

Hệ thống phát hiện khói và cháy thông minh đa thể thức

Hồ Anh Dũng¹, Đoàn Thị Hương Giang², Trần Đình Hùng³,
Ma Khánh Tùng², Nguyễn Huyền Tiên An², Bùi Thị Duyên^{2*}

¹Khoa Công nghệ thông tin, Trường Đại học Công nghệ Đông Á, Đường Trịnh Văn Bô, Nam Từ Liêm, Hà Nội, Việt Nam;

²Khoa Điều khiển và Tự động hóa, Trường Đại học Điện lực, 235 Hoàng Quốc Việt, Bắc Từ Liêm, Hà Nội, Việt Nam;

³Viện Hàng không Vũ trụ Viettel, Tập đoàn Công nghiệp Viễn thông Quân đội, 1 Trần Hữu Dực, Nam Từ Liêm, Hà Nội, Việt Nam.

*Corresponding author: duyenbt@epu.edu.vn

Nhận bài: 12/6/2024; Hoàn thiện: 02/8/2024; Chấp nhận đăng: 12/8/2024; Xuất bản: 25/8/2024.

DOI: <https://doi.org/10.54939/1859-1043.j.mst.97.2024.138-147>

TÓM TẮT

Phát hiện sớm sự cố khói và cháy trong các căn hộ được xem là một trong những yếu tố vô cùng quan trọng, giúp ngăn ngừa hậu quả nghiêm trọng do cháy nổ gây ra về con người và tài sản. Các giải pháp phổ biến hiện nay được áp dụng là dùng các loại cảm biến vật lý như cảm biến khí ga, cảm biến khói và cảm biến nhiệt độ gây ra do cháy. Tuy nhiên, việc sử dụng cảm biến vật lý sẽ có nhược điểm là thời gian phát hiện chậm hơn so với việc kết hợp nhiều luồng thông tin, đặc biệt là kết hợp với thông tin hình ảnh. Bài báo này chúng tôi đề xuất giải pháp phát hiện cháy đa thể thức, sử dụng kết hợp các cảm biến vật lý (Sensor) và cảm biến hình ảnh (Camera). Trong đó, mô hình đề xuất áp dụng kiến thức về trí tuệ nhân tạo (Artificial Intelligent-AI) và Internet vạn vật (IoT) để phát hiện khói và cháy áp dụng trong môi trường trong nhà. Giải pháp phát hiện khói và cháy bằng hình ảnh sử dụng giải thuật chắt lọc tri thức (Knowledge Distillation-KD) từ nhiều mô hình giáo viên YOLO bản đầy đủ cho mô hình YOLO bản rút gọn và cho kết quả phát hiện cao hơn lên tới 10%; mô hình rút gọn có thời gian đáp ứng nhanh hơn so với các mô hình bản đầy đủ lên tới 8.22 (ms) khi sử dụng GPU và 51.56 (ms) khi sử dụng CPU.

Từ khóa: Mạng nơ ron tích chập; Học sâu; Báo cháy; Cảm biến; Phát hiện cháy; Đa thể thức

1. MỞ ĐẦU

Từ thực tế và từ các thông tin cung cấp trên trang thông tin điện tử Cục cảnh sát Phòng cháy, chữa cháy và cứu nạn cứu hộ [1] cho thấy, tình trạng cháy nổ tại các hộ gia đình ngày càng diễn biến phức tạp hơn, gia tăng cả về số lượng lẫn mức độ thiệt hại với những tác động tiêu cực đến tính mạng và tài sản của người dân. Đứng trước những báo động đó, mỗi người dân trong chúng ta cần phải nâng cao ý thức về phòng cháy, chống cháy, chữa cháy, tích cực chủ động trong việc bảo vệ tính mạng và tài sản bản thân, xã hội trước các nguy cơ hiểm họa luôn rình rập. Qua đó, giảm thiểu tối đa các thiệt hại do hỏa hoạn gây ra. Một trong những giải pháp đó là trang bị hệ thống báo cháy, trang bị các trang thiết bị phòng chống cháy nổ trong khu vực sinh sống và nơi làm việc. Các hệ thống phát hiện sớm cháy nổ thông thường hiện nay được bán trên thị trường thường là các thiết bị độc lập để phát hiện khói riêng, khí ga riêng và phát hiện cháy riêng [12]. Các thiết bị này được các công ty hoặc các cửa hàng nhập về bán cho người dân. Tuy nhiên, các thiết bị báo khói và cháy này là các thiết bị rời và mỗi thiết bị độc lập với các thiết bị khác. Chúng có đặc điểm chung là sử dụng nguồn điện chung với điện lưới và sử dụng cảm biến vật lý để phát hiện và báo cháy tại chỗ. Không thể thực hiện kết hợp các luồng thông tin với nhau, khi nguồn lưới có sự cố thì thiết bị dừng hoạt động và không thể thực hiện được chức năng báo cháy. Bên cạnh đó, hệ thống báo cháy chỉ dùng cảm biến vật lý thường phát hiện khi có khói nhiều hoặc cháy đã lớn. Do đó, trong nghiên cứu này chúng tôi đề xuất một hệ thống báo cháy thông minh đa thể thức kết hợp ba loại cảm biến vật lý gồm phát hiện khí ga, phát hiện khói và phát hiện cháy. Ngoài ra, hệ thống phát hiện khói và cháy sử dụng thông tin hình ảnh sẽ giúp phát hiện sớm sự cố cháy

