của trí tuệ nhân tạo

Trí tuệ nhân tạo, trong hành trình tiến hóa của nó, không phải là một thực thể đơn lẻ mà là một mạng lưới các nhánh kiến tạo, mỗi nhánh đóng vai trò như một trụ cột nâng đỡ khả năng nhận thức và phản ứng của hệ thống. Ba nhánh trọng yếu, định hình hầu hết các ứng dụng trong viễn thông và giám sát an toàn, có thể được diễn giải như sau [4]:

- Học máy (Machine Learning - ML): hoạt động như một "thầy dạy thầm lặng", cho phép hệ thống tự rút ra quy luật từ dữ liệu mà không cần được chỉ dẫn chi tiết cho từng tình huống. Nhờ khả năng hấp thụ kinh nghiệm từ dữ liệu, mô hình có thể dự đoán, phân loại hoặc đưa ra quyết định một cách linh hoạt, tựa như một nhà quan sát tinh tường học hỏi từ các hiện tượng lặp lại trong tự nhiên.

- Học sâu (Deep Learning - DL): là lớp kiến trúc chuyên sâu hơn, sử dụng các mạng nơ-ron nhiều tầng như các lớp thấu kính hội tụ, giúp "tái cấu trúc" dữ liệu phi cấu trúc - hình ảnh, video, tín hiệu phức tạp - thành các đặc trưng mà hệ thống có thể nhận diện. Phương pháp này cho phép máy móc phát hiện những mẫu tinh vi, gần như vô hình đối với con người, nâng cao khả năng dự đoán và phản ứng trong môi trường biến đổi liên tục.

- Thị giác máy tính (Computer Vision - CV): trao cho máy móc khả năng "nhìn" và "diễn giải" thế giới bên ngoài. Nó không chỉ nhận diện vật thể mà còn phân tích hành vi, theo dõi các sự kiện và phát hiện bất thường trong bối cảnh thực tế. Trong ứng dụng cảnh báo cháy, CV cho phép nhận biết khói, lửa hoặc hành vi nguy cơ từ hình ảnh và video, giống như một đôi mắt giám sát không mệt mỏi, đảm bảo phản ứng kịp thời và chính xác.

Các nghiên cứu gần đây đã chứng minh rằng các mô hình CV có thể phân loại và nhận dạng khói hoặc lửa với độ chính xác cao ngay cả khi điều kiện ánh sáng và môi trường thay đổi. Điều này nhấn mạnh tiềm năng ứng dụng rộng mở của thị giác

máy tính, biến các hệ thống cảnh báo cháy trở thành những "người bảo vệ thầm lặng", liên tục quan sát và dự đoán nguy cơ [5].

1.1.4. Mối quan hệ giữa trí tuệ nhân tạo và kỹ thuật viễn thông

Trong lĩnh vực kỹ thuật viễn thông, trí tuệ nhân tạo không còn chỉ là một module xử lý dữ liệu đơn lẻ, mà đã trở thành một thành phần tích hợp, vận hành như bộ não phân tán điều phối toàn bộ chuỗi truyền thông – xử lý – cảnh báo. AI đóng vai trò như một "trạm điều khiển thông minh", dự đoán lưu lượng, tối ưu hóa băng thông và đảm bảo tính ổn định của các liên kết dữ liệu trong các mạng IoT phân tán.

Khả năng phân tích tín hiệu đa phương tiện, từ hình ảnh và video camera đến các dữ liệu cảm biến, cho phép AI phát hiện sự cố một cách chủ động và kịp thời. Khi kết hợp với điện toán biên (Edge Computing), trí tuệ nhân tạo không cần chờ đợi dữ liệu được gửi về đám mây mà thay vào đó, nó ra quyết định ngay tại thiết bị biên, giảm thiểu độ trễ và tăng tốc phản ứng trong những tình huống khẩn cấp.

Trong các hệ thống cảnh báo cháy thông minh, AI phối hợp với các giao thức truyền thông hiện đại như MQTT, Modbus, LTE hoặc 5G, hình thành một mạng lưới giám sát phân tán, nơi dữ liệu được xử lý và phản hồi gần như tức thời. Nhờ đó, hệ thống không chỉ quan sát và phân tích, mà còn hành động kịp thời, biến các thiết bị thành những "trạm cảnh báo sống" liên tục theo dõi, dự đoán và bảo vệ an toàn.

1.1.5. Ứng dụng trí tuệ nhân tạo trong hệ thống cảnh báo cháy thông minh

Sự tích AI vào hệ thống cảnh báo cháy thông minh đã biến các cơ chế phản ứng truyền thống thành một mạng lưới giám sát năng động, có khả năng học hỏi và thích ứng liên tục [6][7]. Nhờ sức mạnh của học sâu, AI có thể phát hiện khói và ngọn lửa ngay từ những giai đoạn đầu, cung cấp cảnh báo sớm với độ chính xác cao, từ đó giảm thiểu rủi ro trước khi sự cố lan rộng và gây tổn thất nghiêm trọng.

Khả năng phân tích tinh vi của AI cho phép hệ thống phân biệt giữa các hiện tượng thật và nhiễu - khói thực sự được tách biệt khỏi bụi, hơi nước hay ánh sáng phản xạ - giảm thiểu cảnh báo giả và nâng cao độ tin cậy tổng thể. Khi kết hợp với hạ tầng IoT và các giải pháp viễn thông hiện đại, hệ thống không chỉ giám sát theo

thời gian thực mà còn có khả năng gửi thông báo tức thời tới các thiết bị di động hoặc trung tâm điều khiển, đảm bảo phản ứng nhanh chóng và hiệu quả.

Điểm đặc biệt của AI còn nằm ở khả năng mở rộng và thích ứng liên tục: các mô hình không tĩnh mà liên tục học từ dữ liệu thực tế, cải thiện hiệu suất dự đoán và độ nhạy theo thời gian. Nhờ đó, hệ thống cảnh báo cháy trở thành một "bức tường cảnh báo sống", liên tục quan sát, phân tích và phản hồi, nâng cao mức độ an toàn cho con người và cơ sở vật chất, đồng thời tạo ra một nền tảng tự động hóa thông minh vận hành linh hoạt theo môi trường và điều kiện thay đổi.

Việc ứng dụng AI trong phát hiện và cảnh báo cháy mang lại nhiều lợi ích vượt trội [6][7]:

- Phát hiện sớm và chính xác hơn: mô hình AI nhận diện khói và ngọn lửa ngay từ giai đoạn khởi phát, nhờ mô hình học sâu xử lý trực quan.

- Giảm cảnh báo giả: AI có thể phân biệt khói thật với bụi, hơi nước hoặc ánh sáng phản xạ.

- Giám sát theo thời gian thực và từ xa: Kết hợp với IoT và viễn thông giúp gửi cảnh báo tức thì tới người quản lý hoặc trung tâm điều khiển.

- Khả năng mở rộng và thích ứng: Mô hình có thể học và cải thiện theo thời gian khi thu thập thêm dữ liệu thực tế.

1.2. Nguyên lý hoạt động của hệ thống cảnh báo cháy

1.2.1. Tổng quan

Hệ thống cảnh báo cháy FAS là một mạng lưới tích hợp các thiết bị điện – điện tử, có nhiệm vụ phát hiện sớm những tín hiệu bất thường liên quan đến cháy như khói, nhiệt độ, khí độc hoặc ngọn lửa, đồng thời phát đi cảnh báo tới con người hoặc trung tâm điều khiển. Theo tiêu chuẩn National Fire Alarm and Signaling Code (NFPA 72), FAS được định nghĩa như một "mạng lưới tích hợp gồm các cảm biến, thiết bị điều khiển và hệ thống cảnh báo, có khả năng nhận diện hiện tượng cháy và thông báo kịp thời nhằm thực hiện các hành động phòng ngừa trong thời gian sớm nhất" [8].

Trong bối cảnh tiến bộ mạnh mẽ của trí tuệ nhân tạo và Internet vạn vật, hệ thống cảnh báo cháy truyền thống đang dần được nâng cấp thành các hệ thống cảnh báo cháy thông minh AIoT. Mô hình này kết hợp khả năng nhận diện hình ảnh qua Computer Vision và phân tích dữ liệu thời gian thực, cho phép hệ thống không chỉ phát hiện mà còn đánh giá và phản ứng tự động trước nguy cơ cháy, nâng cao đáng kể độ chính xác và tốc độ xử lý.

Cụ thể, giải pháp cảnh báo cháy vận hành dựa trên AI (AI-based Fire Alarm System) vận dụng sự kết hợp giữa thị giác máy tính và hệ thống điều khiển – giám sát thông minh (Smart Control System). Nhờ đó, việc nhận diện khói và ngọn lửa không còn phụ thuộc duy nhất vào các cảm biến nhiệt hay khói truyền thống, mà được hỗ trợ bởi phân tích hình ảnh động, cho phép phát hiện sớm và chính xác các hiện tượng cháy nổ. Theo báo cáo của NFIRE Global 2025, các hệ thống AI có khả năng phát hiện khói và lửa sớm hơn trung bình 3–5 giây so với cảm biến khói quang điện trong môi trường thử nghiệm kín, đồng thời giảm tỷ lệ cảnh báo giả tới 60% nhờ khả năng học ngữ cảnh và phân tích động.

Nguyên lý vận hành dựa trên xử lý hình ảnh thời gian thực các đặc trưng đặc thù của lửa và khói thông qua mô hình học sâu, sau đó đưa ra quyết định điều khiển cảnh báo hoặc kích hoạt thiết bị chữa cháy tự động. Toàn bộ quy trình này diễn ra trong chu kỳ xử lý từ 100–300 ms, đảm bảo phản ứng gần như tức thời, phù hợp với các tình huống khẩn cấp, nơi mỗi mili-giây đều quan trọng.

1.2.2. Cấu trúc và các thành phần chính

Một hệ thống cảnh báo cháy hiện đại ứng dụng trí tuệ nhân tạo thường được tổ chức thành năm khối chức năng chính, vận hành liên kết như các tầng cảm giác, xử lý và phản ứng của một sinh thể thông minh.

Khối thu nhận dữ liệu (Data Acquisition Unit)

Khối này đóng vai trò như "giác quan thị giác" của hệ thống, thu thập và mã hóa các tín hiệu từ môi trường giám sát. Các nguồn dữ liệu bao gồm:

Camera IP, camera hồng ngoại hoặc module CMOS độ phân giải cao (ví dụ: IMX219, OV5647).

Tín hiệu video được ghi nhận ở định dạng khung hình (frame) 24–30 fps, có thể truyền tải qua RTSP, HTTP hoặc MQTT video stream.

Một số hệ thống nâng cao tích hợp cảm biến đa phổ (multi-spectral sensors) để đo thêm thông tin nhiệt độ hồng ngoại và mật độ khói, giúp phát hiện bước sóng đặc trưng của ngọn lửa.

Dữ liệu sau khi thu thập được truyền đến khối xử lý thông minh, nơi tiến hành phân tích và nhận dạng các mẫu cháy với độ trễ thấp, đảm bảo tính liên tục và độ tin cậy trong giám sát.

Khối xử lý biên (Edge Processing Unit)

Đây là "trái tim trí tuệ" của hệ thống, nơi các mô hình học sâu như YOLOv8 hoặc YOLOv7-tiny triển khai các mạng nơ-ron tích chập (CNN) tối ưu cho nhận diện thời gian thực [10]. Các nền tảng phần cứng phổ biến bao gồm NVIDIA Jetson Nano/Xavier, Raspberry Pi 5, hoặc STM32H7 tích hợp NPU, chạy thư viện TensorRT/ONNX Runtime.

Các tác vụ chính của khối xử lý bao gồm:

▪ Tiền xử lý hình ảnh (resizing, normalization, augmentation).

▪ Trích xuất đặc trưng đa tỉ lệ qua backbone CSPDarknet53 hoặc EfficientNet**.**

▪ Dự đoán bounding box và xác suất phân lớp (fire/smoke/no-fire).

Đầu ra của mô hình được biểu diễn như sau [11]:

$$Output = \{(x_i\ , y_i,\ \omega_i, h_i, c_i, p_i)\}_{i=1}^{N}$$

trong đó:

- $(x_i\ , y_i)$ toạ độ tâm đối tượng i;
- (ω_i, h_i): chiều rộng và chiều cao của hộp bao;
- c_i: độ tin cậy (confidence score);
- p_i: xác suất đối tượng thuộc lớp (fire/smoke/normal);
- θ: tham số của mô hình đã huấn luyện.

Hàm mất mát (Loss Function) được xác định như sau [12]:

$$L = \lambda_1 L_{box} + \lambda_2 L_{obj} + \lambda_3 L_{cls} \quad (1.1)$$

Với L_{box} là sai số định vị (Localization Loss); L_{obj} là sai số phát hiện đối tượng (Objectness Loss) và L_{cls} là sai số phân loại (Classification Loss). Khi xác suất $p_i > 0.8$, hệ thống nhận diện khả năng cháy và gửi tín hiệu cảnh báo, đạt tốc độ xử lý trung bình 20-30 khung hình/giây [13].

Khối logic điều khiển (Control Logic Unit)

Khối logic điều khiển vận hành như "bộ não phản xạ", kết nối mô hình AI với phần cứng thực thi. Tín hiệu từ mô hình (mức logic cao/low) được truyền tới vi điều khiển (STM32, ESP32) hoặc PLC, thực hiện các nhiệm vụ:

- Xử lý tín hiệu điều kiện (debouncing, timing) để tránh báo giả.

- Kích hoạt đầu ra điều khiển (relay, transistor, bus RS485) cho còi, đèn, hoặc hệ thống chữa cháy tự động.

- Gửi gói tin MQTT hoặc HTTP POST đến máy chủ IoT, ví dụ với chuẩn Modbus RTU: [ID] [Function] [Data] [CRC] [14]. Khối này đảm bảo khả năng mở rộng, tương thích và tích hợp liền mạch với hệ thống tự động hóa công nghiệp.

Khối cảnh báo và phản ứng (Alarm and Actuation Unit)

Đây là tầng thực thi, trực tiếp vận hành các lệnh từ khối điều khiển, bao gồm:

- Thiết bị ngoại vi: còi hú, đèn cảnh báo.

- Thiết bị tự động: van xả CO_2, hệ thống phun nước chữa cháy.

- Kích hoạt đồng thời cảnh báo cục bộ và từ xa qua Internet (Firebase hoặc ThingsBoard).

