

Nghiên cứu và phát triển hệ thống cảnh báo xâm nhập sử dụng thuật toán trí tuệ nhân tạo tiên tiến

Research and development of an intrusion warning system using advanced artificial intelligence algorithms

Đặng Hoàng Phúc, Lê Anh Uyên Vũ, Hoàng Đức Quý*

Trường Đại học Giao thông vận tải Thành phố Hồ Chí Minh

*Tác giả liên hệ: quyhd@ut.edu.vn

Ngày nhận bài: 2/10/2023; Ngày chấp nhận đăng: 13/12/2023

Tóm tắt:

Trước các vấn nạn về trộm cắp đang diễn ra hết sức phức tạp, việc có một hệ thống cảnh báo xâm nhập, đặc biệt tại các công trường đang thi công là vô cùng cấp thiết. Trong bài báo này, nhóm tác giả xây dựng một hệ thống cảnh báo xâm nhập tự động và chính xác khi có người xâm nhập vào một khu vực trong các công trường xây dựng. Cụ thể, sử dụng các mô hình học sâu tiên tiến như YOLOv5 và YOLOv8 để có được tọa độ của vật thể, sau đó, tọa độ của đối tượng được tính toán và so sánh với tọa độ khu vực cần giám sát để xác định hành vi vi phạm. Kết quả đạt được trong nghiên cứu này rất tốt, khi mô hình YOLOv5n đạt được độ chính xác trung bình là hơn 91% với độ nhạy hơn 84% và tốc độ xử lý hơn 12 khung hình trên giây, tương tự với mô hình YOLOv8n đạt được độ chính xác trung bình hơn 92% với độ nhạy hơn 82% và tốc độ xử lý hơn 15 khung hình trên giây.

Từ khóa: YOLO; Hệ thống báo động; Phát hiện vật thể; Phát hiện xâm nhập.

Abstract:

Given the extremely complicated problems of theft, having an intrusion warning system, especially at construction sites, is extremely urgent. In this study, we will build an automatic intrusion warning system when someone enters an area on a construction site automatically and accurately. Specifically, we used advanced deep learning models such as YOLOv5 and YOLOv8 to obtain the coordinates of the object and then compared them with the coordinates of the monitored area to determine whether the conduct was a violation. The results achieved in this study were very good when the YOLOv5n model achieved an average accuracy of more than 91% with a sensitivity of more than 84% and a processing speed of more than 12 frames per second, similar to the YOLOv8n model that achieved an average accuracy of more than 92%, with a sensitivity of more than 82% and a processing speed of more than 15 frames per second.

Keywords: YOLO; Alarm system; Object detection; Intrusion detection.

1. Giới thiệu

Hiện nay, vấn nạn trộm cắp trong các công trường xây dựng vẫn là một vấn đề nghiêm trọng đang diễn ra tại nhiều quốc gia trên toàn thế giới. Điều này gây ra thiệt hại lớn về tài sản, đồng thời, ảnh hưởng đến sự an toàn khi có thể gây tai nạn cho các đối tượng xâm nhập bởi điều kiện thi công của các công trường là vô cùng nguy hiểm và tiềm ẩn nhiều rủi ro. Mặt khác,

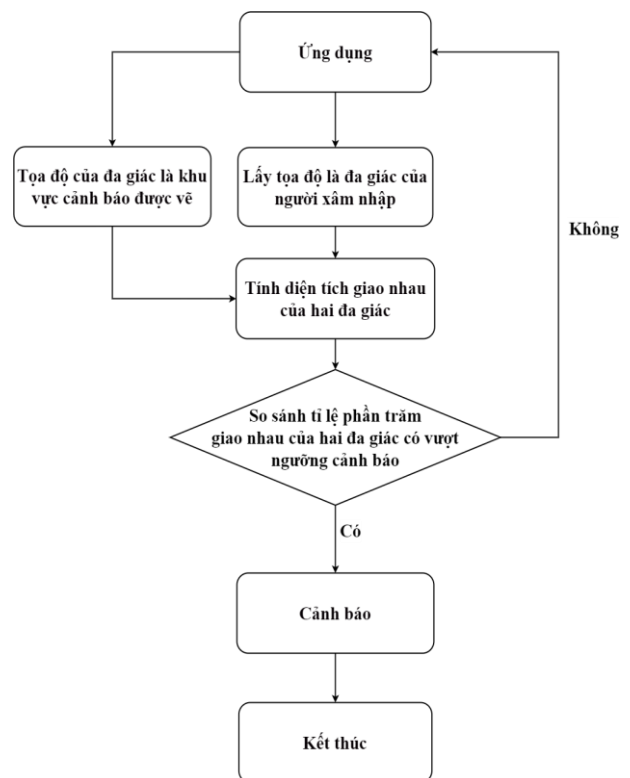
tội phạm đã ngày càng trở nên tinh vi và thông minh hơn, khiến cho tình trạng trộm cắp trở nên phức tạp. Với sự phát triển của công nghệ số, vấn đề trộm cắp đang trở thành mối đe dọa ngày càng lớn. Từ đó, đặt ra các bài toán ngăn chặn và xử lý kịp thời những hành vi trộm cắp, tuy nhiên, để giải quyết vấn đề này cần đến một nguồn nhân lực và phải luôn nêu cao cảnh giác, chủ động trong việc bảo vệ tài sản. Từ đó, có

thể thấy, hệ thống cảnh báo xâm nhập là một giải pháp hiệu quả cho việc giám sát xâm nhập tự động, qua đó, có thể phát hiện kịp thời cũng như ngăn chặn các hành vi bất hợp pháp từ các đối tượng vi phạm một cách tức thời.

Một số nghiên cứu đã được đề xuất cho việc xác định sự xâm nhập tại khu vực công trường xây dựng và đường ray [1-10], các tác giả sử dụng những phương pháp gần giống nhau. Tiêu biểu như Tran và các cộng sự [1] đặt ra bài toán phát hiện con người xâm nhập vào khu vực nguy hiểm sử dụng mô hình YOLOv8, kết hợp phương pháp nhận diện xâm nhập bằng cách xác định một điểm nằm trên hộp giới hạn bao xung quanh vật thể và kiểm tra tọa độ của điểm đó trên những đường thẳng hoặc bên trong khu vực cần cảnh báo. Ngoài ra, Zhao và cộng sự [2] đã đề xuất một số giải pháp cảnh báo xâm nhập sớm dựa trên việc phân loại khu vực nguy hiểm của công trình theo khu vực làm việc của từng loại công việc cụ thể đối với công nhân xây dựng. Hu và các cộng sự [3] đề xuất cách giám sát theo thời gian thực cho các khu vực nguy hiểm trong công trường xây dựng dựa trên mô hình YOLOv5 và căn cứ ngưỡng các phần diện tích giao nhau để cảnh báo, cũng như phân loại công nhân có hoặc không trang bị đồ bảo hộ khi lao động. Một số phương pháp khác trong [11-15], tiêu biểu như Zhang và các cộng sự [12] đề xuất một số phương pháp khi phát hiện vật thể chuyển động trong khung hình và cảnh báo các vật thể bị bỏ rơi. Nhìn chung, các phương pháp kể trên đều hiện hữu các hạn chế nhất định, chẳng hạn, khả năng linh hoạt của các phương pháp cảnh báo chưa thực sự tối ưu, không cho phép tùy chỉnh các ngưỡng xâm nhập, không thể hiệu chỉnh độ nhạy của hệ thống, từ đó, giúp kiểm soát khả năng gửi cảnh báo nhiều lần một cách hiệu quả.