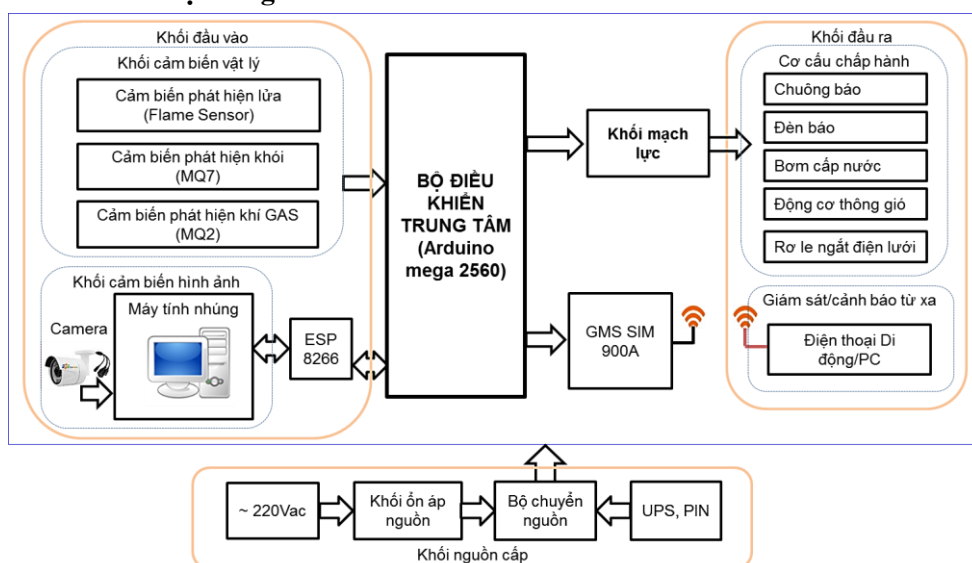
trong các hộ gia đình. Hệ thống đề xuất cung cấp giải pháp chuyển nguồn tự động giúp hệ thống tiếp tục phát hiện sự cố khi tách nguồn điện lưới ra khỏi mạch điều khiển của hệ thống cảnh báo, điều này giúp tăng độ ổn định cho hệ thống đề xuất. Hệ thống cung cấp giải pháp báo cháy tại chỗ và báo cháy từ xa, gửi thông tin về thiết bị di động như điện thoại, máy tính bảng khi người dùng ra khỏi nhà. Chúng tôi cũng đưa ra một số giải pháp hỗ trợ công tác chữa cháy như: ngắt nguồn điện, mở cửa, bật bơm và khuếch tán khí ga.

Ngoài ra, bài toán xử lý ảnh trong hệ thống sử dụng máy tính nhúng, đây là thiết bị có kích thước nhỏ gọn, dễ lắp đặt, kết nối và cài đặt các mô hình xử lý thông tin nói chung cũng như xử lý ảnh nói riêng, tuy nhiên chúng có nhược điểm là cấu hình thấp. Để có phần mềm xử lý hình ảnh dễ dàng cài đặt trên hệ thống, chúng tôi đề xuất sử dụng mô hình YOLO, đây là một trong những mô hình mạng nơ ron tích chập hiệu quả cho bài toán phát hiện và nhận dạng đối tượng trong ảnh. Mô hình này được công bố phiên bản đầu tiên vào năm 2016. Cho đến nay, mô hình YOLO đã có nhiều phiên bản khác nhau được công bố từ YOLO V1-V9 [3-11]. Tuy nhiên, phổ biến và được nhiều nhóm nghiên cứu và thực nghiệm sử dụng hiện nay là YOLO V5 [7]. Khi sử dụng bản đầy đủ của mô hình YOLO thường khó để triển khai trên các thiết bị máy tính nhúng nhỏ gọn như Raspberry, Jetson,... Do vậy, đã có nhiều phiên bản thu gọn của các phiên bản tương ứng như YOLO V5n [7] là bản thu gọn của YOLO V5 [7] bản đầy đủ. Tuy nhiên, các bản thu gọn có thời gian đáp ứng nhanh nhưng độ chính xác không cao. Học chuyển giao có chất lọc tri thức từ các chuyên gia là một trong những giải pháp được đưa ra [1, 14] nhằm nâng cao hiệu quả của các mô hình học sâu. Trong nghiên cứu này, chúng tôi thực hiện đánh giá hiệu quả và thời gian đáp ứng của các phiên bản YOLO V5n với giải pháp học chuyển giao có chất lọc tri thức từ các chuyên gia khác nhau là các phiên bản YOLO đầy đủ gần đây nhất gồm YOLO V5 [7], YOLO V6 [8], YOLO V7 [9] và YOLO V8 [10]. Giải pháp đề xuất được đánh giá trên bộ cơ sở dữ liệu (CSDL) khói và cháy do chúng tôi thu thập và gán nhãn có tên được đặt là EPUfire. Kết quả thử nghiệm cho thấy giải pháp đề xuất đạt kết quả chính xác cao trên bộ CSDL EPUFire.

Phần tiếp theo của bài báo gồm các phần sau đây: Phần 2 mô tả chi tiết giải pháp đề xuất. Kết quả thử nghiệm được trình bày trong phần 3. Phần 4 là mục cuối cùng sẽ trình bày kết luận và hướng phát triển trong thời gian tiếp theo.

2. PHƯƠNG PHÁP ĐỀ XUẤT

2.1. Sơ đồ khối của hệ thống

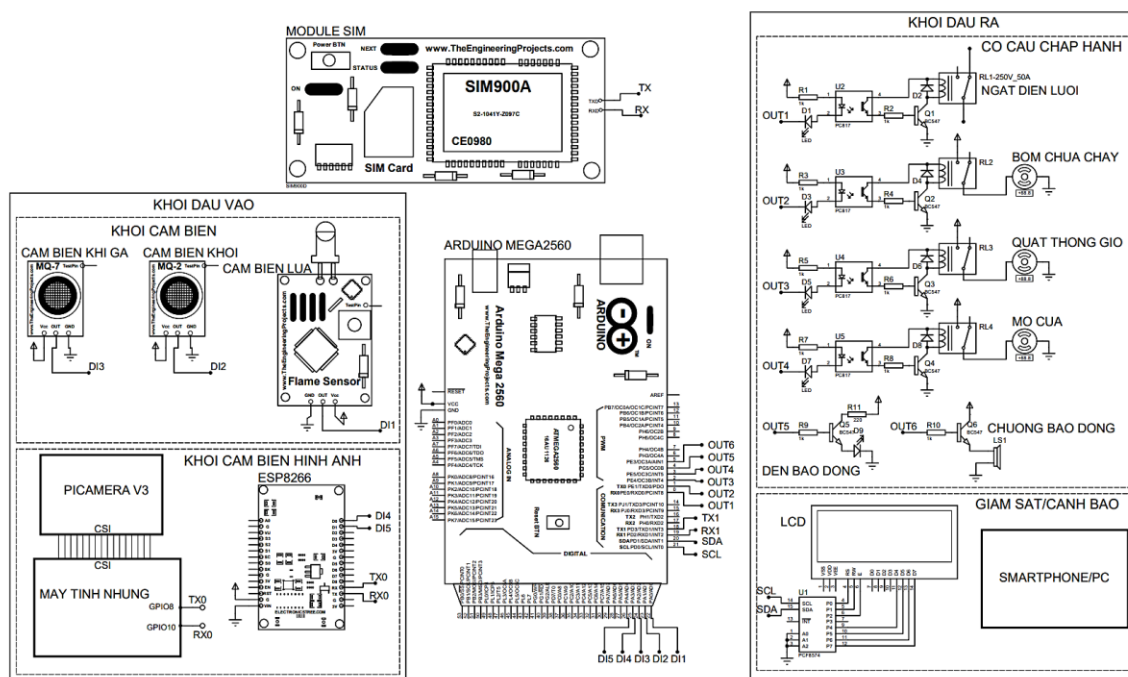


Hình 1. Sơ đồ khối hệ thống phát hiện khói và cháy đa thể thức.