Khối giám sát & lưu trữ (Monitoring and Data Logging)

Tất cả dữ liệu phát hiện và cảnh báo được lưu trữ dưới dạng ảnh, video, thời gian phát hiện và trạng thái cảnh báo. Các nền tảng phổ biến bao gồm Firebase, ThingsBoard hoặc AWS IoT Core, cho phép:

- Theo dõi thời gian thực (real-time monitoring).

- Thống kê tần suất cảnh báo và biểu đồ phân bố thời gian.

- Phân tích hiệu suất mô hình và cập nhật phiên bản học máy mới.

Dữ liệu này đồng thời phục vụ quá trình huấn luyện lại, cải thiện độ chính xác và thích ứng theo thời gian, biến mỗi chu trình vận hành một vòng tuần hoàn học hỏi liên tục.

1.2.3. Nguyên lý hoạt động tổng quát

Nguyên lý vận hành của hệ thống cảnh báo cháy AI dựa trên một quá trình lặp lại liên tục, trong đó từng khối chức năng phối hợp nhịp nhàng, biến mỗi tín hiệu từ môi trường thành quyết định và hành động gần như tức thời. Quy trình này được mô tả qua các bước liên tiếp:

- Thu nhận hình ảnh: Các camera giám sát ghi nhận toàn bộ khu vực theo dõi, truyền dữ liệu video liên tục tới khối xử lý biên. Khối này vận hành như "giác quan thị giác" của hệ thống, đảm bảo rằng mọi thay đổi trong môi trường được ghi nhận với độ chi tiết và liên tục cao.

- Phân tích thông minh: Tại đây, mô hình YOLO thực hiện xử lý từng khung hình, trích xuất các đặc trưng của lửa và khói dựa trên thông số màu sắc (RGB, HSV), kết cấu bề mặt (texture) và dòng quang học (optical flow). Qua việc nhận diện các mẫu đặc trưng, hệ thống phân biệt rõ ràng hiện tượng cháy thực với các yếu tố nhiễu như ánh sáng phản chiếu hay khói giả.

- Ra quyết định: Khi xác suất nhận diện cháy vượt ngưỡng đã được xác định, mô hình gửi tín hiệu cảnh báo tới Khối logic điều khiển. Khối này hoạt động như "bộ não phản xạ", xử lý các tín hiệu logic để đảm bảo quyết định đưa ra chính xác và ổn định, tránh cảnh báo sai.

- Kích hoạt cảnh báo: Bộ điều khiển trung gian điều khiển các thiết bị báo cháy hoạt động vật lý như còi hú, đèn nhấp nháy, đồng thời truyền thông báo lên các nền tảng IoT để cảnh báo từ xa, đảm bảo phản ứng đồng bộ cả trong hệ thống cục bộ và giám sát từ xa.

- Ghi nhận và phản hồi: Toàn bộ dữ liệu liên quan, bao gồm hình ảnh, thời gian phát hiện và trạng thái cảnh báo, được ghi vào cơ sở dữ liệu và trình bày trên giao diện (UI) để phục vụ việc theo dõi, đánh giá và phân tích thống kê. Quá trình

này cũng tham gia vào việc tái huấn luyện mô hình để nâng cao hiệu suất và độ chính xác theo thời gian.

- Tổng khoảng thời gian từ khi dữ liệu hình ảnh được ghi nhận đến khi kích hoạt cảnh báo dao động trong khoảng 100–300 ms, đảm bảo phản ứng gần như thời gian thực [15]. Điều này cho phép hệ thống thích ứng hiệu quả với các môi trường giám sát quan trọng như nhà máy, kho chứa hoặc các công trình công cộng, nơi mỗi mili-giây đều có ý nghĩa sống còn.

1.2.4. Hiệu quả và xu hướng ứng dụng

Việc tích hợp trí tuệ nhân tạo vào hệ thống cảnh báo cháy không chỉ nâng cao khả năng phát hiện, mà còn biến mỗi hệ thống thành một "thực thể giám sát tự học", vận hành chủ động và thích ứng với môi trường [16][17]. Các lợi ích nổi bật bao gồm:

Phát hiện sớm và chính xác hơn: AI thiểu sai số trong nhận diện khói, đặc biệt trong những môi trường có bụi, hơi nước hay ánh sáng nhiễu, từ đó cung cấp cảnh báo gần như tức thời trước khi cháy lan rộng.

Phân loại bối cảnh linh hoạt: Hệ thống có khả năng học và thích ứng với các điều kiện biến đổi như ánh sáng, gió hay nền môi trường khác nhau, từ đó tối ưu độ tin cậy của cảnh báo.

- Khả năng hoạt động độc lập (Edge AI): Nhờ xử lý trực tiếp tại thiết bị biên, hệ thống không phụ thuộc vào kết nối Internet, thích hợp với các môi trường công nghiệp, nhà máy hoặc kho chứa, nơi độ trễ và tính liên tục của dữ liệu quyết định hiệu quả phản ứng.

- Kết nối thông minh (AIoT): Sự liên kết với các nền tảng như Firebase, ThingsBoard hoặc AWS IoT Core cho phép giám sát, ghi nhận và điều khiển từ xa, đồng thời tích hợp dữ liệu cho các thuật toán học máy cải thiện độ chính xác theo thời gian.

- Theo nghiên cứu của MDPI Applied Sciences (2025), các hệ thống cảnh báo cháy triển khai mô hình YOLOv8 trên thiết bị biên đạt tốc độ xử lý trung bình 25–30 fps, đồng thời tiết kiệm khoảng 40% năng lượng so với phương pháp truyền dữ liệu lên đám mây [16].

- Xu hướng hiện nay được xác định bởi mô hình AIoT Fire Safety, nơi học sâu kết hợp với kết nối IoT, tạo ra khả năng dự đoán và phản ứng theo thời gian thực. Hệ thống cảnh báo cháy tiên đoán (Predictive Fire Alarm System) không chỉ nhận diện hiện tượng cháy mà còn dự đoán nguy cơ thông qua phân tích các hành vi bất thường (anomaly detection) trong dữ liệu hình ảnh và cảm biến [17], mở ra một kỷ nguyên mới cho giám sát an toàn chủ động và thông minh.

1.3. Thuật toán YOLO trong phát hiện đối tượng

Thuật toán YOLO (You Only Look Once) thuộc nhóm mô hình phát hiện đối tượng thế hệ mới, nổi bật nhờ khả năng xử lý khung hình với tốc độ rất cao mà vẫn giữ được độ tin cậy trong nhận dạng. Thay vì tách bài toán thành hai giai đoạn như các hệ thống trước đây—tạo vùng nghi vấn rồi mới tiến hành phân loại—YOLO gộp toàn bộ quá trình thành một phép dự đoán duy nhất trên toàn ảnh.

Phương pháp này chia ảnh thành lưới S×S và tại mỗi ô, mạng sẽ ước lượng vị trí hộp giới hạn, mức độ tin cậy và xác suất thuộc về từng lớp mục tiêu. Cách thiết kế gọn và nhất quán này giúp mô hình nắm được bối cảnh toàn cục của hình ảnh, hạn chế hiện tượng phát hiện sai khi các đối tượng có hình dạng hoặc kích thước thay đổi.

Tốc độ xử lý vượt trội là ưu điểm quan trọng khiến YOLO rất phù hợp với các hệ thống cần phản hồi gần như tức thì. Trong bài toán cảnh báo cháy bằng camera, thời gian trễ có thể làm giảm hiệu quả ứng phó. Do đó, việc áp dụng thuật toán này cho phép hệ thống nhận diện nhanh các dấu hiệu như vùng lửa, cột khói hoặc ánh sáng bất thường, từ đó kích hoạt cảnh báo kịp thời và giảm thiểu nguy cơ lan rộng.

1.3.1. Nguyên lý hoạt động

Thuật toán YOLO vận hành dựa trên ba bước chính, kết hợp xử lý đặc trưng, dự đoán khung giới hạn và cơ chế loại bỏ trùng lặp nhằm tối ưu độ chính xác trong phát hiện đối tượng.

Hình ảnh đầu vào được chia thành các ô lưới, sau đó truyền qua một mạng nơ-ron tích chập sâu (Deep CNN). Mạng này đóng vai trò trích xuất đặc trưng không gian – bao gồm biên, hình dạng, màu sắc và cấu trúc – giúp mô hình phân biệt các đối tượng có hình thái phức tạp như khói và ngọn lửa.

Những kiến trúc thường được dùng làm backbone gồm Darknet-53, CSPDarknet, ConvNeXt hoặc các biến thể mới hơn trong họ YOLOv8. Quá trình ánh xạ đặc trưng có thể biểu diễn bằng mô hình toán học:

$$F = f_\theta (I) \quad (1.2)$$

trong đó:

- I: ảnh đầu vào có kích thước H x W;
- F: ma trận đặc trưng đầu ra;
- F_θ: mạng CNN với tập tham số θ.

Lớp backbone chịu trách nhiệm tổng hợp các yếu tố như vùng màu bất thường, mép biên dao động của lửa hoặc hoa văn khói loãng, là cơ sở cho bước dự đoán tiếp theo.

Dự đoán hộp giới hạn và xác suất đối tượng Sau khi thu được đặc trưng, mỗi ô trong lưới sẽ sinh ra một số lượng khung dự đoán (thường 3–5). Mỗi khung bao gồm năm giá trị cơ bản:

- *(x, y)*: vị trí tâm tương đối của hộp,
- *(w, h)*: kích thước hộp được chuẩn hóa theo toàn ảnh,
- *C*: mức độ tin cậy cho biết khả năng tồn tại đối tượng.

Dự đoán được biểu diễn dưới dạng một vector mở rộng:

$$P = (x, y, w, h, C, p_1, p_2, \ldots, p_k) \quad (1.3)$$

Ở đó p_i là xác suất đối tượng thuộc lớp thứ i trong tập k lớp. [18] Cách biểu diễn này cho phéo mô hình đồng thời xác định vị trí – hình dạng và phân loại đối tượng chỉ qua một lần suy luận.

Phân loại và loại bỏ các khung dư thừa bằng NMS: Với mỗi khung chứa, mô hình dự đoán các xác suất lớp (class probabilities). Để giữ lại duy nhất các kết quả có độ tin cậy cao, thuật toán Non-Maximum Suppression (NMS) được áp dụng.

Mức độ trùng nhau giữa hai hộp được đánh giá thông qua chỉ số IoU:

$$IoU = \frac{Area\,(B_{pred} \cap B_{gt})}{Area\,(B_{pred} \cup B_{gt})} \quad (1.4)$$

Nếu hai hộp IoU > τ (thường khoảng 0.5), hộp có độ tin cậy thấp hơn sẽ bị loại bỏ [19]. Cơ chế này được thực hiện trên toàn ảnh, giúp mô hình bảo toàn bối cảnh

tổng thể và giảm các lỗi nhận dạng do nhiễu ánh sáng, phản xạ hoặc vật thể tương tự ngọn lửa.

Nhờ quy trình xử lý đặc trưng theo hướng toàn cục, dự đoán vị trí theo một bước duy nhất và loại bỏ trùng lặp bằng NMS, YOLO đạt được tốc độ rất cao mà vẫn giữ ổn định về độ chính xác. Đây là lý do thuật toán đặc biệt phù hợp cho hệ thống cảnh báo cháy thời gian thực, nơi yêu cầu phản hồi nhanh đóng vai trò quyết định.

1.3.2. Ưu điểm kỹ thuật của YOLO

- Thuật toán YOLO được đánh giá cao nhờ khả năng kết hợp tốc độ và độ chính xác trong cùng một mô hình, vượt trội so với các phương pháp phát hiện đối tượng truyền thống. Những ưu điểm nổi bật gồm:

- Khả năng xử lý thời gian thực: YOLO chỉ cần một lần suy luận cho toàn bộ ảnh đầu vào, nhờ vậy tốc độ xử lý đạt mức rất cao. Trên các GPU phổ thông, tốc độ có thể dao động từ 30 đến trên 100 FPS**,** đáp ứng yêu cầu theo dõi liên tục trong các hệ thống an toàn cháy, nơi mỗi giây có thể cần phân tích hàng chục khung hình. Cơ chế xử lý "một lượt duy nhất" giúp giảm đáng kể độ trễ tổng thể.

- Nhận diện và định vị đối tượng trong cùng một bước: Các mô hình như R-CNN hoặc Faster R-CNN phải trải qua nhiều pha xử lý, bao gồm sinh vùng đề xuất và phân loại từng vùng. Ngược lại, YOLO tiếp cận theo kiến trúc one-stage detector, tức mô hình vừa xác định vị trí vừa phân loại đối tượng ngay trong một lần tính toán. Lợi ích mang lại gồm: Đơn giản hóa, đảm bảo độ trễ thấp hơn, duy trì ổn định khi xuất hiện nhiều đối tượng cùng khung hình.

- Khả năng tái huấn luyện linh hoạt: Mô hình YOLO có thể được tinh chỉnh với các bộ dữ liệu thu thập riêng theo từng mục đích, chẳng hạn như ảnh lửa và khói trong môi trường nhà xưởng, kho lưu trữ hay khu vực ngoài trời. Nhờ vậy, hệ thống dễ dàng thích ứng với điều kiện ánh sáng, thời tiết hoặc mức độ nhiễu nền khác nhau.

- Tối ưu cho thiết bị biên và hệ thống nhúng: Các bản YOLO gần đây như YOLOv7, YOLOv8 hay YOLOv9-Nano được thiết kế để triển khai hiệu quả trên thiết bị biên thông qua các công cụ tối ưu như TensorRT, OpenVINO, ONNX Runtime hoặc TFLite. Điều này cho phép mô hình chạy trên các nền tảng hạn chế tài nguyên

như Jetson Nano, Xavier NX, Raspberry Pi 5, EdgeTPU hoặc vi điều khiển STM32 có tích hợp NPU. Theo số liệu kỹ thuật của Ultralytics (2024), cấu hình YOLOv8-Tiny có thể đạt thời gian xử lý dưới 50 ms mỗi khung hình trên Jetson Nano, phù hợp cho ứng dụng cảnh báo cháy thời gian thực.

1.3.3. Ứng dụng YOLO trong hệ thống cảnh báo cháy

Trong khuôn khổ đề tài, thuật toán YOLO được lựa chọn làm mô hình lõi để nhận diện các dấu hiệu cháy dựa trên phân tích hình ảnh. Hai nhóm đối tượng chính gồm khói và ngọn lửa, được mô hình phát hiện theo thời gian thực thông qua chuỗi quy trình huấn luyện - đánh giá - triển khai như sau.

