

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI  
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH  
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG PHÁT HIỆN NGƯỜI XÂM  
NHẬP VÀ CẢNH BÁO QUA TELEGRAM**

Giảng viên hướng dẫn : ThS. TRẦN PHONG NHÃ  
Sinh viên thực hiện : ĐẶNG VÕ CÔNG THÀNH  
Lớp : CÔNG NGHỆ THÔNG TIN K60  
Khoá : K60

TP. Hồ Chí Minh, năm 2023

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI  
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH  
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG PHÁT HIỆN NGƯỜI XÂM  
NHẬP VÀ CẢNH BÁO QUA TELEGRAM**

Giảng viên hướng dẫn : ThS. TRẦN PHONG NHÃ

Sinh viên thực hiện : ĐẶNG VÕ CÔNG THÀNH

Lớp : CQ CNTT K60

Khoá : K60

TP. Hồ Chí Minh, năm 2023

**NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP**  
**BỘ MÔN: CÔNG NGHỆ THÔNG TIN**  
-----\*\*\*-----

**Mã sinh viên:** 6051071109

**Họ tên SV:** ĐẶNG VÕ CÔNG THÀNH

**Khóa:** K60

**Lớp:** CQ CNTT K60

**1. Tên đề tài**

XÂY DỰNG HỆ THỐNG PHÁT HIỆN NGƯỜI XÂM NHẬP VÀ CẢNH  
BÁO QUA TELEGRAM

**2. Mục đích, yêu cầu**

**a. Mục đích**

Nghiên cứu ứng dụng công nghệ học sâu để hỗ trợ công việc giám sát, phát hiện và cảnh báo người xâm nhập trong không gian của viện bảo tàng, hàng trưng bày của cửa hàng hay bảo vệ an ninh khu vực bảo vệ ngoài giờ hành chính.

**b. Yêu cầu**

- Yêu cầu công nghệ:

+ Sử dụng ngôn ngữ lập trình python (python 3.7.0), Open CV và YOLO

+ Công cụ: PyCharm Community Edition , Telegram.

- Yêu cầu chức năng:

+ Xây dựng mô hình phát hiện đối tượng con người, thuật toán nhận dạng đối tượng trên Yolo v4-tiny.

+ Xây dựng mô hình polygon (khu vực bảo vệ).

- + Xây dựng mô hình chatbot trên telegram và gửi thông báo.
- + Áp dụng mô hình trong thực tế để phát hiện người trên camera trong thời gian thực và gửi cảnh báo qua telegram khi phát hiện xâm nhập.

### **3. Nội dung và phạm vi đề tài**

#### **a. Nội dung đề tài:**

- + Tổng quan đề tài.
- + Xây dựng ứng dụng.
- + Kết quả thực nghiệm.

#### **b. Phạm vi đề tài:**

Tại các khu bảo tàng, các loại hàng hóa trưng bày, hàng mẫu tại các cửa hàng hay bảo vệ an ninh khu vực ngoài giờ hành chính để giúp đỡ tránh việc quan sát số lượng lớn camera của các chú bảo vệ trên màn ảnh.

### **4. Công nghệ, công cụ và ngôn ngữ lập trình**

- Tìm hiểu ngôn ngữ python và một số thư viện hỗ trợ: Open cv, Yolo, Shapely.

### **5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng**

- Nhận được đầu vào là hình ảnh thực từ camera hoặc video.
- Vẽ được khu vực bảo vệ polygon.
- Nhận diện được đối tượng (con người).
- Gửi được cảnh báo qua chatbot thông qua telegram khi có xâm nhập.

## **6. Giảng viên và cán bộ hướng dẫn**

Họ tên: ThS. TRẦN PHONG NHÃ

Đơn vị công tác: TRƯỜNG ĐẠI HỌC GTVT PHÂN HIỆU TẠI TP.HCM

Điện thoại: 0981058040

Email: tpnha@utc2.edu.vn

**Ngày        tháng        năm 2023**  
**Trưởng BM Công nghệ Thông tin**

**Giảng viên hướng dẫn**

**ThS. Trần Phong Nhã**

**ThS. Tần Phong Nhã**

Đã nhận nhiệm vụ DATN

Sinh viên: Đặng Võ Công Thành

Điện thoại: 0564633602

Email: 6051071109@st.utc2.edu.vn

Kí tên

## LỜI CẢM ƠN

Trước hết, em xin bày tỏ lòng biết ơn chân thành và sâu sắc nhất tới giảng viên hướng dẫn chỉ bảo tận tình của thầy Trần Phong Nhã và sự động viên khích lệ của thầy trong suốt quá trình làm nghiên cứu đề án và hoàn thành đề án của em.

Em xin gửi lời cảm ơn chân thành tới các giảng viên Bộ môn Công Nghệ Thông Tin – trường đại học Giao Thông Vận Tải Phân Hiệu Tại TP HCM và các anh chị khóa trước, các bạn lớp công nghệ thông tin k60 đã tận tình chỉ dạy và hướng dẫn cho em trong suốt quá trình học tập và làm đề án tốt nghiệp. Và em cũng xin gửi lời cảm ơn tới gia đình tôi về sự hỗ trợ không thể thiếu của gia đình. Sự khích lệ, động viên, sự quan tâm, chăm sóc của họ đã giúp em vượt qua tất cả khó khăn để theo học chương trình và hoàn thiện bản luận văn cuối khoá này.

Mặc dù đã hết sức cố gắng hoàn thành đề án tốt nghiệp nhưng chắc chắn sẽ không tránh khỏi những sai sót. Kính mong nhận được sự cảm thông, chỉ bảo tận tình của các quý thầy cô và các bạn.

Em xin chân thành cảm ơn!

## **NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

**Tp. Hồ Chí Minh, ngày ..... tháng ..... năm 2023**

**Giảng viên hướng dẫn**

**ThS. TRẦN PHONG NHÃ**

## Mục lục

<b>TỔNG QUAN.....</b>	<b>1</b>
<b>CHƯƠNG 1: CƠ SỞ LÝ THUYẾT .....</b>	<b>5</b>
<b>1.1 Xử lý ảnh và các vấn đề cơ bản liên quan đến xử lý ảnh. ....</b>	<b>5</b>
1.1.1 Xử lý ảnh. ....	5
1.1.2 Các thành phần cơ bản về xử lý ảnh. ....	6
1.1.2.1 Ảnh số và các điểm ảnh. ....	6
1.1.2.2 Nắn chỉnh biến dạng. ....	7
1.1.2.3 Khử nhiễu. ....	8
1.1.2.4 Chỉnh mức xám. ....	9
1.1.2.5 Biên. ....	9
1.1.2.6 Nhận dạng. ....	9
1.1.2.7 Nén ảnh. ....	11
1.1.2 Một số phương pháp xử lý ảnh số. ....	11
1.1.2.1 Các kỹ thuật lọc nhiễu. ....	11
1.1.2.1.1 Kỹ thuật lọc trung bình. ....	11
1.1.2.1.2 Kỹ thuật lọc trung vị. ....	12
1.1.2.1.3 Lọc thông thấp. ....	12
1.1.2.1.4 Lọc thông cao. ....	13
1.1.2.2 Kỹ thuật phân ngưỡng. ....	14
1.1.2.2.1 Kỹ thuật phân ngưỡng tự động. ....	14
1.1.2.2.2 Phương pháp sử dụng các điểm biên. ....	14
1.1.3 Học máy. ....	15
1.1.3.1 Tổng quan. ....	15
1.1.3.2 Phân loại. ....	15
1.1.3.3 Các ứng dụng của học máy. ....	16



1.1.4	Học sâu.....	17
1.1.4.1	Các khái niệm.....	17
1.1.4.2	Cách hoạt động của học sâu.....	18
1.1.4.3	Các ứng dụng phổ biến của học sâu.....	18
<b>1.2</b>	<b>Tổng quan về YOLO, Thư viện OPEN CV.....</b>	<b>19</b>
1.2.1	Kiến trúc mạng YOLO.....	19
1.2.2	Các phiên bản của YOLO.....	27
1.2.3	OpenCV.....	28
1.2.3.1	Tổng quan OpenCV.....	28
1.2.3.2	Cấu trúc tổng quan.....	29
<b>CHƯƠNG 2:</b>	<b>BÀI TOÁN PHÁT HIỆN NGƯỜI XÂM NHẬP .....</b>	<b>31</b>
<b>2.1</b>	<b>Tổng quan các phương pháp nhận diện đối tượng.....</b>	<b>31</b>
2.1.1	Đặc điểm các loại đối tượng và người.....	31
2.1.2	Các phương pháp nhận diện đối tượng hiện nay.....	31
2.1.2.1	Phương pháp dựa trên cơ sở tri thức.....	31
2.1.2.2	Phương pháp dựa trên đặc trưng bất biến.....	32
2.1.2.3	Phương pháp dựa trên so khớp mẫu.....	33
2.1.2.4	Phương pháp dựa trên diện mạo.....	33
2.1.3	Các kỹ thuật phát hiện người.....	34
<b>2.2</b>	<b>Thuật toán phát hiện người bằng mạng neuron tích chập,YOLO.....</b>	<b>36</b>
2.2.1	Kỹ thuật nhận diện đối tượng bằng mạng neuron tích chập.....	36
2.2.2	Các bước thực hiện kỹ thuật nhận diện đối tượng bằng mạng neuron .....	37
2.2.3	Nhận dạng đối tượng bằng YOLO v4-tiny.....	38
<b>2.3</b>	<b>Chatbot Telegram.....</b>	<b>40</b>
2.3.1	Tổng quan Chatbot.....	40

2.3.2	Lợi ích của chatbot telegram.....	41
<b>CHƯƠNG 3: TRIỂN KHAI ỨNG DỤNG .....</b>		<b>42</b>
3.1	Mô hình hệ thống dự kiến.....	42
3.2	Xây dựng chatbot trên telegram.....	43
3.3	Thuật toán tìm người.....	46
3.3.1	Mô hình thuật toán.....	46
3.3.2	Chú thích bằng lời.....	47
3.3.3	Kết quả thực nghiệm.....	47
<b>KẾT LUẬN .....</b>		<b>52</b>
<b>TÀI LIỆU THAM KHẢO .....</b>		<b>54</b>

## Danh mục hình ảnh

Hình 1.1	Quá trình xử lý ảnh .....	5
Hình 1.2	Các bức cơ bản trong quá trình xử lý ảnh .....	5
Hình 1.3	Ảnh số .....	6
Hình 1.4	Độ phân giải ảnh .....	7
Hình 1.5	Ảnh thu nhận và ảnh mong muốn .....	7
Hình 1.6	Ảnh trước và sau khi khử nhiễu .....	8
Hình 1.7	Hình ảnh trước và sau khi chỉnh mức xám .....	9
Hình 1.8	Mô hình lọc thông cao .....	13
Hình 1.9	Các giải thuật học máy .....	16
Hình 1.10	Mối quan hệ giữa học sâu và các lĩnh vực liên quan .....	17
Hình 1.11	Sơ đồ kiến trúc YOLO.....	19
Hình 1.12	Kiến trúc của một output của một modle YOLO.....	21
Hình 1.13	Các bảng đồ đặc trưng của mạng YOLO.....	22
Hình 1.14	Xác định hộp neo cho một vật thể.....	23
Hình 1.15	Minh họa dự đoán khung đối tượng.....	25
Hình 1.16	Non-max suppression.....	26
Hình 1.17	Các thành phần của thư viện OpenCV.....	29
Hình 2.1	Mô hình YOLO.....	35
Hình 2.2	Bảng ma trận RGB.....	37
Hình 2.3	Mạng neuron tích chập tích chập.....	37
Hình 2.4	Mạng draket-53.....	39
Hình 3.1	Mô hình hệ thống dự kiến .....	44
Hình 3.2	Tên botfather .....	43
Hình 3.3	Tạo chatbot .....	44
Hình 3.4	Lấy id .....	45
Hình 3.5	Viết chương trình .....	45
Hình 3.6	Sơ đồ tìm kiếm người trên video .....	46
Hình 3.7	Vẽ và lưu các điểm .....	48

Hình 3.8	Phát hiện đối tượng .....	49
Hình 3.9	Phát cảnh báo .....	50
Hình 3.10	Gửi cảnh báo qua chatbot telegram .....	51

## **TỔNG QUAN**

### **1. Tổng quan về đề tài.**

Cuộc sống hiện đại, công nghệ phát triển biến ước mơ máy móc thay thế con người dần trở nên hiện thực. Những công việc nặng nhọc, độc hại, không an toàn, nhàm chán dần dần sẽ được máy móc thay thế con người thực hiện. Đồng thời công việc sẽ được thực hiện nhanh hơn, nhiều hơn, tốt hơn, chính xác và chuyên nghiệp hơn. Tại các viện bảo tàng hay các nơi trưng bày hàng mẫu hiện nay đa số được trang bị camera giám sát số lượng rất là lớn nhằm đảm bảo an toàn an ninh và an toàn 24/24. Tuy nhiên việc sử dụng camera giám sát số lượng lớn như hiện nay mới chỉ đảm bảo việc quan sát và lưu lại các hình ảnh mà chưa thể tự động phát hiện và cảnh báo các nguy cơ từ đối tượng xâm nhập trái phép vào thời gian cấm và sự lơ đãng thiếu chú ý của các anh bảo vệ hay có ý định đánh cắp tài sản.

Một số dòng camera thông minh có thể phát hiện một số đối tượng chuyển động, tuy nhiên việc phát hiện này chưa đáp ứng yêu cầu chẳng hạn camera giám sát an ninh nhưng phát hiện cả chó, mèo, chuột hay một động vật nào đó chuyển động trước ống kính, điều đó làm giảm độ tin cậy của việc giám sát cũng như gây phiền phức, khó chịu cho chủ nhân của nó. Việc ứng dụng học sâu (Deep Learning) kết hợp cùng các camera giám sát an ninh giúp phát hiện chính xác đối tượng xâm nhập là hướng tiếp cận có thể khắc phục được những hạn chế trên. Phương pháp Deep Learning được kết hợp nhiều với các kỹ thuật của lĩnh vực Thị giác máy tính (Computer Vision) giúp tăng khả năng phát hiện xâm nhập, đạt tỷ lệ phát hiện cao và tỷ lệ cảnh báo sai thấp.

### **2. Mục tiêu.**

Nghiên cứu ứng dụng công nghệ học sâu để hỗ trợ trong công việc giám sát, phát hiện và cảnh báo người xâm nhập trong không gian của viện bảo tàng, hàng trưng bày, bảo vệ an ninh khu vực bảo vệ ngoài giờ hành chính để giúp đỡ các chú bảo vệ tránh sự trung tâm nhìn vào toàn bộ các camera trên màn hình lớn tăng hiệu suất làm việc.

### **3. Phạm vi.**

Tại các khu bảo tàng, các loại hàng hóa trưng bày, hàng mẫu hay bảo vệ an ninh khu vực ngoài giờ hành chính.

### **4. Tổng quan về lĩnh vực nghiên cứu.**

#### **\* Nhận dạng đối tượng (Object Detection), phát hiện người trong ảnh tĩnh và video.**

Để phát hiện người trong ảnh tĩnh hay video thì thường người ta sử dụng các thuật toán nhận dạng đối tượng trên ảnh tĩnh hay video rồi sau đó áp dụng các thuật toán để phát hiện người.

Các thuật toán nhận dạng đối tượng hiện nay phần lớn đều áp dụng cho ảnh tĩnh, do đó để có thể áp dụng cho video thì cần thiết phải tìm cách lấy các ảnh tĩnh từ video. Video thực chất được tạo ra từ các khung hình liên tiếp, do đó nếu tách các khung hình này thành các ảnh tĩnh thì có thể áp dụng các phương pháp nhận dạng cho ảnh tĩnh trên video.

Nhận dạng đối tượng là quá trình phân loại các đối tượng được biểu diễn theo một mô hình nào đó và gán chúng vào một lớp chuyên đề (gán cho đối tượng một tên gọi) dựa trên những quy luật và các mẫu chuẩn. Quá trình nhận dạng dựa vào các mẫu đối tượng đã biết trước được gọi là nhận dạng có kiểm định (hay phân loại có kiểm định), nhận dạng đối tượng không dựa theo mẫu được gọi là nhận dạng không kiểm định (hay phân loại không kiểm định).

Trên thế giới kỹ thuật nhận dạng đối tượng đã được nghiên cứu và ứng dụng rất sớm vào trong lĩnh vực khoa học máy tính từ thế kỷ trước như: nhận dạng vân tay, nhận dạng chữ viết, nhận dạng giọng nói, nhận dạng khuôn mặt, phương tiện giao thông, động vật, đồ vật, có thể kể đến một số nghiên cứu của tác giả J.J.Hull và cộng sự (1992), R. Kimmel và G. Sapiro (2003), Yi Li (2005), Mark Williams Pontin (2007), từ các kết quả nhận dạng đã phục vụ hiệu quả cho các công tác đảm bảo an ninh quốc gia, bảo mật trong các giao dịch tài chính, chăm công, quản lý nhân sự, hành chính,...

Ở Việt Nam, các kỹ thuật nhận dạng đã được nghiên cứu và ứng dụng, điển hình như một số tác giả: Trần Trung Kiên (2013) đã nghiên cứu hệ thống nhận dạng gương

mặt trong video giám sát; Tống Văn Ngọc (2018) đã nghiên cứu nhận dạng và phát hiện hành động người dùng thị giác máy tính; Nguyễn Thị Thủy (2018) đã nghiên cứu phương pháp nhận dạng khuôn mặt người và ứng dụng trong quản lý nhân sự;... và kỹ thuật nhận dạng cũng đã được sử dụng nhận dạng đối tượng trên dữ liệu ảnh viễn thám, điển hình như các nghiên cứu: Trịnh Thị Hoài Thu và cộng sự (2012) đã nghiên cứu ứng dụng phương pháp định hướng đối tượng và phương pháp phân loại dựa vào điểm ảnh trong nhận dạng phân loại đối tượng trên dữ liệu ảnh Worldview...

Có thể thấy trong cách tiếp cận về kỹ thuật nhận dạng, có 3 cách tiếp cận thường được sử dụng là:

- Nhận dạng dựa theo không gian: Trong kỹ thuật này, các đối tượng là các đối tượng định lượng. Mỗi đối tượng được biểu diễn bằng một véc tơ nhiều chiều, mỗi chiều là một tham số thể hiện một đặc điểm của đối tượng đó.
- Nhận dạng dựa vào kỹ thuật mạng neuron tích chập: Mạng neuron tích chập là hệ thống bao gồm nhiều phần tử xử lý đơn giản (mạng neuron tích chập) hoạt động song song. Tính năng của hệ thống này tùy thuộc vào cấu trúc của hệ, các trọng số liên kết mạng neuron tích chập và quá trình tính toán tại các mạng neuron tích chập đơn lẻ.
- Nhận dạng dựa theo cấu trúc: Đối tượng ngoài cách biểu diễn theo định lượng, chúng còn tồn tại ở nhiều kiểu đối tượng mang tính định tính.