Hệ thống phát hiện khói và cháy đa thể thức đề xuất được chúng tôi thể hiện như sơ đồ khối trong hình 1. Thành phần đầu tiên trong hệ thống có khối đầu vào sử dụng hai luồng thông tin: (1) Cảm biến vật lý, gồm: cảm biến phát hiện lửa, cảm biến phát hiện khói và cảm biến phát hiện khí ga; (2) Cảm biến hình ảnh sử dụng camera kết nối trực tiếp với máy tính nhưng để phát hiện và nhận dạng khói và cháy. Thành phần tiếp theo là khối xử lý trung tâm, khối này dùng để kết hợp các luồng thông tin đầu vào khác nhau và chúng tôi lựa chọn sử dụng mô đun Arduino Mega 2560. Khối thứ ba là khối tác động đầu ra. Trong bài báo này chúng tôi thiết kế tác động đầu ra khi phát hiện sự cố gồm: (1) Tác động và/hoặc cảnh báo tại chỗ với chuông báo, đèn báo, bơm cấp nước, động cơ thông gió, động cơ mở cửa, rơ le ngắt điện lưới; (2) Cảnh báo từ xa bằng cách báo tin tới ứng dụng cài đặt trên thiết bị di động.

2.2. Mạch thiết kế phát hiện khói và cháy sử dụng cảm biến vật lý

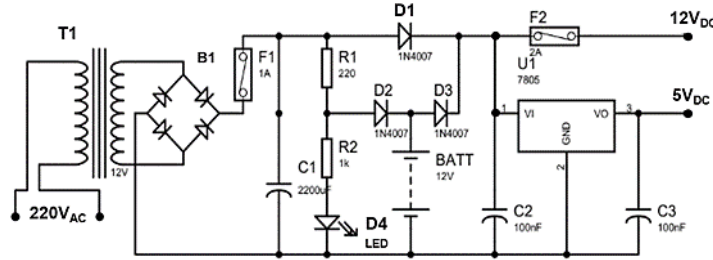
Hệ thống báo cháy và khói đa thể thức là một giải pháp tích hợp sử dụng nhiều loại cảm biến, thiết bị để phát hiện và phản ứng với các tình huống cháy nổ một cách hiệu quả. Cảm biến khói MQ7 phát hiện nồng độ khói trong không khí, cảm biến khí gas MQ2 nhận diện các khí gas dễ cháy như propane, butane và methane; cảm biến lửa Flame Sensor phát hiện ánh sáng hồng ngoại từ ngọn lửa. Khi một trong các cảm biến này phát hiện dấu hiệu bất thường, tín hiệu được gửi tới bộ điều khiển Arduino và bộ điều khiển kích hoạt quạt thông gió, động cơ mở cửa để giảm nồng độ khói, khí gas, và bơm nước để dập tắt lửa. Ngoài ra, hệ thống còn sử dụng mô đun SIM để gửi tin nhắn cảnh báo qua SMS đến các số điện thoại đã được lập trình sẵn, giúp người dùng nhận biết và có phương án kịp thời. Với các thành phần này, hệ thống không chỉ giúp phát hiện và xử lý tình huống cháy nổ một cách hiệu quả mà còn đảm bảo an toàn và giảm thiểu thiệt hại tối đa cho khu vực được bảo vệ. Trong hình 2 là hình ảnh thiết kế chi tiết hệ thống phát hiện khói và cháy sử dụng cảm biến vật lý.



Hình 2. Sơ đồ thiết kế chi tiết hệ thống phát hiện khói và cháy đa thể thức.

Trong các sự cố gây ra khói và cháy có nhiều nguyên nhân do cháy, nổ, hỏng và chập của các thiết bị điện tử trong các hộ gia đình. Khi có sự cố cháy xảy ra, hệ thống cấp điện lưới có thể bị ảnh hưởng (đứt dây, chập mạch,...). Để đảm bảo hệ thống vẫn thực hiện được các chức năng thì việc

duy trì nguồn nuôi là cần thiết. Do vậy, trong bài báo này chúng tôi thiết kế hệ thống UPS tự động chuyển nguồn dự phòng khi có sự cố xảy ra, được minh họa trong hình 3. Trong đó, sơ đồ mạch của UPS có thể cung cấp các điện áp $12V_{DC}$ và $5V_{DC}$ cho mạch điều khiển của hệ thống. Máy biến áp T1 lấy nguồn lưới $220V_{AC}$ hạ áp xuống $12V$ và qua cầu B1 chỉnh lưu tạo ra nguồn 1 chiều. Khi nguồn điện chính hoạt động, pin sẽ được sạc và thu được điện áp $12V_{DC}$ và $5V_{DC}$ ở các đầu ra. Khi có sự cố cháy xảy ra, sẽ gửi tín hiệu ngắt nguồn lưới và chuyển qua nguồn điện dự phòng BATT.

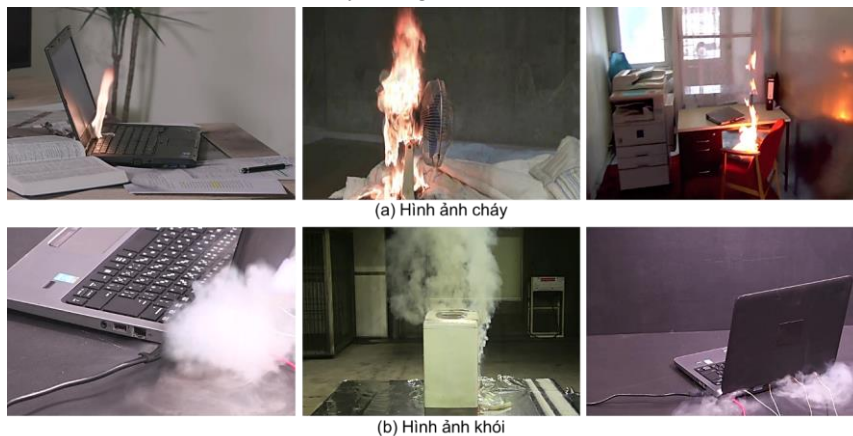


Hình 3. Sơ đồ thiết kế hệ thống chuyển mạch nguồn tự động (UPS).

