ĐẠI HỌC ĐÔNG Á

KHOA CÔNG NGHỆ THÔNG TIN

NGÀNH TRÍ TUỆ NHÂN TẠO VÀ KHOA HỌC DỮ LIỆU

-----------✧✧✧-----------

Logo, company name

Description automatically generated

**ĐỒ ÁN HỌC PHẦN**

THỊ GIÁC MÁY TÍNH 2

ĐỀ TÀI: ỨNG DỤNG YOLO XÂY DỰNG

HỆ THỐNG CẢNH BÁO PHÁT HIỆN XÂM NHẬP

GVHD: Ts. Nguyễn Sĩ Thìn

SVTH: Ngô Văn Úc IDSV: 53133

*Đà Nẵng, tháng 05 năm 2024*

Đà Nẵng, 10/11/2022

Mục lục

[DANH MỤC BẢNG BIỂU 3](#_Toc166703808)

[DANH MỤC HÌNH ẢNH 3](#_Toc166703809)

[CÁC KÍ HIỆU VIẾT TẮT 3](#_Toc166703810)

[TÓM TẮT 3](#_Toc166703811)

[1. TỔNG QUAN VÀ MỤC TIÊU ĐỀ TÀI 3](#_Toc166703812)

[a. Giới thiệu đề tài 3](#_Toc166703813)

[b. Bối cảnh nghiên cứu 4](#_Toc166703814)

[c. Mục đích nghiên cứu 4](#_Toc166703815)

[2. CƠ SỞ LÝ THUYẾT 6](#_Toc166703816)

[a. Giới thiệu 6](#_Toc166703817)

[b. Tìm hiểu YOLO 6](#_Toc166703818)

[c. Hệ thống gửi cảnh báo qua gmail bằng SMTP 10](#_Toc166703819)

[d. Mô hình đề xuất 13](#_Toc166703820)

[3. TRIỂN KHAI HỆ THỐNG 14](#_Toc166703821)

[a. Thực hiện phát hiện đối tượng bằng YOLO 14](#_Toc166703822)

[b. Xây dựng bộ gửi cảnh báo bằng smtplib 16](#_Toc166703823)

[4. THỰC NGHỆM VÀ ĐÁNH GIÁ 17](#_Toc166703824)

[a. Kết quả triển khai 17](#_Toc166703825)

[b. Đánh giá 17](#_Toc166703826)

[5. PHƯƠNG HƯỚNG PHÁT TRIỂN TIẾP THEO 17](#_Toc166703827)

[6. TÀI LIỆU THAM KHẢO 18](#_Toc166703828)

# DANH MỤC BẢNG BIỂU

# DANH MỤC HÌNH ẢNH

[Hình 1. Phát hiện đối tượng bằng YOLO - Nguồn DataCamp 7](#_Toc166704001)

[Hình 2. So sánh hiệu suất của các mô hình tiên tiến [4] 8](#_Toc166704002)

[Hình 3. Kiến trúc mô hình YOLO - Nguồn DataCamp 8](#_Toc166704003)

[Hình 4. Bước xác định grid cells - Nguồn DataCamp 9](#_Toc166704004)

[Hình 5. Dự đoán hộp chứa đối tượng - Nguồn DataCamp 10](#_Toc166704005)

[Hình 6. Tính toán độ chính xác và lọc hộp giới hạn - Nguồn DataCamp 10](#_Toc166704006)

[Hình 7. Chọn ra các vùng có xác suất chứa đối tượng cao nhất - Nguồn DataCamp 11](#_Toc166704007)

[Hình 8. Giao thức SMPT - Nguồn MRTECH 12](#_Toc166704008)

[Hình 9. Mô hình đề xuất 14](#_Toc166704009)

Hình 10. Kết quả triển khai

CÁC KÍ HIỆU VIẾT TẮT

|  |  |
| --- | --- |
| Kí hiệu | Diễn giải |
| CNN | Convolutional Neural Network |
| YOLO | You Look Only Once |
| SMTP | Simple Mail Transfer Protocol |

# TÓM TẮT

Báo cáo này trình bày về việc áp dụng thuật toán YOLO v8 để xây dựng một hệ thống cảnh báo phát hiện xâm nhập hiệu quả. Trong phần Tổng quan và mục tiêu đề tài giới thiệu về các vấn đề cần giải quyết, bối cảnh nghiên cứu, mục đích và cấu trúc tổ chức của báo cáo. Phần cơ sở lí thuyế cung cấp các kiến thức về thuật toán YOLO, cách thức triển khai hệ thống gửi cảnh báo qua Gmail và mô hình đề xuất. Tiếp theo là phần triển khai hệ thống, mô tả quá trình thực hiện phát hiện đối tượng bằng YOLO và xây dựng bộ gửi cảnh báo bằng thư viện smtplib. Kết quả thực nghiệm và đánh giá được trình bàu trong phần tương ứng, kèm theo các đánh giá về hiệu suất và khả năng áp dụng của hệ thống. Cuối cùng, nghiên cứu đề xuất các phương hướng phát triển tiếp theo để nâng cao hiệu suất và tính ứng dụng của hệ thống. Báo cáo này là một nỗ lực hướng đến việc cải thiện khả năng phát hiện và cảnh báo xâm nhập trong các hệ thống an ninh và giám sát.