Trong nghiên cứu này, nhóm tác giả sử dụng một phương pháp khác nhưng đảm bảo sự chính xác cho hệ thống, đó là kết hợp mô hình phát hiện vật thể tiên tiến YOLOv5 [16] và YOLOv8 [17] với phương pháp xác định diện tích giao nhau của hai đa giác để nhận diện xâm nhập của

đối tượng vi phạm. Cụ thể, nếu ngưỡng giao nhau của hai đa giác đạt một ngưỡng nhất định được cài đặt từ trước thì hệ thống thực hiện gửi các cảnh báo như âm thanh và hình ảnh của đối tượng đến thiết bị của người giám sát. Hai đa giác ở đây chính là một đa giác bao quanh vật thể xác định chính xác vị trí và kích thước của vật thể được mô hình phát hiện, đa giác thứ hai là khu vực mong muốn giám sát và thực hiện gửi các cảnh báo. Quan sát Hình 1, có thể thấy phương pháp này xác định sự xâm nhập bằng cách tính toán phần diện tích của hộp giới hạn xung quanh vật thể khi YOLO phát hiện, sau đó, tiếp tục tính toán phần diện tích cắt nhau nằm bên ngoài của hình hộp so với đa giác cần cảnh báo. Sau cùng, thay các phần diện tích vào một công thức và đầu ra chính là phần diện tích vi phạm hay ngưỡng xâm nhập của các vật thể đang vi phạm.



Hình 1. Quá trình phát hiện xâm nhập.

2. Phương pháp nghiên cứu

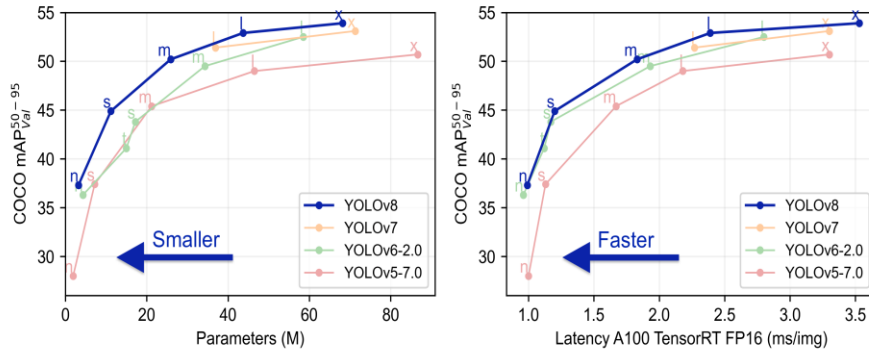
2.1. Mô hình YOLOv8

YOLO (You Only Look Once) là một thuật toán phát hiện đối tượng tiên tiến trong lĩnh vực thị giác máy tính. Với phiên bản mới nhất là

YOLOv8 đã được giới thiệu vào đầu năm 2023 và được cải tiến cấu trúc để đạt kết quả phát hiện đối tượng tốt hơn. YOLOv8 có độ chính xác cao hơn so với các phiên bản trước đó, được đánh giá trên tập dữ liệu COCO và Roboflow 100. Ngoài ra, YOLOv8 còn hỗ trợ đầy đủ các tác vụ về thị giác máy tính như phát hiện vật thể, phân đoạn, ước tính tư thế, theo dõi và phân loại vật thể. Bên cạnh đó, thuật toán này có sự hỗ trợ từ cộng đồng người dùng lớn của giới

chuyên gia trong ngành, điều này, khiến việc sử dụng nó trở nên thuận tiện hơn.

Quan sát Hình 2, với phiên bản YOLOv8m có kích thước trung bình, thế nhưng, nó đạt hơn 50% mAP khi đánh giá trên tập dữ liệu COCO, so với phiên bản YOLOv5m chỉ đạt hơn 45%. Điều này cho thấy, YOLOv8 đạt độ chính xác cao hơn rất nhiều so với các phiên bản được ra mắt trước đây.

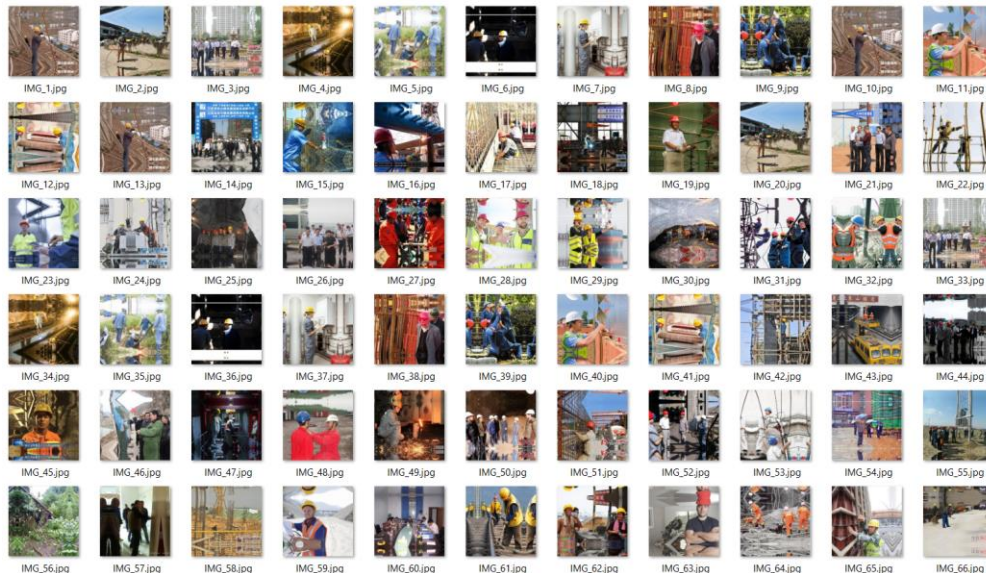


Hình 2. So sánh mô hình YOLOv5 [16] và YOLOv8 [17].

2.2. Tập dữ liệu

Sau khi thu thập các tập dữ liệu người và công nhân xây dựng khác nhau từ tập dữ liệu Penn-Fudan [18] và tập dữ liệu SHWD [19], sẽ có

được thư mục hình ảnh như Hình 3 và dán nhãn cho ảnh. Sau khi hoàn thành, có thể sử dụng 2000 hình ảnh và chia theo tỷ lệ 6:2:2 với 60% dùng cho việc huấn luyện và 40% còn lại dùng để đánh giá mô hình.



Hình 3. Tập dữ liệu huấn luyện tùy chỉnh.

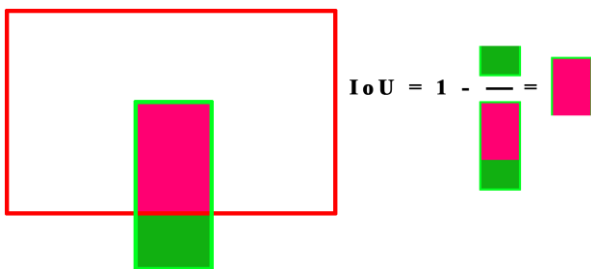
Cụ thể, sử dụng 1200 ảnh (train) trong tập dữ liệu được dán nhãn để huấn luyện mô hình và 400 ảnh (val) còn lại để đánh giá mô hình, 400 ảnh khác (test) đánh giá mô hình sau cùng. Kích

thước ảnh 640 và được chia nhỏ kích thước batch là 16, tiếp theo, tiến hành chạy quá trình huấn luyện trong vòng 100 epoch. Huấn luyện với cả hai mô hình khác nhau là YOLOv5 và

YOLOv8, sau đó, chọn ba mô hình thấp nhất ở hai phiên bản để tiến hành đào tạo và lấy các kết quả thực hiện đánh giá và sử dụng.

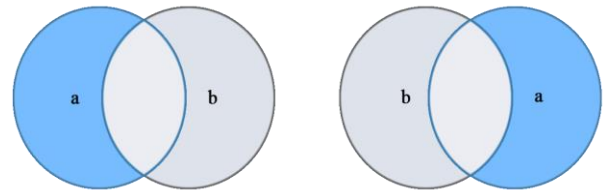
2.3. Phương pháp phát hiện xâm nhập

Phương pháp phát hiện xâm nhập được đề xuất bằng cách tính diện phần khác biệt của hai đa giác khi chúng giao nhau cùng với diện tích của hình hộp chữ nhật bao quanh vật thể và có được tỷ lệ giao nhau của hai đa giác đó. Cụ thể, quát sát Hình 4, khi một hộp giới hạn của vật thể tiến vào bên trong đa giác màu đỏ là khu vực cảnh báo, chúng tách ra thành các phần nằm bên trong và bên ngoài khu vực cảnh báo. Để tính toán cần lấy phần diện tích nằm bên ngoài (màu xanh lục) chia cho diện tích của toàn bộ hộp giới hạn của vật thể, sau đó, lấy giá trị 1 trừ tỷ lệ phép chia, sẽ có được một tỷ lệ giao nhau hay chính là phần diện tích đối tượng vi phạm (phần màu hồng) so với khu vực đang cần giám sát.