2.3. Hệ thống phát hiện khói và cháy sử dụng thông tin hình ảnh

2.3.1. Cơ sở dữ liệu khói và cháy EPUFire

Trong nghiên cứu này, chúng tôi thực hiện cài đặt, thu thập bộ CSDL khói và cháy trong điều kiện trong nhà. Ngoài ra, chúng tôi thực hiện tìm kiếm trên mạng Internet các video, hình ảnh về sự cố khói và cháy trong nhà. Việc kết hợp từ nhiều luồng thông tin sẽ có được bộ CSDL đa dạng, phong phú về sự cố khói, cháy khác nhau tại nhiều thời gian và địa điểm khác nhau. Chúng tôi đặt tên bộ CSDL khói và cháy tự thu thập với tên là EPUFire. Hình vẽ 4 minh họa một số hình ảnh khói và cháy trong bộ CSDL EPUFire của chúng tôi. Bộ CSDL này có đặc điểm là có hình ảnh về các sự cố khói và cháy tại các ngữ cảnh trong nhà, điều kiện nền phức tạp và tại các thời điểm, địa điểm hoàn toàn khác nhau trong nhiều ngày. Bộ CSDL EPUFire thu dưới dạng video và hình ảnh. Với phần dữ liệu tự thu thập chúng tôi sử dụng máy quay là HiK-Vision DS-2CD2643G2-IZS cung cấp ảnh màu RGB với tốc độ thu thập 30 (fps), độ phân giải ảnh 2688×1520 (pixels). Mỗi video có độ dài 1 phút đến 10 phút. Với phần dữ liệu thu thập trên Internet, thông tin gồm cả video và hình ảnh tại nhiều độ phân giải khác nhau. Bộ CSDL tổng hợp này sau khi thu thập được chúng tôi chuyển sang dạng ảnh, lấy mẫu (3fps) và thực hiện gán nhãn bằng công cụ LabelImg với hai loại nhãn là khói và cháy. Tổng số hình ảnh được lấy mẫu để gán nhãn có 7424 ảnh có hình ảnh về cháy, 6939 ảnh có khói, 3240 ảnh có cả khói và cháy. Ngoài ra, nhóm tác giả cũng thu thập thêm các mẫu dữ liệu không có ảnh khói và cháy với số lượng ảnh là 2000 ảnh. Hình 4 sau đây minh họa một số hình ảnh về khói và cháy trong bộ CSDL EPUFire.

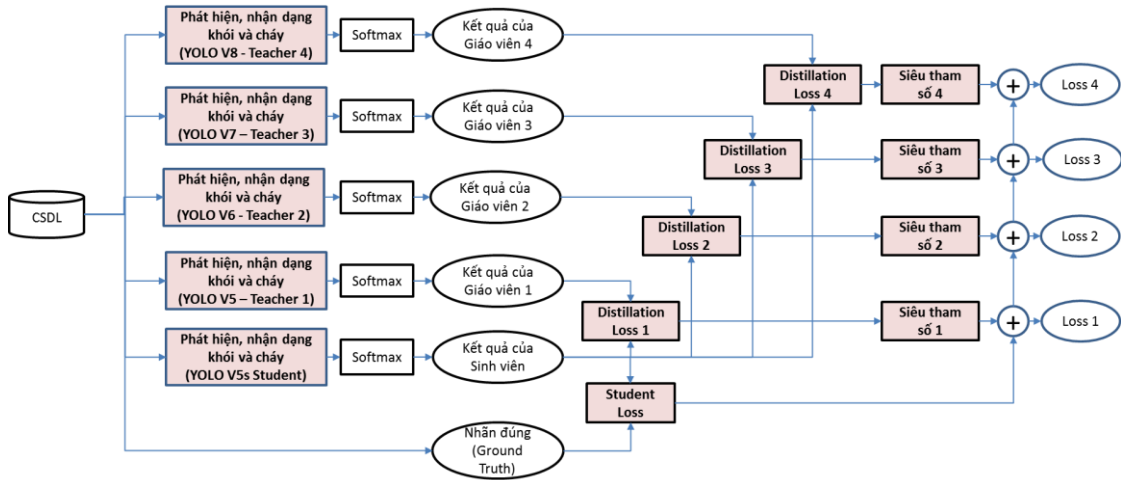


Hình 4. Minh họa một số hình ảnh trong bộ CSDL EPUFire.

Với bộ CSDL khói và cháy, chúng tôi thực hiện gán nhãn với ảnh chỉ có khói thì chỉ có một vùng ảnh khói được khoanh vùng và được gán nhãn là 0, với ảnh chỉ có cháy thì vùng ảnh chứa đám cháy được khoanh vùng và gán nhãn là 1. Ngoài ra, với ảnh có cả khói và cháy chúng tôi sẽ khoanh vùng khói và gán nhãn là 0, khoanh vùng cháy và gán nhãn là 1. Bộ CSDL này sẽ được sử dụng để đánh giá mô hình phát hiện khói và cháy bằng hình ảnh. Toàn bộ quá trình đánh giá định lượng được thực hiện chi tiết trong mục 3. Từ kết quả đánh giá, chúng tôi lựa chọn mô hình và sử dụng toàn bộ CSDL EPUFire này để huấn luyện mô hình thực tế.

2.3.2. Phần mềm phát hiện khói và cháy dùng thông tin hình ảnh

Trong phần phát hiện và nhận dạng khói và cháy sử dụng thông tin hình ảnh, chúng tôi sử dụng chất lọc tri thức cho mô hình YOLO V5 nano từ các phiên bản từ YOLO V5, YOLO V6, YOLO V7 và YOLO V8 như minh họa trong hình 5 sau đây.



Hình 5. Minh họa một số hình ảnh của bộ CSDL FireEPU.

Mô hình chất lọc tri thức chuyển giao trong hình 5 được thực hiện thông qua hàm mất mát cơ sở từ nghiên cứu [1]. Đây là hàm thực hiện kết hợp của hàm mất mát của sinh viên L_s (Student Loss) và hàm mất mát kết hợp giữa hai giáo viên thông qua các siêu tham số. Điểm khác biệt với [1] là trong nghiên cứu này chúng tôi thực hiện chuyển giao tri thức từ lần lượt bốn giáo viên để khảo sát mô hình chuyển giao nào phù hợp với CSDL khói và cháy trong hệ thống đề xuất của chúng tôi trong hình 1.