1. TỔNG QUAN VÀ MỤC TIÊU ĐỀ TÀI
   1. Giới thiệu đề tài

Trong thời đại công nghệ ngày nay, việc bảo vệ an ninh và giám sát là một trong những ưu tiên hàng đầu của các tổ chức và cá nhân. Một trong những thách thức lớn nhất trong lĩnh vực này là khả năng phát hiện và cảnh báo về các hành vi xâm nhập một cách nhanh chóng và hiệu quả. Đồng thời, sự phát triển của trí tuệ nhân tạo và máy học đã mở ra nhiều cơ hội để cải thiện hệ thống an ninh và giám sát.

Đề tài này tập trung vào việc ứng dụng thuật toán YOLO để xây dựng một hệ thống cảnh báo phát hiện xâm nhập, nhằm giải quyết các thách thức trên. YOLO là một thuật toán phát hiện và nhận dạng đối tượng trong ảnh và video với tốc độ cao và độ chính xác đáng kể. Bằng cách kết hợp YOLO với các công nghệ gửi cảnh báo qua email, hệ thống của này có khả năng phát hiện các hành vi xâm nhập và thông báo ngay lập tức.

Đối với các tổ chức và cá nhân quan tâm đến việc cải thiện hệ thống an ninh và giám sát, nghiên cứu này không chỉ cung cấp một giải pháp hiệu quả mà còn mở ra một khả năng mới trong việc sử dụng trí tuệ nhân tạo để bảo vệ tài sản và đảm bảo an toàn cho mọi người

* 1. Bối cảnh nghiên cứu

Trong thời đại số hóa ngày nay, nhu cầu trong việc xây dựng và duy trì các hệ thống an ninh và giám sát ngày càng trở nên cấp thiết. Việc bảo vệ tài sản, ngăn chặn các hành vi phạm pháp, và đảm bảo an toàn cho cộng đồng trở thành ưu tiên hàng đầu của nhiều tổ chức và cá nhân. Tuy nhiên, các hệ thống truyền thống thường gặp khó khăn trong việc phát hiện và đáp ứng kịp thời đối với các hành vi xâm nhập.

Trong bối cảnh này, trí tuệ nhân tạo và máy học đã đem lại nhiều cơ hội mới để cải thiện hiệu suất và hiệu quả của các hệ thống an ninh và giám sát. Trong số các thuật toán phổ biến, YOLO nổi bật với khả năng phát hiện và nhận dạng đối tượng với tốc độ nhanh và độ chính xác cao.

Đề tài này nhấn mạnh vào việc áp dụng YOLO để xây dựng một hệ thống cảnh báo phát hiện xâm nhập. Bằng cách kết hợp khả năng phát hiện đối tượng của YOLO với tính linh hoạt và hiệu suất của việc gửi cảnh báo qua email, hệ thống có khả năng phát hiện và cảnh báo kịp thời về các hành vi xâm nhập, từ đó giúp cải thiện khả năng đáp ứng và giảm thiểu rủi ro cho các tổ chức và cộng đồng.

* 1. Mục đích nghiên cứu

Mục tiêu chính của nghiên cứu này là áp dụng và phát triển thuật toán YOLO để xây dựng một hệ thống cảnh báo phát hiện xâm nhập hiệu quả. Nghiên cứu này đặt ra các mục tiêu cụ thể như sau:

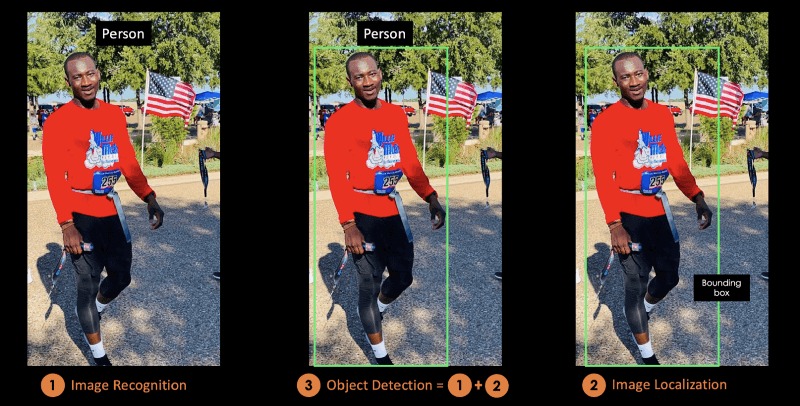
* *Tối ưu hóa quá trình phát hiện đối tượng:* Nghiên cứu tối ưu hóa và cải thiện hiệu suất của thuật toán YOLO trong việc phát hiện đối tượng, bao gồm việc tăng cường độ chính xác và giảm thời gian xử lý.
* *Xây dựng bộ cảnh báo linh hoạt và tin cậy:* Phát triển một hệ thống cảnh báo linh hoạt có khả năng gửi thông báo ngay lập tức qua email khi phát hiện hành vi xâm nhập, đồng thời đảm bảo tính tin cậy và độ ổn định của hệ thống.
* *Đánh giá hiệu suất và tính ứng dụng:* Tiến hành các thử nghiệm và đánh giá hiệu suất của hệ thống trong các điều kiện thực tế để đảm bảo khả năng áp dụng và hiệu quả của nó trong các môi trường thực tế.
* *Đề xuất các phương pháp phát triển tiếp theo:* Dựa trên kết quả thu được, đề xuất các phương hướng phát triển và nâng cấp tiếp theo để cải thiện hiệu suất và tính ứng dụng của hệ thống trong tương lai.

Tổng thể, mục đích của nghiên cứu là đóng góp vào việc phát triển và ứng dụng trí tuệ nhân tạo trong việc cải thiện hệ thống an ninh và giám sát, từ đó nâng cao khả năng phát hiện và đáp ứng kịp thời đối với các hành vi xâm nhập.