Hình 4. Diện tích giao nhau của hai đa giác.

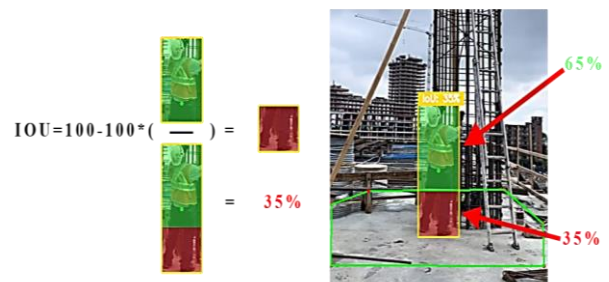
Với phương pháp này cần tính được phần diện tích khác biệt giữa hai đa giác, có thể hiểu đó là phần hộp giới hạn được cắt và nằm bên ngoài đa giác cần cảnh báo hay nói cách khác là hiệu của phần hộp giới hạn trừ đi phần diện tích là khu vực cần cảnh báo. Phép chia của của phần diện tích nằm bên ngoài với diện tích của hộp giới hạn luôn trong khoảng từ 0 đến 1. Khi đó, phần diện tích còn lại chính là phần diện tích đang bị đối tượng xâm phạm và tính phần tỷ lệ giao nhau còn lại. Bằng cách lấy giá trị 1 trừ cho tỷ lệ của hình hộp cắt bên ngoài sẽ tính được tỷ lệ giao nhau của hai đa giác. Tại Hình 5 có thể thấy, phần diện tích cần tính chính là phần diện tích a (màu xanh lam).



Hình 5. Xác định phần khác biệt của đa giác.

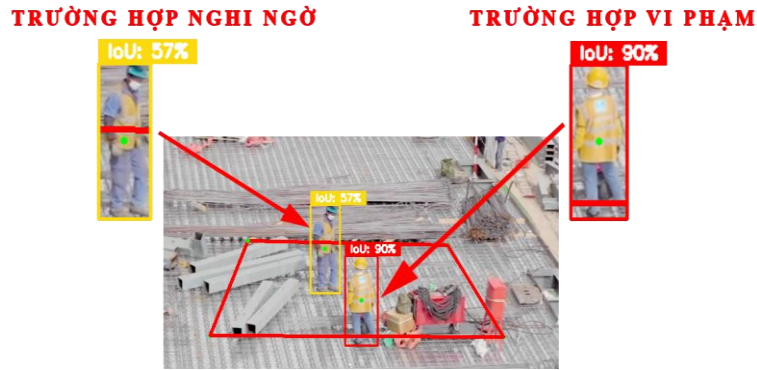
Quan sát Hình 6, sau khi tính toán phần tỷ lệ diện tích giao nhau, sẽ trả về một giá trị giao của hai đa giác. Cần chuyển đổi các giá trị tỷ lệ này sau khi tính toán sang giá trị phần trăm để làm cơ sở cho ngưỡng cảnh báo xâm nhập. Nhân các giá trị này với 100, từ đó, có thể chuyển đổi các tỷ lệ diện tích của các đối tượng vi phạm sang phần trăm và các giá trị phần trăm được sử dụng làm cơ sở so sánh cho hệ thống cảnh báo.

IOU= 100%- DIỆN TÍCH BOUNDING BOX BÊN NGOÀI (%)



Hình 6. Chuyển đổi tỷ lệ sang phần trăm.

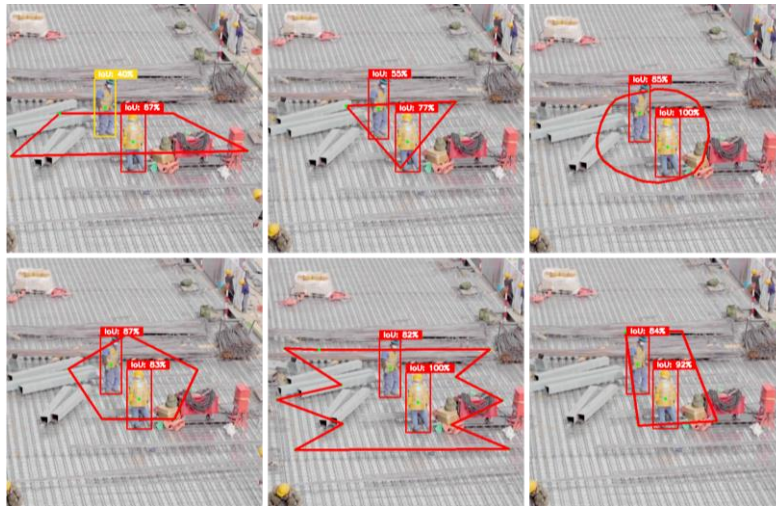
Một số ưu điểm của phương pháp này là chính xác gần như tuyệt đối, linh hoạt và tốc độ xử lý tương đối tốt so với các phương pháp khác. Với phương pháp này, không phải chú ý hình dạng của đa giác trong khu vực cần cảnh báo, có thể là đa giác lồi hoặc đa giác lõm, hay các hình dạng của đa giác được quyết định và đều có thể hoạt động ổn định, tuy nhiên, điều kiện là tại khu vực cần cảnh báo phải là một đa giác và có phần diện tích cụ thể (Hình 7). Từ đó, phần tỷ lệ này đạt một ngưỡng phần trăm giao nhau như cấu hình ban đầu và sẽ có cảnh báo tức thời. Tuy nhiên, cần đảm bảo đa giác cho khu vực cần cảnh báo phải có kích thước lớn hơn hình hộp bao quanh đối tượng, để khi giao nhau với đa giác được vẽ có thể đạt được ngưỡng tối đa khi diện tích của hình hộp bao quanh đối tượng hoàn toàn bên trong khu vực cần cảnh báo.



Hình 7. Hoạt động với các hình dạng khác nhau.

Đối với trường hợp đa giác cần cảnh báo có diện tích khi giao nhau nhỏ hơn phần diện tích của hình hộp bao quanh đối tượng, phải giảm ngưỡng cảnh báo ở mức thấp, ví dụ tối thiểu là 1%, mặc định khi có hình hộp bao quanh đối tượng nằm gần khu vực cần cảnh báo sẽ lập tức cảnh báo. Ngưỡng cảnh báo được quyết định giá trị hợp lý vì phụ thuộc vào vị trí đặt máy ảnh, vị trí cần giám sát xa hay gần hoặc nơi đông người hay ít người qua lại để hiệu chỉnh ngưỡng phù hợp tránh các trường hợp không cảnh báo hoặc gửi cảnh báo quá nhiều lần. Tại Hình 8, giả sử ngưỡng cảnh báo được cấu hình là 55%, tuy nhiên, ngưỡng phần trăm giao nhau của hai

đa giác chưa đạt tới một giá trị là 55%, như trong trường hợp này là 40% (phần màu vàng), thì đây được xem là trường hợp nghi ngờ. Đối với các trường hợp từ ngưỡng 55% trở lên cho đến tối đa là 100% như trong trường hợp này là 67%, thì tỷ lệ giao nhau của các đối tượng so với khu vực cần cảnh báo nằm trong khoảng này sẽ làm cho hệ thống lập tức cảnh báo. Ngoài ra, hiển thị thêm các trường hợp xâm nhập có thể là vi phạm hoặc nghi ngờ ở trong ứng dụng để khi sử dụng có thể giám sát và thống kê các loại trường hợp vi phạm thông qua thuật toán tracking.