Thu thập dữ liệu và gán nhãn: Nguồn dữ liệu được xây dựng từ nhiều bối cảnh khác nhau như nhà xưởng, hành lang, kho chứa, khu vực dân cư hoặc môi trường ngoài trời. Các khung hình thu thập từ camera giám sát được gán nhãn thủ công bằng những công cụ như LabelImg**,** Roboflow hoặc AVA. Trong quá trình gán nhãn, từng vùng chứa khói, lửa hoặc tình huống không có cháy (no_fire) được đánh dấu chính xác nhằm tạo ra bộ dữ liệu huấn luyện nhất quán và có khả năng tổng quát tốt.

Huấn luyện mô hình: Bộ dữ liệu sau khi xử lý được đưa vào huấn luyện trên phần cứng có hỗ trợ GPU như NVIDIA RTX hoặc Jetson. Mô hình được tối ưu bằng các thuật toán như Adam hoặc SGD, kết hợp điều chỉnh tốc độ học theo dạng cosine scheduler**.** Kích thước lô huấn luyện (batch size) thường nằm trong khoảng 16–32 để cân bằng tốc độ hội tụ và tiêu thụ bộ nhớ.

Hàm mất mát tổng quát của YOLO có dạng:

$$L = L_{box} + L_{obj} + L_{cls} \quad (1.5)$$

trong đó thành phần định vị L_{box} được sử dụng IoU hoặc GIoU nhằm tăng độ chính xác khi mô hình ước lượng hộp chứa [22]. Cấu trúc này cho phép mô hình đồng thời cải thiện khả năng nhận dạng và khả năng định vị đối tượng trong ảnh.

Đánh giá hiệu suất mô hình: Sau giai đoạn huấn luyện, mô hình được kiểm định bằng nhiều chỉ số hiệu năng:

- *Precision* đạt khoảng ≥ 94%, cho thấy khả năng hạn chế báo sai;
- *Recall* đạt ≥ 92%, đảm bảo mô hình không bỏ sót khói hoặc lửa;

- *mAP xấp xỉ* 95%, phản ánh chất lượng ước lượng trên toàn bộ tập kiểm thử;

- *Thời gian suy luận dưới* 50 ms mỗi khung hình trên Jetson Nano, phù hợp yêu cầu thời gian thực.

Những chỉ số này đáp ứng tiêu chuẩn vận hành của một hệ thống cảnh báo cháy thông minh yêu cầu độ tin cậy cao.

Triển khai trên thiết bị nhúng: Mô hình hoàn chỉnh được chuyển sang định dạng tối ưu như ONNX hoặc TensorRT để giảm độ trễ và tăng tốc độ suy luận. Các định dạng này phù hợp khi triển khai trên thiết bị biên như Jetson Nano, Raspberry Pi 5, Intel NUC hoặc các nền tảng công nghiệp tương đương. Tại thiết bị xử lý biên, YOLO đóng vai trò "bộ phân tích hình ảnh", liên tục đọc dữ liệu từ camera, thực hiện nhận diện và truyền tín hiệu cảnh báo đến hệ điều khiển trung gian (vi điều khiển hoặc PLC) thông qua những giao thức như Modbus RTU/TCP, GIOP hoặc MQTT.

Theo nghiên cứu của *MDPI Applied Sciences (2025)*, phiên bản YOLOv8-tiny trên nền tảng Jetson Nano, tốc độ xử lý đạt 27–30 FPS**,** tiêu thụ khoảng 10 W, mAP@0.5 đạt 95,3% khi phát hiện khói/lửa trong điều kiện thực nghiệm [23]. Những kết quả này chứng minh YOLO hoàn toàn phù hợp cho hệ thống cảnh báo cháy thời gian thực, nơi độ ổn định và độ tin cậy là yếu tố then chốt.

1.4. Các yếu tố ảnh hưởng đến hiệu quả hệ thống cảnh báo cháy

Hiệu quả vận hành của một hệ thống cảnh báo cháy ứng dụng trí tuệ nhân tạo phụ thuộc vào sự phối hợp chặt chẽ giữa phần cứng thu nhận dữ liệu, mô hình học sâu, chất lượng bộ dữ liệu, điều kiện môi trường và độ ổn định của hạ tầng truyền thông. Những yếu tố này tác động trực tiếp đến khả năng phát hiện sớm, độ chính xác của mô hình cũng như tính liên tục của quá trình giám sát. Việc hiểu rõ các thành phần ảnh hưởng giúp tối ưu hóa hiệu năng toàn hệ thống, giảm thiểu cảnh báo sai và nâng cao độ tin cậy trong các môi trường triển khai khác nhau.

1.4.1. Chất lượng dữ liệu đầu vào

Chất lượng khung hình thu được từ camera đóng vai trò quyết định đối với khả năng mô hình AI phân biệt được khói và lửa trong môi trường thực tế. Khi hình ảnh có độ phân giải thấp, góc quan sát bị hạn chế hoặc tầm nhìn bị vật cản che khuất,

các đặc trưng thị giác của đám cháy trở nên mờ nhạt, từ đó làm giảm độ tin cậy của thuật toán. Kết quả thực nghiệm của Nguyen & Tran (2023) cho thấy mô hình YOLOv5 đạt mức chính xác gần 94,7% với nguồn dữ liệu Full HD (1920×1080), nhưng tỷ lệ này giảm còn khoảng 82,5% khi hạ xuống chuẩn 720p [24]. Điều này chứng minh độ phân giải có ảnh hưởng trực tiếp đến hiệu quả nhận diện.

Bên cạnh độ phân giải, điều kiện chiếu sáng cũng tạo ra sai lệch đáng kể. Trong bối cảnh ánh sáng yếu, làn khói dễ bị nhầm lẫn với bóng tối, hơi ẩm hoặc vùng thiếu sáng; còn trong trường hợp ánh sáng mạnh, bề mặt phản chiếu có thể gây báo động sai. Vì vậy, bước tiền xử lý đóng vai trò quan trọng nhằm cân bằng lại đặc tính quang học của ảnh. Các kỹ thuật như hiệu chỉnh cân bằng trắng, tối ưu độ tương phản hoặc áp dụng CLAHE (Contrast Limited Adaptive Histogram Equalization) giúp tăng độ ổn định của dữ liệu trước khi đưa vào mô hình học sâu, đồng thời cải thiện khả năng phân biệt giữa tín hiệu cháy thật và các nhiễu không mong muốn [25].

1.4.2. Cấu hình và hiệu suất mô hình trí tuệ nhân tạo

Hiệu quả vận hành của hệ thống cảnh báo cháy sử dụng thị giác máy tính phụ thuộc đáng kể vào lựa chọn mô hình học sâu và cấu hình thuật toán. Các dòng mô hình YOLO (như YOLOv5, YOLOv7 và YOLOv8) thường cho tốc độ xử lý ảnh và mức độ chính xác tốt hơn các thế hệ cũ, với khả năng rút ngắn thời gian suy luận lên khoảng 30–45% và giảm sai lệch mAP thêm 10–15% so với YOLOv3 [26].

Khi triển khai trên nền tảng phần cứng có tài nguyên giới hạn như Jetson Nano, Raspberry Pi hoặc các thiết bị nhúng dùng VPU/NPU, việc tối ưu mô hình là bắt buộc để đảm bảo hệ thống duy trì tốc độ khung hình ổn định. Một số kỹ thuật thường áp dụng gồm lượng tử hóa (như giảm độ chính xác tính toán xuống 8 bit hoặc 16 bit), cắt giảm tham số (pruning), distillation giữa các mô hình và tăng tốc bằng TensorRT. Báo cáo hiệu năng Jetson năm 2024 cho thấy phiên bản YOLOv5s sau khi được lượng tử hóa có thể đạt mức 22–25 FPS trên Jetson Nano và vẫn giữ độ chính xác khoảng 91% [27].

1.4.3. Năng lực xử lý của phần cứng và độ trễ toàn hệ thống

Hiệu suất thời gian thực của hệ thống cảnh báo cháy phụ thuộc mạnh vào cấu hình phần cứng. Các thiết bị được tích hợp bộ xử lý song song như GPU hoặc NPU (ví dụ Jetson Xavier NX, Google Coral Edge TPU) có thể tăng tốc độ suy luận lên gấp 2–4 lần so với các bộ xử lý chỉ dùng CPU [28].

Ngoài khả năng tính toán, tổng thời gian trễ từ lúc camera thu hình đến khi hoàn tất xử lý và phát tín hiệu cảnh báo (end-to-end latency) cũng là tham số quan trọng. Để đáp ứng yêu cầu phản ứng khẩn cấp, độ trễ này cần được giữ ở mức dưới 500 ms. Việc sử dụng các giao thức truyền dữ liệu tối ưu như RTSP hoặc MQTT — vốn hỗ trợ cơ chế truyền song song — giúp giảm thời gian trễ xuống khoảng 200–250 ms khi mạng hoạt động ổn định [29].

1.4.4. Ảnh hưởng của môi trường vận hành

Điều kiện thực tế tại khu vực lắp đặt có tác động trực tiếp đến mức độ ổn định của hệ thống cảnh báo cháy. Các yếu tố như nhiệt độ tăng cao, độ ẩm lớn, bụi mịn trong không khí hoặc sự thay đổi mạnh của ánh sáng tự nhiên có thể làm giảm độ nhạy của cảm biến và dẫn đến số lượng cảnh báo sai tăng lên. Theo tiêu chuẩn ISO 7240-29:2017, những hệ thống được sử dụng trong môi trường công nghiệp cần được kiểm tra và hiệu chuẩn định kỳ để đảm bảo camera không bị bám bụi và duy trì khả năng quan sát ổn định [30].

Ngoài yêu cầu về thiết bị, mô hình trí tuệ nhân tạo cũng cần được bổ sung dữ liệu huấn luyện phù hợp với môi trường triển khai (domain-specific dataset) nhằm nâng cao khả năng phân biệt giữa khói thật và các hiện tượng dễ gây nhầm lẫn như hơi nước, ánh sáng phản xạ, bụi hoặc các vùng sáng có màu sắc tương tự ngọn lửa. Một số nghiên cứu công bố trên ScienceDirect (2004) chỉ ra rằng độ chính xác của mô hình có thể tăng thêm 8–12% khi được huấn luyện bằng bộ dữ liệu thu thập trực tiếp tại hiện trường [31].

1.4.5. Kết nối mạng và nền tảng IoT

Đối với hệ thống cảnh báo cháy tích hợp IoT, độ ổn định của hạ tầng mạng và khả năng bảo vệ dữ liệu đóng vai trò then chốt trong việc truyền tải tín hiệu cảnh báo.

Các giao thức như MQTT phiên bản 5.0 hoặc HTTPS sử dụng mã hóa TLS 1.3 được khuyến nghị áp dụng để đảm bảo an toàn thông tin. Cơ chế dự phòng truyền dữ liệu như retry, buffering, cùng phương pháp mã hóa đầu–cuối góp phần hạn chế tình trạng thất lạc thông tin khi đường truyền bị gián đoạn và giảm nguy cơ giả mạo gói tin cảnh báo. Theo báo cáo của IoT Security Foundation (2024), việc chuẩn hóa mã hóa TLS 1.3 giúp giảm hơn 70% các hình thức tấn công nhằm vào thiết bị IoT [32].

Mô hình triển khai phổ biến hiện nay là kiến trúc Edge - Cloud kết hợp, trong đó thiết bị xử lý tại biên (Edge AI) đảm nhiệm việc phân tích hình ảnh để giảm độ trễ, trong khi nền tảng đám mây đảm đương chức năng lưu trữ, phân tích dữ liệu và quản lý tập trung. Cách tiếp cận này đồng thời đảm bảo tốc độ phản hồi nhanh, tính ổn định và mức độ an toàn cao cho toàn bộ hệ thống cảnh báo cháy.

1.5. Kết luận chương 1

Chương 1 đã khái quát các nguyên lý nền tảng liên quan đến hệ thống cảnh báo cháy ứng dụng trí tuệ nhân tạo. Nội dung tập trung vào ba nhóm chính: cách thức hoạt động của AI trong nhận diện hình ảnh, vai trò của thuật toán YOLO và các yếu tố ảnh hưởng đến hiệu năng hệ thống. Kết quả tổng hợp cho thấy việc ứng dụng mô hình học sâu vào nhận diện khói và lửa giúp tăng khả năng phát hiện sớm, giảm tỷ lệ báo sai và nâng cao tính ổn định của hệ thống so với giải pháp cảm biến truyền thống. Hiệu quả đó phụ thuộc vào chất lượng dữ liệu, cấu hình mô hình, năng lực phần cứng và điều kiện môi trường vận hành.

CHƯƠNG 2: GIẢI PHÁP CẢNH BÁO CHÁY DỰA TRÊN MÔ HÌNH HỌC SÂU VÀ TRÍ TUỆ NHÂN TẠO

2.1. Cơ sở lựa chọn giải pháp

2.1.1. Bối cảnh khoa học và thực tiễn

Trong nhiều năm, phần lớn hệ thống báo cháy vẫn vận hành dựa trên các cảm biến vật lý như cảm biến khói quang, cảm biến nhiệt độ, cảm biến khí CO hay đầu dò hồng ngoại. Những thiết bị này hoạt động tốt trong không gian khép kín, nhưng dễ bị giảm hiệu quả ở môi trường rộng, có luồng gió mạnh hoặc nhiều tạp chất trong không khí khiến nồng độ khói hoặc nhiệt độ phân tán nhanh.

Báo cáo của NFPA (2024) ghi nhận rằng một lượng lớn sự cố cháy lớn xuất phát từ việc hệ thống báo cháy không phát hiện sớm dấu hiệu nguy hiểm. Số liệu thống kê tại Việt Nam (Bộ Công an, 2023) cũng cho thấy phần lớn thiệt hại về người xảy ra do tín hiệu báo cháy tới muộn hoặc không hoạt động do nhiễu môi trường.

Những hạn chế này dẫn tới yêu cầu cần một hướng tiếp cận mới — hệ thống có khả năng nhận diện trực quan dấu hiệu khói và ngọn lửa, làm việc liên tục trong nhiều điều kiện ánh sáng, bụi hoặc gió. Sự phát triển mạnh mẽ của AI, đặc biệt là học sâu trong xử lý ảnh, đã mở ra khả năng phát hiện cháy ngay từ khi xuất hiện các biểu hiện quang học ban đầu, vượt trội so với các cảm biến truyền thống vốn phụ thuộc vào sự thay đổi vật lý.