Trong nghiên cứu này sẽ được tổ chức thành bảy phần chính. Phần tóm tắt sẽ cung cấp tổng quan và ngắn gọn về nội dung và mục tiêu của bài nghiên cứu. Phần tổng quan và mục tiêu đề tài sẽ trình bày tổng quan về các vấn đề cần giải quyết và nêu lên ý nghĩa của đề tài, đồng thời nêu lên bối cảnh nghiên cứu hiện tại của vấn đề và mục đích nghiên cứu. Phần tiếp theo sẽ nói về các cơ sở lí thuyết như tìm hiểu YOLO và giải thích thuật toán YOLO và cách hoạt động của nó. Giải thích và trình bày cách thức gửi cảnh báo qua Gmail và các công cụ liên quan, đồng thời đưa ra mô hình đề xuất. Phần triển khai hệ thống sẽ thực hiện phát hiện đối tượng băng YOLO, hướng dẫn về quá trình triển khai thuật toán YOLO. Đồng thời xây dựng bộ gửi cảnh báo bằng smtplib và mô tả cách sử dụng. Phần thực nghiệm và đánh giá sẽ trình bày kết quả thực nghiệm thực tế của hệ thống. Đưa ra phân tích và đánh giá kết quả thu được, bao gồm ưu điểm và hạn chế của hệ thống. Hai Phần cuối là phương hướng phát triển tiếp theo sẽ nói lên những điều cần làm trong tương lai và các tài liệu tham khảo được sử dụng trong nghiên cứu này.

1. CƠ SỞ LÝ THUYẾT
   1. Giới thiệu

Phát hiện đối tượng là một kỹ thuật được sử dụng trong thị giác máy tính để xác định và định vị các đối tượng trong hình ảnh hoặc video.  Bản địa hóa hình ảnh là quá trình xác định vị trí chính xác của một hoặc nhiều đối tượng bằng cách sử dụng các hộp giới hạn, tương ứng với các hình chữ nhật xung quanh đối tượng.  Quá trình này đôi khi bị nhầm lẫn với phân loại hình ảnh hoặc nhận dạng hình ảnh, nhằm mục đích dự đoán loại hình ảnh hoặc đối tượng trong hình ảnh thành một trong các danh mục hoặc lớp.  Hình minh họa dưới đây tương ứng với cách trình bày trực quan của lời giải thích trước đó. Đối tượng được phát hiện trong ảnh là “Person” [1].



Hình 1. Phát hiện đối tượng bằng YOLO - Nguồn DataCamp

* 1. Tìm hiểu YOLO

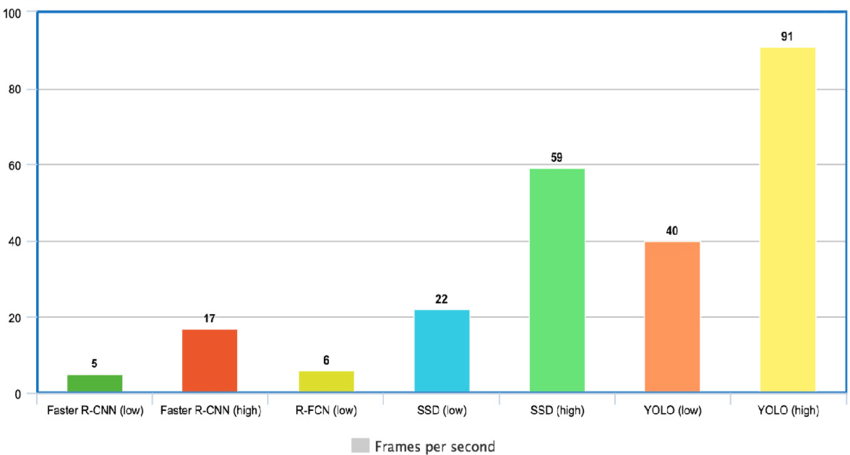
You Only Look Once (YOLO) là một thuật toán phát hiện đối tượng theo thời gian thực tiên tiến được giới thiệu vào năm 2015 bởi [Joseph Redmon](https://arxiv.org/search/cs?searchtype=author&query=Redmon%2C+J) , [Santosh Divvala](https://arxiv.org/search/cs?searchtype=author&query=Divvala%2C+S) , [Ross Girshick](https://arxiv.org/search/cs?searchtype=author&query=Girshick%2C+R) và [Ali Farhadi](https://arxiv.org/search/cs?searchtype=author&query=Farhadi%2C+A) trong bài nghiên cứu nổi tiếng “[You Only Look Once: Thống nhất, Phát hiện đối tượng theo thời gian thực](https://arxiv.org/abs/1506.02640)”.  Các tác giả coi vấn đề phát hiện đối tượng là một vấn đề hồi quy thay vì nhiệm vụ phân loại bằng cách phân tách các hộp giới hạn về mặt không gian và liên kết xác suất với từng hình ảnh được phát hiện bằng cách sử dụng một mạng thần kinh tích chập (CNN) [2].

Thuật toán YOLO là một trong những thuật toán phổ biến và mạnh mẽ nhất trong lĩnh vực phát hiện đối tượng trong ảnh và video. YOLO không chỉ nổi bật với tốc độ cao mà còn đạt được độ chính xác đáng kể.

Điều khiến YOLO trở nên phổ biến trong việc phát hiện đối tượng làdo có độ chính xác cao, tốc độ phát hiện nhanh, khái quát tốt và đặc biệt nó là mã nguồn mở [3].