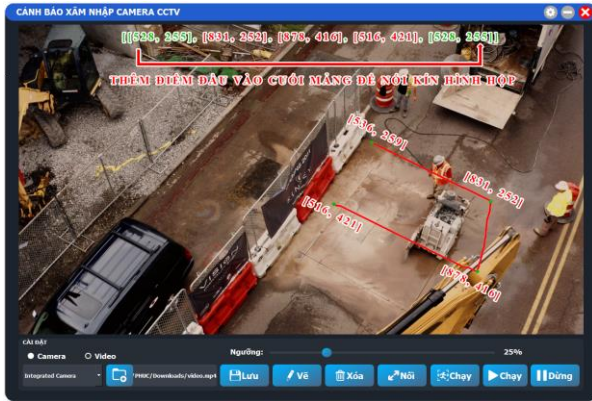
Hình 8. Phát hiện đối tượng xâm nhập.

2.4. Thiết kế ứng dụng

Trong nghiên cứu này, thiết kế một chương trình hoàn chỉnh trên máy tính, cụ thể, nhóm sử dụng thư viện PyQt5 của Python cùng với phần mềm Qt Designer. Với ứng dụng này, cần thực hiện các chức năng cơ bản như hiển thị hình ảnh từ nguồn video đầu vào để phát hiện vật thể trên khung hình từ mô hình YOLO, các chức năng

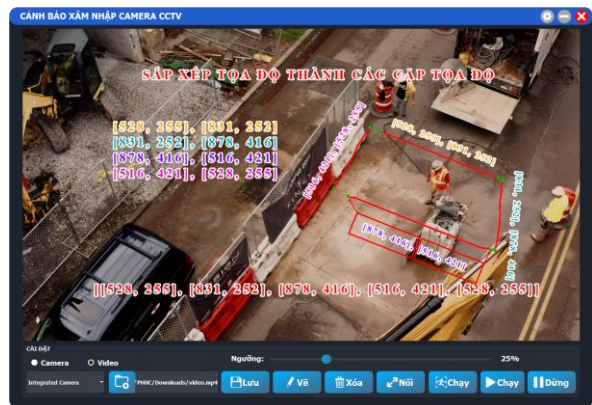
cấu hình về kênh cảnh báo hay cài đặt thông tin người dùng để lưu trữ hình ảnh đối tượng xâm nhập,... và cho phép chọn nguồn từ video hoặc camera kết nối vào máy tính. Ngoài ra, tạo các chức năng để hỗ trợ quá trình xác định khu vực cần cảnh báo để có thể dễ dàng thao tác trong quá trình sử dụng ứng dụng. Với Hình 9 có thể thấy, ứng dụng mô tả chức năng tự động nối kín một đa giác, bằng cách lấy tọa độ đầu tiên trong

danh sách tọa độ đã vẽ và thêm vào cuối danh sách, như vậy, có thể nối điểm cuối vào điểm đầu và tạo thành một đa giác kín hoàn chỉnh.



Hình 9. Nối kín khu vực cảnh báo.

Chức năng lưu tọa độ được thể hiện trong Hình 10, cần xác định các đối số cần thiết để có thể vẽ nên một đường thẳng. Cụ thể, trong trường hợp này, cần nhóm hai cặp tọa độ liền kề nhau từ tọa độ hai điểm để tạo thành một đường thẳng, sau đó lặp qua từng cặp tọa độ trong tọa độ khu vực mà người dùng vẽ, tiếp theo, lưu tọa độ này vào tập tin, khi khởi động chỉ cần để ứng dụng đọc các giá trị này từ tập tin và vẽ lại đa giác đã lưu.



Hình 10. Lưu tọa độ khu vực đã vẽ.

Mặt khác, trong quá trình thiết kế ứng dụng theo như Hình 11, tọa độ ứng dụng nằm trên cùng bên trái là $[0,0]$, tuy nhiên, tọa độ sử dụng để vẽ là một nhãn (label) trong PyQt có kích thước nhỏ hơn so với toàn bộ giao diện ứng dụng. Kích thước để hiển thị hình ảnh là 1280×720 điểm ảnh đang lệch một khoảng $[31,50]$ so với giao diện bao quanh bên ngoài. Vì vậy, cần xử lý tọa độ khi người dùng chọn vào khung hình bằng cách trừ thêm một giá trị tương ứng cho

cả trục x và trục y để tọa độ được chọn trên ứng dụng trùng khớp hoàn toàn so với tọa độ thực tế trên khung hình được chỉ định. Sau khi hoàn thành các chức năng cơ bản cho hệ thống, cần tạo một môi trường ảo và cài đặt các thư viện cần thiết để ứng dụng có thể hoạt động, với việc sử dụng thư viện pyinstaller có thể đóng gói toàn bộ thư viện cần thiết thành một ứng dụng và tạo một chương trình cài đặt trên máy tính.

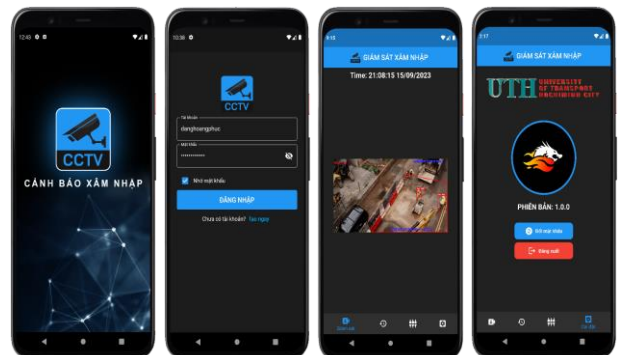
Tọa độ lệch một khoảng $(x,y)=(31,50)$



Xử lý bằng cách trừ đi một khoảng x và y tương ứng khi vẽ trên ứng dụng

Hình 11. Tính toán lại giá trị tọa độ.

Trong hệ thống này, chúng tôi sử dụng Flutter để thiết kế giao diện ứng dụng di động cơ bản như Hình 12. Với cơ sở dữ liệu được sử dụng là MongoDB để lưu trữ các thông tin về lịch sử xâm nhập và thời gian xâm nhập. Cụ thể, xây dựng các chức năng như khi giám sát khung hình hiện tại, cứ sau những giây mặc định, ứng dụng sẽ cập nhật trên cơ sở dữ liệu, có thể xem lại lịch sử vi phạm của các đối tượng xâm nhập và bật hoặc tắt cảnh báo tạm thời khi cần thiết để tránh làm phiền khi gửi cảnh báo liên tục quá nhiều lần và tương tự với các chức năng khác.

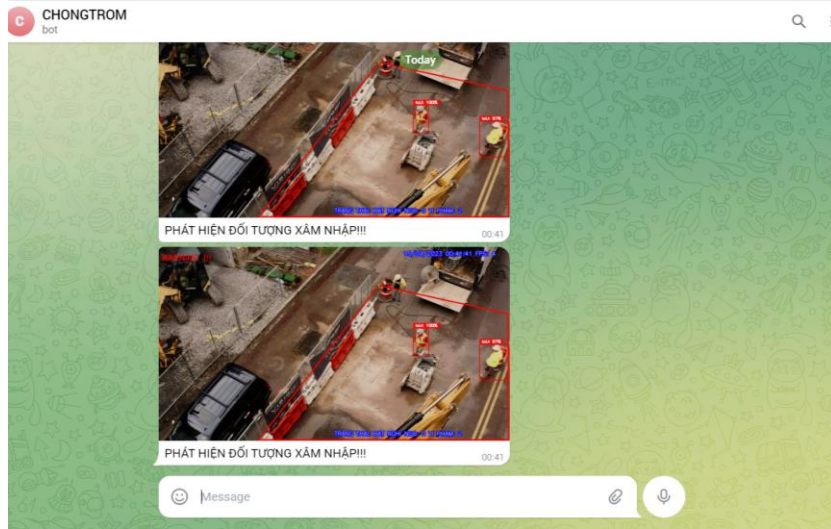


Hình 12. Giao diện ứng dụng di động.

Tuy nhiên, khi có đối tượng xâm nhập, có thể sử dụng Telegram làm kênh cảnh báo chính cho hệ thống như trong Hình 13. Vì tính đơn giản và tiện lợi khi có thể gửi thông báo ngay lập tức

tới ứng dụng nếu đối tượng xâm nhập. Thế nhưng, phải tiến hành cấu hình các thông số cần thiết để nhận cảnh báo như “token” và “id” của nhóm trò chuyện trên ứng dụng để có thể thực

hiện gửi và nhận các cảnh báo xâm nhập khi có đối tượng vi phạm theo thời gian thực một cách nhanh chóng.