Trong lần chuyển giao và chất lọc tri thức đầu tiên giữa giáo viên 1 (YOLO V5) với học sinh L_{KD1} (Distillation Loss 1), các hàm mất mát cơ sở chúng tôi sử dụng là CE (Cross Entropy) [17], hàm mất mát tổng hợp của mô hình thứ nhất (KD1) được biểu diễn như trong công thức (1).

$$L_1(y, \hat{y}^s, \hat{y}^t) = L_s(y, \hat{y}^s) + \alpha_1 L_{KD1}(\hat{y}^{t_1}, \hat{y}^s) = \sum_i^N (y_i \log \hat{y}_i^s) + \alpha \sum_i^N (\hat{y}_i^{t_1} \log \hat{y}_i^s) \quad (1)$$

Hàm chuyển giao thứ hai thực hiện chất lọc tri thức từ hai giáo viên 1 (YOLO V5) và giáo viên 2 (YOLO V6) thông qua hàm mất mát tổng hợp của mô hình thứ hai (KD2) như biểu diễn trong công thức (2) sau đây:

$$\begin{aligned} L_2(y, \hat{y}^s, \hat{y}^t) &= L_1(y, \hat{y}^s, \hat{y}^t) + \alpha_2 L_{KD2}(\hat{y}^{t_2}, \hat{y}^s) = L_s(y, \hat{y}^s) + \alpha_1 L_{KD1}(\hat{y}^{t_1}, \hat{y}^s) \\ &\quad + \alpha_2 L_{KD2}(\hat{y}^{t_2}, \hat{y}^s) \\ &= \sum_i^N (y_i \log \hat{y}_i^s) + \alpha_1 \sum_i^N (\hat{y}_i^{t_1} \log \hat{y}_i^s) + \alpha_2 \sum_i^N (\hat{y}_i^{t_2} \log \hat{y}_i^s) \end{aligned} \quad (2)$$

Hàm chuyển giao thứ ba thực hiện chất lọc tri thức từ ba giáo viên 1 (YOLO V5), giáo viên 2 (YOLO V6) và giáo viên 3 (YOLO V7) thông qua hàm mất mát tổng hợp của mô hình thứ ba (KD3) như biểu diễn trong công thức (3) sau đây:

$$\begin{aligned}
 L_3(y, \widehat{y}^s, \widehat{y}^t) &= L_2(y, \widehat{y}^s, \widehat{y}^t) + \alpha_3 L_{KD3}(\widehat{y}^{t3}, \widehat{y}^s) = L_s(y, \widehat{y}^s) + \alpha_1 L_{KD1}(\widehat{y}^{t1}, \widehat{y}^s) \\
 &\quad + \alpha_2 L_{KD2}(\widehat{y}^{t2}, \widehat{y}^s) + \alpha_3 L_{KD3}(\widehat{y}^{t3}, \widehat{y}^s) \\
 &= \sum_i^N (y_i \log \widehat{y}_i^s) + \alpha_1 \sum_i^N (y_i^{t1} \log \widehat{y}_i^s) + \alpha_2 \sum_i^N (y_i^{t2} \log \widehat{y}_i^s) + \alpha_3 \sum_i^N (y_i^{t3} \log \widehat{y}_i^s)
 \end{aligned} \tag{3}$$

Hàm chuyển giao thứ tư thực hiện chất lọc tri thức từ cả bốn giáo viên gồm: giáo viên 1 (YOLO V5), giáo viên 2 (YOLO V6), giáo viên 3 (YOLO V7) và giáo viên 4 (YOLO V8) thông qua hàm mất mát tổng hợp của mô hình thứ tư (KD4) như biểu diễn trong công thức (4) sau đây:

$$\begin{aligned}
 L_4(y, \widehat{y}^s, \widehat{y}^t) &= L_3(y, \widehat{y}^s, \widehat{y}^t) + \alpha_4 L_{KD4}(\widehat{y}^{t4}, \widehat{y}^s) = L_s(y, \widehat{y}^s) + \alpha_1 L_{KD1}(\widehat{y}^{t1}, \widehat{y}^s) \\
 &\quad + \alpha_2 L_{KD2}(\widehat{y}^{t2}, \widehat{y}^s) + \alpha_3 L_{KD3}(\widehat{y}^{t3}, \widehat{y}^s) + \alpha_4 L_{KD4}(\widehat{y}^{t4}, \widehat{y}^s) \\
 &= \sum_i^N (y_i \log \widehat{y}_i^s) + \alpha_1 \sum_i^N (y_i^{t1} \log \widehat{y}_i^s) + \alpha_2 \sum_i^N (y_i^{t2} \log \widehat{y}_i^s) \\
 &\quad + \alpha_3 \sum_i^N (y_i^{t3} \log \widehat{y}_i^s) + \alpha_4 \sum_i^N (y_i^{t4} \log \widehat{y}_i^s)
 \end{aligned} \tag{4}$$

2.3.3. Thang đo đánh giá (Metric)

Để đánh giá hiệu quả của mô hình YOLO, chúng tôi sử dụng các chỉ tiêu độ chính xác P (Precision), độ triệu hồi R (Recall) và độ chính xác trung bình mAP (Mean Average Precision). Trong đó, độ chính xác thu được trên tập dữ liệu thử nghiệm là thước đo đánh giá độ hiệu quả của mô hình qua các thông số độ chính xác P và độ triệu hồi R. Độ chính xác P được định nghĩa là tỉ lệ của vùng khối và cháy phát hiện từ hệ thống chia cho vùng khối và cháy đúng (Ground Truth) minh họa bởi công thức (5) sau đây:

$$P = \frac{TP}{TP + FP} \tag{5}$$

Độ triệu hồi R được định nghĩa là tỉ lệ của vùng khối và cháy phát hiện đúng trên tổng số vùng có trong ảnh như minh họa bởi công thức (6) sau đây:

$$R = \frac{TP}{TP + FN} \tag{6}$$

Mỗi lớp sẽ tính được độ chính xác trung bình trên K mẫu cho một lần thử nghiệm là:

$$AP_i = \frac{\sum_{k=1}^K P_k}{K}$$

Như vậy, độ chính xác trung bình mAP của N = 2 lớp (tương ứng với lớp khối và lớp cháy) cho mỗi lần thử nghiệm được định nghĩa theo công thức (7) sau đây:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \tag{7}$$

Trong đó, TP (True Positive) là tổng số dự đoán đúng được phân lớp vào đúng; FP (False Positive) tổng số dự đoán sai được phân lớp vào đúng; FN (False Negative) tổng số dự đoán sai được phân lớp vào sai; và TN (True Negative) là tổng số dự đoán đúng được phân lớp vào sai, N là tổng số lớp. Kết quả đánh giá hệ thống sẽ được thực hiện và trình bày chi tiết trong mục 3.