Về tốc độ, YOLO cực kỳ nhanh vì nó không xử lý được các đường ống phức tạp. Nó có thể xử lý hình ảnh ở tốc độ 45 khung hình mỗi giây (FPS). Ngoài ra, YOLO đạt hơn gấp đôi Độ chính xác trung bình (mAP) trung bình so với các hệ thống thời gian thực khác, điều này khiến nó trở thành một ứng cử viên sáng giá cho việc xử lý thời gian thực.

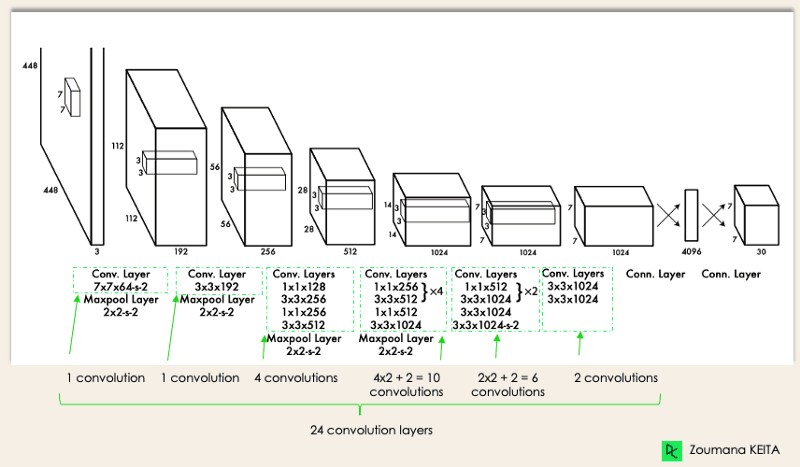
Từ hình ảnh dưới đây cho thấy rằng mô hình YOLO vượt xa các máy dò đối tượng khác với 91 FPS.



Hình 2. So sánh hiệu suất của các mô hình tiên tiến [4]

Với độ chính xác cao, mô hình YOLO vượt xa so với các mô hình tiên tiến khác về hiệu suất. Khả năng khái quát hóa cũng tốt hơn ở các phiên bản mới hơn được cập nhật.

Kiến trúc YOLO tương tự như [GoogleNet](https://arxiv.org/pdf/1409.4842.pdf). Như minh họa bên dưới, nó có tổng thể 24 lớp chập, bốn lớp tổng hợp tối đa và hai lớp được kết nối đầy đủ.



Hình 3. Kiến trúc mô hình YOLO - Nguồn DataCamp

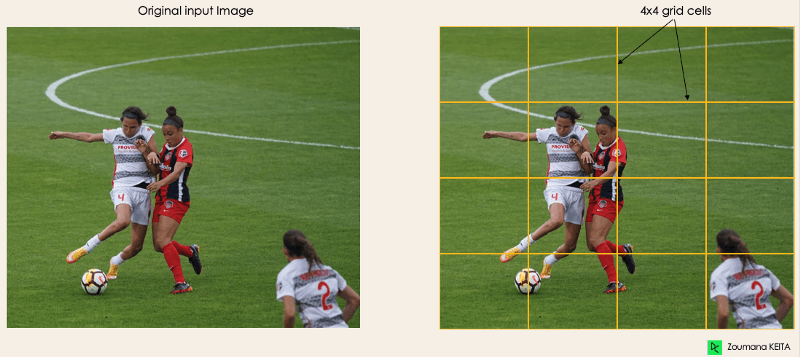
Kiến trúc hoạt động như sau:

* Thay đổi kích thước hình ảnh đầu vào thành 448x448 trước khi đi qua mạng tích chập.
* Tích chập 1x1 trước tiên được áp dụng để giảm số lượng kênh, sau đó là tích chập 3x3 để tạo ra đầu ra hình khối.
* Chức năng kích hoạt dưới mui xe là ReLU, ngoại trừ lớp cuối cùng sử dụng chức năng kích hoạt tuyến tính.
* Một số kỹ thuật bổ sung, chẳng hạn như chuẩn hóa hàng loạt và loại bỏ, tương ứng sẽ chuẩn hóa mô hình và ngăn không cho mô hình khớp quá mức.

YOLO hoạt động bằng cách chia ảnh thành một lưới ô vuông và dự đoán các hộp giới hạn và xác suất đối tượng trong mỗi ô vuông đó. Điều này cho phép YOLO thực hiện phát hiện đối tượng trực tiếp từ toàn bộ ảnh một cách nhanh chóng và hiệu quả.

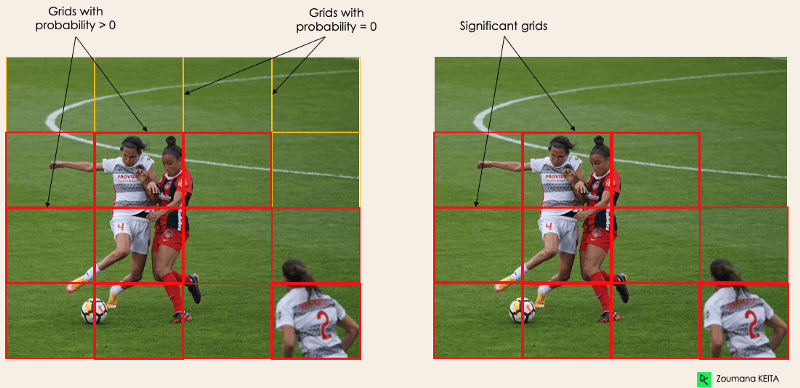
Các bước chính của thuật toán YOLO bao gồm:

* *Xác định grid cells:* Chia ảnh thành lưới ô vuông.