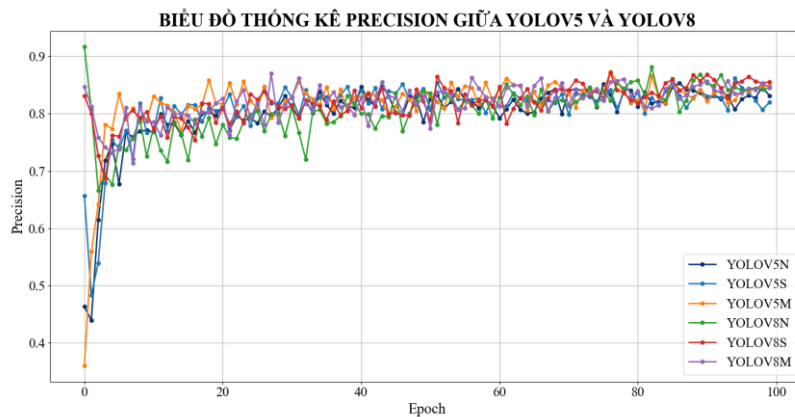
Hình 13. Cảnh báo qua ứng dụng Telegram.

3. Kết quả đạt được

3.1. Đánh giá kết quả

Sau khi huấn luyện ba mô hình cho hai phiên bản YOLOv5 và YOLOv8, trích xuất các chỉ số để tiến hành thống kê như mAP (trung bình của độ chính xác), Recall (độ nhạy), Precision (độ chính xác). Ngoài ra, sẽ nhận được tập tin kết quả để vẽ biểu đồ so sánh. Quan sát Hình 14, có

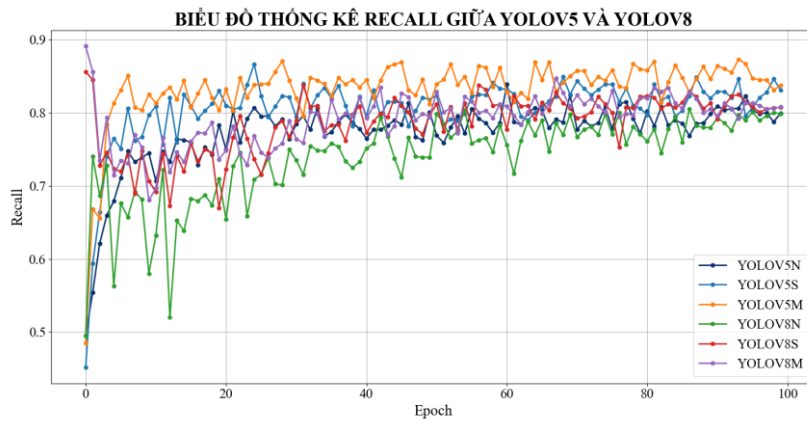
thể nhận xét biểu đồ về độ chính xác của hai phiên bản này, các đường chỉ số của từng mô hình khá bão hòa, với các epoch cuối cùng, cho thấy YOLOv8s có giá trị cao hơn so với các mô hình còn lại. Nếu so sánh với mô hình YOLOv5m thì chỉ gần bằng với YOLOv8s. Điều này cho thấy, YOLOv8 có độ chính xác tương đối cao so với YOLOv5 khi có chênh lệch về độ chính xác tại các mô hình.



Hình 14. Biểu đồ đánh giá Precision.

Dựa vào Hình 15, thấy rằng, chỉ số độ nhạy tương đối cao và có xu hướng bão hòa, tiến dần gần giá trị 1. Tuy nhiên, chỉ số này chưa đạt được mức ổn định nhất khi đường thẳng biểu diễn cho thấy sự dao động lên xuống thất thường, độ chênh lệch giữa các epoch liên kế tương đối cao so với chỉ số độ chính xác. Sau

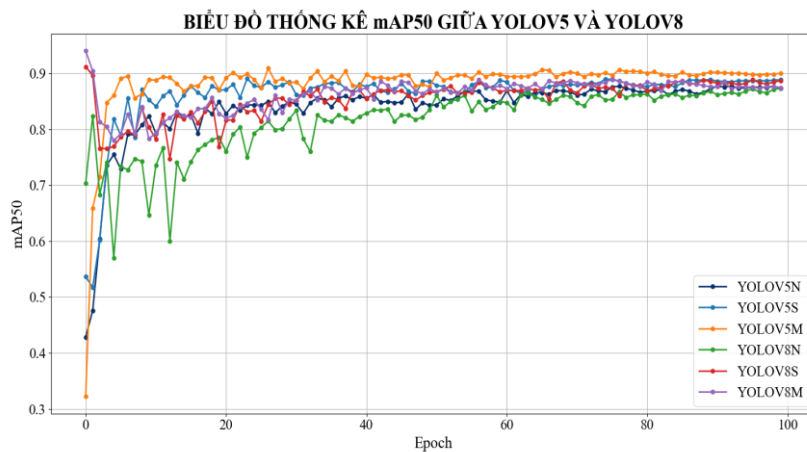
hơn 100 epoch, YOLOv5m đạt giá trị cao nhất, thậm chí lớn hơn so với YOLOv8m. Kết quả có giá trị thấp nhất là YOLOv8n, tất cả các mô hình của YOLOv8 sau khi huấn luyện đều thấp hơn so với toàn bộ mô hình YOLOv5 trong cùng một tập dữ liệu.



Hình 15. Biểu đồ đánh giá Recall.

Tại Hình 16, chỉ số mAP50 hay độ trung bình của độ chính xác ở ngưỡng trùng khớp 50% đã gần hội tụ khi vào các epoch cuối cùng với tất cả các giá trị đều bão hòa và tiệm cận gần giá trị là 1. Đáng đầu kết quả là YOLOv5m khi đạt được khoảng 90%, tương tự với các mô hình còn lại của YOLOv5 đều cho ra kết quả tương đối cao. Điều này, có thể đánh giá mô hình

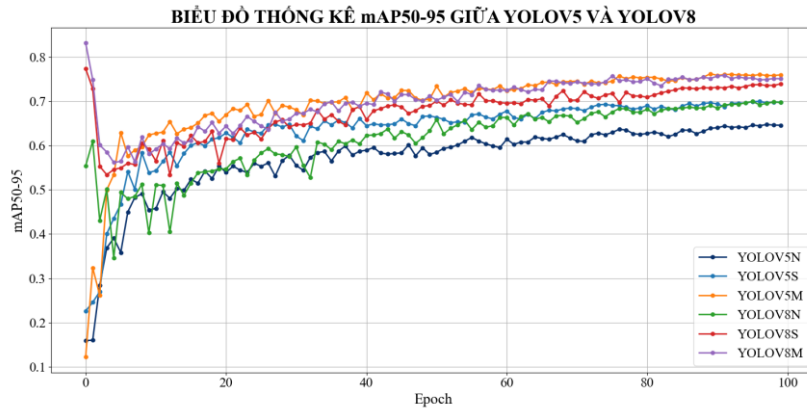
YOLOv5 đạt độ chính xác tốt hơn so với YOLOv8. Tuy nhiên, các kết quả về mAP50 của YOLOv5 và YOLOv8 đều không có sự khác biệt nhiều, sự chênh lệch không lớn ở các epoch cuối cùng cho thấy tất cả các mô hình đều có thể sử dụng trong các ứng dụng thực tế với hiệu suất cao.



Hình 16. Biểu đồ đánh giá mAP50.