#### **\* Các phương pháp nghiên cứu phát hiện đối tượng hiện nay.**

Nhiều phương pháp phát hiện đối tượng khác nhau sử dụng thị giác máy tính đã được phát triển và ứng dụng rộng rãi trong đời sống thực tiễn. Các phương pháp này phát hiện đối tượng với ba bước chính. Bước thứ nhất là dựa vào các thuộc tính của đối tượng như màu sắc, kết cấu bề mặt và hình dạng để trích chọn các đặc trưng ảnh. Bước thứ hai là sử dụng tập dữ liệu mẫu để xác định các tham số cho các bộ nhận dạng đối tượng trong ảnh. Bước thứ 3 là sử dụng bộ nhận dạng để xác định đối tượng trong các ảnh đầu vào bất kỳ.

Việc áp dụng đột phát và nhanh chóng của Deep Learning vào năm 2012 đã đưa vào sự tồn tại các thuật toán và phương pháp phát hiện đối tượng hiện đại và chính xác

cao như R-CNN, Fast-RCNN, Faster-RCNN, RetinaNet và nhanh hơn nhưng rất chính xác như SSD và YOLO.

## **5. Cấu trúc báo cáo thực tập tốt nghiệp.**

### **Tổng quan.**

- Tổng quan về đề tài.
- Mục tiêu.
- Phạm vi.
- Tổng quan về lĩnh vực nghiên cứu.
- Cấu trúc báo cáo đồ án tốt nghiệp.

### **Chương 1: Cơ sở lý thuyết**

- Xử lý ảnh và các vấn đề liên quan đến xử lý ảnh.
- Tổng quan các phương pháp nhận diện đối tượng người trên ảnh và video.
- Tổng quan về YOLO v4-tiny, thư viện OPENCV và chatbot telegram.

### **Chương 2: Bài toán phát hiện**

- Tổng quan các phương pháp nhận diện đối tượng, phát hiện người trên ảnh và video.
- Thuật toán phát hiện người bằng mạng neuron tích chập, YOLO v4-tiny

### **Chương 3: Triển khai ứng dụng**

- Mô hình hệ thống dự kiến.
- Thuật toán tìm kiếm người trên video.

### **Kết luận**

- Kết quả đạt được.
- Hạn chế.
- Hướng phát triển.

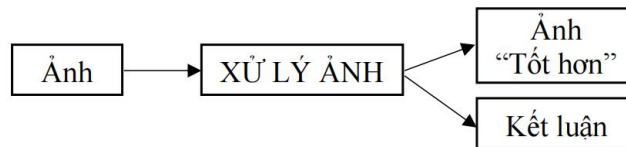


## CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

### 1.1 Xử lý ảnh và các vấn đề cơ bản liên quan đến xử lý ảnh.

#### 1.1.1 Xử lý ảnh.

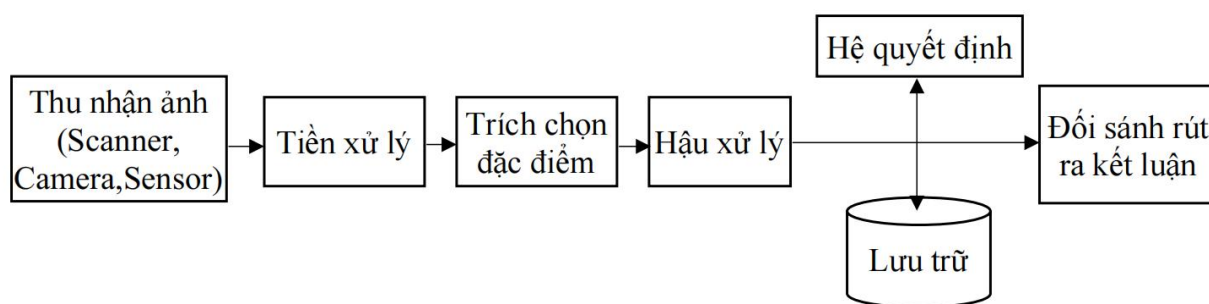
Con người thu nhận thông tin qua các giác quan, trong đó thị giác đóng vai trò quan trọng nhất. Những năm trở lại đây với sự phát triển của phần cứng máy tính, lĩnh vực xử lý ảnh và đồ họa cũng phát triển một cách mạnh mẽ và có nhiều ứng dụng trong cuộc sống. Xử lý ảnh và đồ họa đóng một vai trò quan trọng trong tương tác người máy. Quá trình xử lý ảnh được xem như là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh “tốt hơn” hoặc là một kết luận như trong hình 1.1.



Hình 1.1 Quá trình xử lý ảnh.

Ảnh có thể xem là tập hợp các điểm ảnh trong đó mỗi điểm ảnh được xem như là đặc trưng cường độ sáng hay một dấu hiệu nào đó tại một vị trí nào đó của đối tượng trong không gian và nó có thể xem như một hàm  $n$  biến  $P(c_1, c_2, \dots, c_n)$ . Do đó, ảnh trong xử lý ảnh có thể xem như ảnh  $n$  chiều.

Để xử lý một ảnh đầu vào nói trên, hệ thống xử lý ảnh sẽ sử dụng sơ đồ tổng quát như hình 2. Trong đó ảnh thu nhận từ các thiết bị như scanner, camera sẽ qua bước tiền xử lý, rồi tới bước trích chọn đặc điểm, tiếp đến là bước hậu xử lý, tiếp đến là hệ quyết định và lưu trữ ảnh, cuối cùng là đối sánh rút ra kết luận.

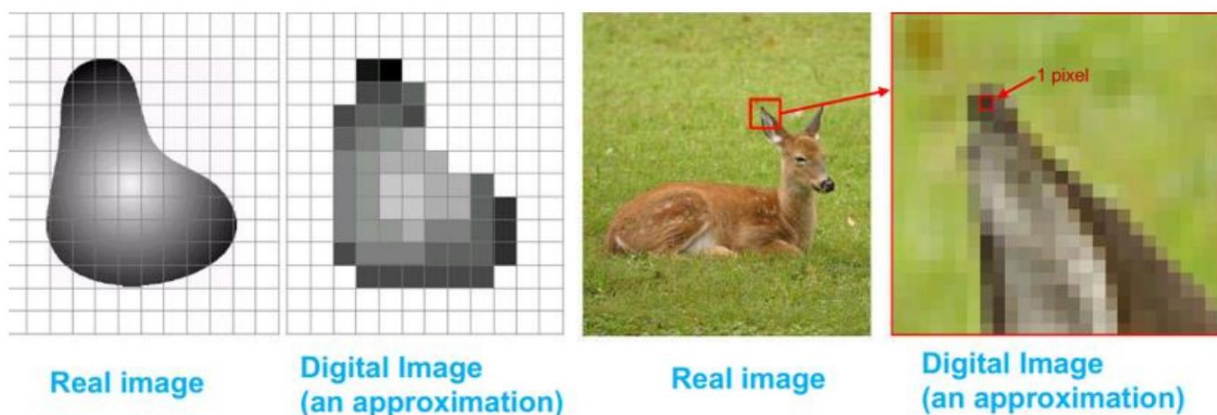


Hình 1.2 Các bước cơ bản trong quá trình xử lý ảnh.

## 1.1.2 Các thành phần cơ bản về xử lý ảnh.

### 1.1.2.1 Ảnh số và các điểm ảnh.

- Điểm ảnh (pixel) có vị trí  $(x,y)$  và có độ xám  $I(x,y)$ .
- Với ảnh màu thì mỗi điểm ảnh sẽ có 3 giá trị tương ứng với độ sáng của các màu đỏ, xanh lục, xanh dương (RGB).
- Ảnh số: "Một hình ảnh có thể được định nghĩa là hàm hai chiều,  $f(x, y)$ , trong đó  $x$  và  $y$  là tọa độ không gian (mặt phẳng) và biên độ của  $f$  tại bất kỳ cặp tọa độ  $(x, y)$  nào được gọi là cường độ hoặc mức độ màu xám của hình ảnh tại điểm đó. Khi  $x, y$  và các giá trị cường độ của  $f$  đều là các đại lượng hữu hạn, rời rạc, thì gọi hình ảnh là hình ảnh kỹ thuật số". Hay có thể hiểu một cách đơn giản rằng "Ảnh số là số hóa làm cho một hình ảnh kỹ thuật số trở thành một xấp xỉ của một cảnh thực".



Hình 1.3 Ảnh số

- Điểm ảnh: "hình ảnh kỹ thuật số chứa một số lượng hữu hạn các hàng và cột của các phần tử. Mỗi phần tử được gọi là pixel".

- Độ phân giải: "độ phân giải là thước đo của chi tiết rõ ràng nhỏ nhất trong ảnh, được tính là số điểm (pixel) trên một đơn vị khoảng cách (dpi)".

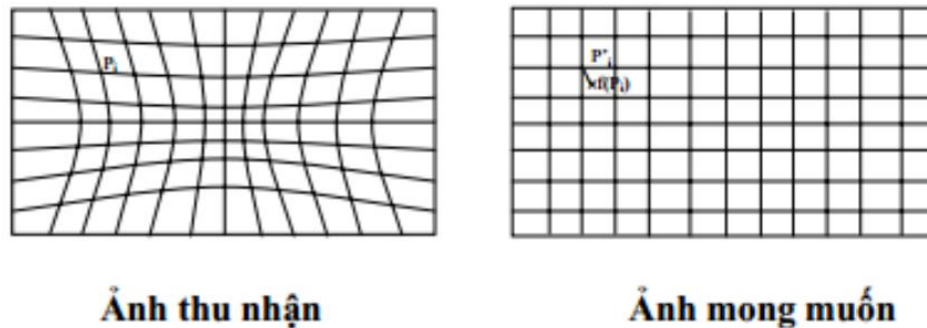


Hình 1.4 Độ phân giải ảnh.

- Như hình 4 tuy cùng kích thước nhưng độ phân giải khác nhau, và độ phân giải càng thấp thì càng mờ. Như ảnh đầu tiên bên trái hình 4 sẽ được hiểu là chiều rộng có 175 điểm ảnh và chiều cao có 256 điểm ảnh.[15]

#### 1.1.2.2 Nắn chỉnh biến dạng.

- Ảnh thu nhận thường bị biến dạng do các thiết bị quang học và điện tử.



Hình 1.5 Ảnh thu nhận và ảnh mong muốn.

Để khắc phục người ta sử dụng các phép chiếu, các phép chiếu thường được xây dựng trên tập các điểm điều khiển.

Giả sử  $(P_i, P_i')$   $i = 1, n$ , có  $n$  các tập điều khiển.

Tìm hàm  $f$  sao cho  $P_i \rightarrow f(P_i)$  sao cho

$$\sum_{i=1}^n \|f(P_i) - P_i'\|^2 \rightarrow \min(1)$$

Giả sử ảnh bị biến đổi chỉ bao gồm: Tịnh tiến, quay, tỷ lệ, biến dạng bậc nhất tuyến tính. Khi đó hàm  $f$  có dạng:

Từ (1), (2) ta có

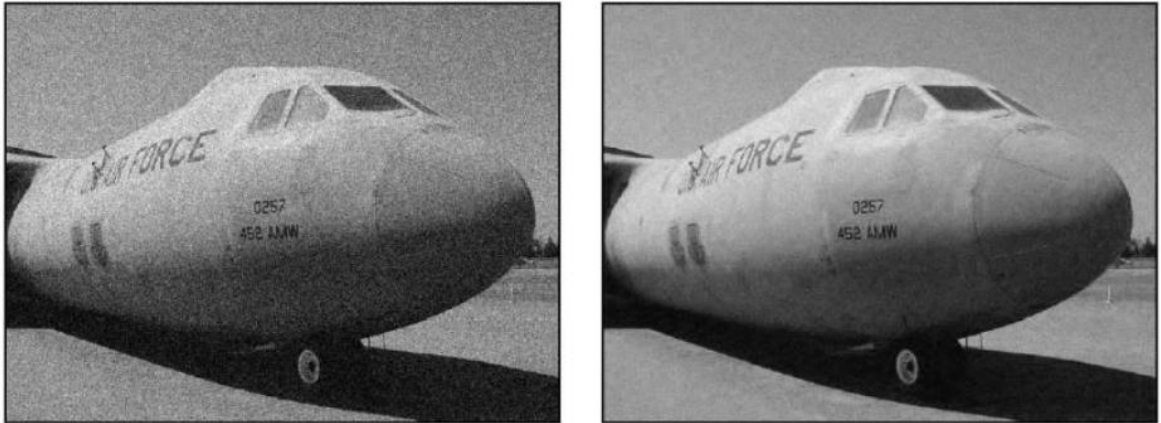
$$\phi = \sum_{i=1}^n (f(P_i) - P_i')^2 = \sum_{i=1}^n (a_1x_i + b_1y_i + c_1 - x_i')^2 + (a_2x_i + b_2y_i + c_2 - y_i')^2 \quad (3)$$

Để cho  $\phi \rightarrow \min$  cần thỏa mãn điều kiện trong công thức (4)

$$\begin{cases} \frac{\partial \phi}{\partial a_1} = 0 \\ \frac{\partial \phi}{\partial b_1} = 0 \\ \frac{\partial \phi}{\partial c_1} = 0 \end{cases} \Leftrightarrow \begin{cases} \sum_{i=1}^n a_1x_i^2 + \sum_{i=1}^n b_1x_iy_i + \sum_{i=1}^n c_1x_i = \sum_{i=1}^n x_i x_i' \\ \sum_{i=1}^n a_1x_iy_i + \sum_{i=1}^n x_iy_i^2 + \sum_{i=1}^n c_1y_i = \sum_{i=1}^n y_i x_i' \\ \sum_{i=1}^n a_1x_i + \sum_{i=1}^n b_1y_i + nc_1 = \sum_{i=1}^n x_i' \end{cases} \quad (4)$$

Giải hệ phương trình tuyến tính trên ta tìm được các ẩn số  $a_1, b_1, c_1$  và  $a_2, b_2, c_2$  và sau đó xác định được hàm  $f$ .

### 1.1.2.3 Khử nhiễu.



Hình 1.6 Ảnh trước và sau khi khử nhiễu.

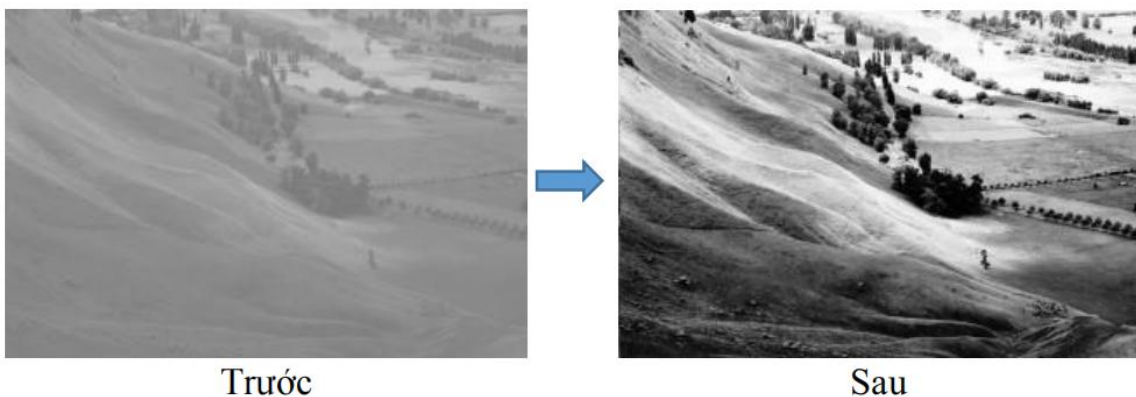
Có 2 loại nhiễu cơ bản trong quá trình thu nhận ảnh:

- Nhiễu hệ thống: là nhiễu có quy luật có thể khử bằng các phép biến đổi.
- Nhiễu ngẫu nhiên: vết bản không rõ nguyên nhân  $\rightarrow$  khắc phục bằng các phép lọc.

#### 1.1.2.4 Chỉnh mức xám.

Nhằm khắc phục tính không đồng đều của hệ thống gây ra. Thông thường có 2 hướng tiếp cận:

- Giảm số mức xám: Thực hiện bằng cách nhóm các mức xám gần nhau thành một bó. Trường hợp chỉ có 2 mức xám thì chính là chuyển về ảnh đen trắng. Ứng dụng: in ảnh màu ra máy in đen trắng.
- Tăng số mức xám: Thực hiện nội suy ra các mức xám trung gian bằng kỹ thuật nội suy. Kỹ thuật này nhằm tăng cường độ mịn cho ảnh.



Hình 1.7 Hình ảnh trước và sau khi chỉnh mức xám.

#### 1.1.2.5 Biên.

Biên là vấn đề chủ yếu trong phân tích ảnh vì các điểm trích chọn trong quá trình phân tích ảnh đều dựa vào biên. Mỗi điểm ảnh có thể là biên nếu ở đó có sự thay đổi đột ngột về mức xám. Tập hợp các điểm biên tạo thành biên hay đường bao quanh của ảnh.

#### 1.1.2.6 Nhận dạng.

Nhận dạng tự động (automatic recognition), mô tả đối tượng, phân loại và phân nhóm các mẫu là những vấn đề quan trọng trong thị giác máy, được ứng dụng trong nhiều ngành khoa học khác nhau. Tuy nhiên, một câu hỏi đặt ra là: mẫu (pattern) là gì? Watanabe, một trong những người đi đầu trong lĩnh vực này đã định nghĩa: “Ngược lại với hỗn loạn (chaos), mẫu là một thực thể (entity), được xác định một cách áng chừng đồ tín hiệu tiếng nói. Khi biết một mẫu nào đó, để nhận dạng hoặc phân loại mẫu đó có thể:

- Hoặc phân loại có mẫu (supervised classification), chẳng hạn phân tích phân biệt (discriminant analysis), trong đó mẫu đầu vào được định danh như một thành phần của một lớp đã xác định.
- Hoặc phân loại không có mẫu (unsupervised classification hay clustering) trong đó các mẫu được gán vào các lớp khác nhau dựa trên một tiêu chuẩn đồng dạng nào đó. Các lớp này cho đến thời điểm phân loại vẫn chưa biết hay chưa được định danh.
  - Hệ thống nhận dạng tự động bao gồm ba khâu tương ứng với ba giai đoạn chủ yếu sau đây:
    - + Thu nhận dữ liệu và tiền xử lý.
    - + Biểu diễn dữ liệu.
    - + Nhận dạng, ra quyết định.
  - Bốn cách tiếp cận khác nhau trong lý thuyết nhận dạng là:
    - + Đối sánh mẫu dựa trên các đặc trưng được trích chọn.
    - + Phân loại thống kê.
    - + Đối sánh cấu trúc.
    - + Phân loại dựa trên mạng neuron tích chập nhân tạo.
      - Trong các ứng dụng rõ ràng là không thể chỉ dùng có một cách tiếp cận đơn lẻ để phân loại “tối ưu” do vậy cần sử dụng cùng một lúc nhiều phương pháp và cách tiếp cận khác nhau. Do vậy, các phương thức phân loại tổ hợp hay được sử dụng khi nhận dạng và nay đã có những kết quả có triển vọng dựa trên thiết kế các hệ thống lai (hybrid system) bao gồm nhiều mô hình kết hợp.
      - Việc giải quyết bài toán nhận dạng trong những ứng dụng mới, nảy sinh trong cuộc sống không chỉ tạo ra những thách thức về thuật giải, mà còn đặt ra những yêu cầu về tốc độ tính toán. Đặc điểm chung của tất cả những ứng dụng đó là những đặc điểm đặc trưng cần thiết thường là nhiều, không thể do chuyên gia đề xuất, mà phải được trích chọn dựa trên các thủ tục phân tích dữ liệu.