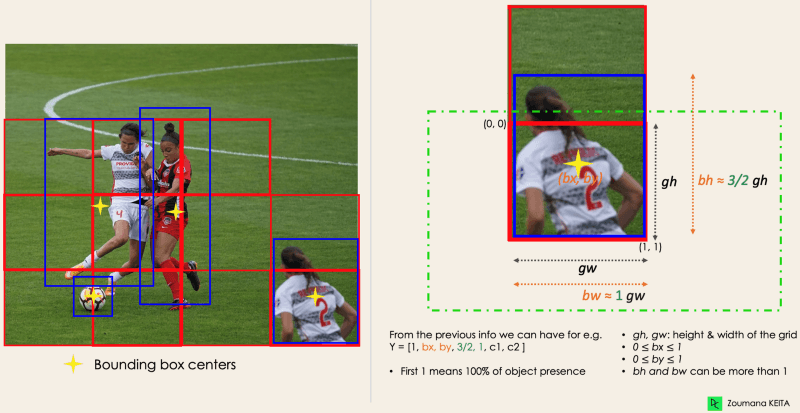
Hình 4. Bước xác định grid cells - Nguồn DataCamp

* *Dự đoán bounding boxes và xác suất đối tượng:* Dự đoán các hộp giới hạn và xác suất của đối tượng trong mỗi ô vuông.



Hình 5. Dự đoán hộp chứa đối tượng - Nguồn DataCamp

* *Tính toán độ chính xác và lọc các bounding boxes:* Sử dụng một ngưỡng xác suất để lọc ra các đối tượng có xác suất đủ cao.



Hình 6. Tính toán độ chính xác và lọc hộp giới hạn - Nguồn DataCamp

* *Post-processing:* Áp dụng các kỹ thuật như Non-maximum Suppression (NMS) để loại bỏ các bounding boxes trùng lặp và chọn ra các bounding box cuối cùng.



Hình 7. Chọn ra các vùng có xác suất chứa đối tượng cao nhất - Nguồn DataCamp

YOLO cung cấp một phương pháp đơn giản và hiệu quả để thực hiện phát hiện đối tượng trong ảnh và video, làm cho nó trở thành một lựa chọn phổ biến cho nhiều ứng dụng trong thực tế, từ an ninh đến ô tô tự lái và nhận diện vật thể trong video.

* 1. Hệ thống gửi cảnh báo qua gmail bằng SMTP

SMTP (là viết tắt của Simple Mail Transfer Protocol) là một giao thức để xử lý trình gửi và định tuyến email giữa các Mail Server. Python cung cấp **smtplib** Module, mà định nghĩa một đối tượng SMTP Client Session có thể được sử dụng để gửi email tới bất kỳ thiết bị internet nào với một SMTP hoặc ESMTP Listener [5].

Việc thiết kế một hệ thống cảnh báo qua Gmail là một phương thức miễn phí, an toàn và nhanh chóng. Việc thiết lập quy trình này cần phải thực hiện một số bước nhất định để có thể gửi cảnh báo thành công.

Giao thức smtp như sau:

Simple Mail Transfer Protocol (SMTP) là giao thức truyền tải e-mail qua mạng, SMTP cho phép chuyển thông điệp mail từ mail server của người gửi đến mail server của người nhận. Thông điệp mail ở định dạng ASCII 7 bit. SMTP được định nghĩa trong RFC 821, điều chỉnh trong RFC 1123 và mở rộng trong RFC 2821 với tên gọi ESMTP (Extended SMTP). SMTP sử dụng cổng 25 trên nền TCP.  
Trong RFC 822 định nghĩa tiêu chuẩn của định dạng thông điệp maill, gồm 2 phần: dòng tiêu đề (header) và phần thân (body) [6].

Mỗi tiêu đề có một dòng tiêu đề *from,to, subject* và một số dòng tiêu đề khác từ lệnh SMTP. Sau tiêu đề là một dòng trống.

A diagram of mail server

Description automatically generatedPhần thân chứa các thông điệp định dạng ở mã ASCII. Mỗi dòng thông thông điệp được kết thúc bằng một dấu chấm. Một hệ thống e-mail có ba thành phần chính: user agents, mail server và giao thức SMTP.

Hình 8. Giao thức SMPT - Nguồn MRTECH

User agentcho phép người dùng đọc, trả lời, chuyển tiếp, lưu và soạn thông điệp mail. User agent là những phần mềm gửi mail như Microsoft’s Outlook, Apple Mail (những phần mềm này sử dụng giao diện GUI).Mail server là thành phần cốt lỗi trong hạ tầng hệ thống e-mail. Khi người dùng soạn xong thông điệp thì user agent sẽ gửi thông điệp đến mail server và thông điệp được đặt trong hàng đợi (message queue), sau đó sẽ gửi đến mail server của người dùng khác và được lưu tại mailbox.Giao thức SMTP đảm nhiệm việc truyền tải thông điệp từ mail server của người gửi đến mail server của người nhận. SMTP thiết lập kết nối TCP trên cổng 25 [6].

