

-----☆-----  
Số: \_\_\_\_\_/BKĐT

Khoa: **Điện – Điện tử**

Bộ Môn: **Viễn Thông**

## NHIỆM VỤ LUẬN VĂN TỐT NGHIỆP

- Họ và tên: Nguyễn Hoàng Khởi MSSV: 1611663
- Ngành: Điện – Điện tử Chuyên ngành: Kỹ thuật Điện tử - Truyền thông
- Đề tài: **Thuật toán học sâu – phát hiện thiết bị bảo hộ lao động cá nhân với YOLOv4**
- Nhiệm vụ:
  - Tìm hiểu về lý thuyết về học máy và học sâu.
  - Tìm hiểu thuật toán YOLO, các phiên bản của thuật toán này
  - Huấn luyện mạng YOLOv4 với bộ dữ liệu tự thu thập
  - Thiết kế và mô phỏng mô hình phát hiện thiết bị bảo hộ lao động cá nhân với YOLOv4
  - Đánh giá mô hình, kết luận, viết cuốn luận văn.
- Ngày giao nhiệm vụ luận văn: **15/10/2020**
- Ngày hoàn thành nhiệm vụ: **11/01/2021**
- Họ và tên người hướng dẫn: **TS. Võ Tuấn Kiệt** Phản hướng dẫn  
**100%**

BM Viễn Thông, Khoa Điện – Điện Tử

Nội dung và yêu cầu LVTN đã được thông qua Bộ Môn.

*TP.HCM, ngày 11 tháng 1 năm 2021*

**CHỦ NHIỆM BỘ MÔN**

**NGƯỜI HƯỚNG DẪN CHÍNH**

PGS. TS. Hà Hoàng Kha

TS. Võ Tuấn Kiệt

**PHẦN DÀNH CHO KHOA, BỘ MÔN:**

Người duyệt (chấm sơ bộ): .....

Đơn vị: .....

Ngày bảo vệ : .....

Điểm tổng kết: .....

Nơi lưu trữ luận văn: .....

## LỜI CẢM ƠN

Lời đầu tiên xin gửi lời cảm ơn chân thành đến thầy Võ Tuấn Kiệt đã tận tình chỉ dẫn, giúp đỡ, định hướng cho em trong suốt thời gian thực tập, đề cương luận văn và luận văn tốt nghiệp đã tạo điều kiện thuận lợi để em chọn và hoàn thành đề tài này.

Bên cạnh đó xin cảm ơn quý thầy cô bộ môn Điện-điện tử đã trang bị cho em những kiến thức nền tảng, bổ ích trong khoảng thời gian học tập tại trường.

Tôi xin gửi lời biết ơn sâu sắc đến cha mẹ, bạn bè và gia đình đã luôn quan tâm ủng hộ tôi trong suốt thời gian qua. Đó chính là nguồn động lực vô cùng lớn giúp tôi vượt qua những khó khăn và thử thách.

Cuối cùng, xin gửi lời cảm ơn đến các thành viên trong câu lạc bộ Pay It Forward và những người bạn trong suốt quá trình học tại trường Đại học Bách Khoa vì đã sẵn sàng tiếp nhận và chia sẻ kiến thức chuyên môn. Dựa vào đó để tôi có thể hoàn thành luận văn lần này.

Tp. Hồ Chí Minh, ngày 11 tháng 1 năm 2021

*Sinh viên*

*Nguyễn Hoàng Khởi*

## LỜI CAM ĐOAN

Tôi tên: Nguyễn Hoàng Khởi là sinh viên chuyên ngành Kỹ thuật Điện tử – Truyền thông, khóa 2016, tại Đại học Quốc gia thành phố Hồ Chí Minh – Trường Đại học Bách Khoa. Tôi xin cam đoan những nội dung sau đều là sự thật:

- i. Công trình nghiên cứu này hoàn toàn do chính tôi thực hiện.
- ii. Các tài liệu và trích dẫn trong luận văn này được tham khảo từ các nguồn thực tế, có uy tín và độ chính xác cao.
- iii. Các số liệu và kết quả của công trình này được tôi tự thực hiện một cách độc lập và trung thực.