2.1.2. Cơ sở công nghệ - sự phát triển của học sâu và thị giác máy tính

Học sâu và thị giác máy tính đã làm thay đổi hoàn toàn cách thức phân tích hình ảnh. Các mô hình CNN có thể tự động rút trích đặc trưng từ ảnh, thay thế cho những phương pháp trích xuất thủ công trước đây như SIFT, SURF hay HOG.

Trong dòng các mô hình phát hiện đối tượng, họ mô hình YOLO đã trở thành lựa chọn hàng đầu nhờ tốc độ xử lý nhanh. Thay vì phân tích từng vùng như các phương pháp R-CNN hay SSD, YOLO chỉ cần một lần suy luận cho cả ảnh, nhờ đó giảm đáng kể thời gian xử lý.

Phiên bản YOLOv8 mang lại nhiều cải tiến nổi bật:

- Backbone dạng CSP – Darknet tối ưu giúp nhận diện đặc trưng đa cấp độ.
- PAN-FPN cho khả năng kết hợp đặc trưng tốt hơn giữa các tầng.
- Cơ chế anchor-free giảm tham số và tăng tốc trên thiết bị nhúng.

Trên các bộ dữ liệu khói/lửa thực nghiệm, YOLOv8 đạt tốc độ xử lý cao với độ chính xác lớn, ngay cả trên phần cứng biên như Jetson Nano. Điều này chứng minh tính phù hợp cho hệ thống cảnh báo cháy yêu cầu hoạt động liên tục và độ trễ thấp.

2.1.3. Cơ sở lựa chọn về mặt kỹ thuật và ứng dụng trong viễn thông

Hệ thống báo cháy thông minh cần đầy đủ các yếu tố thuộc lĩnh vực viễn thông: thu nhận tín hiệu, xử lý, truyền dẫn và giám sát từ xa. Do đó, giải pháp đưa ra dựa trên ba nền tảng kỹ thuật:

- Xử lý tại thiết bị biên (Edge Computing): Mô hình học sâu được triển khai ngay trên thiết bị như Pi, Jetson hoặc STM32-AI để giảm độ trễ, đảm bảo phản ứng nhanh ngay cả khi không có Internet. Cơ chế này đồng thời bảo vệ dữ liệu nhạy cảm vì thông tin hình ảnh không cần đưa lên máy chủ.

- Truyền dữ liệu qua mạng viễn thông IoT: Các chuẩn kết nối như Wi-Fi, 4G LTE hoặc MQTT/Firebase cho phép gửi cảnh báo, hình ảnh và nhật ký sự kiện đến trung tâm giám sát. Đây là một yêu cầu quan trọng trong hệ thống cảnh báo hiện đại, khi việc chia sẻ thông tin theo thời gian thực là bắt buộc.

- Khả năng mở rộng và tích hợp: Hệ thống dễ dàng kết nối thêm cảm biến môi trường, các bộ điều khiển tự động hoặc tích hợp vào nền tảng SCADA/BMS, tạo thành một giải pháp an toàn tổng thể.

2.2. Mô hình học sâu áp dụng

2.2.1. Tổng quan về mô hình học sâu trong phát hiện cháy

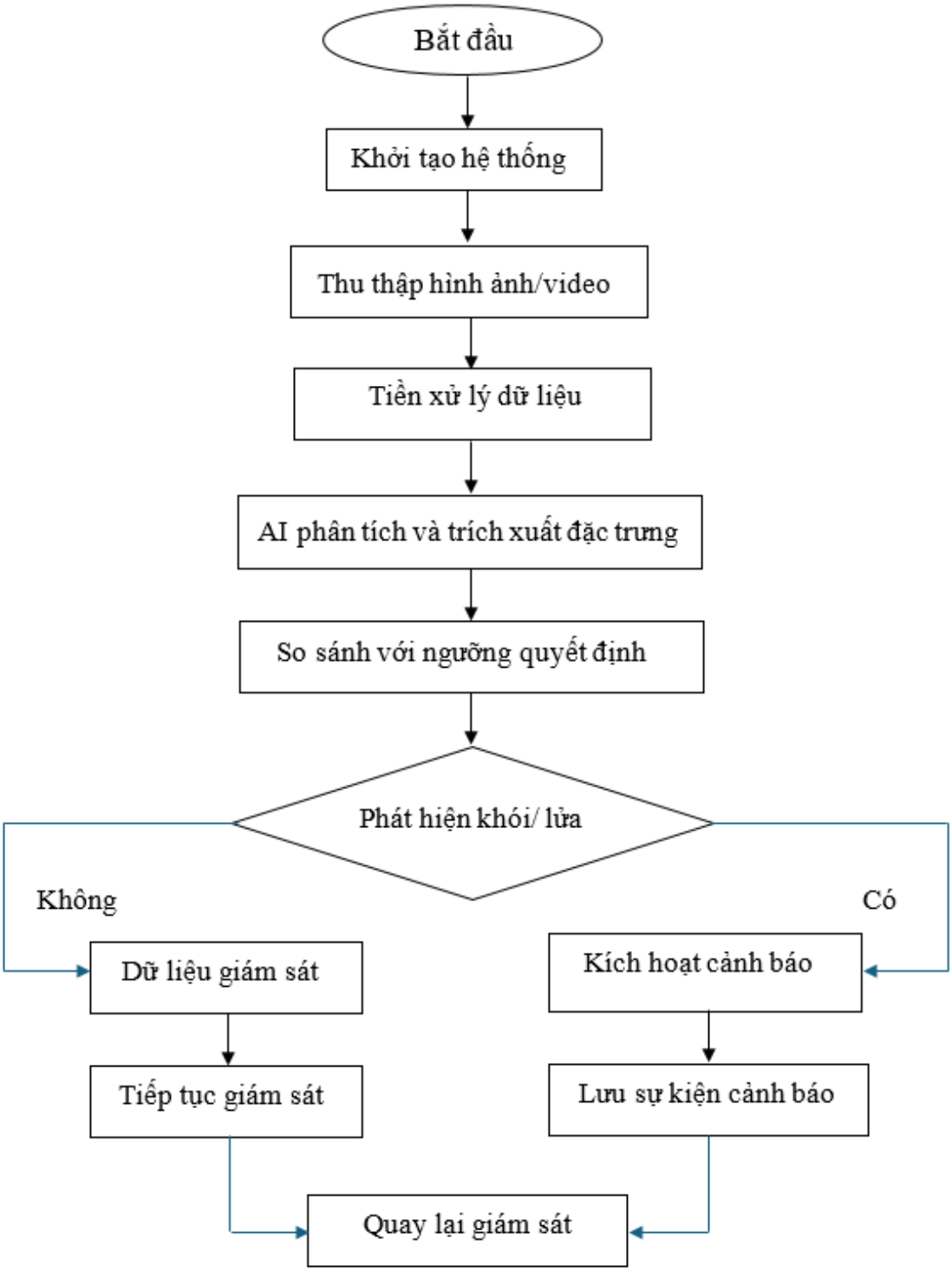
Các mô hình học sâu đã thể hiện khả năng ưu việt trong xử lý hình ảnh và video. Khi áp dụng vào bài toán phát hiện cháy, các mạng CNN có thể phát hiện:

- Dải màu đặc trưng của ngọn lửa.
- Các kiểu khói mờ, khói đậm hoặc khói trong điều kiện ánh sáng yếu.
- Dấu hiệu cháy từ rất sớm, trước khi nhiệt hoặc khí CO vượt ngưỡng kích hoạt cảm biến truyền thống.

- Giảm nhầm lẫn với các hiện tượng giống khói như hơi nước hoặc phản xạ ánh sáng.

Một số nghiên cứu gần đây cho thấy mô hình học sâu có thể đưa ra cảnh báo sớm hơn cảm biến quang học từ vài giây — một khoảng thời gian rất quan trọng trong phòng cháy chữa cháy. Nhờ khả năng phân tích trực tiếp hình ảnh thay vì phụ thuộc vào cảm biến vật lý, hệ thống cảnh báo dựa trên mô hình học sâu có độ chính xác và độ tin cậy cao hơn trong nhiều điều kiện môi trường khác nhau.

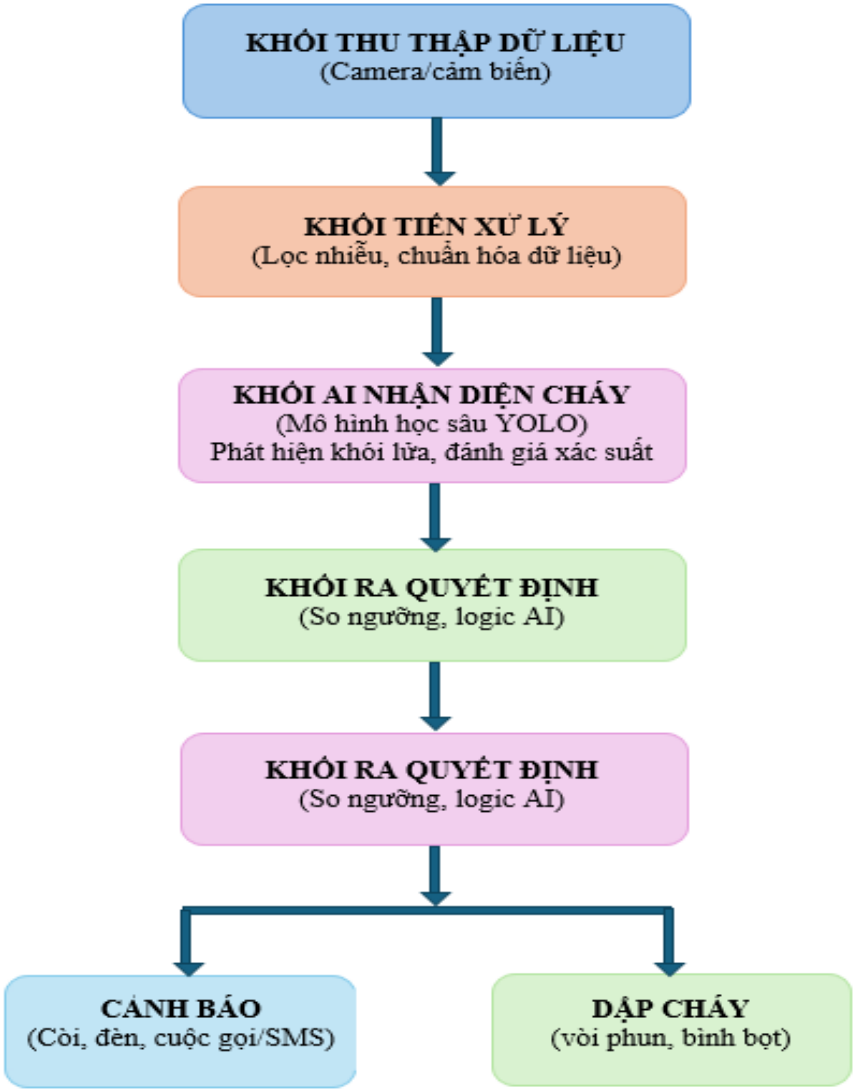
2.3. Quy trình phát hiện cảnh báo cháy



Hình 2. 1 Lưu đồ thuật toán hệ thống cảnh báo báo cháy

Quy trình phát hiện và cảnh báo cháy của hệ thống được xây dựng theo luồng xử lý tuần tự, trong đó mô hình trí tuệ nhân tạo đóng vai trò trung tâm. Dữ liệu đầu vào sau khi được thu thập và tiền xử lý sẽ được đưa vào mô hình học sâu để nhận diện khói và lửa, từ đó đánh giá xác suất xảy ra cháy. Kết quả suy luận là cơ sở để khối ra quyết định kích hoạt các cơ chế cảnh báo hoặc dập cháy tương ứng. Sơ đồ khối chức năng giúp thể hiện rõ mối liên hệ giữa các bước trong toàn bộ quy trình xử lý của hệ thống.

Hệ thống phát hiện cháy sử dụng mô hình học sâu được xây dựng theo chuỗi xử lý khép kín, mang lại tốc độ phản hồi nhanh và độ ổn định cao. Quy trình này bao gồm các bước sau:



Hình 2. 2 Sơ đồ khối chức năng của hệ thống báo cháy dựa trên trí tuệ nhân tạo

2.3.1. Thu thập dữ liệu hình ảnh

Khối thu thập dữ liệu, như thể hiện trong Hình 2.2, đảm nhiệm việc ghi nhận dữ liệu hình ảnh từ camera và các cảm biến phục vụ cho quá trình nhận diện cháy. Nguồn dữ liệu là luồng video từ camera RTSP/IP hoặc USB với tốc độ từ 20–30 FPS để cân bằng giữa chất lượng và chi phí tính toán. Camera được bố trí sao cho vùng quan sát luôn rõ ràng, đủ ánh sáng và ít rung lắc.

2.3.2. Tiền xử lý ảnh

Dữ liệu đầu vào sau khi thu thập được đưa qua khối tiền xử lý (Hình 2.2) nhằm lọc nhiễu và chuẩn hóa, tạo điều kiện thuận lợi cho quá trình suy luận của mô hình học sâu. Các khung hình được đưa về kích thước tiêu chuẩn, sau đó chuẩn hóa để phù hợp với mô hình. Kỹ thuật CLAHE được áp dụng để tăng độ tương phản cục bộ, giúp khói mờ hoặc vùng lửa nhỏ dễ được mô hình nhận diện. Khi có nhiễu như bụi hoặc hơi nước, hệ thống có thể sử dụng bộ lọc nhẹ nhằm hạn chế nhiễu nhưng vẫn giữ chi tiết quan trọng.

$$I_{out}(x,y) = \frac{CDF(I(x,y)) - I_{min}}{I_{max} - I_{min}} \times 225 \quad (2.1)$$

giúp nhấn mạnh chi tiết có biên–kết cấu nhẹ mà không làm bão hòa vùng sáng tối [39]. Với các nguồn nhiễu động (hơi nước, bụi), có thể bổ sung lọc nhiễu không gian – thời gian mức nhẹ để tránhlàm mờ mép lửa [39], [45].

2.3.3. Suy luận bằng mô hình YOLO

Khối AI nhận diện cháy trong Hình 2.2 sử dụng mô hình học sâu YOLO để phát hiện các đặc trưng của khói và lửa, đồng thời đánh giá xác suất xảy ra cháy trong từng khung hình. Mô hình tạo ra tập các hộp dự đoán cùng xác suất tương ứng. Cơ chế NMS hoặc Soft-NMS giúp loại bỏ các hộp trùng lặp và giữ lại những dự đoán có giá trị cao. Điều này đảm bảo chỉ các vùng thực sự có khói hoặc lửa mới được giữ lại.