Phương thức hoạt động của SMTP:

* Bước 1: An khởi động useragent của mình, cung cấp địa chỉ e-mail của Bình, soạn thông điệp và chỉ thị user agent gửi mail.
* Bước 2: User agent của An gửi thông điệp đến mail server của An và thông điệp được đặt trong hàng đợi.
* Bước 3: SNMP client chạy trên mail server của An phát hiện ra thông điệp trong hàng đợi và tiến hành mở kết nối TCP đến **SMTP server** chạy trên mail server của Bình.
* Bước 4: Sau khi thực hiện bắt tay chào hỏi (handshaking), **SMTP client** của An sẽ gửi thông điệp của An đến kết nối TCP.
* Bước 5: Tại mail server của Bình, SMTP server nhận được thông điệp và lưu lại trên mailbox.
* A diagram of mail server

  Description automatically generatedBước 6: Khi Bình khởi động user agent của mình thì sẽ thấy mail của An trong mailbox [6].

SMTP không sử dụng các mail server trung gian để gửi thư, mà chỉ sử dụng một kết nối TCP trực tiếp giữa hai mail server ngay cả khi hai mail server cách nhau một khoản cách rất xa. SMTP truyền thông điệp mail qua ba pha: handshaking (chào hỏi), truyền thông điệp, đóng kết nối [6].

Pha handshaking: đầu tiên, SMTP client (chạy trên mail server bên gửi) thiết lập kết nối TCP trên cổng 25 tới SMTP server (chạy trên mail server bên nhận). Nếu SMTP server gặp sự cố thì client sẽ cố gắng kết nối lại. Khi kết nối được thiết lập, SMTP server và SMTP client thực hiện một số chào hỏi, SMTP client sẽ cho biết địa chỉ email của người gửi và địa chỉ email của người nhận [6].

Pha truyền thông điệp: sau cuộc chào hỏi, SMTP client tiến hành gửi thông điệp trên đường truyền tin cậy TCP.

Pha đóng kết nối: Nếu client muốn gửi tiếp tục gửi thông điệp thì lặp lại quá trình trên trong cùng kết nối TCP, còn không thì chỉ thị TCP đóng kết nối [6].

* 1. Mô hình đề xuất

A diagram of a video camera

Description automatically generated

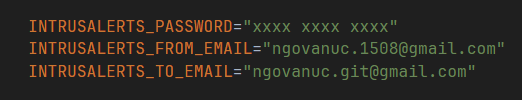
Hình 9. Mô hình đề xuất

Trong nghiên cứu này tập trung vào ứng dụng YOLO để thực hiện phát hiện người. YOLO có độ chính xác cao và nhanh trong việc nhận diện người, điều này là phù hợp vì việc phát hiện xâm nhập trên thời gian thực là tốn nhiều tài nguyên và rất nhiều thông tin được sinh ra trong một khoảng thời gian rất ngắn.

Đầu tiên sẽ đọc camera từ các nguồn camera trực tiếp như CCTV, camera trên các máy tính hoặc thậm chí là trên các điện thoại, tuy nhiên với mỗi nơi cần phải cấu hình đầu vào để có thể phù hợp với thiết bị đầu vào. Sau đó hình ảnh trong camera sẽ được liên tục đưa vào mô hình YOLO, nếu phát hiện có người trong khung hình camera, sẽ vẽ một hộp khung nơi vùng chứa người xâm nhập, hình ảnh này sẽ được lưu vào hệ thống và một cảnh báo bằng Gmail sẽ được tự động gửi đi. Câu hỏi đặt ra ở đây đó là khi nào thì chúng ta sẽ gửi cảnh báo đi. Việc liên tục gửi cảnh báo trong một thời gian ngắn sẽ có thể không phải là phương án tốt. Chúng ta cần xem xét nếu trong trường hợp đối tượng xâm nhập đó không thay đổi thì chúng ta chỉ cần gửi một lần, hoặc một phương án đưa ra có thể sau một khoảng thời gian nhất định sẽ gửi đi cảnh báo tiếp theo.

1. TRIỂN KHAI HỆ THỐNG
   1. Thực hiện phát hiện đối tượng bằng YOLO

*Thiết lập biến môi trường*

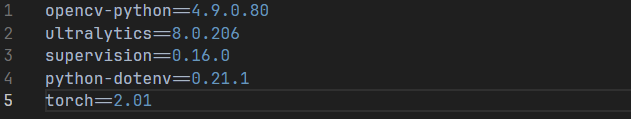


Tạo một file với đuôi .env để thiết lập các biến môi trường, “INTRUSALERT\_PASSWORD” chứa mật khẩu của google app pasword. Để tạo được mật khẩu điều kiện cần là phải có tài khoản Google, đã bật chức năng xác thực hai yếu tố. Tạo mật khẩu tại đây [Link](https://accounts.google.com/v3/signin/challenge/pwd?TL=ALv_Gf9D0DMnIGV2ZqcqkCkZuSFa-R_meJ-WUDNUbl_83iyiYgDmzi37lEmYh-4D&cid=2&continue=https%3A%2F%2Fmyaccount.google.com%2Fapppasswords%3Fpli%3D1%26rapt%3DAEjHL4Oy9L76YbSES2dJTBc7epHmJ_nx8T0jKOxPqVp_XdYxxgXaEXfkuSZTUMgvqu9wyEGNla-osyLYyIlYfzOaNkHtEAKq4biQ9FHS6T2ZQHSZV0kR3t8&flowName=GlifWebSignIn&followup=https%3A%2F%2Fmyaccount.google.com%2Fapppasswords%3Fpli%3D1%26rapt%3DAEjHL4Oy9L76YbSES2dJTBc7epHmJ_nx8T0jKOxPqVp_XdYxxgXaEXfkuSZTUMgvqu9wyEGNla-osyLYyIlYfzOaNkHtEAKq4biQ9FHS6T2ZQHSZV0kR3t8&ifkv=AaSxoQxDp-2Kn4DDTf6pD2ViDCG4vLkTaUO-AI_lKXje1Cpcy6x6yUuJJ0vtjxDiKY56Aqr3ux6pSQ&osid=1&rart=ANgoxcc0kWImS5Mo4j_HCI39X7dVi8AxRozSYbQcka_KZHPRx5AwSurp_DnZtgPASapiBIOOSIjjHMNURqSxPKr5yDY6D6pcUnAVD-ii8Jvu12NaPFA9c9M&rpbg=1&service=accountsettings).