Tp. Hồ Chí Minh, ngày 11 tháng 1 năm 2021

*Sinh viên*

*Nguyễn Hoàng Khởi*

## TÓM TẮT ĐỀ TÀI LUẬN VĂN

Trong luận văn này tập trung xây dựng một mô hình phát hiện thiết bị bảo hộ lao động cá nhân, mục tiêu của đề tài là phát hiện được cá nhân có đang mang thiết bị bảo hộ lao động (nón, áo, găng tay) hay không. Đây không phải là một chủ đề mới mà đã được nghiên cứu trong nhiều năm gần đây, với nhiều mô hình phát hiện đối tượng đã được đề xuất nhưng để đưa vào thực nghiệm trong thực tế thì đề tài vẫn còn đang có hướng phát triển tốt. Việc phát hiện chính xác được người lao động có đang mang thiết bị bảo hộ lao động hay không có thể ứng dụng sau này trong mô hình phát hiện và cảnh báo nguy hiểm tại khu vực làm việc nguy hiểm như công trường.

Trong quá trình thực hiện đề tài luận văn đề xuất sử dụng mô hình mạng YOLOv4-tiny nhằm phát hiện người lao động và mô hình mạng YOLOv4-tiny\_3l nhằm xác định các đối tượng là thiết bị bảo hộ lao động. Với sự kết hợp của hai mạng này luận văn sẽ đưa qua một bước đánh giá để xác định người lao động có mang thiết bị bảo hộ lao động hay không.

Để đánh giá kết quả của mô hình luận văn đã thực nghiệm trong các điều kiện khác nhau. Từ đó đưa ra kết luận mô hình có khả năng phát hiện thiết bị bảo hộ lao động trong khoảng cách từ 2.5-7(m), mô hình còn có khả năng phát hiện khi người lao động đang có hành động giữ thay vì mang thiết bị bảo hộ. Tốc độ xử lý trung bình của mô hình là 3.61 FPS gần đạt mức thời gian thực.

## **ABSTRACT**

In this thesis, focusing on building a model of detecting personal protective equipment, the goal of the topic is to detect individuals who are wearing personal protective equipment (hat, shirt, gloves) or not. This topic that has been studied for many years, with many object detection models being proposed but to put into practice in real-world, the topic is still developing. Accurate detection of whether or not workers are wearing protective equipment can be applied later in a detection and warning model in difficult work areas such as construction sites.

During the implementation of the thesis, the thesis proposes to use the YOLOv4-tiny network model to detect worker and the YOLOv4-tiny\_3l network model to identify objects as labor protection equipment. With the combination of these two networks, the thesis will go through an evaluation step to determine whether employees wear protective equipment or not.

To evaluate the results of the thesis model under different conditions. From there, it is concluded that the model is capable of detecting protective equipment in the distance from 2.5-7 (m), the model also has the ability to detect when the employee is taking action instead of carrying protective equipment. The average processing speed of the model is 3.61 FPS, which is close to the real-time level.

# MỤC LỤC

LỜI CẢM ƠN .....	i
LỜI CAM ĐOAN .....	ii
TÓM TẮT ĐỀ TÀI LUẬN VĂN .....	iii
ABSTRACT.....	iv
DOANH MỤC HÌNH ẢNH .....	ix
DOANH MỤC BẢNG.....	xiii
DOANH MỤC TỪ VIẾT TẮT.....	xiv
CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI. ....	1
1.1 Đặt vấn đề. ....	1
1.2 Mục tiêu của luận văn. ....	2
1.3 Đối tượng và phạm vi nghiên cứu.....	2
1.3.1 Đối tượng .....	3
1.3.2 Phạm vi nghiên cứu.....	3
1.4 Các đóng góp của luận văn .....	3
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VỀ HỌC MÁY VÀ HỌC SÂU.....	4
2.1 Định nghĩa và bài toán của học máy .....	4
2.1.1 Định nghĩa.....	4
2.1.2 Bài toán học có giám sát và học không giám sát. ....	4
2.2 Hai bài toán cơ bản của học máy. ....	5
2.2.1 bài toán hồi quy (regression).....	5
2.2.2 Bài toán phân loại. ....	6
2.3 Định nghĩa về học sâu .....	8
2.3.1 Định nghĩa về học sâu.....	8
2.3.2 Một số kiểu mạng thông dụng trong học sâu .....	8