2.3.4. Giao thức (Protocol)

Trong bài báo này, chúng tôi chia bộ CSDL EPUFire thành 10 phần bằng nhau. Sau đó, chúng

tôi sử dụng phương pháp “Leave-one-subject-out” [1, 2] để thực hiện 10 lần thử nghiệm. Trong đó, tại mỗi lần thử nghiệm, tám phần dữ liệu được sử dụng để huấn luyện mô hình, một phần để validation và một phần để đánh giá (test) mô hình. Tại mỗi lần thử nghiệm này sẽ có độ chính xác P_j, R_j, mAP_j ($j=(1,..,M), M=10$). Độ chính xác của một CSDL đạt được bằng cách lấy trung bình của 10 lần đánh giá mô hình với độ chính xác P (Precision) như trong công thức (8) sau đây:

$$P = \frac{\sum_{j=1}^{M=10} P_j}{M} \quad (8)$$

Độ triệu hồi R (recall) của cả cơ sở dữ liệu được tính như trong công thức (9) sau đây:

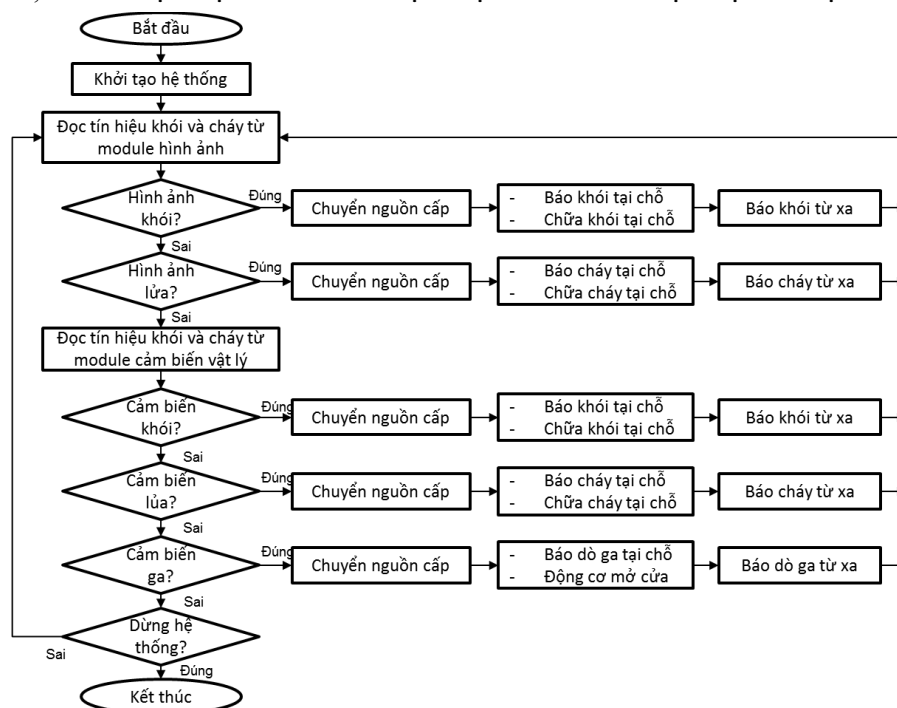
$$R = \frac{\sum_{j=1}^{M=10} R_j}{M} \quad (9)$$

Độ chính xác trung bình mAP được tính trung bình cho tất cả các lần thử nghiệm như trong công thức (10) sau:

$$mAP = \frac{\sum_{j=1}^{M=10} mAP_j}{M} \quad (10)$$

2.4. Lưu đồ thuật toán hệ thống phát hiện khói và cháy đa thể thức

Lưu đồ thuật toán của hệ thống được minh họa như trong hình 6 sau đây. Trong đó, hệ thống phát hiện khói và cháy bằng hình ảnh được ưu tiên phát hiện và tác động cao hơn hệ thống dùng cảm biến vật lý. Bất cứ khi nào phát hiện có sự cố khói hoặc cháy thì sẽ thực hiện chuyển nguồn cấp đầu tiên, sau đó thực hiện báo và chữa sự cố tại chỗ rồi mới thực hiện báo sự cố từ xa.



Hình 6. Lưu đồ thuật toán hệ thống phát hiện khói và cháy đa thể thức.

3. KẾT QUẢ THỬ NGHIỆM

Phần mềm thực hiện trên máy tính CPU Intel Core i5-11400H, GPU NVIDIA GeForce GTX 1650, bộ nhớ Ram 8 GB. Ngôn ngữ lập trình sử dụng là ngôn ngữ Python. Thử nghiệm cài đặt với kích thước mẻ (batch size) là 64; tốc độ huấn luyện mô hình (learning rate) là 10^{-4} ; số lần lặp (epochs) là 100. Các thử nghiệm thực hiện sử dụng cho các giáo viên 1 (YOLO V5), 2 (YOLO

V6), 3 (YOLO V7), và 4 (YOLO V8); sinh viên là mô hình YOLO V5n. Các đánh giá được tiến hành gồm: (1) Độ chính xác phát hiện khói và cháy trên các phiên bản YOLO khác nhau; (2) Độ chính xác phát hiện khói và cháy trên các phiên bản chuyển giao tri thức (KD) từ các phiên bản YOLO khác nhau; (3) Thời gian đáp ứng của mô hình KD (Time cost).

3.1. Độ chính xác phát hiện khói và cháy với mô hình YOLO

Trong phần này chúng tôi sẽ thực hiện đánh giá độ chính xác độc lập của từng phiên bản YOLO trên bộ CSDL khói và cháy EPUFire gồm: YOLO V5n, YOLO V5-V8. Đây là những phiên bản mới nhất của mô hình YOLO đến thời điểm hiện tại. Kết quả được thể hiện với ba tiêu chí đánh giá là Precision, Recall và mAP như trong bảng 1 sau đây.

Bảng 1. Độ chính xác phát hiện khói và cháy bằng hình ảnh.