Theo đó, như trong Hình 17, đánh giá chỉ số mAP50-95 hay nói cách khác là trung bình độ chính xác ở các ngưỡng khác nhau từ 50% đến 95%, tương đối cao ở các epoch cuối cùng. Tuy nhiên, đường cong của các biểu đồ khá rõ rệt khi so với đường cong của chỉ số mAP50, như vậy, chỉ số mAP50-95 có kết quả không tốt so với chỉ số mAP50. Để đường cong này ít biến động trên biểu đồ, các chỉ số của từng epoch phải hội tụ và chênh lệch thấp nhất. Đồ thị có thể bắt đầu từ giá trị thấp nhất sau đó tăng tiệm cận 1

và giữ đều tại giá trị đó hay tại bất kỳ các giá trị khác để đối chiếu qua trục mAP50-95 luôn luôn gần bằng giá trị 1. Mô hình YOLOv5m đạt kết quả cao nhất với hơn 76%, tiếp đến là mô hình YOLOv8m đạt hơn 75%, tương tự với các mô hình còn lại, các kết quả đều tương đồng nhau. Đối với mô hình có kết quả thấp nhất là mô hình YOLOv5n, kết quả đạt được chênh lệch tương đối khá nhiều khi trung bình đạt khoảng hơn 62% so với mô hình YOLOv8n là gần 70%.



Hình 17. Biểu đồ đánh giá mAP50-95.

3.2. Đánh giá độ chính xác mô hình

Dựa vào Bảng 1 để có góc nhìn tổng quan về tất cả các mô hình sau khi huấn luyện. Với độ chính xác xuất từ chỉ số mAP50 trên tập dữ liệu kiểm tra (test) và trích từ 20% của toàn bộ tập dữ liệu với 400 hình ảnh dùng để đánh giá.

YOLOv5n và YOLOv5s có các giá trị về tham số cũng như GFLOPS tương đối thấp, phần hiệu suất tương đối cao. Tuy nhiên, mô hình YOLOv5m cho kết quả tốt với hơn 20.85 triệu tham số và độ chính xác đạt khoảng 92%, FPS trung bình đạt từ 3 FPS.

Bảng 1. Đánh giá các mô hình đề xuất.

Phương thức	Chính xác (%)	Tham số (triệu)	GFLOPS	FPS (f/s)
YOLOv5n	91.7	1.76	4.1	12.91
YOLOv5s	92.5	7.01	15.8	7.38
YOLOv5m	92.9	20.85	47.9	3.3
YOLOv8n	92.2	3	8.1	15.62
YOLOv8s	92.3	11.12	28.4	7.88
YOLOv8m	92.1	25.84	78.7	3.82
YOLOv5x [3]	90.8	87.7	218.8	17
YOLOv5l [3]	90.6	47	115.4	37

Đối với YOLOv8, các chỉ số như số lượng tham số, GFLOPS và FPS có phần hơn so với phiên bản YOLOv5, thế nhưng, chỉ số FPS thấp hơn so với YOLOv5. Cụ thể, mô hình YOLOv8m cho kết quả khá tốt với khoảng hơn 25.84 triệu tham số, GFLOPS đạt 78.7 và độ chính xác đạt hơn 92% với FPS trung bình khoảng từ 3.82 FPS. Tương tự với tất cả các chỉ số của những mô hình YOLOv8 còn lại, với các chỉ số như FPS, GFLOPS, tham số cao hơn rất nhiều so với các mô hình YOLOv5. Điều này cho thấy tốc độ xử lý của YOLOv8 tốt hơn so với YOLOv5 bởi độ chính xác tương đối cao. Hơn thế, khi so

sánh với công trình [3], hiệu suất của phương pháp đề xuất cho thấy sự vượt trội trong độ chính xác, tốc độ xử lý nhờ sự tối ưu trong việc tinh chỉnh kiến trúc mô hình, thông số huấn luyện. Cụ thể, mô hình đề xuất YOLOv5n và YOLOv8n đạt độ chính xác lần lượt 91.7% và 92.2% khi so sánh với YOLOv5x (90.8%) và YOLOv5l (90.6%) của nghiên cứu [3].

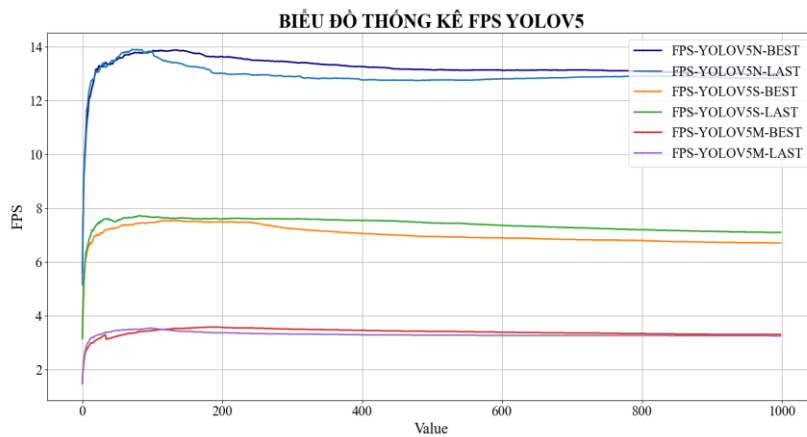
3.3. Đánh giá hiệu suất mô hình

Với thông số đánh giá mô hình, sử dụng máy tính Windows, vi xử lý Intel I5-1135G7, đồ họa tích hợp Intel Iris Xe Graphics. Nhóm tác giả

tiến hành chạy mô hình và phát hiện liên tục nhiều đối tượng người đi lại trong một khung hình nhất định, cứ mỗi một vòng lặp tiếp theo sẽ thực hiện lưu lại giá trị FPS của mô hình cho đến khi đủ 1000 giá trị để thực hiện đánh giá. Điều này tương tự cho toàn bộ mô hình còn lại, lấy 1000 giá trị cho tất cả mô hình đã được huấn luyện với hai phiên bản YOLOv5 và YOLOv8.

Qua Hình 18 cho thấy chỉ số FPS của YOLOv5 tương đối ổn định, không biến động nhiều ở từng epoch khác nhau theo thời gian.

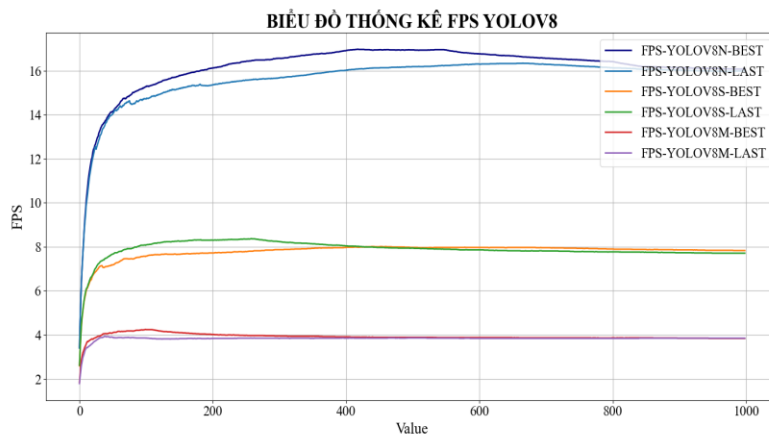
Cả hai phiên bản đều được thống kê trên ba mô hình thấp nhất từ YOLOv5 và tiến hành đánh giá kết quả ở mỗi mô hình bao gồm hai giá trị được trích từ tập tin sau khi huấn luyện với hơn 100 epoch bao gồm “best.pt” và “last.pt”. Với “best.pt” là tập tin trọng số cho mô hình có hiệu suất tốt nhất trên tập dữ liệu kiểm tra, được sử dụng để đánh giá độ chính xác của mô hình và lưu lại tại thời điểm mô hình đạt kết quả tốt nhất. “last.pt” là tệp trọng số cho mô hình ở trạng thái cuối cùng của quá trình huấn luyện.



Hình 18. Biểu đồ đánh giá FPS của YOLOv5.