“INTRUSALERT\_FROM\_EMAIL” là tài khoản google đã dùng để tạo mật khẩu, ở đây sẽ được dùng làm tài khoản gửi thư đi. “INTRUSALERT\_TO\_EMAIL” là tài khoản google sẽ nhận được cảnh báo từ hệ thống cảnh báo xâm nhập. Mail người nhận có thể là bất kì ai miễn là tên tài khoản chính xác.

*Cài đặt các thư viện Python cần thiết*

Tạo một tệp với tên “requirements.txt”, ở đây sẽ liệt kê các thư viện Python cần dùng cho dự án này, các thư viện và phiên bản như sau:



Cần khởi chạy bằng cách gõ *pip install -r requirements.txt* trong powershell hoặc terminal trong trình soạn thảo mã tùy chọn.

*Xây dựng hệ thống nhận diện bằng YOLO*

Tạo một class và các hàm khởi tạo cùng các biến cần thiết.

[Link code GitHub](UDA_THI%20GIAC%20MAY%20TINH%20II/DO%20AN%20THI%20GIAC%20MAY%20TINH%20II/src/intrusion%20alert%20system/detection.py)

Dưới đây là giải thích code:

* + - capture\_index: là vị trí camera đầu vào trên máy tính, mặc định là 0. Đối với các thiết bị có nhiều đầu vào có thể lựa chọn vị trí phù hợp trên thiết bị đó.
    - currentIntruderDetection: số lượng xâm nhập được phát hiện, mặc định khi khởi tạo bằng 0.
    - email\_notification: đây là một đối tượng thông báo email được khởi tạo, sẽ đề cập sau.
    - model: tải trọng số cho mô hình.
    - box\_anotation: dùng để vẽ bounding box khi đối tượng được phát hiện.
    - label\_anotator: hiển thị nhãn của đối tượng được phát hiện.
    - device: dùng CPU để thực hiện quá trình phát hiện được diễn ra nhanh hơn.

*Xây dựng hàm nhận diện*

[Link code GitHub](file:///E:\UDA_LEARNING\UDA_THI%20GIAC%20MAY%20TINH%20II\DO%20AN%20THI%20GIAC%20MAY%20TINH%20II\UDA_THI%20GIAC%20MAY%20TINH%20II\DO%20AN%20THI%20GIAC%20MAY%20TINH%20II\src\intrusion%20alert%20system\detection.py)

Dưới đây là giải thích code:

* + - hàm predict nhận đầu vào là một hình ảnh.
    - biến result lưu kết quả nhận diện của mô hình YOLO.
    - Biến detection sẽ trả về hình ảnh được chuyển thành đối tượng Supervision.
    - Biến detection sẽ lọc các đối tượng được trả về có nhãn là 0, tức là trong hình ảnh có chứa người xâm nhập.
    - Hàm này trả về biến detection là các đối tượng xâm nhập được phát hiện.

*Vẽ bounding box cho đối tượng xâm nhập*

[Link code GitHub](file:///E:\UDA_LEARNING\UDA_THI%20GIAC%20MAY%20TINH%20II\DO%20AN%20THI%20GIAC%20MAY%20TINH%20II\UDA_THI%20GIAC%20MAY%20TINH%20II\DO%20AN%20THI%20GIAC%20MAY%20TINH%20II\src\intrusion%20alert%20system\detection.py)

Dưới đây là giải thích code:

* + - Hàm plot\_bboxes nhận vào danh sách các đối tượng Supervision được lọc các class\_id bằng 0.
    - Các đối tượng này được truyền vào hàm annotate để vẽ các bounding box.
    - Hàm trả về các hình ảnh đã được vẽ đối tượng phát hiện trong bounding box.

*Xây dựng vòng lặp để phát hiện đối tượng xâm nhập*

[Link code GitHub](file:///E:\UDA_LEARNING\UDA_THI%20GIAC%20MAY%20TINH%20II\DO%20AN%20THI%20GIAC%20MAY%20TINH%20II\UDA_THI%20GIAC%20MAY%20TINH%20II\DO%20AN%20THI%20GIAC%20MAY%20TINH%20II\src\intrusion%20alert%20system\detection.py)

Dưới đây là giải thích code:

* + - Đầu vào sẽ dùng thư viện opencv-python để đọc ảnh từ camera.
    - Sử dụng vòng lặp vô hạn while True để tạo quá trình phân tích.
    - Trong mỗi khung hình truyền vào sẽ được truyền tới hàm prediction để phát hiện người.
    - Nếu có đối tượng xâm nhập sẽ vẽ bounding box sau đó cắt ảnh nơi khu vực chứa đối tượng và lưu vào một thư mục mang tên images. Đồng thời sử dụng đối tượng gửi cảnh báo để gửi mail tới người nhận.
  1. Xây dựng bộ gửi cảnh báo bằng smtplib

[Link code GitHub](UDA_THI%20GIAC%20MAY%20TINH%20II/DO%20AN%20THI%20GIAC%20MAY%20TINH%20II/src/intrusion%20alert%20system/notifications.py)

Khởi tạo một class Notification, thiết lập các hàm và biến khởi tạo. Hàm *authenticate* sẽ khởi tạo server bằng giao thức SMTP bằng thư viện smtplib. Tại đây sẽ khởi chạy server và đăng nhập vào hệ thống bằng tài khoản và mật khẩu của người gửi. Hàm *send\_email*  thực hiện chức năng gửi email tới người nhận bằng các thông điệp, đồng thời có các tệp đính kèm như hình ảnh phát hiện đối tượng xâm nhập để cảnh báo người nhận. Sau khi gửi thành công thì thoát server.