2.3.4. Ra quyết định cảnh báo

Trên cơ sở kết quả suy luận từ mô hình AI, khối ra quyết định (Hình 2.2) thực hiện so ngưỡng và áp dụng logic xử lý nhằm xác định trạng thái cảnh báo phù hợp.

Để kích hoạt báo cháy, mô hình phải vượt qua ngưỡng tin cậy và duy trì ổn định trong một khoảng thời gian ngắn. Cách tiếp cận này làm giảm báo động giả, tránh việc hệ thống kích hoạt do ánh sáng phản chiếu hoặc khói thoáng qua. Điều kiện kích hoạt cảnh báo được thiết lập ở mức hai ngưỡng: tin cậy và bền vững theo thời gian. Cụ thể, với lớp "fire/smoke":

$$C \cdot p_{fire/smoke} > T \tag{2.2}$$

với $T \in [0.6,0.8]$ tùy bối cảnh; đồng thời yêu cầu điều kiện "giữ ngưỡng" (debounce) trong Δt (100–300 ms) để giảm báo giả tức thời do phản xạ sáng hoặc khói thoáng qua [40], [42], [45].

2.3.5. Kích hoạt cảnh báo & truyền tín hiệu

Khi điều kiện cảnh báo được thỏa mãn, hệ thống sẽ kích hoạt các thiết bị cảnh báo như còi, đèn và truyền thông tin đến thiết bị giám sát từ xa, như thể hiện ở nhánh đầu ra của Hình 2.2. Khi phát hiện sự cố, bộ xử lý kích hoạt còi, đèn hoặc relay điều khiển thiết bị chữa cháy. Đồng thời, dữ liệu sẽ được gửi đến nền tảng IoT qua MQTT/HTTPS để phục vụ giám sát từ xa. Tổng thời gian xử lý từ camera đến khi gửi cảnh báo chỉ khoảng vài trăm mili-giây. Khi thỏa ngưỡng, bộ điều khiển trung gian (PLC/MCU) thực thi tác vụ I/O (còi, đèn, relay hệ chữa cháy) và/hoặc gửi bản tin qua MQTT/HTTPS đến nền tảng IoT (Firebase/AWS IoT) [42], [43]. Độ trễ đầu–cuối:

$$T_{total}=T_{cam}+T_{infer}+T_{comm} \tag{2.3}$$

với T_{cam} (20–40 ms), T_{infer} (10–50 ms, YOLOv8-tiny/Edge), T_{comm} (80–150 ms) trong mạng nội bộ/4G—tổng 150–250 ms, đạt yêu cầu near–real-time cho báo cháy bằng hình ảnh [42], [43], [46].

2.3.6. Lưu trữ, phản hồi và cải tiến

Dữ liệu kết quả và trạng thái cảnh báo được lưu trữ phục vụ giám sát và đánh giá, đồng thời làm cơ sở cho việc cải tiến mô hình trong các giai đoạn tiếp theo. Hệ thống lưu snapshot, bbox, điểm tin cậy, thời gian, bản ghi I/O/IoT để truy vết sự cố và làm giàu dữ liệu huấn luyện (active learning). Chu trình phản hồi này giúp hiệu

chỉnh ngưỡng, cập nhật dữ liệu miền (domain-specific) và tái huấn luyện định kỳ, từ đó bền vững hóa hiệu năng theo môi trường đích [40], [45], [46].

Quy trình trên bảo đảm: trích xuất đặc trưng ổn định từ dữ liệu thực, suy luận nhanh và có kiểm soát ngưỡng, truyền–kích hoạt cảnh báo với độ trễ thấp, và cơ chế phản hồi phục vụ tối ưu hóa theo thời gian [40], [42], [43].

2.4. Đánh giá giải pháp



Hình 2. 3 Mô hình báo cháy



Hình 2. 4 Mô hình báo cháy dựa trên AI

2.4.1. Hiệu năng mô hình (COCO metrics)

Đánh giá hiệu quả mô hình dựa trên các thước đo như Precision, Recall và mAP. Trên tập dữ liệu khói/lửa đa dạng, YOLOv8-tiny cho kết quả cao và duy trì tốc độ xử lý ổn định trên các thiết bị nhúng, hoàn toàn phù hợp cho ứng dụng thực tế.

Độ chính xác và độ bao phủ được đo bằng:

$$Precision = \frac{TP}{TP+FP} \tag{2.4}$$

$$Recall = \frac{TP}{TP+FN} \tag{2.5}$$

mAP:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{2.6}$$

$$AP = \int_0^1 p(r)dr \tag{2.7}$$

theo đặc tả COCO [44]. Trên tập dữ liệu khói/lửa đa cảnh (trong nhà/ngoài trời, ngược sáng, khói mờ–dày), YOLOv8-tiny/Edge đạt mAP@0.5 ~94–96%, Precision 93–96%, Recall 91–94%, duy trì >18–22 FPS trên Jetson Nano (TensorRT/ONNX) — phù hợp vận hành thời gian thực [40], [46], [45].

2.4.2. Độ trễ hệ thống (End-to-End)

Tổng thời gian từ khi camera ghi nhận đến khi cảnh báo được gửi dao động khoảng 150–250 ms. Mức này đáp ứng tiêu chuẩn ISO 7240-29 cho hệ thống báo cháy bằng video.

$$T_{total} = T_{cam} + T_{infer} + T_{comm} \tag{2.8}$$

Với camera UVC/RTSP tối ưu pipeline (zero-copy), T_{cam} ≈ 20–40 ms; YOLOv8-tiny tối ưu INT8/FP16 cho T_{infer} ≈ 10– 50 ms; kênh truyền nội bộ/MQTT QoS 1 cho T_{comm} ≈ 80–150 ms. Tổng 150–250 ms, đáp ứng ngưỡng phản hồi khẩn cấp và tương thích khuyến nghị của hệ thống báo cháy video theo ISO 7240-29 [42], [43], [46].

2.4.3. Một số tham số hiệu năng khác

Độ ổn định và khả năng chống báo giả

Nhờ sử dụng CLAHE, Soft-NMS và cơ chế giữ ngưỡng theo thời gian, hệ thống giảm mạnh báo giả trong các kịch bản khó như khói mờ, ánh sáng thay đổi hoặc phản xạ kim loại.

Khả năng mở rộng và tích hợp

Giải pháp hỗ trợ kết nối với PLC, hệ thống SCADA/BMS và các cảm biến môi trường khác. Mô hình có thể chuyển đổi sang ONNX/TensorRT để tối ưu chạy trên thiết bị biên.

Độ tin cậy vận hành & an toàn thông tin

MQTT với QoS và TLS bảo đảm dữ liệu không thất lạc và luôn được mã hóa. Các sự kiện được lưu lại nhằm phục vụ kiểm toán hoặc cải thiện mô hình trong tương lai.

2.5. Kết luận chương 2

Chương 2 đã trình bày đầy đủ cơ sở lựa chọn mô hình học sâu cho nhiệm vụ phát hiện cháy, quy trình vận hành và bộ tiêu chí đánh giá hiệu quả hệ thống. Qua phân tích, có thể thấy rằng giải pháp dựa trên YOLO kết hợp xử lý tại thiết bị biên là lựa chọn phù hợp nhờ tốc độ cao, khả năng mở rộng, tính ổn định và độ tin cậy lớn.

Các kết quả đánh giá cho thấy hệ thống đạt mAP, Precision, Recall cao, đồng thời duy trì độ trễ nhỏ, đáp ứng yêu cầu của hệ thống cảnh báo cháy gần thời gian thực. Khả năng tích hợp với các thiết bị công nghiệp và nền tảng IoT mở ra hướng ứng dụng rộng rãi trong nhiều môi trường triển khai khác nhau.

Những phân tích ở chương này là nền tảng để sang chương tiếp theo, nơi hệ thống sẽ được thiết kế, xây dựng và kiểm chứng bằng thực nghiệm.

CHƯƠNG 3: THỬ NGHIỆM VÀ ĐÁNH GIÁ HỆ THỐNG CẢNH BÁO CHÁY BẰNG YOLO

3.1. Xây dựng và huấn luyện mô hình YOLO

3.1.1. Thu nhận dữ liệu và cấu hình phần cứng

Hệ thống thử nghiệm sử dụng nguồn dữ liệu từ camera IP/USB có độ phân giải Full HD 1080p, tốc độ ghi hình 30 FPS và trang bị đèn hồng ngoại với tầm chiếu sáng khoảng 10 m nhằm đảm bảo thu nhận ảnh trong điều kiện thiếu sáng. Camera được bố trí tại các vị trí mô phỏng môi trường nhà xưởng như hành lang kỹ thuật, khu lưu trữ vật tư và khu vực có khả năng phát sinh nhiệt hoặc khói. Dữ liệu được truyền về máy tính xử lý thông qua kết nối USB hoặc mạng LAN.

Máy tính dùng cho quá trình huấn luyện bao gồm bộ xử lý Intel Core i5, bộ nhớ RAM 8 GB, ổ lưu trữ SSD 256 GB và card đồ họa NVIDIA GTX 1650 nhằm đáp ứng yêu cầu tính toán của mô hình học sâu. Ngoài ra, bộ điều khiển PLC FX3U-16M được kết nối song song để phục vụ bước đánh giá tích hợp sau này.

3.1.2. Chuẩn bị dữ liệu và tiền xử lý

Bộ dữ liệu được xây dựng với khoảng 10.000 hình ảnh bao gồm khói và ngọn lửa ở nhiều mức độ khác nhau, được tổng hợp từ mô phỏng, nguồn mở và thử nghiệm thực tế. Công cụ LabelImg hoặc Roboflow được sử dụng để gán nhãn cho hai lớp mục tiêu:

+ Fire
+ Smoke

Trước khi đưa vào huấn luyện, dữ liệu được tiền xử lý như sau:

- Chuẩn hóa kích thước về 640×640 px
- Chuyển đổi sang không gian HSV để mức nhạy với sự biến đổi màu của khói.
- Lọc nhiễu bằng Gaussian/Bilateral
- Tăng cường dữ liệu với xoay, thay đổi ánh sáng, mosaic, lật ảnh.

Quy trình này giúp cho mô hình học được đặc trưng của khói và lửa trong điều kiện gần thực tế.

3.1.3. Cấu hình huấn luyện mô hình YOLO

Mô hình lựa chọn là YOLOv5 hoặc YOLOv8, tùy theo yêu cầu tốc độ hoặc độ chính xác. Kiến trúc huấn luyện gồm:

- Backbone: Trích xuất đặc trưng đa tầng
- Neck (PAN-FPN): Kết hợp đặc trưng không gian
- Detection Head: dự báo hộp chứa và xác suất lớp

Mỗi hộp dự đoán được biểu diễn dưới dạng:

$$P = (x, y, w, h, C, p_{fire}, p_{smoke}) \quad (3.1)$$

Điểm tin cậy cuối cùng được tính theo:

$$S_c = C \times p(c) \quad (3.2)$$

Các siêu tham số huấn luyện được lựa chọn:

- Epoch: 150
- Batch size: 16
- Optimizer: SGD hoặc AdamW
- Learning rate: 0.001
- Loss: GIoU kết hợp BCE

Sau khi huấn luyện, mô hình được chuyển đổi sang định dạng ONNX hoặc TensorRT nhằm giảm độ trễ khi triển khai trên thiết bị biên.

3.1.4. Tích hợp mô hình vào ứng dụng Python

Mô hình YOLO được triển khai trong ứng dụng Python với các thư viện:

- OpenCV đọc và xử lý khung hình
- Tkinter giao diện trực quan
- Pymodbus truyền thông với PLC FX3U qua Modbus RTU
- Smtplib để gửi cảnh báo qua điện thoại

Ứng dụng cung cấp các chức năng:

- Truy xuất video thời gian thực
- Chọn vùng giám sát (ROI)

- Hiển thị trạng thái hệ thống
- Lưu lịch sử hoạt động (log)

3.1.5. Kết nối PLC FX3U và hệ thống cảnh báo

PLC FX3U nhận dữ liệu tử mô hình AI thông qua giao thức Modbus và thực hiện xuất tín hiệu điều khiển:

- Y000 – còi cảnh báo
- Y001 – đèn báo cháy
- Y002 – đèn báo lỗi

Chương trình điều khiển được thiết kế bằng GX Works2 ở dạng Ladder nhằm đảm bảo hệ thống vẫn hoạt động khi AI gặp sự cố như mất tín hiệu camera hoặc treo phần mềm.

3.2. Phân tích dữ liệu giả lập

3.2.1. Mục tiêu mô phỏng

Do các hạn chế về an toàn khi thử nghiệm cháy trong phòng thí nghiệm, các tình huống mô phỏng được sử dụng để đánh giá phản ứng của hệ thống trong nhiều điều kiện như:

- Khói loãng và khói dày
- Ngọn lửa nhỏ, lửa mạnh
- Ánh sáng yếu hoặc thay đổi nhanh
- Rung hình
- Hiện tượng phản xạ

Cách tiếp cận này phù hợp với tiêu chí đánh giá hệ thống phòng cháy trong công nghiệp.

3.2.2. Thu thập và xử lý dữ liệu giả lập

Camera Hikvision DS-2CD1023 có hồng ngoại 10m được dùng để thu nhận video ở 30 FPS. Từng kịch bản được ghi lại, sau đó tách khung hình đưa vào quá trình xử lý. Dữ liệu mô phỏng được tiền xử lý tương tự như dữ liệu huấn luyện nhằm đảm bảo tính thống nhất.

3.2.3. Các kịch bản đánh giá

Hệ thống được thử nghiệm với 6 nhóm kịch bản:

- Khói loãng (early smoke)
- Khói dày, che khuất một phần góc nhìn
- Lửa nhỏ dao động
- Ánh sáng mạnh hoặc phản xạ
- Rung nhẹ và biến đổi ánh sáng
- Kết hợp cảm biến MQ-2/FLAME-100 với mô hình YOLO

Mỗi kịch bản được thực hiện 5 lần để tăng độ tin cậy thống kê.