Các giá trị cho từng mô hình của YOLOv5 tương đối ổn định và đều nhau theo từng thời gian khác nhau. FPS dao động trong khoảng 13 đến 14 FPS đối với mô hình thấp nhất là YOLOv5n, tùy thuộc vào tập tin chứa tham số sau khi huấn luyện. Đối với YOLOv5s chỉ số FPS dao động từ 6 đến 7 khung hình, tương tự với mô hình YOLOv5m, FPS dao động trong

khoảng từ 3 đến 4 FPS. Dựa vào Hình 19, nhận thấy các chỉ số FPS của mô hình YOLOv8 cao tương đối nhiều so với các mô hình YOLOv5. Đối với mô hình thấp nhất YOLOv8n trung bình đạt khoảng hơn 16 khung hình trên giây hiệu năng tương đối cao, với mô hình YOLOv8s trung bình đạt hơn 6 FPS và YOLOv8m đạt khoảng 8 FPS.



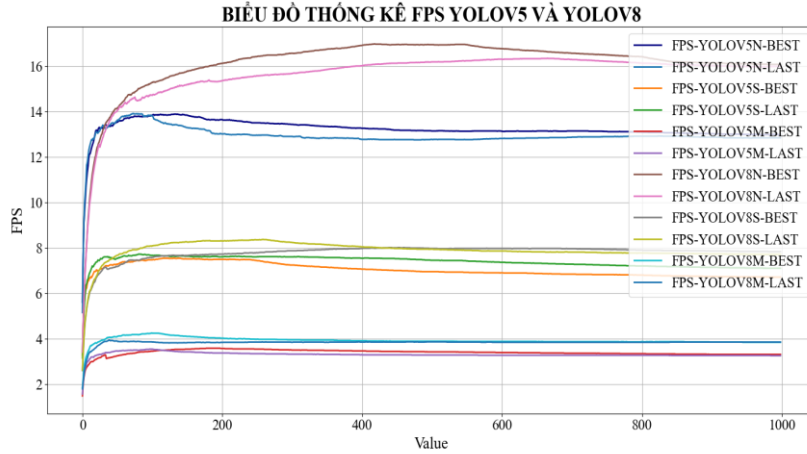
Hình 19. Biểu đồ đánh giá FPS YOLOv8.

Để thấy rõ sự khác biệt, có thể thực hiện so sánh cùng với cả hai phiên bản YOLOv5 và

YOLOv8. Theo Hình 20, tất cả các đường thẳng biểu diễn chỉ số FPS của cả ba phiên bản

YOLOv8 đều có giá trị lớn hơn so với YOLOv5. Với mô hình thấp nhất, YOLOv8 cho số khung hình cao nhất có thể đạt hơn 16 khung hình trên giây, ngoài ra, số FPS trung bình đạt khoảng 15 khung hình trên giây so với mô hình YOLOv5n đạt khoảng 13 khung hình. Tương

tự, mô hình YOLOv8s cho ra số khung hình trên một giây trung bình đạt khoảng 8 khung hình. YOLOv8m đạt được kết quả tương đối tốt khi số khung hình cho ra đạt khoảng hơn 4 khung hình trên giây.

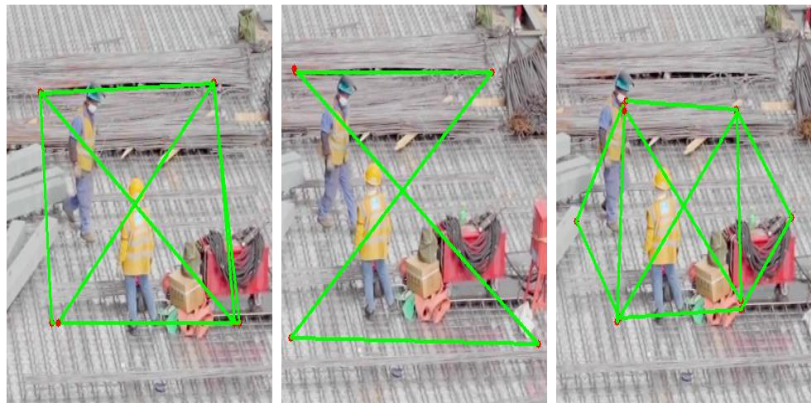


Hình 20. Biểu đồ đánh giá FPS của cả hai.

3.4. Đánh giá phương pháp

Phương pháp phát hiện xâm nhập dựa trên phương pháp phân mảnh giao nhau, với tốc độ xử lý tốt hơn so với các phương pháp khác. Tuy nhiên, hiện hữu một vài nhược điểm, chẳng hạn thể hiện tại Hình 21. Nếu khu vực cần cảnh báo có dạng đặc thù, cắt lẩn nhau, hệ thống không

thể phát hiện các trường hợp vi phạm, cho nên khu vực cần cảnh báo là phần đa giác có diện tích nhất định. Để tránh gặp các vấn đề này, cần xác định hình dạng của khu vực cảnh báo một cách đơn giản và hợp lý, tránh các hình dạng quá đặc thù, cắt lẩn nhau, chồng lẩn,... để không bỏ qua các trường hợp vi phạm khi hệ thống không cảnh báo như mong muốn.



Hình 21. Nhược điểm của phương pháp.

3.5. So sánh thực nghiệm

Giả sử thử nghiệm trong môi trường công trình xây dựng, mô hình xuất kết quả phát hiện tương đối tốt, các vật thể là người đi qua các khu vực cảnh báo đều được phát hiện và vẽ lên một hình hộp chữ nhật bao quanh đối tượng, phân loại

thành các trường hợp vi phạm (Hình 22). Ngoài ra, có thể thấy, mô hình vẫn tồn tại một số nhược điểm như các vật thể khi đứng quá gần cạnh nhau đôi khi không thể phát hiện chính xác, đầy đủ các đối tượng và tạo nên toàn bộ hình hộp bao quanh các đối tượng đó. Mặt khác, với phương pháp tính tỷ lệ giao nhau của đa

giác cho thấy độ chính xác tương đối cao, chi hoạt động với các khu vực cảnh báo có hình dạng nhất định, với các trường hợp đa giác cần cảnh báo cắt lẫn nhau, hệ thống có thể không hoạt động chính xác. Tương tự như trong Hình 23, hệ thống đã có thể phát hiện được ba đối tượng người là công nhân xây dựng đang xâm nhập vào khu vực đã chỉ định. Trước đó, chỉ cần xác định khu vực cần cảnh báo chính xác bằng cách khoanh vùng đầy đủ và rõ rệt khu vực mong muốn giám sát, nhưng trong trường hợp này, ngưỡng cảnh báo được hiệu chỉnh là 70%, có hai đối tượng vi phạm khi đạt giá trị tối đa là 100%, đối với đối tượng còn lại chỉ đạt giá trị 12%. Do đó, trường hợp này chưa đạt đến ngưỡng 70%, chỉ xem đây là trường hợp nghi ngờ do đối tượng chưa thực sự ở bên trong khu vực đã chỉ định và chưa đạt tới ngưỡng cảnh báo mong muốn.



Hình 22. Phát hiện hầu hết các đối tượng vi phạm.

Các kết quả khi thử nghiệm từ những nguồn ảnh với nhiều môi trường khác nhau xuất ra các kết quả tương đối tốt. Với các điều kiện môi trường khác nhau, hệ thống đều có thể phát hiện tương đối đầy đủ các đối tượng xâm phạm. Phương pháp này có độ phức tạp tương đối cao khi phải tính toán phần trăm diện tích giao nhau cho tất cả các đa giác chứa đối tượng và cập nhật trên khung hình, tốc độ xử lý của phương pháp rất tốt khi có thể xử lý nhiều đối tượng cùng lúc, hiệu năng vẫn tương đối cao và gần như là có thể phân loại chính xác được các trường hợp xâm phạm theo thời gian thực. Ngoài ra, trong hệ thống này, nhóm dùng mô hình thấp nhất là YOLOv8n để cho thấy hiệu năng sử dụng là tốt

nhất, độ chính xác khá cao cho việc phát hiện các đối tượng xâm nhập, cũng có thể sử dụng các mô hình khác với kiến trúc phức tạp hơn khi tăng độ chính xác cho hệ thống. Tuy nhiên, có thể ảnh hưởng đến hiệu suất hoạt động của hệ thống do mô hình càng cao, đòi hỏi cấu hình phần cứng tốt hơn để có thể đạt kết quả phát hiện các đối tượng vi phạm là tốt nhất.



Hình 23. Phân loại các trường hợp vi phạm.