1. THỰC NGHỆM VÀ ĐÁNH GIÁ
   1. Kết quả triển khai

Dự án này đã thành công triển khai hệ thống phát hiện xâm nhập bằng mô hình YOLO v8, đồng thời gửi thành công cảnh báo có chứa đính kèm hình ảnh qua mail. Dưới đây là một số hình ảnh ví dụ kết quả triển khai dự án, những hình ảnh này đã được gửi thành công qua mail.

CácA person wearing a black and white plaid jacket

Description automatically generatedA person wearing a plaid shirt

Description automatically generatedA person wearing a hoodie

Description automatically generated khung ảnh sau khi được detect bằng mô hình YOLO v8 xong thì thực hiện vẽ hộp ô vuông để bao quanh nơi có chứa người xâm nhập, sau đó thì lưu những hình ảnh đó lại.

HÌnh 10. Kết quả triển khai

* 1. Đánh giá

Hệ thống có thể phát hiện kết quả và gửi cảnh báo trong thời gian rất ngắn, gần như ngay khi có sự xâm nhập, mô hình liền có thể nhận dạng và gửi cảnh báo thành công. Tuy nhiên dự án được chạy trong môi trường thử nghiệm và các điều kiện lí tưởng khá thuận lợi, trong môi trường quá tối hoặc thiếu ánh sáng, mô hình có thể có những sai sót nhất định và hiệu quả kém ổn định. Ngoài ra có sự nhầm lẫn trong quá tình nhận diện, các vật thể có hình dạng giống người ví dụ như áo hoặc các vật thể hình dạng tương tự sẽ làm nhiễu kết quả. Mặc dù còn tồn tại những sai sót tuy nhiên không đáng kể trong việc nhận diện đối tượng.

1. PHƯƠNG HƯỚNG PHÁT TRIỂN TIẾP THEO

Trong tương lai đề tài này tập trung vào việc cải tiến và mở rộng các chức năng của hệ thống. Một số ý tưởng cải tiến trong tương lai có thể xem xét như:

* + - Mở rộng chức năng cảnh báo thông minh: mở rộng hệ thống để có khả năng phát hiện các hành vi đáng ngờ và phản ứng phù hợp, chẳng hạn như việc gửi cảnh báo đến nhiều người được chỉ định, quản lí hoặc các phương pháp bảo mật tự động.
    - Tăng cường tính linh hoạt và cấu hình: phát triển giao diện người dùng thân thiện hơn và tùy chỉnh hệ thống, bao gồm cả việc thay đổi ngưỡng nhận dạng, kích thước vùng quan sát và các thông số khác để phù hợp với môi trường cụ thể.
    - Kiểm thử và tối ưu hóa hệ thống: tiến hành các bài kiểm tra kỹ lưỡng để đảm bảo tính ổn định và hiệu suất của hệ thống trong nhiều điều kiện khác nhau. Từ đó tối ưu hóa các thông số và cấu hình để đạt được kết quả tốt nhất.
    - Triển khai thực tế và đánh giá hiệu suất: triển khai hệ thống trong môi trường thực tế và thu thập dữ liệu về hệ thống và hiệu suất trong nhiều điều kiện khác nhau. Từ đó tối ưu hóa và các thông số và cấu hình để đạt được kết quả tốt nhất.

1. TÀI LIỆU THAM KHẢO

[1] Giải thích về phát hiện đối tượng YOLO, tác giả [Keitazoumana](https://www.datacamp.com/portfolio/keitazoumana), xuất bản tháng 9/2022. <https://www.datacamp.com/blog/yolo-object-detection-explained>.

[2] Yolo là gì? Giải thích về phát hiện đối tượng YOLO, tác giả [Keitazoumana](https://www.datacamp.com/portfolio/keitazoumana), xuất bản tháng 9/2022. <https://www.datacamp.com/blog/yolo-object-detection-explained>.

[3] Điều gì khiến YOLO trở nên phổ biến trong việc phát hiện đối tượng? tác giả [Keitazoumana](https://www.datacamp.com/portfolio/keitazoumana), xuất bản tháng 9/2022. <https://www.datacamp.com/blog/yolo-object-detection-explained>.

[4] Sánchez Hernández, Sergio & Romero, H & Morales, A. (2020). A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework. IOP Conference Series: Materials Science and Engineering. 844. 012024. 10.1088/1757-899X/844/1/012024.

[5] Gửi Email sử dụng SMTP với Python. Vietjack. <https://www.vietjack.com/python/gui_email_trong_python.jsp>.

[6] Giao thức SMTP, tác giả Mrtech xuất bản ngày 7/10/2014. <https://www.mrtech.vn/2014/07/giao-thuc-smtp_10.html>.

|  |  |
| --- | --- |
| **Điểm** | **Nhận xét của giáo viên** |
|  |  |

|  |  |
| --- | --- |
| **Chữ kí sinh viên** | **Chữ kí giảng viên** |