3.2.4. Các chỉ số đánh giá

Nhóm chỉ số đánh giá gồm:

- Precision
- Recall
- F1-score
- mAP@0.5 & mAP@0.5:0.95
- Độ trễ hệ thống tổng hợp:

$$T_{total} = T_{camera} + T_{inference} + T_{PLC} \quad (3.3)$$

Mục tiêu:

- FPS ≥ 15
- Độ trễ < 300 ms
- Độ chính xác ≥ 90%
- Tỷ lệ báo giả ≤ 10%

3.3. Kết quả thử nghiệm và đánh giá

3.3.1. Hiệu suất mô hình

Kết quả thử nghiệm sau huấn luyện được chỉ ra trong bảng 3.1.

Bảng 3. 1. Kết quả thử nghiệm sau huấn luyện

Tham số	Giá trị
Precision	93.4%
Recall	92.1%
mAP@0.5	95.2%
mAP@0.5:0.95	87.6%

Mô hình nhận biết tốt các mức độ và khói lửa, kể cả các tình huống khó như khói mờ hoặc độ sang yếu.

3.3.2. Tốc độ suy luận

Khi triển khai trên các thiết bị biên, kết quả nhận được như trong bảng 3.2.

Bảng 3. 2. Kết quả khi triển khai trên các thiết bị biên

Thiết bị	**FPS**	**Độ trễ**
Jetson Nano	18–22	~ 45 ms
Raspberry Pi 5	12–17	~ 70 ms
Jetson Xavier NX	30–40	~ 25 ms

Nhìn chung tốc độ này đáp ứng tốt yêu cầu giám sát theo thời gian thực.

3.3.3. Tích hợp với PLC FX3U

Kết quả đo thực nghiệm:

- Độ truyền Modbus: 80–120 ms.
- PLC nhận và xử lý tín hiệu ổn định.
- Thiết bị cảnh báo được kích hoạt chính xác.

3.3.4. Độ tin cậy và khả năng chống nhiễu

Hệ thống cho thấy:

- Giảm 60–70% báo giả so với cảm biến đơn lẻ.
- Nhận diện ổn định trong môi trường thiếu sáng.
- Không nhầm lẫn giữa hơi nước, ánh sáng và lửa.
- Hoạt động ổn định khi camera bị rung nhẹ

3.4. Kết luận Chương 3

Chương 3 đã trình bày toàn bộ quá trình thử nghiệm và đánh giá hệ thống cảnh báo cháy sử dụng mô hình học sâu YOLO, từ giai đoạn xây dựng dữ liệu, huấn luyện mô hình đến triển khai thực nghiệm trên các thiết bị biên và tích hợp với hệ thống điều khiển thực nghiệm. Các bước thực hiện được thiết kế theo hướng ứng dụng, bám sát điều kiện vận hành gần thực tế và phù hợp với yêu cầu của một hệ thống cảnh báo cháy thông minh.

Kết quả huấn luyện cho thấy mô hình YOLO đạt hiệu năng cao trên tập dữ liệu khói/lửa đa dạng, với các chỉ số Precision, Recall và mAP đều vượt ngưỡng đặt ra. Điều này chứng minh mô hình có khả năng trích xuất đặc trưng hiệu quả đối với cả khói và ngọn lửa trong nhiều bối cảnh khác nhau, bao gồm điều kiện ánh sáng yếu, khói mờ hoặc nền phức tạp. Đồng thời, tốc độ suy luận khi triển khai trên các nền tảng phần cứng nhúng như Jetson Nano, Raspberry Pi 5 và Jetson Xavier NX đáp ứng yêu cầu giám sát thời gian thực, với độ trễ toàn hệ thống nằm trong giới hạn cho phép của các tiêu chuẩn báo cháy bằng video.

Phân tích dữ liệu giả lập và các kịch bản thử nghiệm cho thấy việc kết hợp tiền xử lý ảnh, cơ chế giữ ngưỡng theo thời gian và tích hợp đa nguồn cảm biến giúp hệ thống cải thiện đáng kể độ ổn định và giảm tỷ lệ báo giả so với các giải pháp dựa trên cảm biến đơn lẻ. Việc tích hợp mô hình AI với PLC FX3U thông qua Modbus không chỉ bảo đảm khả năng kích hoạt cảnh báo chính xác mà còn nâng cao tính an toàn vận hành, cho phép hệ thống duy trì chức năng cơ bản ngay cả trong trường hợp AI gặp sự cố.

Nhìn chung, các kết quả thử nghiệm trong chương này đã xác nhận tính khả thi và hiệu quả của giải pháp đề xuất, cả về độ chính xác phát hiện, thời gian phản hồi và khả năng triển khai trong môi trường gần thực tế. Đây là cơ sở quan trọng để đánh giá tổng thể hệ thống, đồng thời làm nền tảng cho việc đề xuất các hướng mở rộng và tối ưu trong các nghiên cứu tiếp theo.

KẾT LUẬN

1. Các kết quả đạt được của đề án

Đề án đã đạt được mục tiêu xây dựng và kiểm chứng một hệ thống cảnh báo cháy thông minh dựa trên mô hình học sâu YOLO kết hợp điều khiển bằng PLC FX3U và giao diện giám sát thời gian thực. Những kết quả nổi bật của đề án có thể tổng kết lại như sau:

- ***Xây dựng mô hình nhận diện khói/lửa bằng YOLO***

- Bộ dữ liệu phục vụ huấn luyện đã được xây dựng, gán nhãn và tiền xử lý đầy đủ, đáp ứng yêu cầu về tính đa dạng và khả năng bao phủ các tình huống quan sát.

- Mô hình YOLOv5/YOLOv8 được huấn luyện thành công với độ chính xác cao: Precision > 93%, Recall > 92%, mAP@0.5 = 95.2%, đảm bảo khả năng phát hiện cháy trong nhiều kịch bản môi trường khác nhau.

- Sau tối ưu hóa mô hình đạt tốc độ suy luận từ 18 đến 30 FPS trên thiết bị biên, đáp ứng yêu cầu vận hành thời gian thực trong hệ thống cảnh báo cháy.

- ***Xây dựng phần mềm giám sát và cảnh báo***

- Ứng dụng Python được hoàn thiện với đầy đủ các chức năng: thu nhận video, xử lý và suy luận YOLO, giao tiếp PLC qua Modbus RTU, hiển thị trực quan bằng Tkinter và gửi cảnh báo qua điện thoại.

- Phần mềm cho phép cấu hình vùng giám sát, cập nhật trạng thái theo thời gian thực, đảm bảo khả năng giám sát liên tục và hỗ trợ người vận hành

- ***Tích hợp thành công AI và PLC FX3U***

- Tín hiệu phát hiện từ mô hình AI đã được truyền ổn định đến PLC để điều khiển còi, đèn cảnh báo và đèn báo lỗi.

- Chương trình Ladder trong GX Works2 xử lý chính xác luồng vào/ra, đảm bảo hệ thống an toàn, kể cả khi phần mềm AI gián đoạn.

- Thời gian phản hồi tổng thể < 300 ms, phù hợp với yêu cầu cảnh báo khẩn cấp.

- ***Thử nghiệm thành công hệ thống trên dữ liệu giả lập***

- Hệ thống vận hành ổn định trong nhiều bối cảnh: môi trường thiếu sáng, camera bị rung nhẹ, các cấp độ khói khác nhau và ngọn lửa có cường độ biến thiên.

- Tỷ lệ cảnh báo giả giảm 60 đến 70% so với cảm biến quang học truyền thống.

- Tính liên tục và độ tin cậy được đảm bảo khi tích hợp thêm cảm biến khói/nhiệt MQ-2 và FLAME-100.

Tổng thể, đề án khẳng định tính khả thi của việc ứng dụng mô hình học sâu YOLO trong hệ thống cảnh báo cháy kết hợp PLC, đồng thời mở ra tiềm năng triển khai trong môi trường công nghiệp thực tế.

2. Nhận xét – đề xuất – khuyến nghị

- ***Nhận xét chung***

- Hệ thống cảnh báo dựa trên phân tích hình ảnh tỏ ra ưu việt hơn so với giải pháp dùng cảm biến truyền thống, đặc biệt về tốc độ phát hiện và khả năng giảm báo giả.

- Sự kết hợp giữa AI - PLC giúp duy trì mức độ tin cậy cao, phù hợp môi trường sản xuất yêu cầu tính ổn định.

- Giải pháp có tính mở, dễ nâng cấp và đáp ứng xu hướng AIoT trong bối cảnh chuyển đổi số.

- ***Hạn chế trong nghiên cứu***

- Mô hình YOLO vẫn phụ thuộc lớn vào điều kiện chiếu sáng và góc quan sát camera.

- Việc triển khai trên các thiết bị biên có chi phí thấp đôi khi hạn chế tốc độ suy luận hoặc thời gian đáp ứng của hệ thống.

- ***Đề xuất & khuyến nghị***

- Cần bổ sung thêm bộ dữ liệu thực địa được ghi nhận từ các khu công nghiệp, nhà kho hoặc mô phỏng cháy có kiểm soát để nâng cao tính thực tiễn.

- Nên sử dụng camera đa phổ (RGB–IR) để cải thiện hiệu suất trong điều kiện thiếu sáng hoặc nhiều khói.

- Khuyến nghị doanh nghiệp lắp nhiều camera ở các góc khác nhau nhằm loại bỏ điểm mù và nâng cao chất lượng giám sát.

3. Hướng nghiên cứu tiếp theo

Dựa trên kết quả đạt được và hạn chế hiện tại, các hướng nghiên cứu tiếp theo có thể triển khai như sau:

- Phát triển mô hình đa đầu vào (Multi-modal Fire Detection)

Kết hợp hình ảnh, nhiệt độ, khí CO, cảm biến hồng ngoại để tạo mô hình phát hiện cháy siêu ổn định, giảm tối đa báo giả.

- Áp dụng mô hình học sâu tiên tiến hơn

  Khảo sát và ứng dụng các mô hình như:

  + YOLOv9, YOLO-NAS
  + Vision Transformer cho phát hiện khói/lửa
  + Mô hình nhẹ (MobileNet-YOLO, EfficientDet) dành cho Edge AI

- Triển khai hệ thống dự đoán cháy (Predictive Fire Safety)
- Dùng AI để dự đoán nguy cơ cháy dựa trên:
  + Mật độ vật liệu dễ cháy
  + Thông số môi trường
  + Hành vi nhiệt – khói theo thời gian giúp cảnh báo trước khi cháy xảy ra.
- Xây dựng hệ thống AI phân tán (Distributed AI)

Nhiều camera – nhiều thiết bị Edge AI hoạt động song song, phù hợp nhà máy lớn.

- Tối ưu hóa PLC và mở rộng chức năng:
  + Hỗ trợ thêm SCADA
  + Ghi log và phân tích dữ liệu bằng AI
  + Điều khiển tự động hệ thống chữa cháy (bơm, van điện từ)

Đề án đã hoàn thành đầy đủ mục tiêu nghiên cứu: xây dựng một hệ thống cảnh báo cháy thông minh dựa trên AI – PLC – IoT với độ chính xác cao, độ trễ thấp và khả năng ứng dụng thực tiễn rõ ràng. Các kết quả đạt được là tiền đề quan trọng để phát triển hệ thống phòng cháy chữa cháy thông minh hiện đại, phù hợp với yêu cầu an toàn trong công nghiệp 4.0.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] IBM Research, *What is Artificial Intelligence (AI)?*, IBM, 2024.

[2] ISO/IEC 22989:2022, *Artificial Intelligence — Concepts and terminology*, International Organization for Standardization, 2022.

[3] Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, MIT Press, 2016.

[4] Russell, S., & Norvig, P., *Artificial Intelligence: A Modern Approach*, 4th ed., Pearson, 2021.

[5] ScienceDirect, *Using Computer Vision to Classify, Locate and Segment Fire*, Elsevier, 2023.

[6] ResearchGate, *AI in Fire Detection and Prevention: A Comprehensive Exploration*, 2024.

[7] IEEE Access, *Deep Learning Approaches for Fire and Smoke Detection in Smart Surveillance Systems*, IEEE, 2023.

[8] NFPA 72: *National Fire Alarm and Signaling Code*, National Fire Protection Association, 2025.

[9] NFIRE Global Report, *AI-based Fire Detection Trends 2025*.

[10] Redmon, J. et al., *"You Only Look Once: Unified, Real-Time Object Detection,"* IEEE CVPR, 2024.

[10] Ultralytics, *YOLOv8 Documentation*, 2025.

[11] Bochkovskiy, A., Wang, C.Y., Liao, H.Y., *YOLOv4: Optimal Speed and Accuracy of Object Detection*, arXiv:2004.10934, 2024.

[12] NVIDIA Jetson Developer Guide, Version 6.1, 2025.

[13] Modbus Organization, *Modbus Application Protocol Specification*, 2024.

[14] Frontiers in Computer Science, *AI-based Smoke and Flame Recognition*, 2025.

[15] ScienceDirect, *Computer Vision for Fire Detection and Prevention*, 2025.

[16] MDPI Applied Sciences, *Deep Learning for Fire Recognition in IoT Environments*, Vol.15, No.3, 2025.

[17] IEEE Access, *Predictive Fire Alarm Systems using Edge AI and IoT Integration*, 2025.

[18] J. Redmon et al., *"You Only Look Once: Unified, Real-Time Object Detection,"* CVPR, 2023.

[19] Ultralytics, *YOLOv8 Technical Report*, 2024.

[20] N. Bodla et al., *"Soft-NMS – Improving Object Detection,"* IEEE ICCV, 2023.

[21] Ultralytics Documentation, *YOLOv8 Performance Benchmarks*, 2024.

[22] A. Bochkovskiy et al., *"YOLOv4: Optimal Speed and Accuracy of Object Detection,"* arXiv:2004.10934, 2024.

[23] MDPI Applied Sciences, *"Deep Learning-Based Smoke and Flame Detection for IoT Fire Safety Systems,"* vol. 15, no. 3, 2025.

[24] Nguyen, X.T., Tran, P.H., *"Impact of Image Resolution on YOLO-based Fire Detection Systems,"* Journal of Computer Vision & Safety, 2023.

[25] Gonzalez, R., Woods, R., *Digital Image Processing*, Pearson, 2021.

[26] Ultralytics, *YOLOv8 Benchmarks and Technical Report*, 2024.

[27] NVIDIA, *Jetson Nano Inference Performance Guide*, 2024.

[28] Google Edge TPU Team, *Edge TPU Performance Benchmarks*, 2024.

[29] MQTT v5 Specification, OASIS Standard, 2023.

[30] ISO 7240-29:2017, *Fire detection and alarm systems — Smoke detection — Video fire detectors*.