### 1.1.2.7 Nén ảnh.

Nhằm giảm thiểu không gian lưu trữ. Thường được tiến hành theo cả hai cách khuynh hướng là nén có bảo toàn và không bảo toàn thông tin. Nén không bảo toàn thì thường có khả năng nén cao hơn nhưng khả năng phục hồi thì kém hơn. Trên cơ sở hai khuynh hướng, có 4 cách tiếp cận cơ bản trong nén ảnh:

Nén ảnh thống kê: Kỹ thuật nén này dựa vào việc thống kê tần suất xuất hiện của giá trị các điểm ảnh, trên cơ sở đó mà có chiến lược mã hóa thích hợp. Một ví dụ điển hình cho kỹ thuật mã hóa này là.

Nén ảnh không gian: Kỹ thuật này dựa vào vị trí không gian của các điểm ảnh để tiến hành mã hóa. Kỹ thuật lợi dụng sự giống nhau của các điểm ảnh trong các vùng gần nhau. Ví dụ cho kỹ thuật này là mã nén.

Nén ảnh sử dụng phép biến đổi: Đây là kỹ thuật tiếp cận theo hướng nén không bảo toàn và do vậy, kỹ thuật thường nên hiệu quả hơn chính là tiếp cận theo kỹ thuật nén này.

## 1.1.2 Một số phương pháp xử lý ảnh số.

### 1.1.2.1 Các kỹ thuật lọc nhiễu.

#### 1.1.2.1.1 Kỹ thuật lọc trung bình.

Với lọc trung bình, mỗi điểm ảnh được thay thế bằng trung bình trọng số của các điểm lân cận.

Tư tưởng của thuật toán lọc trung bình: Sử dụng một cửa sổ lọc (ma trận  $3 \times 3$ ) quét qua lần lượt từng điểm ảnh của ảnh đầu vào input. Tại vị trí mỗi điểm ảnh lấy giá trị của các điểm ảnh tương ứng trong vùng  $3 \times 3$  của ảnh gốc "lấp" vào ma trận lọc. Giá trị điểm ảnh của ảnh đầu ra là giá trị trung bình của tất cả các điểm ảnh trong cửa sổ lọc. Việc tính toán này khá đơn giản với hai bước gồm tính tổng các thành phần trong cửa sổ lọc và sau đó chia tổng này cho số các phần tử của cửa sổ lọc.

Ảnh đầu vào là  $I(x,y)$ ,  $T$  là ma trận mẫu.

Tính  $I(x,y)$ ,  $T$

Tính  $\bar{I}_{(x,y)} = \frac{I_{(x,y)} * T}{M}$  trong đó  $M$  là tổng giá trị trọng số của  $T$

So sánh với ngưỡng  $\theta$  để tính lại  $I(x,y)$  như sau:

$$I_{(x,y)} = \begin{cases} I_{(x,y)} & \text{if } |I_{(x,y)} - \bar{I}_{(x,y)}| \leq \theta \\ \bar{I}_{(x,y)} & \text{if } |I_{(x,y)} - \bar{I}_{(x,y)}| > \theta \end{cases}$$

#### 1.1.2.1.2 Kỹ thuật lọc trung vị.

Trung vị được viết bởi công thức:  $v(m,n) = \text{Trungvi}(y(m - k, n-l))$  với  $(k,l) \in W$  hoặc: cho một dãy  $x_1, x_2, \dots, x_n$  được sắp xếp theo một trật tự khi đó  $x_n$ : điểm trung vị được tính như sau:

$$X_n = X\left(\frac{n}{2} + 1\right) \text{ nếu } n \text{ là lẻ hoặc } X_n = \frac{X\left(\frac{n}{2} + 1\right) + X\left(\frac{n}{2}\right)}{2} \text{ nếu } n \text{ là chẵn}$$

Kỹ thuật này đòi hỏi các điểm ảnh trong cửa sổ phải xếp theo thứ tự tăng dần hay giảm dần so với giá trị trung vị. Kích thước cửa sổ thường được chọn sao cho số điểm ảnh trong cửa sổ ảnh là lẻ. Các cửa sổ thường dùng là  $3 \times 3, 5 \times 5, 7 \times 7$ .

Thuật toán lọc trung vị:

B1: với mỗi điểm ảnh  $I(x,y)$  ta lấy cửa sổ  $W \times W$

B2: sắp xếp các giá trị điểm ảnh trong vòng cửa sổ theo một trật tự

B3: tính  $I_{tv}$  theo công thức ở trên

B4: hiệu chỉnh lại  $I(x,y)$

$$I_{(x,y)} = I_{(x,y)} \text{ nếu } |I_{(x,y)} - I_{tv}| \leq \theta \text{ hoặc } I_{(x,y)} = I_{tv} \text{ nếu } |I_{(x,y)} - I_{tv}| > \theta$$

Lọc trung vị là phi tuyến vì:

$$\text{Trungvi}(x(m)+y(m)) \neq \text{trungvi}(x(m)) + \text{trungvi}(y(m)).$$

Hữu ích cho việc loại bỏ các điểm ảnh hay các hàng mà vẫn bảo toàn độ phân giải.

Hiệu quả giảm đi khi số điểm nhiều trong cửa sổ lớn hơn hay bằng một nửa số điểm trong cửa sổ.

#### 1.1.2.1.3 Lọc thông thấp.

Lọc thông thấp thường được sử dụng để làm trơn nhiễu. Trong kỹ thuật này người ta thường dùng một số nhân chập sau:

$$H_a = \frac{1}{8} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad H_b = \frac{1}{(b+2)^2} \begin{bmatrix} 1 & b & 1 \\ b & b^2 & b \\ 1 & b & 1 \end{bmatrix}$$



Đễ dàng nhận thấy khi  $b=1$   $H_b$  chính là nhân chập  $H_1$  (lọc trung bình). Để hiểu rõ hơn bản chất khử nhiễu cộng của các bộ lọc này, viết lại phương trình thu nhận ảnh dưới dạng:

$$X_{qs}[m,n] = X_{goc}[m,n] + \eta[m,n]$$

Trong đó  $\eta[m,n]$  là nhiễu cộng có phương sai  $\sigma_n^2$ . Như vậy theo cách tính của lọc trung bình ta có:

$$Y(m,n) = \frac{1}{N_w} \sum \sum_{(k,l) \in w} X_{goc}(m-k, n-l) + \eta m,n$$

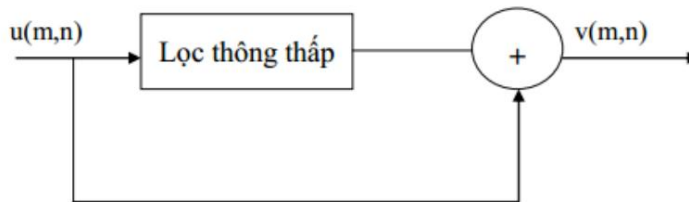
Hay

$$Y(m,n) = \frac{1}{N_w} \sum \sum_{(k,l) \in w} X_{goc}(m-k, n-l) + \frac{\sigma_n^2}{N_w}$$

Như vậy nhiễu trong ảnh giảm đi  $N_w$  lần.

#### 1.1.2.1.4 Lọc thông cao.

Lọc thông cao được định nghĩa:  $h_{HP} = \delta(m,n) - h_{LP}(m,n)$  với  $h_{LP}(m,n)$  là lọc thông thấp. Bộ lọc thông cao có thể được cài đặt như sau:



Hình 1.8 Mô hình lọc thông cao.

Bộ lọc thông cao dùng trong trích chọn biên và làm trơn ảnh. Có thể thấy biên là điểm có độ biến thiên nhanh về giá trị mức xám. Theo quan điểm về tần số tín hiệu, như vậy các điểm biên ứng với các thành phần tần số cao. Do vậy chúng ta có thể dùng bộ lọc thông cao để cải thiện: lọc các thành phần tần số thấp và chỉ giữ lại thành phần tần số cao. Vì thế lọc thông cao thường dùng làm trơn biên trước khi tiến hành các thao tác với biên ảnh.

Một số mặt nạ dùng trong lọc thông cao

$$H_1 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad H_2 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad H_3 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -1 \\ 1 & -2 & 1 \end{bmatrix}$$

Các nhân chập thông cao có đặc tính chung là tổng các hệ số của bộ lọc bằng 1.

### 1.1.2.2 Kỹ thuật phân ngưỡng.

#### 1.1.2.2.1 Kỹ thuật phân ngưỡng tự động.

Cơ sở của kỹ thuật này dựa theo nguyên lý trong vật lý. Dựa vào nguyên lý thống kê (entropy), dựa vào toán học, dựa vào các điểm cực trị địa phương để tách.

- Giả sử có ảnh  $I(M*N)$ .
- $G$  là số mức xám của ảnh (trên lý thuyết).
- Gọi  $t(g)$  là số điểm ảnh có mức xám  $\leq g$  momen quán tính trung bình có mức xám nhỏ hơn hoặc bằng các mức xám  $g$ .

$$M(g) = \frac{1}{t(g)} \sum_{i=0}^g ih(i)$$

$$T(g) = \sum_{i=0}^g H(i)$$

Hàm  $f: g \rightarrow f(g)$  hàm được tính như sau

$$f(g) = \frac{t(g)}{M_{xn} - t(g)} [M(g) - M(G-1)]^2$$

Tìm ra một giá trị  $\theta$  nào đó sao cho  $f$  đạt max khi đó  $\theta$  là ngưỡng cần tìm ( $f(\theta) = \max \Rightarrow \theta$  là ngưỡng).

#### 1.1.2.2.2 Phương pháp sử dụng các điểm biên.

Điểm biên là điểm mà ở đó có sự thay đổi đột ngột về giá trị mức xám. Nó là điểm nằm ở biên giới của các đối tượng ảnh hay giữa các đối tượng ảnh và nền. Do mức xám của các điểm biên sẽ thể hiện được các vùng tốt hơn nên biểu đồ mức xám của các điểm biên sẽ cho kết quả chính xác hơn so với biểu đồ mức xám tổng thể. Việc xác định ngưỡng dựa trên toán tử dò biên vô hướng laplace. Ngưỡng được xác định trước hết bằng cách tính laplace của ảnh đầu vào. Cách đơn giản nhất là nhân chập với mặt nạ sau đây:

$$H = \begin{bmatrix} 0 & -1 & 0 \\ 4 & -1 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Lúc này ta có một biểu đồ mức xám của ảnh ban đầu mà ta chỉ quan tâm tới các điểm ảnh có giá trị laplace lớn, những điểm ảnh trong nhóm 85% hoặc lớn hơn sẽ nằm trong biểu đồ này, còn các điểm khác thì không. Ngưỡng vừa sử dụng sẽ được tìm trong biểu đồ mức xám vừa tìm được.

### **1.1.3 Học máy.**

#### **1.1.3.1 Tổng quan.**

Học máy (Machine Learning) là một tập con của AI (Artificial Intelligence – Trí tuệ nhân tạo). Theo định nghĩa của Tom Mitchell, Một chương trình máy tính được cho là học để thực hiện một nhiệm vụ T từ kinh nghiệm E, nếu hiệu suất thực hiện công việc T của nó được đo bởi chỉ số hiệu suất P và được cải thiện bởi kinh nghiệm E theo thời gian.

#### **1.1.3.2 Phân loại.**

Có hai loại phương pháp học máy chính:

- Phương pháp quy nạp: Máy học/phân biệt các khái niệm dựa trên dữ liệu đã thu thập được trước đó. Phương pháp này cho phép tận dụng được nguồn dữ liệu rất nhiều và sẵn có.
- Phương pháp suy diễn: Máy học/phân biệt các khái niệm dựa vào các luật. Phương pháp này cho phép tận dụng được các kiến thức chuyên ngành để hỗ trợ máy tính.

Hiện nay, các thuật toán đều cố gắng tận dụng được ưu điểm của hai phương pháp này.

#### **1.1.3.3 Các giải thuật học máy.**

- Học có giám sát (Supervised Learning): Một thuật toán machine learning được gọi là học có giám sát nếu việc xây dựng mô hình dự đoán mối quan hệ giữa đầu vào và đầu ra được thực hiện dựa trên các cặp (đầu vào, đầu ra) đã biết trong tập huấn luyện.
- + Học có giám sát là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (input, outcome) đã biết từ trước.

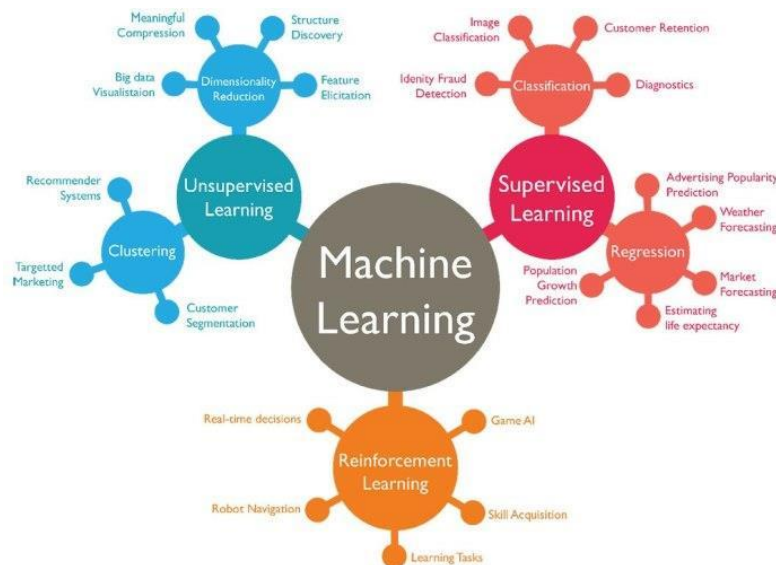
+ Cặp dữ liệu này còn được gọi là (data, label), tức (dữ liệu, nhãn).

+ Học có giám sát là nhóm phổ biến nhất trong các thuật toán Machine Learning.

- Học không giám sát (Unsupervised Learning): Các thuật toán mà dữ liệu huấn luyện chỉ bao gồm các dữ liệu đầu vào mà không có đầu ra tương ứng. Các thuật toán machine learning có thể không dự đoán được đầu ra nhưng vẫn trích xuất được những thông tin quan trọng dựa trên mối liên quan giữa các điểm dữ liệu.

- Học bán giám sát (Semi-Supervised Learning): Là những thuật toán mà tập huấn luyện bao gồm các cặp (đầu vào, đầu ra) và dữ liệu khác chỉ có đầu vào.

- Học củng cố (Reinforcement Learning): Là các thuật toán machine learning có thể không yêu cầu dữ liệu huấn luyện mà mô hình học cách ra quyết định bằng cách giao tiếp với môi trường xung quanh. Các thuật toán này liên tục ra quyết định và nhận phản hồi từ môi trường để tự củng cố hành vi.[3]



Hình 1.9 Các giải thuật học máy.

#### 1.1.3.4 Các ứng dụng của học máy.

Học máy có ứng dụng rộng khắp trong các ngành khoa học/sản xuất, đặc biệt những ngành cần phân tích khối lượng dữ liệu khổng lồ. Một số ứng dụng phổ biến của học máy như:

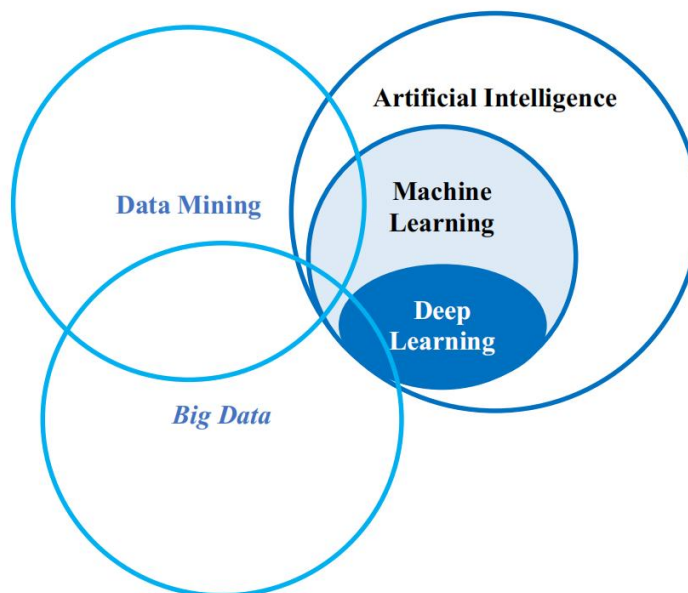
- Xử lý ngôn ngữ tự nhiên (Natural Language Processing): xử lý văn bản, giao tiếp người - máy, ...

- Nhận dạng (Pattern Recognition): nhận dạng tiếng nói, chữ viết tay, vân tay, thị giác máy (Computer Vision) ...
- Tìm kiếm (Search Engine)
- Chẩn đoán trong y tế: phân tích ảnh X-quang, các hệ chuyên gia chẩn đoán tự động.
- Tin sinh học: phân loại chuỗi gene, quá trình hình thành gene/protein.
- Vật lý: phân tích ảnh thiên văn, tác động giữa các hạt ...
- Phát hiện gian lận tài chính (financial fraud): gian lận thẻ tín dụng.
- Phân tích thị trường chứng khoán (stock market analysis).
- Chơi trò chơi: tự động chơi cờ, hành động của các nhân vật ảo.
- Robot: là tổng hợp của rất nhiều ngành khoa học, trong đó học máy tạo nên hệ thần kinh/bộ não của người máy.[18]

#### 1.1.4 Học sâu.

##### 1.1.4.1 Các khái niệm.

Học sâu (Deep Learning): Là một nhóm thuật toán nhỏ của học máy lấy ý tưởng dựa trên mạng neuron tích chập (Neural Network) của con người. Học sâu thường yêu cầu lượng dữ liệu lớn và nguồn tài nguyên sử dụng nhiều hơn các phương pháp thông thường, tuy nhiên cho độ chính xác cao hơn.



Hình 1.10 Mối quan hệ giữa học sâu và các lĩnh vực liên quan.

#### 1.1.4.2 Cách hoạt động của học sâu.

Cách thức hoạt động của thuật toán học sâu diễn ra như sau: Các dòng thông tin sẽ được trải qua nhiều lớp cho đến lớp sau cùng. Lấy quy trình học của con người làm ví dụ cụ thể. Qua các lớp đầu tiên sẽ tập trung vào việc học các khái niệm cụ thể hơn trong khi các lớp sâu hơn sẽ sử dụng thông tin đã học để nghiên cứu và phân tích sâu hơn trong các khái niệm trừu tượng. Quy trình xây dựng biểu diễn dữ liệu này được gọi là trích xuất tính năng.