4. Kết luận

Công nghệ luôn phát triển và đổi mới, các hành vi trộm cắp càng tinh vi hơn. Theo đó, các tai nạn trong lao động, các tình huống xâm nhập vào các công trường, khu vực sửa chữa, khu vực nguy hiểm,... vẫn tồn tại và đôi khi để lại hậu quả vô cùng nghiêm trọng. Vì vậy, trong nghiên cứu này, nhóm tác giả đề xuất một hệ thống cảnh báo xâm nhập vào vùng nguy hiểm sử dụng thuật toán trí tuệ nhân tạo tiên tiến để đảm bảo sự an ninh và an toàn tại nơi làm việc hoặc nơi ở một cách tốt nhất, tự động và chính xác.

Kết quả thực nghiệm cho thấy độ chính xác của phương pháp đề xuất tương đối cao: YOLOv5n với độ chính xác trung bình có thể đạt hơn 91%, độ nhạy hơn 84% và tốc độ xử lý đạt hơn 12 khung hình trên giây, tương tự với mô hình YOLOv8n có độ chính xác trung bình đạt hơn 92%, độ nhạy hơn 82% và tốc độ xử lý có thể đạt 15 khung hình trên giây. Mặt khác, hệ thống đề xuất có thể phát hiện hầu hết các đối tượng xâm nhập khi tiến gần một khu vực cần cảnh báo và phát ra các tín hiệu cảnh báo gửi đến người giám sát. Ngoài ra, để tăng tính hiệu quả của hệ thống tốt hơn, có thể cần kết hợp bổ sung các phương pháp cảnh báo khác để

tăng sự linh hoạt và tính chính xác, nhằm đảm bảo các lần cảnh báo có vi phạm tránh sai sót trong việc phát hiện các đối tượng một cách tối đa. Do đó, cần thu thập đa dạng hơn về tập dữ liệu trong các điều kiện môi trường khác nhau như ánh sáng, tư thế, hành vi, góc độ khác nhau,... để có thể giúp mô hình tăng độ chính xác một cách tốt nhất, đảm bảo chất lượng trong những lần cảnh báo, tránh gây bất tiện, tạo nên trải nghiệm không tốt cho người sử dụng. Những vấn đề này được tiếp tục nghiên cứu và thảo luận trong các nghiên cứu tiếp theo để nâng cao hiệu quả khi nhận diện của mô hình và tính ổn định hệ thống phát hiện sự xâm nhập.

Lời cảm ơn

Nghiên cứu này được tài trợ bởi Trường Đại học Giao thông vận tải Thành phố Hồ Chí Minh trong đề tài mã số KH2309.

Tài liệu tham khảo

- [1] S. V. -T. Tran et al., "A human detection approach for intrusion in hazardous areas using 4D-BIM-Based spatial-temporal analysis and computer vision," *Build.*, vol. 13, no. 9, 2023, Art. no. 2313, doi: 10.3390/buildings13092313.
- [2] J. Zhao, Y. Xu, W. Zhu, M. Liu, and J. Zhao, "Real-time early safety warning for personnel intrusion behavior on construction sites using a CNN Model," *Build.*, vol. 13, no. 9, 2023, Art. no. 2206, doi: 10.3390/buildings13092206.
- [3] R. Hu, F. Li, T. Wang, H. Fan and L. Dong, "Research on intrusion identification of hazardous construction areas based on machine vision," *J. Phys.: Conf. Ser.*, vol. 2435, 2022, Art. no. 012015, doi: 10.1088/1742-6596/2435/1/012015.
- [4] Z. Zhang , C. Trivedi, X.Liu, "Automated detection of grade-crossing-trespassing near misses based on computer vision analysis of surveillance video data," *Saf. Sci.*, vol. 110, part B, pp. 276-285 Dec. 2018, doi: 10.1016/j.ssci.2017.11.023.
- [5] H. Pan, Y. Li, H. Wang, X. Tian, "Railway obstacle intrusion detection based on convolution neural network multitask learning," *Electron.*, vol.11, no. 17, 2022, Art. no. 2697, doi: 10.3390/electronics11172697.
- [6] Z. Zhang, A. Zaman, J. Xu, X. Liu, "Artificial intelligence-aided railroad trespassing detection and data analytics: Methodology and a case study," *Accid. Anal. Prevent.*, vol. 168, Apr. 2022, Art. no. 106594, doi: 10.1016/j.aap.2022.106594.
- [7] Karthik Arumugam, Harsh Ingle, Yogesh Rajgure, "Railway trespassing detection and alert system using deep learning - CNN, YOLO," *Int. Res. J. Eng. Technol.*, vol. 8, no. 12, pp. 1501-1505, Dec. 2021. [Online]. Available: <https://www.irjet.net/archives/V8/i12/IRJET-V8I12257.pdf>.
- [8] C. Meng, Z. Wang, L. Shi, Y. Gao, Y. Tao, and L. Wei, "SDRC-YOLO: A novel foreign object intrusion detection algorithm in railway scenarios," *Electron.*, vol. 12, no. 5, 2023, Art. no. 1256, doi: 10.3390/electronics12051256.
- [9] Y. Wang and P. Yu, "A fast intrusion detection method for high-speed railway clearance based on low-cost embedded GPUs," *Sensors*, vol. 21, no. 21, 2021, Art. no. 7279, doi: 10.3390/s21217279.
- [10] G. T. Kim, Y. Lee, K. Lee, H. H. Ko, "Development of specific area intrusion detection system using YOLO in CCTV video," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 8S2, pp. 852-856, Jun. 2019. [Online]. Available: <https://www.ijtee.org/wp-content/uploads/papers/v8i8s2/H11440688S219.pdf>.
- [11] S. Patel, N. Patel, S. Deshpande, A. Siddiqui, "Ship intrusion detection using custom object detection system with YOLO algorithm," *Int. Res. J. Eng. Technol.*, vol. 8, no. 1, pp. 669-677, Jan. 2021. [Online]. Available: <https://www.irjet.net/archives/V8/i1/IRJET-V8I1124.pdf>.
- [12] T. Zhang et al., "Recent advances in video analytics for rail network surveillance for security, trespass and suicide prevention - A Survey," *Sensors*, vol. 22, no. 12, 2022, Art. no. 4324, doi: 0.3390/s22124324.

- [13] G. Setiarso, A. M. Hirzan, and A. Hartanto, “Human recognition with yolo to reduce false alarms in the internet-of-things-based motion detection system,” *J. Adv. Comput. Technol. Appl.*, vol. 5, no. 1, pp. 1-12, 2023. [Online]. Available: <https://jacta.utem.edu.my/jacta/article/view/5275>.
- [14] T. Erlina and M. Fikri, “A YOLO algorithm-based visitor detection system for small retail stores using single board computer,” *J. Appl. Eng. Technol. Science*, vol. 4, no. 2, pp. 908–920, 2023, doi: 10.37385/jaets.v4i2.1872.
- [15] S. Liu, N. Zhang, and G. Yu, “Lightweight Security Wear Detection Method Based on YOLOv5,” *Wire. Commun. Mobi. Comput.*, vol. 2022, 2022, Art. no. 1319029, doi: 10.1155/2022/1319.
- [16] G. Jocher. *YOLOv5 by Ultralytics v7.0*. (2022) Accessed: Jun 2, 2023. [Online]. Available: <https://zenodo.org/records/7347926>.
- [17] G. Jocher, A. Chaurasia and J. Qiu. *Ultralytics YOLO v8.0.0*. (2023). Accessed: Jun 8, 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>.
- [18] L. Wang, J. Shi, G. Song and I. Shen, “Object detection combining recognition and segmentation,” in *Asian Conf. Comput. Vis. 2007, Lect. Not. Comput. Sci.*, vol. 4843, Heidelberg, Berlin, Germany: Springer, 2007, pp.189-199, doi: 10.1007/978-3-540-76386-4_17.
- [19] M. Gochoo, Apr. 2021, “Safety Helmet Wearing Dataset”, Mendeley Data. [Online]. Available: <https://data.mendeley.com/datasets/9rcv8mm682/1>.