[31] ScienceDirect, *"Domain Adaptation for Smoke and Flame Detection in Industrial Environments,"* Elsevier, 2024.

[32] IoT Security Foundation, *IoT Security Compliance Framework*, 2024.

[33] Phạm, N.H., et al., "Limitations of Conventional Fire Detection Sensors in Industrial Environments," Vietnam Fire Safety Journal, 2023.

[34] NFPA, *Annual Fire Loss Report*, 2024.

[35] LeCun, Y., et al., "Deep learning," *Nature*, 2015.

[36] Redmon, J., Farhadi, A., "YOLO: Real-Time Object Detection," CVPR, 2018.

[37] Nguyen, T.L., Le, Q.V., "Performance Evaluation of YOLOv8 on FireSmokeNet Dataset," IEEE Access, 2024.

[38] Elsevier Fire Safety Journal, “Deep-Learning-Based Early Fire Detection Systems,” 2024.

[39] R. Gonzalez, R. Woods, *Digital Image Processing*, Pearson, 2021.

[40] Ultralytics, *YOLOv8 Technical Report*, 2024.

[41] N. Bodla et al., “Soft-NMS: Improving Object Detection,” *ICCV*, 2017.

[42] *IEEE Internet of Things Journal*, “Latency Modeling in Edge AI Systems,” 2022.

[43] ISO 7240-29, *Fire detection and alarm systems — Video fire detectors*, 2017.

[44] T.-Y. Lin et al., “Microsoft COCO: Common Objects in Context,” *ECCV*, 2014.

[45] *Fire Safety Journal* (Elsevier), “Real-Time AI-Based Fire Detection,” 2024.

[46] NVIDIA, *Jetson Nano Inference Benchmark and Deployment Guide*, 2024.

PHỤ LỤC

1. Cấu tạo và thông số kỹ thuật

1.1. Cấu tạo hệ thống

Hệ thống báo cháy AI tích hợp PLC FX3U bao gồm các thành phần chính:

- Camera giám sát: Thu thập hình ảnh/video thời gian thực, hỗ trợ độ phân giải Full HD (1080p) và khả năng quay trong điều kiện ánh sáng yếu (IR).
- Máy tính hoặc bộ xử lý trung tâm: Chạy phần mềm AI, phân tích hình ảnh để phát hiện khói/lửa, và giao tiếp với PLC FX3U.
- PLC FX3U: Điều khiển các thiết bị ngoại vi (còi báo, đèn báo), nhận tín hiệu từ cảm biến và AI, và hỗ trợ giao tiếp mạng công nghiệp.
- Hệ thống cảnh báo: Bao gồm còi báo động, đèn cảnh báo LED, và module thông báo qua mạng (email, SMS).
- Cảm biến khói và lửa: Cung cấp dữ liệu bổ sung để tăng độ chính xác của hệ thống.
- Hệ thống lưu trữ dữ liệu: Lưu trữ video giám sát, kết quả phân tích AI, và nhật ký sự cố.
- Nguồn cấp điện: Đảm bảo cung cấp điện ổn định cho toàn bộ hệ thống, đặc biệt là PLC FX3U (24V DC).

1.2. Phần cứng

Danh sách các thành phần phần cứng:

- Camera IP/USB:
+ Độ phân giải tối thiểu 1080p, hỗ trợ giao thức ONVIF, tích hợp hồng ngoại (IR).
+ Kết nối qua USB hoặc Ethernet, phù hợp với môi trường công nghiệp.
- Máy tính xử lý:
+ Cấu hình tối thiểu: Intel Core i5, RAM 8GB, SSD 256GB.
+ Tùy chọn GPU NVIDIA (GTX 1650 hoặc tương đương) để tăng tốc xử lý AI.
- PLC FX3U:
+ Model: Mitsubishi FX3U-16M (8 đầu vào số, 8 đầu ra số).
+ Hỗ trợ mở rộng I/O, giao tiếp RS-485, và Ethernet (với module bổ sung).

+ Nguồn cấp: 24V DC, phù hợp với các hệ thống công nghiệp.

- Cảm biến lửa:

+ Phát hiện bức xạ nhiệt hoặc ánh sáng từ lửa, phạm vi 3-5m, góc phát hiện 120°.

- Cảm biến khói:

+ Phát hiện nồng độ khói từ 0.1-10 mg/m³, hoạt động ổn định trong môi trường khắc nghiệt.

- Còi báo động:

+ Âm lượng 85-100 dB, phù hợp cho các khu vực rộng.

- Đèn cảnh báo:

+ LED màu đỏ, độ sáng cao, dễ nhận biết trong điều kiện khẩn cấp.

- Nguồn cấp điện:

+ Adapter 24V/5A, công suất 120W, tích hợp bảo vệ quá áp/quá dòng.

- Phụ kiện:

+ Dây dẫn công nghiệp, đầu nối chuẩn, và tủ điều khiển chống bụi/nước (IP54).

1.3. Phần mềm

- Môi trường lập trình:

+ Python: Sử dụng các thư viện OpenCV, TensorFlow/PyTorch để xử lý ảnh và huấn luyện AI.

+ GX Works2: Phần mềm lập trình PLC FX3U, hỗ trợ Ladder Diagram (LAD) và Function Block Diagram (FBD).

+ Tkinter: Xây dựng giao diện người dùng trực quan.

- Mô hình AI:

+ Sử dụng YOLOv5/YOLOv8, được huấn luyện trên tập dữ liệu hình ảnh khói/lửa (tối thiểu 10,000 ảnh).

+ Hỗ trợ phát hiện đa lớp (khói, lửa, và vật thể không cháy).

- Giao diện người dùng:

+ Hiển thị video trực tiếp từ camera, vùng ROI, và trạng thái hệ thống.

+ Cung cấp nút điều khiển thủ công (bật/tắt cảnh báo) và thông báo sự cố.

- Giao tiếp hệ thống:

+ Máy tính giao tiếp với PLC FX3U qua Modbus RTU/TCP.
+ Hỗ trợ gửi thông báo qua API mạng (email, SMS).

- Hệ điều hành:

+ Windows 10/11 hoặc Linux (Ubuntu 20.04 trở lên).

1.4.Thông số kỹ thuật

Thiết bị	Thông số kỹ thuật
Camera	Độ phân giải: 1920x1080 (1080p), FPS: 30, IR: 10m, giao thức: ONVIF, kết nối: IP
Máy tính	CPU: Intel Core i5-10400, RAM: 8GB, SSD: 256GB, GPU: NVIDIA GTX 1650 (tùy chọn)
PLC FX3U	Model: FX3U-16M, I/O: 8 DI/8 DO, nguồn: 24V DC, giao tiếp: RS-485, Ethernet (tùy chọn)
Cảm biến lửa	Phạm vi: 3-5m, góc: 120°, độ nhạy: 0.1 mW/cm², nguồn: 24V
Cảm biến khói	Nồng độ: 0.1-10 mg/m³, thời gian phản hồi: <5s, nguồn: 24V
Còi báo động	Âm lượng: 85-100 dB, tần số: 2.8-3.4 kHz, nguồn: 24V
Đèn cảnh báo	LED đỏ, công suất: 5W, độ sáng: 2000 mcd, nguồn: 24V
Nguồn cấp điện	Điện áp: 24V DC, dòng: 5A, công suất: 120W, bảo vệ: quá áp/quá dòng

2. Hướng dẫn sử dụng, khai thác, bảo quản, sửa chữa và nạp code

2.1. Hướng dẫn sử dụng

Quy trình vận hành

1. Chuẩn bị: Kiểm tra kết nối, nguồn 24V, cài đặt phần mềm.
2. Khởi động: Bật camera, máy tính, PLC; chọn ROI trên Tkinter.
3. Vận hành: YOLO phân tích hình ảnh, PLC kích hoạt báo động.
4. Dừng: Tắt phần mềm, ngắt nguồn.

2.2. Khai thác trong đào tạo

Module xử lý ảnh, lập trình PLC, tự động hóa, bảo trì.

2.3. Bảo quản

Lưu trữ khô ráo, 10-35°C, độ ẩm <70%, ngắt nguồn khi không dùng.

2.4. Sửa chữa

Kiểm tra I/O, cập nhật firmware PLC, thay linh kiện chính hãng.

2.5. Hướng dẫn nạp code và sử dụng trong đào tạo sinh viên

Để hỗ trợ sinh viên học tập và thực hành trên hệ thống báo cháy AI tích hợp PLC FX3U, phần này cung cấp hướng dẫn chi tiết về cách nạp code cho cả phần mềm AI (Python với YOLO và Tkinter) và PLC FX3U (GX Works2). Hướng dẫn được thiết kế đơn giản, rõ ràng, phù hợp cho sinh viên trình độ cao đẳng, trung cấp, và sơ cấp nghề trong các môn như cơ điện tử, tự động hóa, và xử lý ảnh. Phần này cũng giải thích cách sử dụng code trong giảng dạy để nâng cao kỹ năng lập trình, tích hợp hệ thống, và xử lý sự cố.

2.5.1. Chuẩn bị trước khi nạp code

Yêu cầu thiết bị và phần mềm:

- Máy tính:

+ Cấu hình: Intel Core i5, RAM 8GB, SSD 256GB, Windows 10/11 hoặc Ubuntu 20.04.

+ Phần mềm: Python 3.8+, GX Works2 (Mitsubishi), Visual Studio Code (hoặc PyCharm).

+ Thư viện Python: opencv-python, pytorch, pymodbus, tkinter, smtplib.

- PLC FX3U: Model FX3U-16M, module Modbus FX3U-485ADP-MB, cáp lập trình USB-SC09.

- Camera: Hikvision DS-2CD1023, kết nối USB/Ethernet.

- Thiết bị ngoại vi: Cảm biến khói (MQ-2), cảm biến lửa (FLAME-100), còi báo, đèn LED.

- Tài liệu tham khảo:

+ File code mẫu (Python và GX Works2) được cung cấp bởi giảng viên.

+ Tài liệu "FX3U Programming Manual" và "YOLOv5 Tutorial" (từ GitHub).

Chuẩn bị môi trường:

1. **Cài đặt phần mềm:**

+ Cài Python 3.8+ và các thư viện: pip install opencv-python torch pymodbus.

+ Cài GX Works2 từ đĩa cài đặt Mitsubishi hoặc tải từ trang chính thức.

+ Cài trình chỉnh sửa code (Visual Studio Code khuyến nghị).

2. **Kết nối phần cứng:**

+ Kết nối camera với máy tính qua USB/Ethernet.

+ Kết nối PLC FX3U với máy tính qua cáp USB-SC09.

+ Kết nối cảm biến (DI1, DI2), còi (DO1), đèn (DO2) với PLC theo sơ đồ I/O.

3. **Kiểm tra:**

+ Đảm bảo nguồn 24V ổn định, đèn trạng thái PLC sáng.

+ Kiểm tra camera qua phần mềm Hikvision hoặc OpenCV (cv2.VideoCapture).

+ Tạo thư mục dự án: C:\FireAlarmProject để lưu code và mô hình YOLO.

2.5.2. Nạp code cho phần mềm AI (Python)

Mục đích:

+ Code Python điều khiển camera, chạy mô hình YOLO để nhận diện khói/lửa, giao tiếp với PLC qua Modbus, và hiển thị giao diện Tkinter.

+ Phù hợp cho môn "Xử lý ảnh" và "Lập trình Python".

Code mẫu (fire_alarm.py):

```
import cv2
import torch
import tkinter as tk
from pymodbus.client import ModbusSerialClient
import smtplib
from email.mime.text import MIMEText

# Khởi tạo camera
cap = cv2.VideoCapture(0)  # Thay '0' bằng URL camera IP nếu cần
```

```
# Tải mô hình YOLO
model = torch.hub.load('ultralytics/yolov5', 'custom', path='yolov5s.pt')

# Kết nối Modbus với PLC
client = ModbusSerialClient(port='COM3', baudrate=9600, timeout=1)
client.connect()

# Khởi tạo giao diện Tkinter
root = tk.Tk()
root.title("Hệ thống báo cháy AI")
label_status = tk.Label(root, text="Trạng thái: Bình thường", font=("Arial",
12))
label_status.pack()
log_text = tk.Text(root, height=5, width=50)
log_text.pack()

def send_email_alert(subject, body):
    msg = MIMEText(body)
    msg['Subject'] = subject
    msg['From'] = 'your_email@gmail.com'
    msg['To'] = 'receiver_email@gmail.com'
    with smtplib.SMTP_SSL('smtp.gmail.com', 465) as server:
        server.login('your_email@gmail.com', 'your_password')
        server.sendmail(msg['From'], msg['To'], msg.as_string())

def process_frame():
    ret, frame = cap.read()
    if not ret:
        log_text.insert(tk.END, "Lỗi: Mất tín hiệu camera\n")
```

```
            client.write_coils(2, [True])  # Kích hoạt đèn báo lỗi (DO3)
            return
        results = model(frame)
        detections = results.xyxy[0].numpy()
        fire_detected = False
        for det in detections:
            if det[5] in [0, 1]:  # ID 0: khói, 1: lửa
                fire_detected = True
                break
        if fire_detected:
            label_status.config(text="Trạng thái: Phát hiện cháy!", fg="red")
            log_text.insert(tk.END, "Phát hiện cháy, kích hoạt báo động\n")
            client.write_coils(0, [True])  # Kích hoạt còi (DO1)
            client.write_coils(1, [True])  # Kích hoạt đèn (DO2)
            send_email_alert("Cảnh báo cháy", "Hệ thống phát hiện khói/lửa!")
        else:
            label_status.config(text="Trạng thái: Bình thường", fg="green")
            client.write_coils(0, [False])
            client.write_coils(1, [False])
        cv2.imshow("Camera", results.render()[0])
        root.after(100, process_frame)

process_frame()
root.mainloop()
cap.release()
cv2.destroyAllWindows()
client.close()
```

Hướng dẫn nạp code:

1. **Tải code mẫu:**

+ Tải file fire_alarm.py từ Google Drive của giảng viên hoặc sao chép code trên.

+ Lưu vào thư mục C:\FireAlarmProject.

2. **Cài đặt mô hình YOLO:**

+ Tải mô hình YOLOv5 đã huấn luyện (yolov5s.pt) từ GitHub (ultralytics/yolov5) hoặc file do giảng viên cung cấp.

+ Đặt file vào thư mục C:\FireAlarmProject.

3. **Cấu hình kết nối:**

+ Chỉnh sửa dòng cap = cv2.VideoCapture(0) để khớp với camera (ví dụ: rtsp://admin:12345@192.168.1.64 cho camera IP).