Kiến trúc phức tạp của việc học sâu được cung cấp từ mạng lưới thần kinh sâu với khả năng thực hiện trích xuất tính năng tự độ truy xuất, xử lý dữ liệu,...được Model thể hiện rõ.

#### 1.1.4.3 Các ứng dụng phổ biến của học sâu.

**Trợ lý ảo** Cho dù đó là Alexa hay Siri hay Cortana, những trợ lý ảo của các nhà cung cấp dịch vụ trực tuyến đều sử dụng học sâu để giúp hiểu lời nói của người dùng và ngôn ngữ con người sử dụng khi họ tương tác với máy.

**Dịch thuật** Theo cách tương tự, thuật toán học sâu có thể tự động dịch giữa các ngôn ngữ. Điều này có thể hỗ trợ mạnh mẽ cho khách du lịch, doanh nhân và những người làm việc trong chính phủ.

**Máy bay không người lái và xe ô tô tự hành** Cách một chiếc xe tự hành “nhìn” được thực tế đường đi và di chuyển, dừng lại, tránh một quả bóng trên đường hoặc xe khác là thông qua các thuật toán học sâu. Các thuật toán càng nhận được nhiều dữ liệu thì càng có khả năng hành động giống như con người trong quá trình xử lý thông tin.

**Chatbots và dịch vụ chatbots** Chatbots hỗ trợ dịch vụ chăm sóc khách hàng cho rất nhiều công ty để có thể đáp ứng một cách tối ưu những câu hỏi của khách hàng với số lượng ngày càng tăng nhờ vào việc học sâu.

**Nhận dạng khuôn mặt** Học sâu được sử dụng để nhận diện khuôn mặt không chỉ vì mục đích bảo mật mà còn cho việc gắn thẻ mọi người trên các bài đăng trên Facebook. Những thách thức đối với thuật toán học sâu trong nhận diện khuôn mặt là nhận biết chính người đó ngay cả khi họ đã thay đổi kiểu tóc, cạo râu hoặc khi hình ảnh được chụp trong điều kiện thiếu ánh sáng.

**Tô màu hình ảnh** Chuyển đổi hình ảnh đen trắng thành màu trước – đây là một nhiệm vụ được thực hiện tỉ mỉ bởi bàn tay con người. Ngày nay, các thuật toán học sâu có thể sử dụng ngữ cảnh và các đối tượng trong các hình ảnh để tô màu chúng với kết quả thật ấn tượng và chính xác.

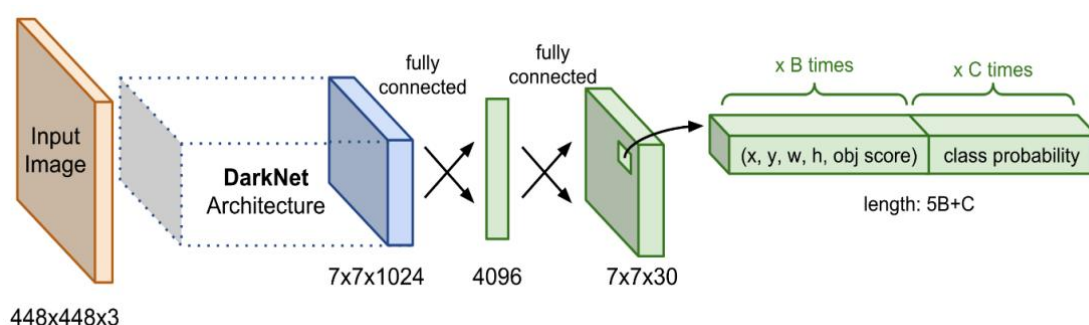
**Y học và dược phẩm** Chẩn đoán chính xác bệnh tật và khối u, đồng thời kê đơn các loại thuốc phù hợp nhất bộ gen của mỗi bệnh nhân. Deep learning trong lĩnh vực y tế đã nhận được sự đầu tư của nhiều công ty dược phẩm và y tế lớn.

**Mua sắm và giải trí được cá nhân hóa** Việc cá nhân hóa thông tin người dùng giúp các hệ thống thương mại điện tử có thể đưa ra các đề xuất cho những gì người dùng nên xem tiếp theo và những đề xuất đó thường chính xác là những gì người dùng cần,... Đó chính là ứng dụng của học sâu trong các ứng dụng mua sắm và giải trí.[3]

## 1.2 Tổng quan về YOLO, Thư viện OPEN CV và chatbot Telegram.

### 1.2.1 Kiến trúc mạng YOLO.

Kiến trúc mạng YOLO bao gồm: Mạng cơ sở (base network) là các mạng tích chập làm nhiệm vụ trích xuất đặc trưng. Phần phía sau là những lớp bổ sung (Extra layers) được áp dụng để phát hiện vật thể trên bản đồ đặc trưng (feature map) của mạng cơ sở. Mạng cơ sở của YOLO sử dụng chủ yếu là các lớp tích chập và các lớp kết nối đầy đủ. Các kiến trúc YOLO cũng khá đa dạng và có thể tùy biến thành các phiên bản cho nhiều hình dạng đầu vào khác nhau.



Hình 1.11 Sơ đồ kiến trúc YOLO.

Kiến trúc mạng YOLO trong hình 1.11 bao gồm: Thành phần Darknet Architecture được gọi là mạng cơ sở có tác dụng trích xuất đặc trưng. Đầu ra (output) của mạng cơ sở là một bản đồ đặc trưng có kích thước  $7*7*1024$  sẽ được sử dụng làm

đầu vào (input) cho các lớp bổ sung có tác dụng dự đoán nhãn và tọa độ khung giới hạn của vật thể.

Hiện tại YOLO đang hỗ trợ 2 đầu vào chính là 416\*416 và 608\*608. Mỗi một đầu vào sẽ có một thiết kế các lớp riêng phù hợp với hình dạng (shape) của đầu vào. Sau khi đi qua các lớp tích chập thì hình dạng giảm dần theo cấp số nhân là 2. Cuối cùng chúng ta thu được một bản đồ đặc trưng có kích thước tương đối nhỏ để dự báo vật thể trên từng ô của bản đồ đặc trưng.

Kích thước của bản đồ đặc trưng sẽ phụ thuộc vào đầu vào. Đối với đầu vào 416\*416 thì bản đồ đặc trưng có các kích thước là 13\*13, 26\*26 và 52\*52. Và khi đầu vào là 608\*608 sẽ tạo ra bản đồ đặc trưng 19\*19, 38\*38, 72\*72.

### **Đầu ra (output) của YOLO**

Đầu ra của mô hình YOLO là một véc tơ sẽ bao gồm các thành phần:

$$y^T = [\rho_0, \underbrace{\langle t_x, t_y, t_w, t_h \rangle}_{\text{bounding box}}, \underbrace{\langle \rho_1, \rho_2, \dots, \rho_c \rangle}_{\text{scores of c classes}}]$$

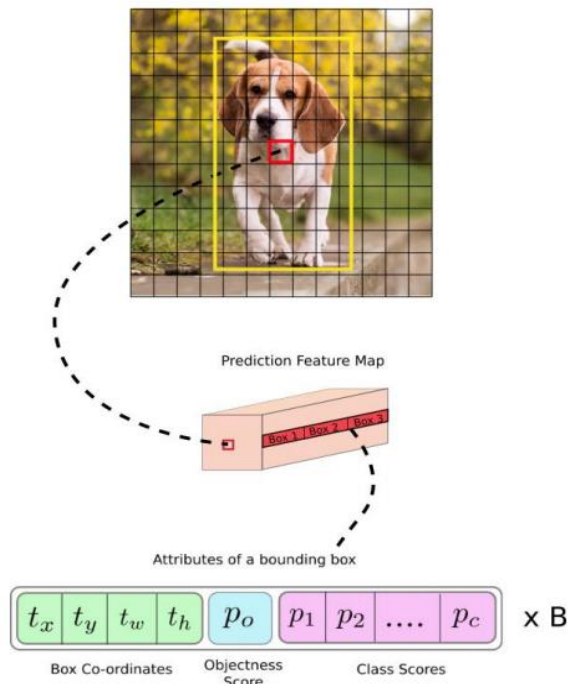
Trong đó:

- $\rho_0$  là xác suất dự báo vật thể xuất hiện trong khung giới hạn.
- Bounding box là xác suất dự báo vật thể xuất hiện trong khung giới hạn.
- Score of c classes là vector phân phối xác suất dự báo của các class.

Việc hiểu đầu ra rất quan trọng để có thể cấu hình tham số chuẩn xác khi huấn luyện model qua các open source như darknet. Như vậy đầu ra sẽ được xác định theo số lượng classes theo công thức  $(n\_class + 5)$ . Nếu huấn luyện 80 classes thì sẽ có đầu ra là 85. Trường hợp áp dụng 3 anchors/cell thì số lượng tham số đầu ra sẽ là:

$$(n\_class + 5) \times 3 = 85 \times 3 = 255$$



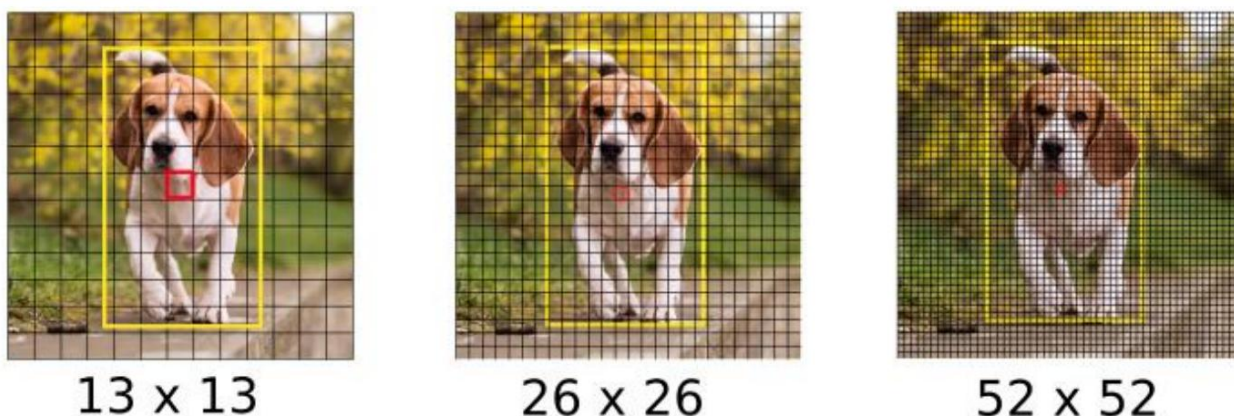


Hình 1.12 Kiến trúc của một output của một model YOLO.

Trong hình 1.12 hình ảnh gốc là một bản đồ đặc trưng kích thước 13\*13. Trên mỗi một ô của bản đồ đặc trưng chúng ta lựa chọn ra 3 hộp neo với kích thước khác nhau lần lượt là Box 1, Box 2, Box 3 sao cho tâm của các hộp neo trùng với ô. Khi đó đầu ra của YOLO là một véc tơ kết hợp của 3 khung giới hạn. Các thuộc tính của một khung giới hạn được mô tả như dòng cuối cùng trong hình.

### Dự báo trên nhiều bản đồ đặc trưng

YOLO dự báo trên nhiều bản đồ đặc trưng. Những bản đồ đặc trưng ban đầu có kích thước nhỏ giúp dự báo được các đối tượng kích thước lớn. Những bản đồ đặc trưng sau có kích thước lớn hơn trong khi hộp neo được giữ cố định kích thước nên sẽ giúp dự báo các vật thể kích thước nhỏ.



Hình 1.13 Các bản đồ đặc trưng của mạng YOLO.

Hình 1.13 các bản đồ đặc trưng của mạng YOLO với hình dạng đầu vào là  $416 \times 416$ , đầu ra là 3 bản đồ đặc trưng có kích thước lần lượt là  $13 \times 13$ ,  $26 \times 26$  và  $52 \times 52$ . Trên mỗi một ô của các bản đồ đặc trưng chúng ta sẽ áp dụng 3 hộp neo để dự đoán vật thể. Như vậy số lượng các hộp neo khác nhau trong một mô hình YOLO sẽ là 9 (3 bản đồ đặc trưng  $\times$  3 hộp neo).

Đồng thời trên một bản đồ đặc trưng hình vuông  $S \times S$ , mô hình YOLO sinh ra một số lượng hộp neo là:  $S \times S \times 3$ . Như vậy số lượng hộp neo trên một bức ảnh sẽ là:  $(13 \times 13 + 26 \times 26 + 52 \times 52) \times 3 = 10647$  (hộp neo). Đây là một số lượng rất lớn và là nguyên nhân khiến quá trình huấn luyện mô hình YOLO vô cùng chậm bởi chúng ta cần dự báo đồng thời nhãn và khung giới hạn trên đồng thời 10647 khung giới hạn. Một số lưu ý khi huấn luyện YOLO:

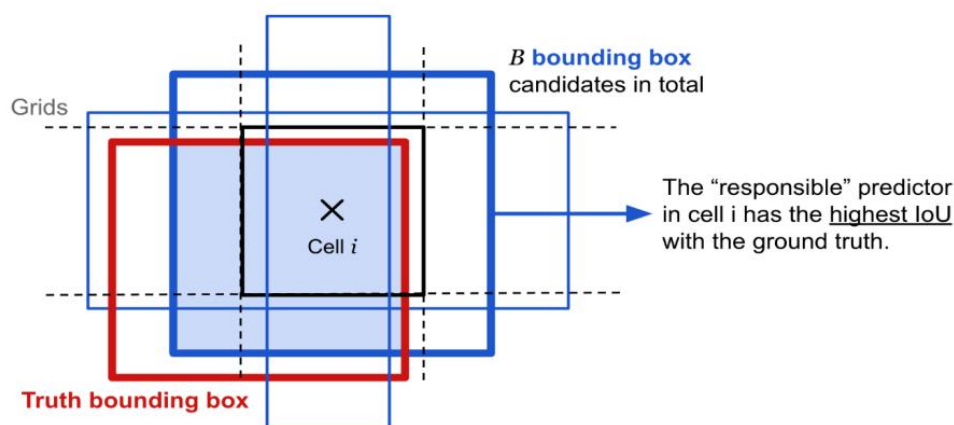
- + Khi huấn luyện YOLO sẽ cần phải có RAM dung lượng lớn hơn để lưu được 10647 khung giới hạn như trong kiến trúc này.
- + Không thể thiết lập các `batch_size` quá lớn như trong các mô hình classification vì rất dễ bị đầy bộ nhớ. Gói darknet của YOLO đã chia nhỏ một batch thành các subdivisions cho vừa với RAM.
- + Thời gian xử lý của một bước trên YOLO lâu hơn rất rất nhiều lần so với các mô hình classification. Do đó nên thiết lập các bước giới hạn huấn luyện cho YOLO nhỏ. Đối với các tác vụ nhận diện dưới 5 classes, dưới 5000 bước là có thể thu được nghiệm tạm chấp

nhận được. Các mô hình có nhiều classes hơn có thể tăng số lượng các bước theo cấp số nhân.

### Hộp neo (Anchor box):

Để tìm được khung giới hạn cho vật thể, YOLO sẽ cần các hộp neo làm cơ sở ước lượng. Những hộp neo này sẽ được xác định trước và sẽ bao quanh vật thể một cách tương đối chính xác. Sau này thuật toán hồi quy khung giới hạn sẽ tinh chỉnh lại hộp neo để tạo ra khung giới hạn dự đoán cho vật thể. Trong một mô hình YOLO:

+ Mỗi một vật thể trong hình ảnh huấn luyện được phân bố về một hộp neo. Trong trường hợp có từ 2 hộp neo trở lên cùng bao quanh vật thể thì chúng ta sẽ xác định hộp neo mà có IoU (intersection over union) với khung giới hạn thực tế là cao nhất.



Hình 1.14 Xác định hộp neo cho một vật thể.

Trong hình 1.14 từ Cell i ta xác định được 3 hộp neo viền xanh như trong hình. Cả 3 hộp neo này đều giao nhau với khung giới hạn của vật thể. Tuy nhiên chỉ hộp neo có đường viền dày nhất màu xanh được lựa chọn làm hộp neo cho vật thể bởi nó có IoU so với khung giới hạn thực tế là cao nhất.

+ Mỗi một vật thể trong hình ảnh huấn luyện được phân bố về một ô trên bản đồ đặc trưng mà chứa điểm giữa của vật thể. Chẳng hạn như hình chú chó trong hình 3.13 sẽ được phân về cho ô màu đỏ vì điểm giữa của ảnh chú chó rơi vào đúng ô này. Từ ô ta sẽ xác định các hộp neo bao quanh hình ảnh chú chó. Như vậy khi xác định một vật thể sẽ cần xác định 2 thành phần gắn liền với nó là (ô, hộp neo). Không chỉ riêng mình ô hoặc chỉ mình hộp neo.

### Hàm mất mát (loss function):

Hàm mất mát của YOLO chia thành 2 phần:  $L_{loc}$ (localization loss) đo lường sai số của khung giới hạn và  $L_{cls}$ (confidence loss) đo lường sai số của phân phối xác suất các classes.

$$L_{loc} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$
$$L_{cls} = \underbrace{\sum_{i=0}^{S^2} \sum_{j=0}^B (1_{ij}^{obj} + \lambda_{noobj} (1 - 1_{ij}^{obj})) (C_{ij} - \hat{C}_{ij})^2}_{cell\ contain\ object} + \underbrace{\sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (p_i(c) - \hat{p}_i)^2}_{probability\ distribution\ classes}$$
$$L = L_{loc} + L_{cls}$$

Trong đó:

$1_i^{obj}$ : Hàm indicator có giá trị 0,1 nhằm xác định xem ô thứ i có chứa vật thể hay không. Bằng 1 nếu chứa vật thể và 0 nếu không chứa. Cho biết khung giới hạn thứ j của ô thứ i có phải là bounding box của vật thể được dự đoán hay không? (xem hình 14).

$C_{ij}$ : Điểm tin cậy của ô thứ i,  $P(\text{contain object}) * \text{IoU}(\text{predict bbox}, \text{ground truth bbox})$ .

$\hat{C}_{ij}$ : Điểm tự tin dự đoán.

C: Tập hợp tất cả các lớp.

$P_i(c)$ : Xác suất có điều kiện, có hay không ô thứ i có chứa một đối tượng của lớp  $c \in C$ .

$\hat{p}_i(c)$ : xác suất có điều kiện dự đoán.

$L_{loc}$  là hàm mất mát của khung giới hạn dự báo so với thực tế.

$L_{cls}$  là hàm mất mát của phân phối xác suất. Trong đó tổng đầu tiên là mất mát của dự đoán có vật thể trong ô hay không? Và tổng thứ 2 là mất mát của phân phối xác suất nếu có vật thể trong ô. Ngoài ra để điều chỉnh hàm mất mát (loss function) trong trường hợp dự đoán sai khung giới hạn ta thông qua hệ số điều chỉnh  $\lambda_{coord}$  và muốn giảm nhẹ hàm mất mát trong trường hợp ô không chứa vật thể bằng hệ số điều chỉnh  $\lambda_{noobj}$

### Dự báo khung giới hạn (bounding box):

Để dự báo khung giới hạn cho một vật thể chúng ta dựa trên một phép biến đổi từ hộp neo và ô.