Mô hình	Precision (%)	Recall (%)	mAP (%)
YOLO V5n	89.2	86.1	88.6
YOLO V5	96.5	96.7	96.0
YOLO V6	97.5	96.4	97.0
YOLO V7	97.6	98.2	98.7
YOLO V8	98.3	97.7	98.1

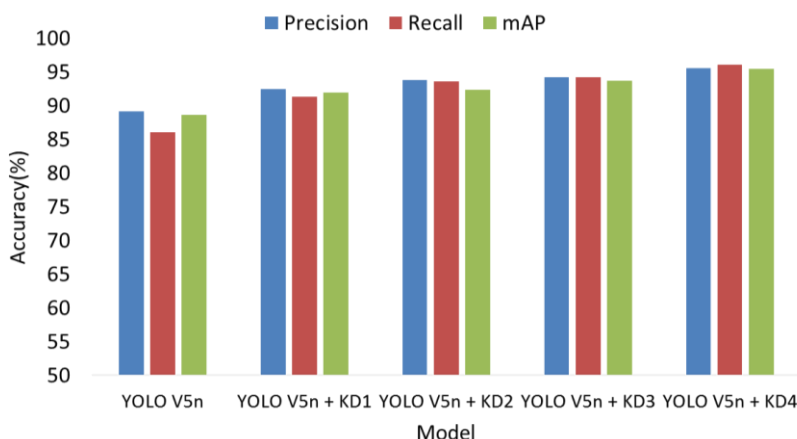
Kết quả trong bảng 1 này cho thấy:

- Các phiên bản YOLO V5n với cấu hình nhẹ thì đạt kết quả thấp hơn hẳn (89.2%, 86.1% và 88.6% trong hàng 2 của bảng 1) so với phiên bản YOLO V5 cấu hình đầy đủ (96.5%, 96.7%, và 96.0% trong hàng 3 của bảng 1).

- Các mô hình YOLO cấu hình đầy đủ có phiên bản cao hơn đạt độ chính xác cao hơn so với phiên bản thấp hơn trên cả ba tiêu chí đánh giá Precision, Recall và mAP (Từ hàng 3 đến hàng 6 của bảng 1). Trong đó, YOLO V8 đạt kết quả cao nhất trong hàng 8 của bảng 1 các giá trị (98.3%, 97.7% và 98.1%).

- Các mô hình YOLO V5-V8 có độ chính xác cao này sẽ được sử dụng với vai trò lần lượt là các giáo viên để huấn luyện chuyển giao tri thức cho mô hình sinh viên có cấu hình rút gọn YOLO V5n. Chi tiết đánh giá được thực hiện trong mục 3.2 sau đây.

3.2. Độ chính xác phát hiện khói và cháy dùng mô hình chất lọc tri thức KD



Hình 7. Độ chính xác của mô hình YOLO V5n và YOLO V5n - KD.

Hình 7 biểu diễn các kết quả của mô hình YOLO V5n với bản nguyên gốc và bản có sử dụng các chiến lược học tập có chất lọc tri thức chuyên gia. Trong đó, ba cột đầu là kết quả YOLO V5n gốc sẽ có kết quả thấp nhất, lần lượt ba cột tiếp theo lần lượt là kết quả của mô hình YOLO V5n có học tri thức chuyên gia KD1 đến KD4. Kết quả cho thấy:

- Sử dụng học chuyển giao có chất lọc tri thức chuyên gia cải thiện kết quả trên cả ba tiêu chí đánh giá cho mô hình YOLO V5n. Các cột YOLO V5n-KD1 đến Các cột YOLO V5n-KD4 đạt giá trị cao hơn ba cột đầu tiên (YOLO V5n).

- Việc sử dụng càng nhiều chuyên gia khi học chuyển giao thì hiệu quả của mô hình YOLO V5n KD đạt được kết quả cao hơn YOLO V5n. Trong đó, Precision tăng từ 3.3% đến 6.4%, Recall tăng từ 5.2% đến 10% và mAP tăng từ 3.3% đến 6.9% khi sử dụng lần lượt từ 1 đến 4 giáo viên. Từ kết quả trên, chúng tôi thực hiện mô hình chất lọc tri thức từ 04 giáo viên YOLO V5n-KD4 để triển khai cho mô hình hệ thống phát hiện sự cố khói và sự cố cháy thông minh đa thể thức thực tế tại địa chỉ [16] cũng như thực hiện đánh giá thời gian đáp ứng như trình bày tại mục 3.3 sau đây.

3.3. Thời gian đáp ứng của mô hình phát hiện khói và cháy

Phần này chúng tôi thực hiện tính toán thời gian đáp ứng cho phát hiện khói và cháy trên từng khung hình và so sánh giữa phiên bản YOLO V5n-KD4 (hàng 2 - bảng 2) với các mô hình đầy đủ từ YOLO V5 (hàng 3 - bảng 2) đến YOLO V8 (hàng 6 - bảng 2) khi chạy trên máy tính sử dụng CPU Intel Core i5-11400H, GPU NVIDIA GeForce GTX 1650, bộ nhớ Ram 8 GB.

Bảng 2. Thời gian đáp ứng cho từng khung hình trên các phiên bản YOLO khác nhau.

Mô hình	Tốc độ xử lý với CPU	Tốc độ xử lý với GPU
YOLO V5n + KD4	26.25 ms	15.17 ms
YOLO V5	74.45 ms	22.18 ms
YOLO V6	75.37 ms	21.57 ms
YOLO V7	77.81 ms	23.39 ms
YOLO V8	76.69 ms	22.19 ms

Kết quả cho thấy mô hình YOLO V5n – KD4 có thời gian đáp ứng cho mỗi khung hình nhanh hơn (với 26,25 (ms) khi chạy trên CPU và 15,17 (ms) khi thực hiện trên GPU) so với các phiên bản đầy đủ YOLO V5 – V8. Mô hình YOLO V7 có thời gian đáp ứng cao nhất trên cả CPU và GPU (với 77,81 (ms) khi chạy trên CPU và 23,39 (ms) khi thực hiện trên GPU).