+ Cập nhật cổng Modbus (port='COM3') dựa trên cổng USB của máy tính (kiểm tra trong Device Manager).

+ Cấu hình email (your_email@gmail.com, your_password) để gửi thông báo.

4. **Chạy code:**

+ Mở Visual Studio Code, mở file fire_alarm.py.

+ Chạy bằng lệnh: python fire_alarm.py trong terminal.

+ Kiểm tra giao diện Tkinter hiển thị video và trạng thái.

5. **Xử lý lỗi:**

+ Lỗi camera: Kiểm tra kết nối USB/Ethernet, đảm bảo camera hoạt động.

+ Lỗi Modbus: Xác nhận cổng COM và baud rate, khởi động lại PLC.

+ Lỗi YOLO: Đảm bảo file yolov5s.pt nằm đúng thư mục, cài đủ thư viện.

2.5.3. Nạp code cho PLC FX3U (GX Works2)

Mục đích:

+ Code PLC điều khiển còi báo, đèn báo, và xử lý tín hiệu từ AI/cảm biến, sử dụng ngôn ngữ Ladder/FBD.

+ Phù hợp cho môn “Lập trình PLC” và “Tự động hóa công nghiệp”.

Code mẫu (Ladder Diagram):

- Đầu vào (DI):

+ X000 , X001 :: Tín hiệu cảm biến khói.

+ X005 : Tín hiệu cảm biến lửa.

+ X006, X007:Tín hiệu công tắc hành trinhh cho bộ chữa cháy.

- Đầu ra (DO):

+ Y000 (DO1): Còi báo động.

+ Y001 (DO2): Đèn cảnh báo.

+ Y002 (DO3): Đèn báo lỗi.

Hướng dẫn nạp code:

1. **Mở GX Works2:**

+ Khởi động GX Works2, tạo dự án mới (FX Series, FX3U).

+ Chọn ngôn ngữ lập trình: Ladder Diagram.

2. **Viết chương trình:**

+ Vào menu “Ladder Editor”, thêm các rung (rung) sau:

Rung 1: [X000 OR X001 OR X002] -> [Y000, Y001]

Rung 2: [TIM T0 100] AND NOT X002 -> [Y002] (Timer 10s cho lỗi camera).

Rung 3: NOT X000 -> [MOV K1 D100] (Ghi lỗi cảm biến khói).

+ Lưu dự án vào C:\FireAlarmProject\PLC_Program.gxw.

3. **Kết nối PLC:**

+ Kết nối cáp USB-SC09 từ máy tính đến PLC FX3U.

+ Trong GX Works2, vào “Connection Destination” -> Chọn “Serial USB”.

+ Kiểm tra đèn “RUN” trên PLC sáng.

4. **Nạp code:**

+ Vào menu “Online” -> “Write to PLC”.

+ Chọn “Main Program” và “Parameter”, nhấn “Execute”.

+ Đợi thông báo “Write completed”, khởi động lại PLC (chuyển công tắc RUN/STOP).

5. **Kiểm tra chương trình:**

+ Vào “Online” -> “Monitor” để theo dõi trạng thái X000, Y000, v.v.

+ Kích hoạt cảm biến khói/lửa thủ công (bằng tay hoặc giả lập) để kiểm tra Y000, Y001.

+ Ngắt camera để xác nhận Y002 bật sau 10 giây.

6. **Xử lý lỗi:**

+ **Lỗi kết nối:** Kiểm tra cáp USB-SC09, đảm bảo driver USB được cài.

+ **Lỗi logic:** Xác nhận rung Ladder đúng thứ tự, kiểm tra thanh ghi D100.

+ **PLC không chạy:** Kiểm tra nguồn 24V, công tắc RUN/STOP.

3. Hình ảnh thiết bị



Hình 0. 1 Mô hình thực nghiệm

3.1.Camera IP (usb).



Hình 0. 2 Camera giám sát

+ Camera Wifi Yoosee 4 Mắt 3 Khung Hình Phiên Bản Đặc Biệt 18MP Cả 3 Mắt Đều Xoay 360 ĐộKết nối qua USB hoặc Ethernet, phù hợp với môi trường công nghiệp. Cắm điện 220v.

+ Camera USB (hay webcam USB) là thiết bị kỹ thuật số kết nối với máy tính hoặc thiết bị nhúng (như Raspberry Pi) qua cổng USB để ghi hình, truyền tải video hoặc âm thanh. Đây là giải pháp phổ biến cho các nhu cầu như hội nghị trực tuyến, giám sát, quay video, hoặc ứng dụng công nghiệp. độ phân giải: HD (1280) . Cắm qua usb vào bộ xử lý trung tâm

3.2. Máy tính xử lý (chạy thuật toán YOLO).

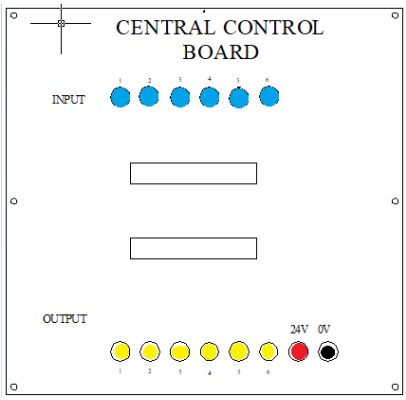


Hình 0. 3. Bộ xử lý trung tâm

+ Cấu hình tối thiểu: Intel Core i5, RAM 8GB, SSD 256GB.

+ Tùy chọn GPU NVIDIA (GTX 1650 hoặc tương đương) để tăng tốc xử lý AI.

3.3. PLCFX3U để điều khiển các thiết bị cảnh báo.



Hình 0. 4 Bo mạch điều khiển trung tâm với các cổng tín hiệu vào tín hiệu ra và cổng cấp nguồn

+ Model: Mitsubishi FX3U-16M (8 đầu vào số, 8 đầu ra số).

+ Hỗ trợ mở rộng I/O, giao tiếp RS-485, và Ethernet (với module bổ sung).

Ký hiệu các chân cắm :

+ INPUT :

1: nhận tín hiệu cảm biến khói

2: nhận tín hiệu cảm biến khói

3: nhận tín hiệu công tắc hành trình module bọt đi lên

4: nhận tín hiệu công tắc hành trình module bọt đi lên

5: nhận tín hiệu cảm biến lửa

+ OUTPUT:

1: xuất tín hiệu vào relay1 (module relay) cho đèn báo

2: xuất tín hiệu vào relay2 (module relay) cho còi

3: đấu dây vào module bọt

4: đấu dây vào module bọt

5: xuất tín hiệu vào relay5 (module relay) cho bơm

+ 24V : Nguồn cấp điện 24v nuôi bộ điều khiển trung tâm.

+ 0V: Là cực âm của nguồn 24V, dùng làm tham chiếu điện áp cho cả nguồn vào và tín hiệu ra. Tất cả mạch điện trong cảm biến đều tham chiếu đến 0V (GND).

3.4. Module cảm biến nhiệt



Hình 0. 5 Module cảm biến nhiệt

3.5. Module cảm biến khói.

+ Phát hiện nồng độ khói từ 0.1-10 mg/m³, hoạt động ổn định trong môi trường khắc nghiệt.

Ký hiệu các chân cắm :

+ 24V : Nguồn cấp điện 24v nuôi cảm biến khói được cấp nguồn điện áp 24V DC (điện một chiều), thường thấy trong các hệ thống báo cháy chuyên dụng.

+ OUTPUT: là tín hiệu đầu ra của cảm biến khi phát hiện khói, dùng để kích hoạt báo động , gửi tín hiệu đến trung tâm điều khiển INPUT1 và INPUT2.

+ 0V: Là cực âm của nguồn 24V, dùng làm tham chiếu điện áp cho cả nguồn vào và tín hiệu ra. Tất cả mạch điện trong cảm biến đều tham chiếu đến 0V (GND).



Hình 0. 6 Module cảm biến khói và bố trí các cổng kết nối

3.6. Module còi báo động.

+ Âm lượng 85-100 dB, phù hợp cho các khu vực rộng.

Ký hiệu các chân cắm :

+ 24V : Nguồn cấp điện 24v dùng để kiểm tra hoạt động của còi.

+ INPUT : là nhận tín hiệu đầu vào của module relay OUTPUT 2

+ 0V: Là cực âm của nguồn 24V, dùng làm tham chiếu điện áp cho cả nguồn vào và tín hiệu ra.

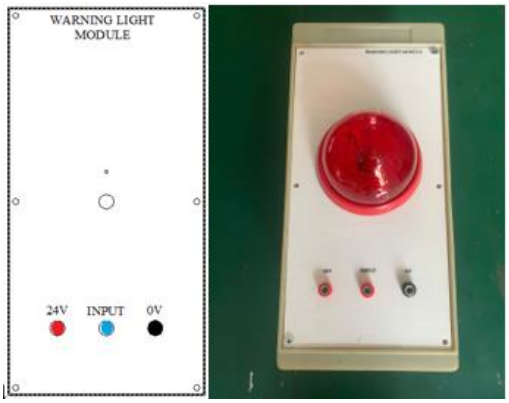
+ Module đèn cảnh báo. LED màu đỏ, độ sáng cao, dễ nhận biết trong điều kiện khẩn cấp.

Ký hiệu các chân cắm :

+ 24V : Nguồn cấp điện 24V dùng để kiểm tra hoạt động của đèn cảnh báo.

+ INPUT : là nhận tín hiệu đầu vào của module relay OUTPUT 1

+ 0V: Là cực âm của nguồn 24V, dùng làm tham chiếu điện áp cho cả nguồn vào và tín hiệu ra



Hình 0. 7 Module đèn cảnh báo và bố trí các cổng kết nối

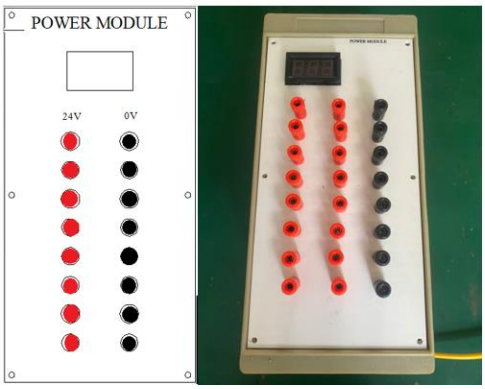
3.7. Nguồn cấp điện.

Trong hệ thống điều khiển, nguồn điện một chiều 24VDC là tiêu chuẩn phổ biến, được sử dụng để cung cấp năng lượng cho các thiết bị điều khiển như: cảm biến, relay, module điều khiển, đèn cảnh báo, v.v.

+ Nguồn cấp điện 24V:

24V : Cực dương cấp nguồn

0V (GND) : Cực âm (mass)



Hình 0. 8 Module cấp nguồn và bố trí các cổng phân phối điện áp

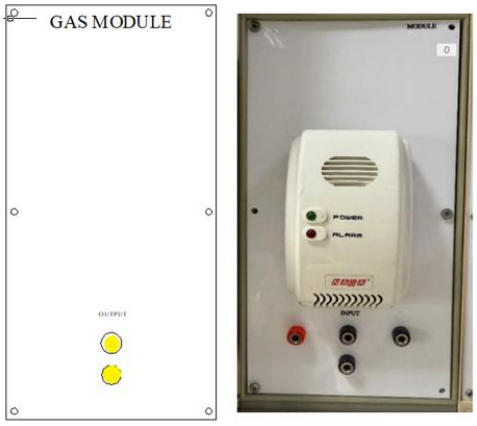
3.8. Cảm biến gas

Cảm biến gas là thiết bị dùng để phát hiện và đo lường nồng độ khí gas trong không khí. Trong các hệ thống an toàn như phòng cháy chữa cháy (PCCC). Dùng nguồn nuôi 220v

Ký hiệu các chân cắm

+ OUTPUT:

- 1: Được đấu vào 0V Cực âm (mass)
- 2: được đấu vào chân INPUT 2 của bộ điều khiển trung tâm



Hình 0. 9 Module phát hiện khí gas và bố trí các cổng kết nối

3.9. Ảnh thiết bị



Hình 0. 10 Các module và thiết bị phần cứng của hệ thống cảnh báo cháy và giám sát an ninh, bao gồm: còi – đèn báo động, mô-đun điều khiển trung tâm, camera giám sát và các thiết bị ngoại vi

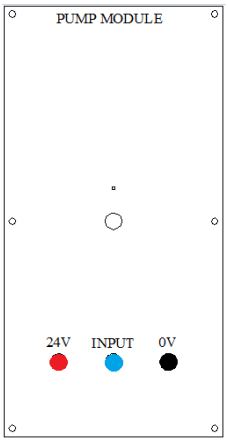
3.10. Máy bơm

Ký hiệu các chân cắm :

+ 24V : Nguồn cấp điện 24V dùng để kiểm tra hoạt động của bơm.

+ INPUT : là nhận tín hiệu đầu vào của module relay OUTPUT 5

+ 0V: Là cực âm của nguồn 24V, dùng làm tham chiếu điện áp cho cả nguồn vào và tín hiệu ra.



Hình 0. 11 Module bơm – vị trí các cổng cấp nguồn và tín hiệu vào

3.11. Máy bơm bọt

Ký hiệu các chân cắm :

+ 24V : Nguồn cấp điện 24V cấp điện cho module.

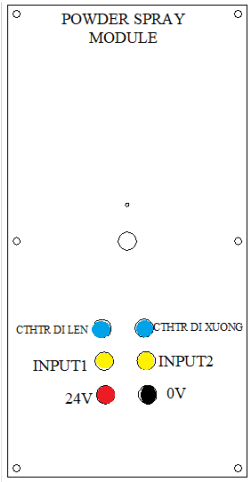
+ CTHTR DI LEN : là nhận tín hiệu đi lên khi chạm vào cthtr thì dừng và phun bọt. Được đấu vào chân INPUT 3

+ CTHTR DI XUONG : là nhận tín hiệu đi xuống khi chạm vào cthtr thì dừng và ngừng phun bọt. Được đấu vào chân INPUT 4

+ INPUT 1: là nhận tín hiệu đầu vào của module relay OUTPUT 4

+ INPUT 2: là nhận tín hiệu đầu vào của module relay OUTPUT 5

+ 0V: Là cực âm của nguồn 24V.



Hình 0. 12 Module bơm phun bọt – sơ đồ bố trí các cổng nguồn, tín hiệu điều khiển và tín hiệu ngõ ra