YOLO dự đoán khung giới hạn sao cho nó sẽ không lệch khỏi vị trí trung tâm quá nhiều. Nếu khung giới hạn dự đoán có thể đặt vào bất kỳ phần nào của hình ảnh, như trong mạng regional proposal network, việc huấn luyện mô hình có thể trở nên không ổn định.

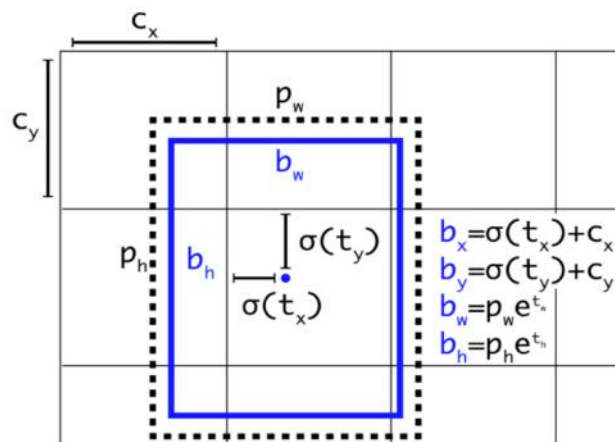
Cho một hộp neo có kích thước  $(P_w, P_h)$  tại cell nằm trên feature map với góc trên cùng bên trái của nó là  $(c_x, c_y)$ , mô hình dự đoán 4 tham số  $(t_x, t_y, t_w, t_h)$  trong đó 2 tham số đầu là độ lệch (offset) so với góc trên cùng bên trái của cell và 2 tham số sau là tỷ lệ so với hộp neo. Và các tham số này sẽ giúp xác định khung giới hạn dự đoán  $b$  có tâm  $(b_x, b_y)$  và kích thước  $(b_w, b_h)$  thông qua hàm sigmoid và hàm exponential như các công thức bên dưới:

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$

$$b_w = p_w e^{t_w}$$

$$b_h = p_h e^{t_h}$$



Hình 1.15 Minh họa dự đoán khung đối tượng.

Hình 1.15 công thức ước lượng khung giới hạn từ hộp neo. Hình chữ nhật nét đứt bên ngoài là hộp neo có kích thước là  $(P_w, P_h)$ . Tọa độ của một khung giới hạn sẽ được xác định dựa trên đồng thời cả hộp neo và cell mà nó thuộc về. Điều này giúp kiểm soát vị trí của khung giới hạn dự đoán đâu đó quanh vị trí của ô và khung giới hạn mà không vượt quá xa ra bên ngoài giới hạn này.

### **Non-max suppression:**

Do thuật toán YOLO dự báo ra rất nhiều khung giới hạn trên một bức ảnh nên đối với những cell có vị trí gần nhau, khả năng các khung hình bị overlap là rất cao. Trong trường hợp đó YOLO sẽ cần đến non-max suppression để giảm bớt số lượng các khung hình được sinh ra một cách đáng kể.



Hình 1.16 Non-max suppression.

Trong hình 1.16 từ 3 khung giới hạn ban đầu cùng bao quanh chiếc xe đã giảm xuống còn một khung giới hạn cuối cùng. Các bước của non-max suppression:

- + Bước 1: Đầu tiên chúng ta sẽ tìm cách giảm bớt số lượng các khung giới hạn bằng cách lọc bỏ toàn bộ những khung giới hạn có xác suất chứa vật thể nhỏ hơn một ngưỡng nào đó, thường là 0.5.
- + Bước 2: Đối với các khung giới hạn giao nhau, non-max suppression sẽ lựa chọn ra một khung giới hạn có xác suất chứa vật thể là lớn nhất. Sau đó tính toán chỉ số giao thoa IoU với các khung giới hạn còn lại.

Nếu chỉ số này lớn hơn ngưỡng threshold thì điều đó chứng tỏ 2 khung giới hạn đang chồng lên nhau rất cao. Chúng ta sẽ xóa các khung giới hạn có xác suất thấp hơn và

giữ lại khung giới hạn có xác suất cao nhất. Cuối cùng, thu được một khung giới hạn duy nhất cho một vật thể.[7]

### 1.2.2 Các phiên bản của YOLO.

Mô hình YOLO được mô tả lần đầu tiên bởi Joseph Redmon và các cộng sự. Được công bố trong bài viết năm 2015. YOLO có 3 phiên bản là YOLOv1, YOLOv2, YOLOv3.

\* **YOLO v1:** Sử dụng Framework Darknet được train trên tập ImageNet-1000. Nó không thể tìm thấy các object nhỏ nếu chúng xuất hiện dưới dạng một cụm. Phiên bản này gặp khó khăn trong việc phát hiện các đối tượng nếu hình ảnh có kích thước khác với hình ảnh được train.

\* **YOLOv2:** Đặt tên là YOLO9000 đã được Joseph Redmon và Ali Farhadi công bố vào cuối năm 2016. Cải tiến chính của phiên bản này tốt hơn, nhanh hơn, tiên tiến hơn để bắt kịp faster R-CNN (phương pháp sử dụng Region Proposal Network), xử lý được những vấn đề gặp phải của YOLOv1. Sự thay đổi của YOLOv2 so với YOLOv1:

- **Batch Normalization:** Giảm sự thay đổi giá trị unit trong hidden layer, do đó sẽ cải thiện được tính ổn định của Neural Network.

- **Higher Resolution Classifier:** Kích thước đầu vào trong YOLOv2 được tăng từ  $224 \times 224$  lên  $448 \times 448$ .

- **Anchor boxes:** Dự đoán khung giới hạn và được thiết kế cho tập dữ liệu đã cho sử dụng clustering.

- **Fine-Grained Features:** YOLOv2 chia ảnh thành  $13 \times 13$  grid cells, do đó có thể phát hiện được những object nhỏ hơn, đồng thời cũng hiệu quả với các object lớn.

- **Multi-Scale Training:** YOLOv1 có điểm yếu là phát hiện các đối tượng với các kích cỡ đầu vào khác nhau. Điều này được giải quyết bằng YOLO v2, nó được train với kích thước ảnh ngẫu nhiên trong khoảng  $320 \times 320$  đến  $608 \times 608$ .

- **Darknet 19:** YOLOv2 sử dụng Darknet 19 với 19 convolutional layers, 5 max pooling layers và 1 softmax layer.

\* **YOLO v3:** Công bố vào tháng 4 năm 2018 với việc phát hiện, phân loại chính xác đối tượng, và được xử lý thời gian thực. Những cải tiến chính của YOLOv3 so với hai phiên bản trước gồm:

- **Bounding Box Predictions:** Cung cấp score mỗi khung giới hạn sử dụng logistic regression.



- **Class Predictions:** Sử dụng logistic classifiers cho mọi class thay vì softmax.
- **Feature Pyramid Networks (FPN):** Giới thiệu residual block và FPN.
- **Darknet-53:** YOLOv3 sử dụng Darknet 53 với 53 convolutional layers.
- \* **YOLO v4:** Phiên bản YOLOv4 ra mắt vào năm 2020 và là phiên bản tiếp theo của YOLO. Nó kết hợp các kỹ thuật tiên tiến như CSPDarknet53 (một kiến trúc mạng nâng cao), PANet (Path Aggregation Network), và các cải tiến khác để tăng độ chính xác và tốc độ suy luận. YOLOv4 cũng hỗ trợ việc huấn luyện trên nhiều nền tảng khác nhau và có khả năng phát hiện đối tượng chính xác trên nhiều loại dữ liệu.
- \* **YOLO v5** là một phiên bản nâng cấp và độc lập của YOLO được phát triển bởi Alexey Bochkovskiy và các thành viên của Ultra-Light-Weight Object Detection (ULWOD) team. Phiên bản này được công bố vào năm 2020 và mang lại nhiều cải tiến về hiệu suất và chất lượng phát hiện đối tượng.
- \* **YOLO v6** YOLOv6 (còn được gọi là MT-YOLOv6) là một mô hình phát hiện đối tượng một giai đoạn dựa trên kiến trúc YOLO. Mô hình được phát triển bởi các nhà nghiên cứu tại Meituan. Điểm đáng chú ý ở YOLOv6 là YOLOv6 đạt được hiệu suất mạnh hơn YOLOv5 khi được so sánh với bộ dữ liệu MS COCO.
- \* **YOLOv7** là trình phát hiện đối tượng thời gian thực một giai đoạn. Nó đã được giới thiệu với gia đình YOLO vào ngày 22 tháng 7. Hơn nữa, YOLOv7 đang là công cụ phát hiện đối tượng thời gian thực nhanh nhất và chính xác nhất cho đến nay. YOLOv7 đã thiết lập một tiêu chuẩn quan trọng trong việc nâng cao hiệu suất của nó.

### 1.2.3 OpenCV.

#### 1.2.3.1 Tổng quan OpenCV.

OpenCv (Open Source Computer Vision) là một thư viện mã nguồn mở về thị giác máy với hơn 500 hàm và hơn 2500 các thuật toán đã được tối ưu về XLA, và các vấn đề liên quan tới thị giác máy. OpenCV được thiết kế một cách tối ưu, sử dụng tối đa mạnh của các dòng chip đa lõi... để thực hiện các phép tính toán trong thời gian thực, nghĩa là tốc độ đáp ứng của nó thể đủ nhanh cho các ứng dụng thông thường.



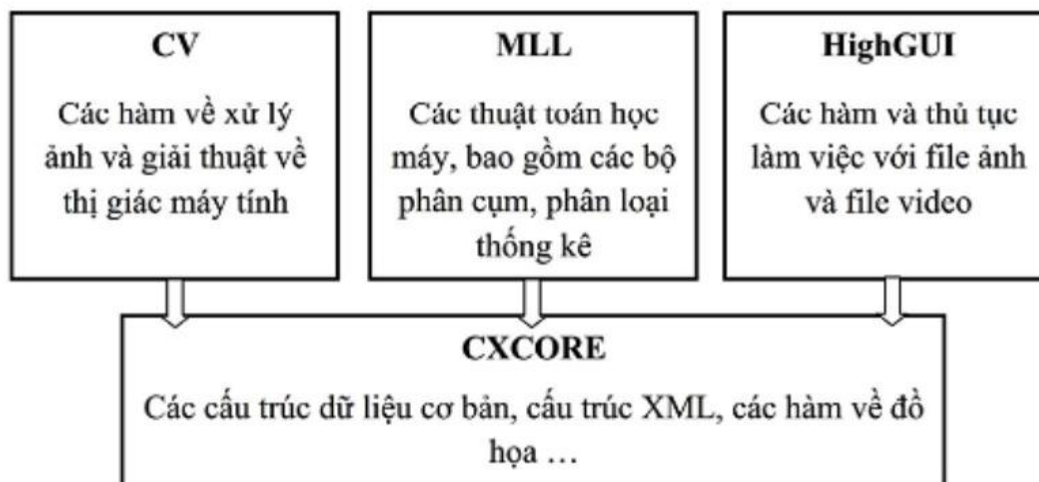
OpenCv là thư viện được thiết kế để chạy trên nhiều nền tảng khác nhau (cross-platform), nghĩa là nó có thể chạy trên hệ điều hành Window, Linux, Mac, iOS... Việc sử dụng thư viện OpenCv tuân theo các quy định về sử dụng phần mềm mã nguồn mở BSD do đó bạn có thể sử dụng thư viện này một cách miễn phí cho các mục đích phi thương mại lẫn thương mại.

Dự án về OpenCv được khởi động từ những năm 1999, đến năm 2000 nó được giới thiệu trong một hội nghị của IEEE về các vấn đề trong thị giác máy và nhận dạng, tuy nhiên bản OpenCV 1.0 mãi tới tận năm 2006 mới chính thức được công bố và năm 2008 bản 1.1 (prerelease) mới được ra đời. Tháng 10 năm 2009, bản OpenCV thế hệ thứ hai ra đời (thường gọi là phiên bản 2.x), phiên bản này có giao diện của C++ (khác với phiên bản trước có giao diện của C) và có nhiều điểm khác biệt so với phiên bản thứ nhất.

Thư viện OpenCV ban đầu được sự hỗ trợ từ Intel, sau đó được hỗ trợ bởi Willow Garage, một phòng thí nghiệm chuyên nghiên cứu về công nghệ robot. Cho đến nay, OpenCV vẫn là thư viện mở, được phát triển bởi nguồn quỹ không lợi nhuận (none-profit foundation) và được sự hưởng ứng rất lớn của cộng đồng

#### 1.2.3.2 Cấu trúc tổng quan.

Cấu trúc tổng quan của OpenCV bao gồm 5 phần chính. Hình vẽ dưới đây, mô tả 4 trong 5 thành phần đó.



Hình 1.17 Các thành phần của thư viện OpenCV.

Phần CV bao gồm các thư viện cơ bản về XLA và các giải thuật về thị giác máy tính. MLL là bộ thư viện về các thuật toán học máy, bao gồm rất nhiều bộ phân cụm và phân loại thống kê. HighGUI chứa đựng những thủ tục vào ra, các chức năng về lưu trữ cũng như đọc các file ảnh và video. Phần thứ 4, Cxcore chứa đựng các cấu trúc dữ liệu cơ bản (ví dụ như cấu trúc XML, các cây dữ liệu...). Phần cuối cùng là CvAux, phần này bao gồm cả thư viện cho việc phát hiện, theo dõi và nhận dạng đối tượng (khuôn mặt, mắt...).

## **CHƯƠNG 2: BÀI TOÁN PHÁT HIỆN NGƯỜI XÂM NHẬP.**

### **2.1 Tổng quan các phương pháp nhận diện đối tượng.**

#### **2.1.1 Đặc điểm các loại đối tượng và người.**

Các đặc điểm của đối tượng được trích chọn tùy theo mục đích nhận dạng trong quá trình xử lý ảnh. Có thể nêu ra một số đặc điểm của ảnh sau đây:

- Đặc điểm không gian: Phân bố mức xám, phân bố xác suất, biên độ, điểm uốn...
- Đặc điểm biến đổi: Các đặc điểm loại này được trích chọn bằng việc thực hiện lọc vùng (zonal filtering). Các bộ vùng được gọi là “mặt nạ đặc điểm” (feature mask) thường là các khe hẹp với hình dạng khác nhau (chữ nhật, tam giác, cung tròn...)
- Đặc điểm biên và đường biên: Đặc trưng cho đường biên của đối tượng và do vậy rất hữu ích trong việc trích chọn các thuộc tính bất biến được dùng khi nhận dạng đối tượng. Các đặc điểm này có thể được trích chọn nhờ toán tử gradient, toán tử Laplace, toán tử “chéo không” (zero crossing)...

Việc trích chọn hiệu quả các đặc điểm giúp cho việc nhận dạng các đối tượng ảnh chính xác, với tốc độ tính toán cao và dung lượng nhớ lưu trữ giảm xuống.

#### **2.1.2 Các phương pháp nhận diện đối tượng hiện nay.**

Có nhiều hướng tiếp cận và phương pháp khác nhau liên quan đến vấn đề nhận dạng. Theo Ming-Hsuan Yang, có thể phân loại thành bốn hướng tiếp cận chính:

- Hướng tiếp cận dựa trên cơ sở tri thức.
- Hướng tiếp cận dựa trên các đặc trưng bất biến.
- Hướng tiếp cận dựa trên đối sánh mẫu.
- Hướng tiếp cận dựa vào diện mạo xuất hiện phương pháp này thường dùng một mô hình máy học nên còn được gọi là phương pháp dựa trên cơ sở máy học.[7]

##### **2.1.2.1 Phương pháp dựa trên cơ sở tri thức.**

Mã hóa các hiểu biết của con người về đối tượng thành các luật. Thông thường các luật mô tả quan hệ của các đặc trưng.

Trong phương pháp này, các luật sẽ phụ thuộc rất lớn vào tri thức của những tác giả nghiên cứu. Đây là phương pháp dạng từ trên xuống. Dễ dàng xây dựng các luật cơ bản để mô tả các đặc trưng của đối tượng và các quan hệ tương ứng. Ví dụ, một khuôn mặt thường có hai mắt đối xứng nhau qua trục thẳng đứng ở giữa khuôn mặt và có một mũi, một miệng. Các quan hệ của các đặc trưng có thể được mô tả như quan hệ về khoảng cách và vị trí. Thông thường các tác giả sẽ trích đặc trưng của khuôn mặt trước tiên để có được các ứng viên, sau đó các ứng viên này sẽ được nhận dạng thông qua các luật để biết ứng viên nào là khuôn mặt (face) và ứng viên nào không phải khuôn mặt (none-face). Thường áp dụng quá trình xác định để giảm số lượng nhận dạng sai.

Một vấn đề khá phức tạp khi dùng hướng tiếp cận này là làm sao chuyển từ tri thức con người sang các luật một cách hiệu quả. Nếu các luật này quá chi tiết thì khi nhận dạng có thể nhận dạng thiếu các đối tượng có trong ảnh, vì những đối tượng này không thể thỏa mãn tất cả các luật đưa ra. Nhưng các luật tổng quát quá thì có thể chúng ta sẽ nhận dạng lầm một vùng nào đó không phải là đối tượng mà lại nhận dạng là đối tượng và cũng khó mở rộng yêu cầu từ bài toán để nhận dạng các đối tượng có nhiều tư thế khác nhau.

#### **2.1.2.2 Phương pháp dựa trên đặc trưng bất biến.**

Mục tiêu các thuật toán đi tìm các đặc trưng mô tả cấu trúc đối tượng, các đặc trưng này sẽ không thay đổi khi vị trí đối tượng, vị trí đặt thiết bị thu hình hoặc điều kiện ánh sáng thay đổi. Đây là hướng tiếp cận theo kiểu dưới lên. Các tác giả cố gắng tìm các đặc trưng không thay đổi của đối tượng để nhận dạng đối tượng. Dựa trên nhận xét thực tế, con người dễ dàng nhận biết các đối tượng trong tư thế khác nhau và điều kiện ánh sáng khác nhau, thì phải tồn tại các thuộc tính hay đặc trưng không thay đổi. Có nhiều nghiên cứu đầu tiên nhận dạng các đặc trưng đối tượng rồi chỉ ra có đối tượng trong ảnh hay không. Ví dụ: Các đặc trưng như: lông mày, mắt, mũi, miệng và đường viền của tóc được trích bằng phương pháp xác định cạnh. Trên cơ sở các đặc trưng này, thực hiện việc xây dựng một mô hình thống kê để mô tả quan hệ của các đặc trưng này và nhận dạng sự tồn tại của khuôn mặt trong ảnh.