Với kết quả trên, chúng tôi sử dụng mô hình YOLO V5n + KD4 đã huấn luyện trên toàn bộ CSDL khói và cháy EPUFire. Trong đó mô hình huấn luyện trên máy tính sử dụng CPU Intel Core i5-11400H, GPU NVIDIA GeForce GTX 1650, bộ nhớ Ram 8 GB. Sau đó, mô hình YOLO V5n được cài đặt trên Jetson Nano A02, CPU 4 core 2,1GHz, RAM 4GB, GPU 128 core Maxwell. Camera sử dụng cho hệ thống thực tế là loại Raspberry Pi Camera Module V3. Độ phân giải 1920x1080 (pixels), tốc độ thu thập ảnh của camera này là 50 (fps). Hệ thống phát hiện khói và cháy bằng hình ảnh sử dụng mô hình YOLO V5n của chúng tôi đề xuất khi cài đặt trên Jetson Nano a02 có thời gian đáp ứng thực tế đo được xấp xỉ 27,85 (fps).

4. KẾT LUẬN

Bài báo đã thiết kế hệ thống báo cháy, báo khói thông minh đa thể thức kết hợp cảm biến vật lý và cảm biến hình ảnh. Hệ thống đề xuất của chúng tôi có khả năng chuyển nguồn tự động, giúp tăng tính an toàn của mạch, kéo dài thời gian giúp hệ thống báo, khắc phục và chữa cháy khi có phát hiện sự cố cháy nổ. Chúng tôi đã định nghĩa, thu thập và xử lý được một bộ CSDL về khói và cháy trong điều kiện trong nhà có tên là EPUFire. Ngoài ra, nghiên cứu cũng đề xuất một giải pháp để xử lý hình ảnh sử dụng mô hình YOLO V5n có thời gian đáp ứng nhanh xấp xỉ 26 (ms) trên CPU và 15 (ms) trên GPU. Trong thời gian sắp tới, chúng tôi sẽ tiếp tục thu thập thêm dữ liệu và đánh giá hệ thống thực trên nhiều dữ liệu khói và cháy hơn. Chúng tôi sẽ tìm cách kết hợp thông tin ảnh màu (RGB) với thông tin chuyển động của khói và cháy để nâng cao hiệu quả của quá trình phát hiện. Tích hợp hệ thống để có mô hình nhỏ gọn hơn, dễ dàng lắp đặt và triển khai trong thực tế.

Lời cảm ơn: Nghiên cứu này được hỗ trợ bởi đề tài cnkh của người học năm 2024 mã số ĐTNH.39/2024 cấp Trường Đại học Điện lực.

TÀI LIỆU THAM KHẢO

- [1]. Đoàn Thị Hương Giang, Hồ Anh Dũng, Nguyễn Ngọc Trung, Nguyễn Trung Hiếu, “*Improvement of performance of human detection in abnormal crowd using knowledge distillation for YOLO network*,” Tạp chí Khoa học và Công nghệ Trường Đại học Công nghiệp Hà Nội, Tập 60 - Số 4, pp. 39-44, (2024), doi: 10.57001/huinh5804.2024.124.
- [2]. Huong-Giang Doan, Ngoc-Trung Nguyen, “*New blender-based augmentation method with quantitative evaluation of CNNs for hand gesture recognition*”, Indonesian Journal of Electrical Engineering and Computer Science, Vol. 30, No. 2, pp. 796-806, (2023). DOI: 10.11591/ijeecs.v30.i2.pp796-806, (2023).
- [3]. Redmon Joseph, Divvala Santosh, Girshick Ross, Farhadi Ali. “*YOLOv1: You Only Look Once: Unified, Real-Time Object Detection*”. 779-788. 10.1109/CVPR.2016.91, (2016).
- [4]. Redmon Joseph, Farhadi Ali. “*YOLO9000: Better, Faster, Stronger*”. 6517-6525. 10.1109/CVPR.2017.690, (2017).
- [5]. J. Redmon, A. Farhadi. “*Yolov3: An incremental improvement*”. CoRR journal, Vol. abs/1804.02767, pp. 1-6, (2018).
- [6]. A. Bochkovskiy, C.Y. Wang, H.Y. M. Liao. “*Yolov4: Optimal speed and accuracy of object detection*”. ArXiv journal, Vol. abs/2004.10934, 2020, pp. 1-17, (2004).
- [7]. Glenn Jocher. “*Yolov5 in pytorch*”. <https://github.com/ultralytics/yolov5>, (2020).
- [8]. Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, Yiduo Li, Bo Zhang, Yufei Liang, Linyuan Zhou, Xiaoming Xu, Xiangxiang Chu, Xiaoming Wei, Xiaolin Wei. “*YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. Computer Vision and Pattern Recognition*”, (2022).
- [9]. Wang Chien-Yao, Bochkovskiy Alexey, Liao Hong-yuan. “*YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*”. 10.48550/arXiv.2207.02696, (2022).
- [10]. Jocher, G.; Chaurasia, A.; Qiu, J. “*YOLO by Ultralytics*,”. GitHub. (2023). Available online: <https://github.com/ultralytics/ultralytics>.
- [11]. Wang, C., Yeh, I., & Liao, H. “*YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information*”. ArXiv, abs/2402.13616, (2024).
- [12]. <https://thietbidienthongminhata.com/>
- [13]. R. Y. Rubinstein, “*Optimization of computer simulation models with rare events*,” European Journal of Operational Research, vol. 99, no. 1, pp. 89–112, (1997).
- [14]. G. E. Hinton, O. Vinyals, and J. Dean, “*Distilling the knowledge in a neural network*,” ArXiv, vol. abs/1503.02531, (2015).
- [15]. <http://canhsatpccc.gov.vn/Home/tabid/40/language/vi-VN/default.aspx>
- [16]. <https://youtu.be/yedwW8nTQsU>

ABSTRACT

Multimodality fire and smoke detection system

Early smoke and fire detection is extremely important for humans and property to prevent serious consequences. A common solution is to utilize physical sensors such as gas detection sensors, smoke detection sensors, and temperature detection sensors caused by fire. However, the detection time of physical sensors is slower than combining multiple cues, especially combining with computer vision. In this paper, we propose a multi-modal fire and smoke detection solution that combines physical sensors (Sensor) and image sensors (Camera). In particular, our proposed method applies artificial intelligence (AI) and Internet of Things (IoT) to detect smoke and fire in the indoor environment. The knowledge distillation algorithm (KD) transfers from the full version of YOLO teacher models to the reduced version of YOLO model, whose detection accuracy is 10% smaller than the full version. The KD model is simpler, so it has a faster response time than the full model up to 8.22 (ms) and 51.56 (ms) when it runs on GPU and CPU, respectively.

Keywords: Convolution neural network; Deep learning; Fire warning; Sensor; Fire detection; Multi modalities.