Một vấn đề của các thuật toán theo hướng tiếp cận đặc trưng cần phải điều chỉnh cho phù hợp điều kiện ánh sáng, nhiễu và bị che khuất. Đôi khi bóng của đối tượng sẽ tạo thêm cạnh mới, mà cạnh này lại rõ hơn cạnh thật sự của nó, vì thế nếu dùng cạnh để nhận dạng sẽ gặp khó khăn.

### **2.1.2.3 Phương pháp dựa trên so khớp mẫu.**

Trong so khớp mẫu, các mẫu chuẩn của đối tượng sẽ được nhận dạng trước hoặc nhận dạng các tham số thông qua một hàm. Từ một ảnh đưa vào, tính các giá trị tương quan so với các mẫu chuẩn. Thông qua các giá trị tương quan này mà các tác giả quyết định có hay không có tồn tại đối tượng trong ảnh. Hướng tiếp cận này có lợi thế là rất dễ cài đặt, nhưng không hiệu quả khi tỷ lệ, tư thế và hình dáng thay đổi. Nhiều độ phân giải, đa tỷ lệ, các mẫu con và các mẫu biến dạng được xem xét thành bất biến về tỷ lệ và hình dáng.

### **2.1.2.4 Phương pháp dựa trên diện mạo.**

Trái ngược với các phương pháp so khớp mẫu với các mẫu đã được định nghĩa trước bởi những chuyên gia, các mẫu trong hướng tiếp cận này được học từ các ảnh mẫu. Một cách tổng quát, các phương pháp tiếp cận theo hướng tiếp cận này áp dụng các kỹ thuật theo hướng xác suất thống kê và máy học để tìm những đặc tính liên quan của đối tượng và không phải là đối tượng. Các đặc tính đã được học ở trong hình thái các mô hình phân bố hay các hàm biệt số nên dùng có thể dùng các đặc tính này để nhận dạng đối tượng. Đồng thời, bài toán giảm số chiều thường được quan tâm để tăng hiệu quả tính toán cũng như hiệu quả nhận dạng.

Các tiếp cận khác trong hướng tiếp cận dựa trên diện mạo là tìm một hàm biệt số (mặt phẳng quyết định, siêu phẳng để tách dữ liệu, hàm ngưỡng) để phân biệt hai lớp dữ liệu: đối tượng và không phải là đối tượng. Bình thường, các mẫu ảnh được chiếu vào không gian có số chiều thấp hơn, rồi sau đó dùng một hàm biệt số (dựa trên các độ đo khoảng cách) để phát hiện, hoặc xây dựng mặt quyết định phi tuyến bằng mạng neuron tích chập đa tầng. Hoặc dùng SVM (Support Vector Machine) và các phương thức kernel, chiếu hoàn toàn các mẫu vào không gian có số chiều cao hơn để dữ liệu bị rời rạc

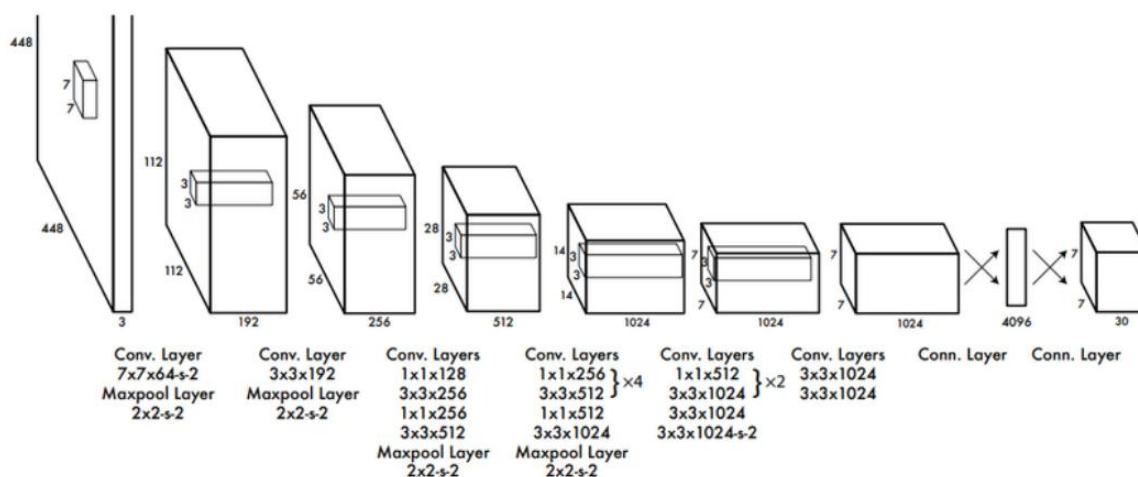
hoàn toàn và có thể dùng một mặt phẳng quyết định phát hiện các mẫu đối tượng và không phải là đối tượng. Có nhiều mô hình máy học được áp dụng trong hướng tiếp cận này.

### **2.1.3 Các kỹ thuật phát hiện người.**

Thời gian qua trên thế giới có hàng loạt các công trình nghiên cứu nhằm giải quyết bài toán phát hiện người trong ảnh. Kỹ thuật xử lý theo nhiều hướng khác nhau, chủ yếu dựa trên cách thức trích chọn đặc trưng và nhận dạng đối tượng. Cơ bản có các hướng tiếp cận chính như sau:

1. Dựa trên các đặc trưng biến đổi Wavelet, Haar - Like và phân loại đa cấp: Wavelet là phép biến đổi được sử dụng để chuẩn hóa các vùng liên thông. Sử dụng phương pháp trích chọn đặc trưng Wavelet Haar để chọn tập đặc trưng cho ảnh đầu vào. Các đặc trưng trích chọn được chứng minh là bất biến.
2. Dựa trên đặc trưng Histogram có hướng (HOG - Histogram of Oriented gradient): HOG là một phân bố biểu đồ mức xám được sử dụng để trích chọn đặc trưng của ảnh. HOG tỏ ra khá hiệu quả trong các bài toán phát hiện người trong ảnh, HOG được đề xuất bởi Bill Triggs và Navel Dalai vào năm 2005 tại Viện Nghiên cứu INRIA. HOG có ưu điểm là có thể tính toán nhanh, đặc trưng này giúp cho hệ thống hoạt động hiệu quả ở môi trường điều kiện chiếu sáng khác nhau vì HOG tương đối độc lập với điều kiện chiếu sáng.
3. Hướng tiếp cận phát hiện từng phần rồi tổ hợp lại, trong đó cho phép tiến hành đồng thời các công đoạn (kỹ thuật Top - Down): người trong ảnh được mô hình hóa thành từng bộ phận. Phát hiện từng phần của đối tượng người (ví dụ: đầu, thân trên, thân dưới,...) sau đó tổng hợp kết quả, kết luận có phải là người hay không.
4. Hướng tiếp cận phát hiện toàn bộ đối tượng (Full body detection) dựa trên các đặc trưng tổng thể của đối tượng để tìm kiếm: phát hiện người trong các cửa sổ tìm kiếm địa phương nếu thỏa mãn các tiêu chí nhất định. Hạn chế của phương pháp này là hiệu suất dễ bị ảnh hưởng bởi nền lộn xộn và sự che lấp.

5. Nhận dạng đối tượng sử dụng YOLOv4 (phiên bản thứ 4 của mạng YOLO): YOLO (You Only Look Once) là một mô hình mạng neuron tích chập tích chập cho việc phát hiện, nhận dạng, phân loại đối tượng. YOLO được tạo ra từ việc kết hợp giữa các lớp tích chập và các lớp kết nối. Trong đó các lớp tích chập sẽ trích xuất ra các đặc trưng của ảnh, còn các lớp kết nối đầy đủ sẽ dự đoán ra xác suất đó và tọa độ của đối tượng.



Hình 2.1 Mô hình YOLO.

### Đánh giá hiệu quả các kỹ thuật áp dụng:

Các hướng nghiên cứu đưa ra cơ bản giải quyết bài toán tìm người trong ảnh tuy nhiên tùy vào từng trường hợp vẫn còn những hạn chế như: đối tượng xuất hiện với các đặc trưng màu sắc, hình dạng, góc độ khác nhau; đối tượng xuất hiện với số lượng lớn các động tác khác nhau; sự thay đổi về quần áo; nhiều nền phức tạp; điều kiện chiếu sáng thay đổi; sự che lấp, tỷ lệ khác nhau;...

Thuật toán HOG chỉ phát hiện được người theo phương diện thẳng mặt có đầy đủ đầu, thân, tay, chân mô phỏng đủ các bộ phận và dáng đi, đứng của người. Khó phát hiện người không đầy đủ các yếu tố hoặc đứng nghiêng.

YOLOv4-tiny cực kỳ nhanh chóng và chính xác. Trong MAP đo được ở 0,5 IOU YOLOv4-tiny ngang bằng với Focal Loss nhưng nhanh hơn khoảng 4 lần. Hơn nữa, bạn có thể dễ dàng đánh đổi giữa tốc độ và độ chính xác chỉ bằng cách thay đổi kích thước của mô hình, không cần đào tạo lại. Tốc độ 30 FPS (Frame per second), có độ chính xác cao nhất trên tập COCO Dataset (COCO Dataset: là tập dữ liệu nhận dạng hình ảnh, phân

đoạn và phụ đề mới. COCO có một số tính năng: Phân loại đối tượng; Nhận biết trong ngữ cảnh; Nhiều đối tượng trên mỗi hình ảnh; Hơn 300.000 hình ảnh; Hơn 2 triệu phiên bản; 80 loại đối tượng; 5 chú thích cho mỗi hình ảnh; Các điểm chính trên 100.000 người).

## **2.2 Thuật toán phát hiện người bằng mạng neuron tích chập, YOLO.**

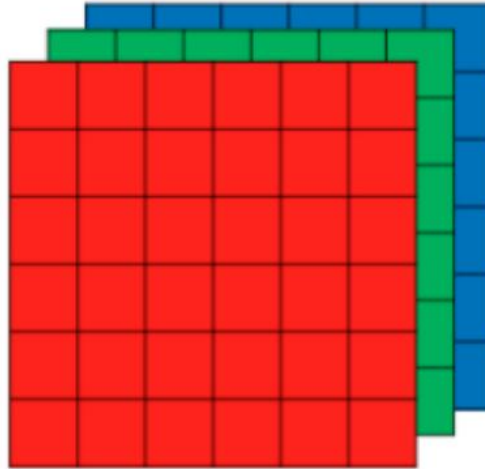
### **2.2.1 Kỹ thuật nhận diện đối tượng bằng mạng neuron tích chập.**

Convolutional Neural Network (CNN – Mạng neuron tích chập tích chập) là một trong những mô hình Deep Learning tiên tiến giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao hiện nay.

Sự ra đời của mạng neuron tích chập tích chập là dựa trên ý tưởng cải tiến cách thức các mạng neuron tích chập nhân tạo truyền thống học thông tin trong ảnh. Do sử dụng các liên kết đầy đủ giữa các điểm ảnh vào node, các mạng neuron tích chập nhân tạo truyền thẳng (Feedforward Neural Network) bị hạn chế rất nhiều bởi kích thước của ảnh, ảnh càng lớn thì số lượng liên kết càng tăng nhanh, kéo theo sự bùng nổ khối lượng tính toán. Ngoài ra, sự liên kết đầy đủ này cũng là sự dư thừa với mỗi bức ảnh, các thông tin chủ yếu thể hiện qua sự phụ thuộc giữa các điểm ảnh với những điểm xung quanh nó mà không quan tâm nhiều đến các điểm ảnh ở cách xa nhau. Mạng neuron tích chập tích chập với kiến trúc thay đổi, có khả năng xây dựng liên kết chỉ sử dụng một phần cục bộ trong ảnh kết nối đến node trong lớp tiếp theo thay vì toàn bộ ảnh như trong mạng neuron tích chập truyền thẳng.

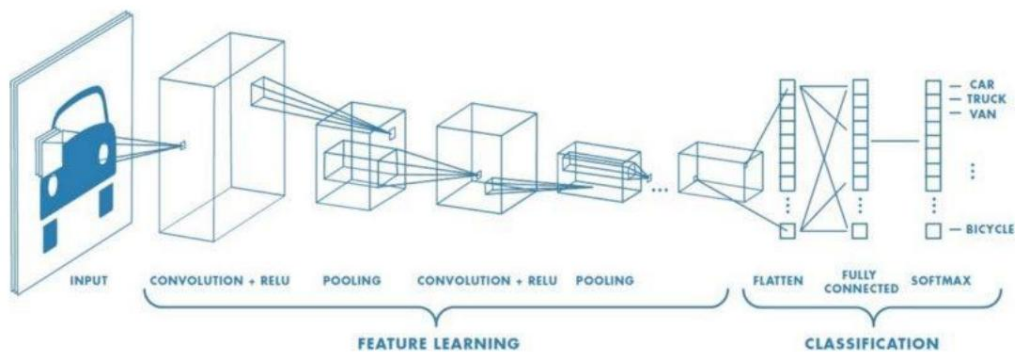
Mạng neuron tích chập tích chập là một phương thức rất hay được sử dụng để nhận dạng hình ảnh, phân loại ảnh, nhận diện đối tượng, nhận diện khuôn mặt,... Mạng neuron tích chập tích chập thực hiện phân loại ảnh bằng các bước nhận ảnh đầu vào, xử lý và phân loại nó dưới dạng các nhãn. Máy tính nhìn nhận dữ liệu đầu vào như một mảng các điểm ảnh (pixel) dựa trên độ phân giải của ảnh. Dựa vào nó máy tính nhìn nhận ảnh dưới dạng  $h \times w \times d$  (h: height, w: width, d: dimension). 1 ảnh  $6 \times 6 \times 3$  nghĩa là ảnh có kích thước  $6 \times 6$  và có 3 kênh màu (RGB) còn ảnh  $4 \times 4 \times 1$  là ảnh có kích thước  $4 \times 4$  và có một kênh màu xám (grayscale).





Hình 2.2 Bảng ma trận RGB.

Để mạng neuron tích chập tích chập thực hiện huấn luyện (train) và kiểm tra (test), mỗi ảnh đầu vào sẽ thông qua một số lớp tích chập với bộ lọc (kernel), Pooling, lớp kết nối đầy đủ (fully connected layers) và thực hiện hàm softmax để phân loại 1 đối tượng. Hình 19 thể hiện đầy đủ quá trình từ nhận dữ liệu cho đến phân loại đối tượng.[9]



Hình 2.3 Mạng neuron tích chập với nhiều lớp tích chập.

### 2.2.2 Các bước thực hiện kỹ thuật nhận diện đối tượng bằng mạng neuron .

- Đưa ảnh đầu vào vào trong lớp tích chập.
- Chọn các tham số, áp các bộ lọc với stride, padding nếu cần thiết. Thực hiện tích chập trên ảnh và thực hiện hàm ReLU.
- Thực hiện pooling để giảm kích cỡ ảnh.
- Thêm các lớp tích chập nữa cho đến khi đạt được kết quả phù hợp.
- Dàn phẳng kết quả và đưa vào lớp kết nối đầy đủ.

- Thực hiện các activation function và phân loại ảnh.[19]

### **2.2.3 Nhận dạng đối tượng bằng YOLO v4-tiny.**

Trong YOLO version 4 tác giả áp dụng một mạng feature extractor là darknet-53. Mạng này gồm 53 convolutional layers kết nối liên tiếp, mỗi layer được theo sau bởi một batch normalization và một activation Leaky Relu. Để giảm kích thước của output sau mỗi convolution layer, tác giả YOLOv4 đã thực hiện down sample bằng các filter với kích thước là 2. Điều này có tác dụng giảm thiểu số lượng tham số cho mô hình.

Model	Train	Test	mAP	FLOPS	FPS	Cfg	Weights
SSD300	COCO trainval	test-dev	41.2	-	46	link	
SSD500	COCO trainval	test-dev	46.5	-	19	link	
YOLOv2 608x608	COCO trainval	test-dev	48.1	62.94 Bn	40	cfg	weights
Tiny YOLO	COCO trainval	test-dev	23.7	5.41 Bn	244	cfg	weights
SSD321	COCO trainval	test-dev	45.4	-	16	link	
DSSD321	COCO trainval	test-dev	46.1	-	12	link	
R-FCN	COCO trainval	test-dev	51.9	-	12	link	
SSD513	COCO trainval	test-dev	50.4	-	8	link	
DSSD513	COCO trainval	test-dev	53.3	-	6	link	
FPN FRCN	COCO trainval	test-dev	59.1	-	6	link	
Retinanet-50-500	COCO trainval	test-dev	50.9	-	14	link	
Retinanet-101-500	COCO trainval	test-dev	53.1	-	11	link	
Retinanet-101-800	COCO trainval	test-dev	57.5	-	5	link	
YOLOv3-320	COCO trainval	test-dev	51.5	38.97 Bn	45	cfg	weights
YOLOv3-416	COCO trainval	test-dev	55.3	65.86 Bn	35	cfg	weights
YOLOv3-608	COCO trainval	test-dev	57.9	140.69 Bn	20	cfg	weights
YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220	cfg	weights
YOLOv3-spp	COCO trainval	test-dev	60.6	141.45 Bn	20	cfg	weights

Hình 2.4 Mạng draknet-53.

Kiến trúc của mô hình YOLOv4 đã có nhiều cải tiến so với các phiên bản trước nhằm cải thiện độ chính xác dự đoán giúp cân bằng được tốc độ và độ chính xác. Trong phiên bản YOLOv4 được thực hiện với nhiều tỉ lệ khác nhau giúp mô hình có thể phát hiện được các đối tượng có nhiều kích thước khác nhau.

Tại mỗi tỉ lệ mô hình sẽ cùng dự đoán các thông tin giống nhau sau đó sử dụng thuật toán Non-max suppression để xác kết quả cuối cùng.

### Tối ưu YOLOv4-Tiny

**1. Độ chính xác:** YOLOv4 thường có độ chính xác cao hơn so với YOLOv4-tiny. Với việc sử dụng một mạng neuron tích chập tích chập (CNN) sâu hơn và số lượng lớp phân

loại lớn hơn, YOLOv4 có khả năng nhận dạng chính xác các đối tượng với độ tin cậy cao hơn.

**2. Tốc độ xử lý:** YOLOv4-tiny có tốc độ xử lý nhanh hơn so với YOLOv4. YOLOv4-tiny sử dụng một mạng neuron tích chập tích chập nhẹ hơn và ít lớp hơn, dẫn đến việc giảm thiểu thời gian tính toán và tăng tốc độ xử lý.

**3. Yêu cầu tài nguyên:** YOLOv4-tiny yêu cầu ít tài nguyên hơn so với YOLOv4. Với mạng neuron tích chập nhẹ hơn, YOLOv4-tiny có thể chạy trên các thiết bị có tài nguyên hạn chế, như các hệ thống nhúng và các thiết bị di động.

**4. Ứng dụng thực tế:** YOLOv4-tiny thích hợp cho các ứng dụng có yêu cầu về tốc độ xử lý cao, chẳng hạn như theo dõi đối tượng trên camera an ninh, hệ thống giám sát giao thông hoặc ứng dụng thời gian thực trên các thiết bị nhúng. Trong khi đó, YOLOv4 thích hợp cho các ứng dụng yêu cầu độ chính xác cao hơn, chẳng hạn như phát hiện đối tượng trong ảnh y tế hoặc phân tích video cao cấp.

Tùy thuộc vào yêu cầu và hạn chế của ứng dụng cụ thể, bạn có thể lựa chọn giữa YOLOv4 và YOLOv4-tiny để đạt được sự cân bằng giữa độ chính xác, tốc độ xử lý và yêu cầu tài nguyên.

## **2.3 Chatbot Telegram.**

### **2.3.1 Tổng quan Chatbot.**

**Bot Telegram** là một tính năng giống như robot được cài đặt trên ứng dụng nhắn tin Telegram giúp người dùng tạo lập và quản lý một cách dễ dàng.

Người tạo ra Bot Telegram có thể thêm các chức năng cho Bot như: cập nhật tin tức, tạo GIF, trả lời tin nhắn,...

Hiện tại, việc tạo Bot trên Telegram là cực kỳ dễ dàng và bất cứ ai cũng có thể làm được cả. Việc khó nhất trong cách tạo lập Bot Telegram chính là thêm các chức năng giúp Bot gửi thông báo cho bạn.

### **2.3.2 Lợi ích của chatbot telegram.**

Việc lựa chọn việc sử dụng chatbot trên Telegram có thể phụ thuộc vào nhiều yếu tố khác nhau và tùy thuộc vào mục đích cụ thể của người sử dụng hoặc doanh nghiệp. Dưới đây là một số lý do mà người ta có thể lựa chọn sử dụng chatbot trên Telegram:

1. Tính năng tin nhắn: Telegram là một ứng dụng nhắn tin được thiết kế để gửi và nhận tin nhắn nhanh chóng. Điều này làm cho việc tích hợp chatbot vào Telegram trở nên thuận tiện và tạo ra một trải nghiệm tương tác tức thì cho người dùng. Và đặc biệt nó hoàn toàn miễn phí và không mất phí.

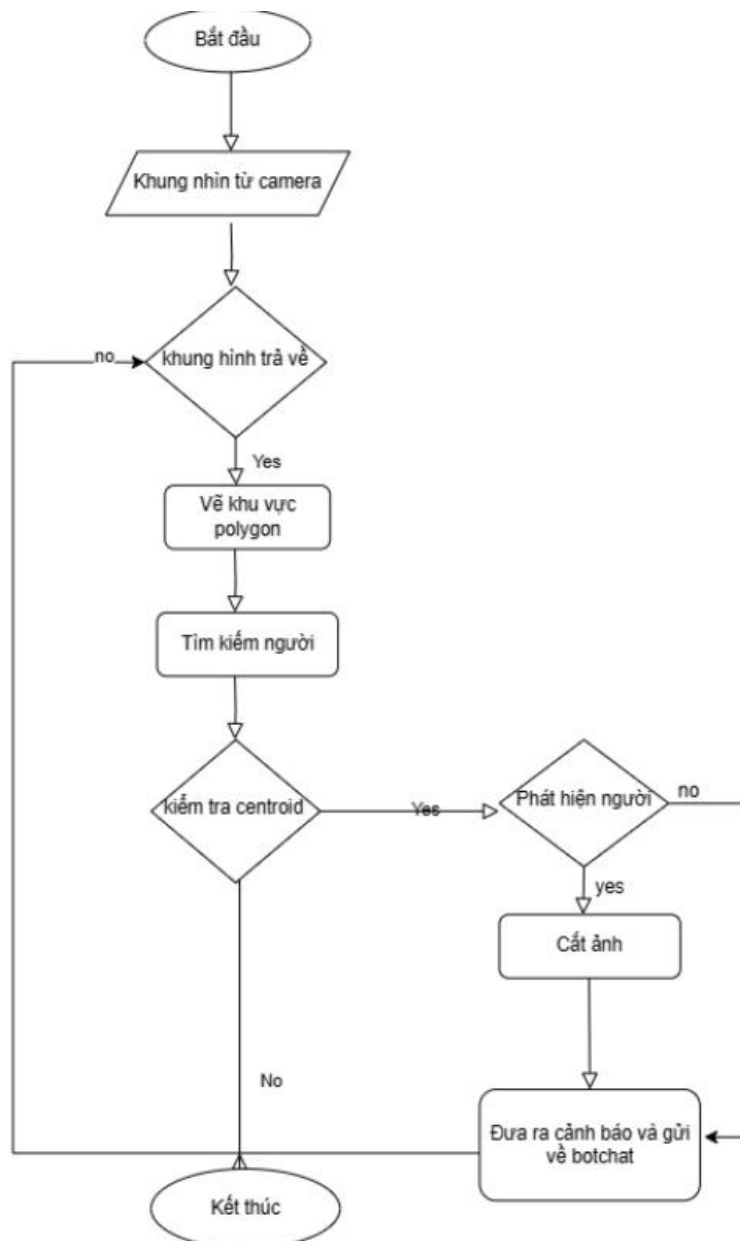
2. Quyền riêng tư và bảo mật: Telegram được biết đến với việc cung cấp mức độ bảo mật cao hơn so với nhiều nền tảng mạng xã hội khác. Nó sử dụng mã hóa mạnh mẽ cho tin nhắn và cho phép người dùng tự hủy tin nhắn trong khoảng thời gian nhất định. Điều này làm cho Telegram trở thành lựa chọn hữu ích khi thông tin nhạy cảm hoặc dữ liệu cá nhân được truyền qua chatbot.

3. Khả năng tích hợp: Telegram cung cấp các API mạnh mẽ cho phép các nhà phát triển xây dựng và tích hợp chatbot vào nền tảng này. Điều này cho phép doanh nghiệp tùy chỉnh và mở rộng tính năng của chatbot theo nhu cầu cụ thể của họ.

4. Độ phổ biến: Telegram đã trở thành một trong những ứng dụng nhắn tin phổ biến trên thế giới, với hàng triệu người dùng. Việc sử dụng chatbot trên một nền tảng phổ biến như Telegram có thể giúp doanh nghiệp tiếp cận được một lượng lớn khách hàng tiềm năng.

## CHƯƠNG 3: TRIỂN KHAI ỨNG DỤNG.

### 3.1 Mô hình hệ thống dự kiến.



Hình 3.1 Mô hình hệ thống dự kiến.

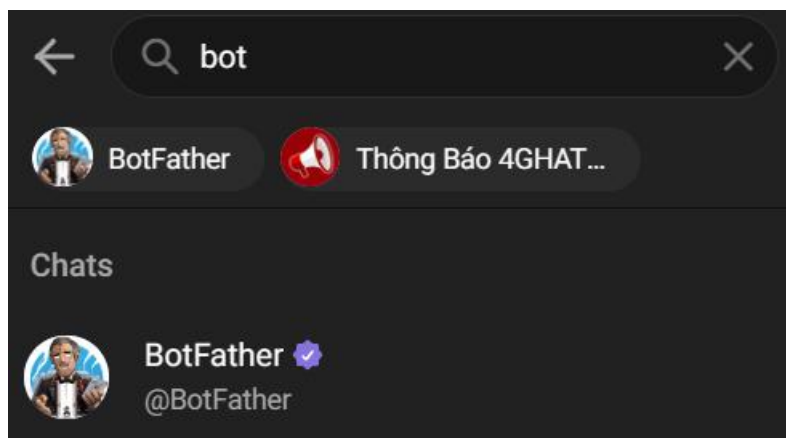
Trong mô hình hệ thống đề xuất: Sau khi nhận dữ liệu từ camera, máy tính sẽ xử lý từng khung ảnh để phát hiện đối tượng là người sử dụng thuật toán YOLOv4-tiny, trong trường hợp phát hiện ra người trong khung hình. Nếu đối tượng người được phát hiện nằm trong đa giác điểm centroid có thuộc khu vực polygon đa giác bảo vệ hay không nếu

có thì hệ thống sẽ thực hiện cảnh báo qua tin nhắn thông qua botchat telegram,...cho người có trách nhiệm xử lý (lãnh đạo, bảo vệ, người trực ca, nhân viên,...) đồng thời lưu lại danh sách đối tượng được trích xuất.

### 3.2 Xây dựng chatbot trên telegram.

**B1:** Tiến hành tạo tài khoản telegram.

**B2:** Tìm kiếm và tạo Botfather trên telegram (chọn bot có tích xanh).

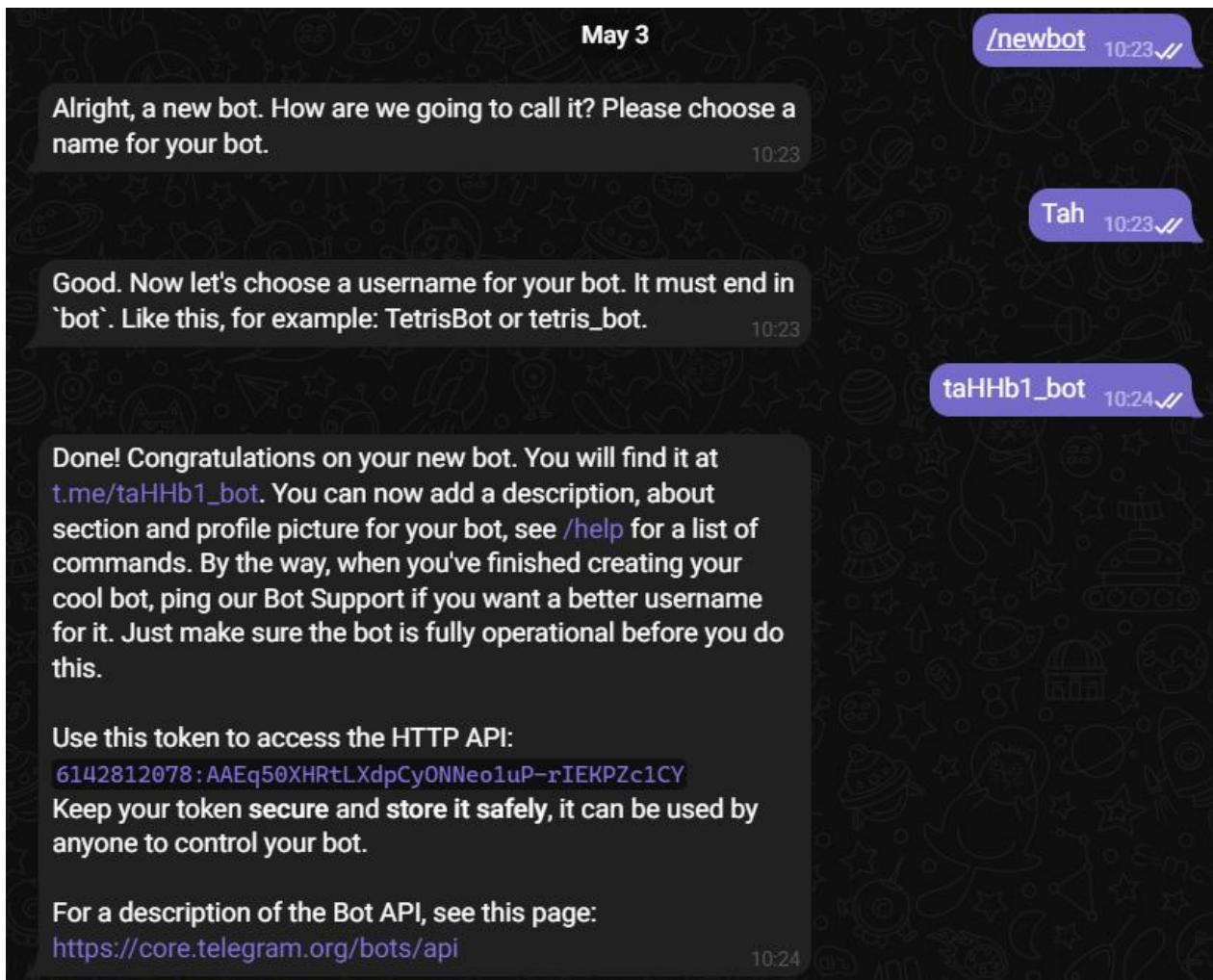


Hình 3.2 Tên botfather.

Để tạo bot người dùng sẽ nhập lần lượt các dòng lệnh:

/start → tên bot → username





Hình 3.3 Tạo chatbot.

**B3:** Tiến hành lấy id của chatbot:

- Bằng cách truy cập vào đường link: [https://api.telegram.org/bot\[TOKEN\]/getUpdates](https://api.telegram.org/bot[TOKEN]/getUpdates) và thay thế mã token mà botfather đã cung cấp ở B2.
- Sau đó ta vào lại chatbot mà ta đã vừa tạo gửi một tin nhắn bất kì để lấy được id



```
{
  "ok": true,
  "result": [
    {
      "update_id": 411339251,
      "message": {
        "message_id": 71,
        "from": {
          "id": 6253730735,
          "is_bot": false,
          "first_name": "DagT",
          "last_name": "TaH",
          "language_code": "en"
        },
        "chat": {
          "id": 6253730735,
          "first_name": "DagT",
          "last_name": "TaH",
          "type": "private"
        },
        "date": 1687543903,
        "text": "hii"
      }
    }
  ]
}
```

Hình 3.4 Lấy id.

**B4:** Cài đặt thư viện: python-telegram-bot.

**B5:** Viết chương trình gửi ảnh và dòng thông báo.

```
def send_telegram(photo_path="alert.png"):
    try:
        my_token = "6142812078:AAEq50XHRtLXdpCyONNeo1uP-rIEKPZc1CY"
        bot = telegram.Bot(token=my_token)
        bot.sendPhoto(chat_id="6253730735", photo=open(photo_path, "rb"),
caption="Có xâm nhập, nguy hiểm!")
    except Exception as ex:
        print("Can not send message telegram ", ex)

    print("Send sucess")
```

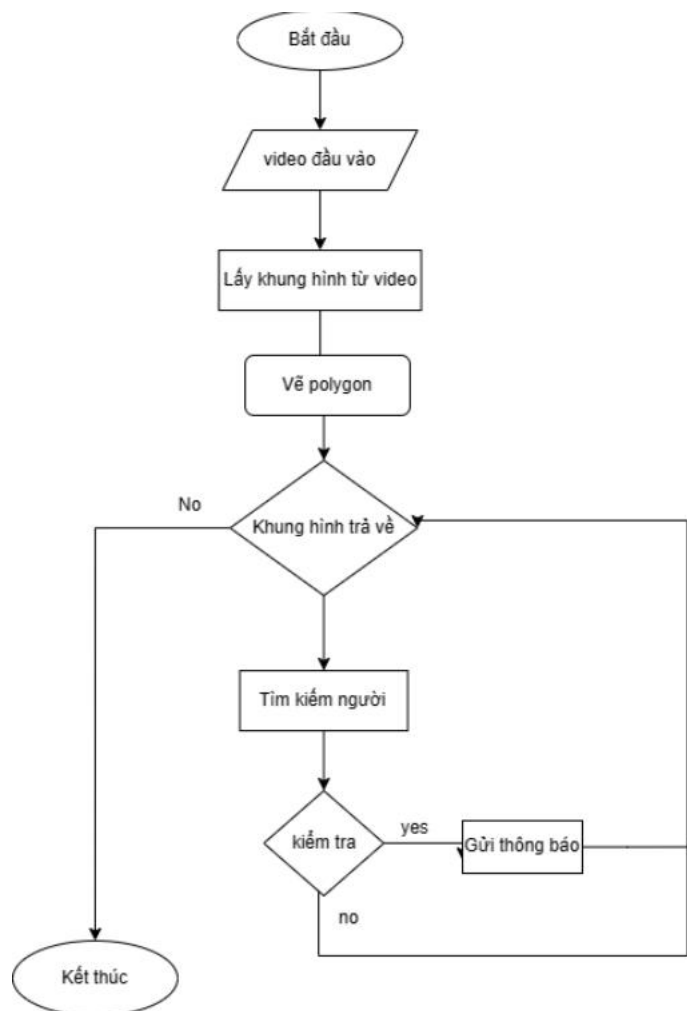
Hình 3.5 Viết chương trình.

### 3.3 Thuật toán tìm người.

Để tìm người bằng YOLOv4-tiny trong đồ án này em sử dụng ngôn ngữ lập trình Python trên Pycharm, đồng thời sử dụng các bộ thư viện mã nguồn mở như: OpenCV, Numpy, Shapely, imutils, python-telegram-bot.

Để sử dụng thuật toán YOLOv4-tiny, trong đồ án sử dụng một số tệp được tải xuống bao gồm: Tệp các trọng số đã được đào tạo trước của YOLOv4-tiny: yolov4-tiny.weights, tệp cấu hình: yolov4-tiny.cfg, tệp class tên: classnames.txt.

#### 3.3.1 Mô hình thuật toán.



Hình 3.6 Sơ đồ tìm kiếm người trên video.

### 3.3.2 Chú thích bằng lời.

Thuật toán tìm người trên video được thực hiện qua các bước sau

- + B1: Nhận video đầu vào.
- + B2: Lấy khung hình từ video.
- + B3: Vẽ đa giác bảo vệ.
- + B4: Kiểm tra nếu còn khung hình thì sang B5 ngược lại thì kết thúc.
- + B5: Tìm kiếm người sử dụng thuật toán YOLO v4-tiny.
- + B6: Kiểm tra nếu centroid thuộc polygon thì gửi thông báo nếu không thì chuyển sang bước B4.

### 3.3.3 Kết quả thực nghiệm.

Như đã nêu tại hình 3.6, trong trong đồ án này chỉ dừng lại ở mức tìm người trong video bằng YOLOv4-tiny khi tìm thấy người. Do cấu hình máy tính cá nhân thực hiện nên đề tài có nhiều hạn chế nên mô hình chạy được trong thời gian thực tuy nhiên cấu hình máy yếu nên sẽ xảy ra hiện tượng giật lag nhẹ khi đối tượng di chuyển. Dưới đây là quá trình xử lý video đã được trích xuất từ camera của laptop và đã tiến hành cho người dùng vẽ các điểm để chuẩn bị việc tạo ra khu vực bảo vệ bằng cách dùng các nút trên bàn phím:

- Khi người dùng nhấp chuột trái trên màn hình để vẽ các điểm, khi muốn lưu lại kết quả thì ấn phím S, khi muốn xóa để vẽ lại thì ấn phím C và sau khi tinh chỉnh vừa ý đúng yêu cầu thì tiến hành ấn phím Q để thoát và chuyển quá trình giám sát.

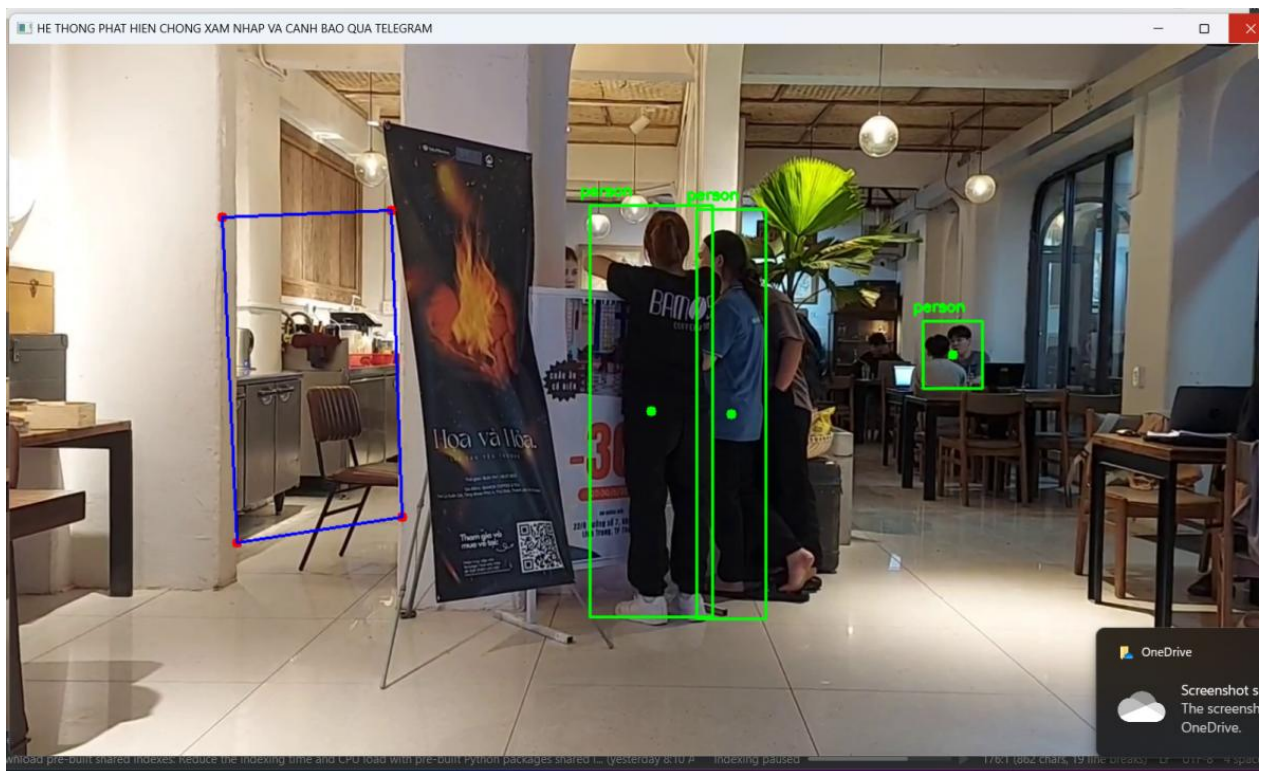
- Thuận tiện cho người dùng chỉ cần vẽ 1 lần không cần phải vẽ lại nhiều lần khi khởi động hệ thống.



Hình 3.7 Vẽ và lưu các điểm.

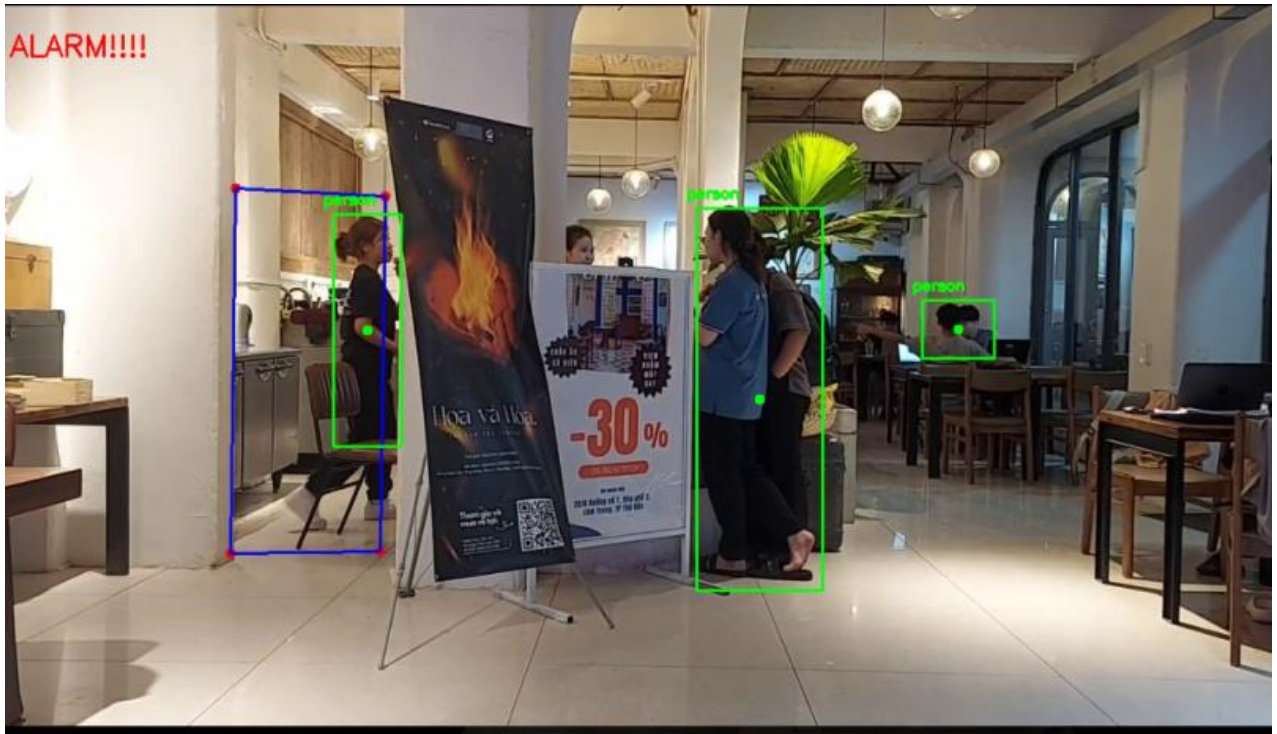
Như trong màn hình hình 3.7 ta thấy được ta đã tiến hành vẽ được các điểm để chuẩn bị cho quá trình giám sát hình 3.8. Nhìn hình 3.8 ta thấy được hệ thống đã lấy được các điểm point mà ta đã tạo ở hình 3.7 và khi ta ấn phím D thì hệ thống tự động nối liền điểm đầu và điểm cuối lại với nhau để tạo ra một đa giác polygon khu vực bảo vệ chống xâm nhập như theo yêu cầu bài.

Chương trình cũng đã nhận diện được các đối tượng là con người theo đúng yêu cầu của chúng ta và tính toán điểm centroid cho mỗi đối tượng.



Hình 3.8 Phát hiện đối tượng.

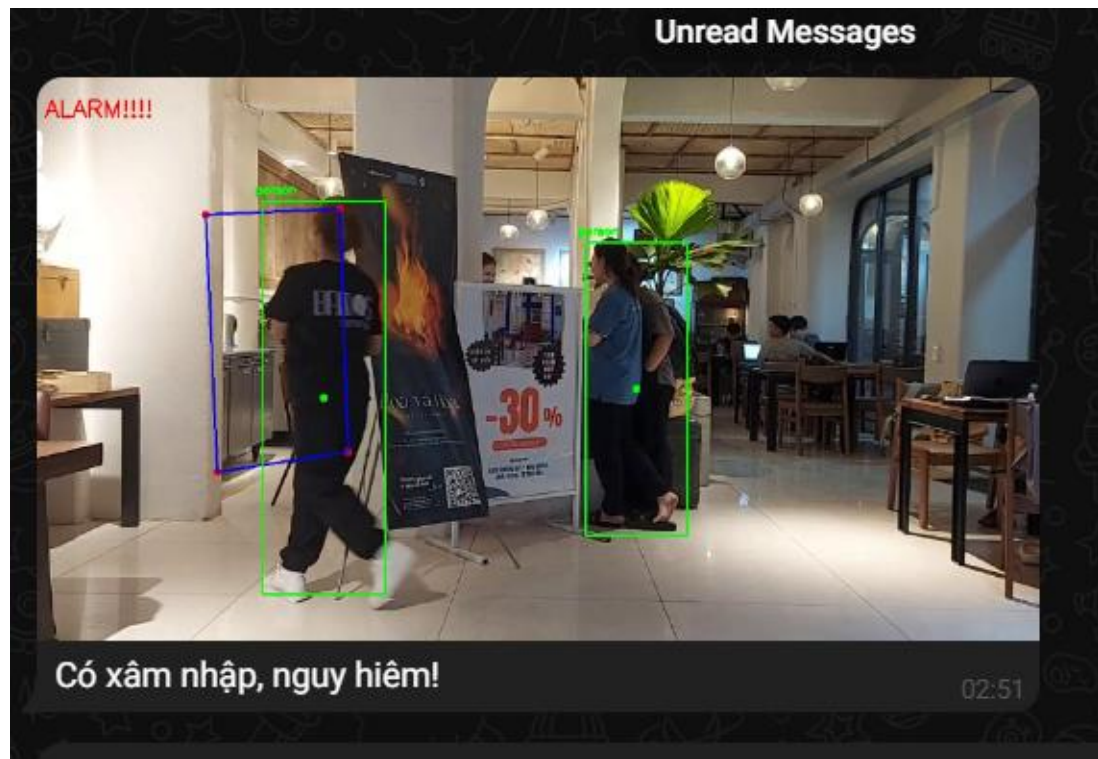
Sau khi đã nhận diện được đối tượng thì hệ thống đã đi tiến hành kiểm tra đối tượng có xâm nhập vào vùng cần bảo vệ hay không nếu có thì sẽ tiến hành hiện dòng chữ ALARM!!! lên trên màn hình như hình 3.9.



Hình 3.9 Phát cảnh báo.

Lúc này sau khi nhận được thông báo thì hệ thống sẽ tiến hành gửi một tin nhắn đến botchat trên telegram và đính kèm một dòng thông báo “có xâm nhập, nguy hiểm!” như hình 3.10 để người có trách nhiệm sẽ phụ trách việc xử lý sự việc.





Hình 3.10 Gửi cảnh báo qua chatbot telegram.

Sau khi nhận được thông báo thì hệ thống sẽ tiến hành tiếp tục gửi lại thông báo qua chatbot sau mỗi 5s tránh việc spam liên tục.

## KẾT LUẬN

### 1. Kết quả đạt được.

Sử dụng một số video trích xuất từ camera sau khi thực hiện tìm người bằng thuật toán YOLOv4-tiny cho kết quả chính như sau:

- + Nhận được video đầu vào.
- + Vẽ được lớp bảo vệ và lưu trữ ra 1 file.txt riêng biệt.
- + Phát hiện được đối tượng là con người.
- + Xây dựng được khu vực cần giám sát.
- + Theo dõi và cảnh báo khi có hành vi xâm nhập.
- + Gửi được tin nhắn cnahr báo là hình ảnh cho việc xâm nhập qua telegram để người có chức vụ xử lý.
- + Chương trình hệ thống chạy nhanh, tốc độ xử lý cao nhưng bù lại là độ chính xác không cao cho các đối tượng nhỏ kém đó cũng là một nhược điểm của thuật toán này.

### 2. Hạn chế.

- Trong khuôn khổ đề tài này em đã cố gắng phát huy hết năng lực bản thân, tìm hiểu, nghiên cứu tài liệu, thu thập dữ liệu với sự hướng dẫn của Giảng viên TS. Trần Phong Nhã, thầy cô giáo thuộc Khoa Công nghệ thông tin, trường đại học Giao Thông Vận Tải phân hiệu tại TPHCM cùng sự giúp đỡ của các bạn học viên lớp công nghệ thông tin k60 để hoàn thành cơ bản các nội dung đề ra tại đề tài này, đảm bảo chất lượng và đúng thời gian quy định.

- Tuy nhiên do một số lý do khách quan, chủ quan nên đề án này chỉ dừng lại ở mức tìm người trong video bằng YOLOv4-tiny khi tìm thấy người xâm nhập thì mới tiến hành phát cảnh báo. Ngoài ra còn một số hạn chế, cụ thể:

- + Chỉ thao tác trên camera LAPTOP hoặc video, chất lượng một số video không tốt nên khả năng tìm người trong một số trường hợp cho kết quả với độ chính xác cho ra kết quả chưa cao.
- + Do trình độ năng lực của bản thân còn nhiều hạn chế, điều kiện máy móc trang thiết bị phục vụ nghiên cứu và thử nghiệm chưa đảm bảo với thời gian nghiên cứu chưa nhiều



nên em chưa tìm hiểu, khám phá và phát huy hết tính năng ưu việt của thuật toán YOLOv4-tiny hay cũng như YOLO v4.

### **3. Hướng phát triển.**

Hướng phát triển của đề tài: Để áp dụng được kết quả đề tài này vào thực tế cần gọt giũa một số mã nguồn chương trình đảm bảo tối ưu, bổ sung các chức năng mới như:

- + Hệ thống phát hiện được đối tượng nào đang nằm trong khu vực bảo vệ và có tính năng tự động bám theo khu vực đối tượng cần bảo vệ ứng dụng cho việc giám sát và trông trẻ em, theo dõi thú cưng khi các nhân vật di chuyển theo thời gian hoặc muốn tạo thêm nhiều khu vực bảo vệ trên cùng một màn hình.

- + Ngoài ra thì nó cũng phải nhận diện được đối tượng nằm trong khu vực bảo vệ và cảnh báo khi vật thể mà cần bảo vệ biến mất thì phải gửi gửi cảnh báo và phát ra âm thanh cảnh báo.

- + Có thể nâng cấp thuật toán lên YOLOv4, YOLOv5 hoặc YOLOv7 thậm chí YOLOv8 để tăng tốc độ nhận dạng và cho kết quả tốt hơn.

- + Đầu tư camera thực tế, camera công nghệ mới hiện nay cho hình ảnh chất lượng cao, rõ nét trong điều kiện thời tiết xấu hoặc ban đêm, trang bị máy móc để có thể train với dữ liệu lớn, đặc biệt là trang bị máy chủ cấu hình cao có card GPU mạnh để có thể chạy trên các công nghệ mới.

Ngoài ra cần hoàn thiện quy trình xử lý để xuất dữ liệu ra các thiết bị ngoại vi phục vụ việc cảnh báo tự động đến người có trách nhiệm xử lý theo các kịch bản cảnh báo được thiết lập các ngưỡng. Xây dựng quy chế vận hành hệ thống đảm bảo phát huy tối đa hiệu quả khi đưa vào áp dụng thực tế tại các khu vực cần bảo vệ như viện bảo tàng, các cửa hàng hay trưng bày hàng mẫu, các cơ quan các đơn vị ngoài giờ hành chính để tăng năng suất làm việc giảm tối thiểu thời gian giám sát vào số lượng lớn camera.

## TÀI LIỆU THAM KHẢO

- [1] TS. Phan Ngọc Hoàng, Slide Computer Vision Advanced, 2019.
- [2] TS. Bùi Thu Trang, Slide Machine Learning, 2019.
- [3] Vũ Hữu Tiệp, Machine Learning cơ bản, 2017.
- [4] TS. Đỗ Năng Toàn, TS. Phạm Việt Bình, Giáo trình Xử lý ảnh, Đại học Thái Nguyên, năm 2007.
- [5] PGS.TS Nguyễn Quang Hoan, Giáo trình Xử lý ảnh, Học viện bưu chính viễn thông, năm 2006.
- [6] Lê Thị Lê Duyên, Mạng neuron tích chập và ứng dụng giải bài toán nhận dạng hành động trong một đoạn video, 2017.
- [7] Vũ Đình Thiện và các tác giả, "Sử dụng mạng YOLOv3 trong bài toán phát hiện và nhận dạng đối tượng trên ảnh", Kỷ yếu Hội nghị Khoa học Công nghệ Đại học Thái Nguyên lần thứ X, tr. 543-551, 2020.
- [8] Trương Công Lợi, Nhận dạng khuôn mặt sử dụng phương pháp biến đổi Eigenfaces và mạng neuron , Đại học Đà Nẵng, 2019.
- [9] Nguyễn Trường Tân, “Ứng dụng mạng neuron để phân loại khuôn mặt”, Đại học Đà Nẵng, 2018.
- [10] Nguyễn Văn Hùng, Nguyễn Văn Xuất, Lê Mạnh Cường. “Một phương pháp phát hiện đối tượng ứng dụng trong hệ thống tự động bám mục tiêu”, Viện Vũ khí, Học viện Kỹ thuật Quân sự, Bộ Quốc phòng, 2015.
- [11] Nguyễn Thị Thủy, “Phương pháp nhận dạng khuôn mặt người và ứng dụng trong quản lý nhân sự”, Đại học Công nghệ - Đại học Quốc gia Hà Nội, 2018.
- [12] Tống Văn Ngọc, “Nhận dạng và phát hiện hành động người dùng thị giác máy tính”, Đại học Sư phạm Kỹ thuật TP. Hồ Chí Minh, 2018.
- [13] Đỗ Văn Dương, Nghiên cứu phương pháp nhận dạng tự động một số đối tượng và xây dựng cơ sở dữ liệu 3D bằng dữ liệu ảnh thu nhận từ thiết bị bay không người lái, 2018.
- [14] Hoàng Kiếm, Nguyễn Hồng Sơn, Đào Minh Sơn, "Ứng dụng mạng noron nhân tạo trong hệ thống xử lý biểu mẫu tự động", 2019, truy cập 15/05/2023..

- [15] Nguyễn Tiến Đạt, “Ảnh số và điểm ảnh”, [Online]: [viblo.asia/p/tuan-1-gioi-thieu-xu-ly-anh-yMnKMdEQ57P](http://viblo.asia/p/tuan-1-gioi-thieu-xu-ly-anh-yMnKMdEQ57P), truy cập 10/05/2023.
- [16] Banghn, “Nhận dạng mặt người – Các hướng tiếp cận”, [Online]: <https://bloghnb.wordpress.com/tag/cac-huong-tiep-can-nhan-dang>, truy cập 12/05/2023.
- [17] Phạm Anh Phương, Ngô Quốc Tạo, Lương Chi Mai, "Trích chọn đặc trưng wavelet Haar kết hợp với SVM cho việc nhận dạng chữ viết tay tiếng Việt", 2018, truy cập 12/05/2023.
- [18] Nguyễn Thanh Tuấn, "oDeep Learning cơ bản", 2019.
- [19] Phạm Đình Khánh, “YOLO You Only Look Once”, [Online]: [phamdinhhkhanh.github.io/2020/03/09/DarknetAlgorithm.html%C3%A1o-bounding-box](http://phamdinhhkhanh.github.io/2020/03/09/DarknetAlgorithm.html%C3%A1o-bounding-box), truy cập 11/05/2023.
- [20] Phạm Duy Tùng, “Tìm hiểu single shot object detectors”, [Online]: [phamduytung.com/blog/2018-12-06-what-do-we-learn-from-single-shot-object-detection](http://phamduytung.com/blog/2018-12-06-what-do-we-learn-from-single-shot-object-detection), truy cập 17/05/2023.
- [21] FPT Software – AI phát hiện người xâm nhập, [Online]: [codelearn.io/sharing/ai-phat-hien-nguoi-xam-nhap-p2](http://codelearn.io/sharing/ai-phat-hien-nguoi-xam-nhap-p2), truy cập 26/05/2023.