2.4 Các quy ước ký hiệu cơ bản của một mạng thần kinh (neural network). ....	13
2.4.1. Lớp (Layer): .....	13
2.4.2. Đơn vị (Units): .....	13
2.4.3. Trọng số và sai số: .....	14
2.5 Một số phép toán cơ bản trong mạng thần kinh nhân tạo .....	14
2.5.1 Phép tính tích chập trong mạng thần kinh .....	14
2.5.2 Phép đệm 0 .....	15
2.5.3 Phép gộp (pooling).....	16
2.5.4 Lớp kết nối đầy đủ .....	16
2.5.5 Hàm softmax .....	17
2.6 Phương pháp giải bài toán mạng thần kinh cơ bản. ....	17
2.6.1 Hàm mất mát (loss function).....	18
2.6.2 Giải thuật lan truyền ngược (Back-propagation). ....	18
2.6.3 Các thuật toán tối ưu nâng cao. ....	20
2.7 Vấn đề quá khớp (Overfitting) và không khớp (underfitting) và phương pháp giải quyết.....	20
2.7.1 Vấn đề quá khớp và không khớp .....	21
2.7.2 Giải pháp .....	22
2.8 Một số phương pháp kiểm tra độ chính xác của mô hình mạng .....	24
2.8.1. Gradient checking. ....	25
2.8.2 Đánh giá bằng chỉ số IoU (intersection of union).....	25
2.8.3 Đánh giá bằng chỉ số F1 score. ....	26
2.9 Kết luận chương 2 .....	27
<b>CHƯƠNG 3. CƠ SỞ LÝ THUYẾT MÔ HÌNH MẠNG PHÁT HIỆN ĐỐI</b>	
<b>TƯỢNG YOLOv4 VÀ HUẤN LUYỆN MÔ HÌNH TRÊN TẬP DỮ LIỆU ..</b>	<b>28</b>
3.1 Lý thuyết thuật toán YOLOv4 (you look only one). ....	28

3.2 Huấn luyện thuật toán YOLOv4 nhận diện thiết bị bảo hộ lao động. ....	29
3.2.1 Thu thập dữ liệu và xử lý dữ liệu .....	30
3.2.2 Huấn luyện mạng trên google colab. ....	34
3.2.3 Cấu hình tập tin cấu trúc mạng YOLOv4 .....	39
3.2.4 Kết quả huấn luyện .....	39
3.3 Kết luận chương 3. ....	40
CHƯƠNG 4. PHÁT HIỆN THIẾT BỊ BẢO HỘ LAO ĐỘNG CÁ NHÂN .....	41
4.1 Đánh giá các phương pháp đã thực hiện. ....	41
Phương pháp 1 .....	41
Phương pháp 2 .....	43
4.1.1. Sơ đồ khối phương pháp 2 .....	45
4.1.2. Kết quả thực hiện. ....	46
4.2 Phương pháp 3 nhận dạng thiết bị bảo hộ lao động cá nhân với YOLOv4 được huấn luyện với 6 đối tượng.....	49
4.2.1 Sơ đồ giải thuật theo phương pháp 3 .....	49
4.2.2 Chương trình nhận dạng thiết bị bảo hộ lao động cá nhân .....	50
4.3 Kết thúc chương 4 .....	59
CHƯƠNG 5. KẾT QUẢ THỰC HIỆN VÀ ĐÁNH GIÁ MÔ HÌNH .....	60
5.1 Kết quả thực hiện .....	60
5.2 Đánh giá. ....	63
5.2.1 Đánh giá mô hình phát hiện đối tượng với thuật toán YOLOv4 .....	63
5.2.2 Đánh giá mô hình nhận dạng thiết bị bảo hộ cá nhân trong thực tế. ....	65
5.3 Kết luận chương 5. ....	65
CHƯƠNG.6 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	67
6.1 Kết luận. ....	67



6.2 Hướng phát triển. ....	67
PHỤ LỤC A.....	69
TÀI LIỆU THAM KHẢO.....	73

## DOANH MỤC HÌNH ẢNH

Hình 2-2-1: Ví dụ về hồi quy tuyến tính trong không gian 2D. ....	6
Hình 2-2-2: Hàm Sigmoid.....	7
Hình 2-3-1 Kiến trúc mạng LeNet-5.....	8
Hình 2-3-2: Kiến trúc mạng Alexnet .....	9
Hình 2-3-3: Kiến trúc mạng VGG-16.....	9
Hình 2-3-4: Kiến trúc mạng inception .....	10
Hình 2-3-5: Sơ đồ biểu diễn kiến trúc bên trong của một tế bào LSTM. ..	11
Hình 2-4-1: Cấu trúc của một mạng thần kinh điển hình. ....	13
Hình 2-5-1: Một mô hình tính tích chập .....	15
Hình 2-5-2: Phép đệm 0.....	15
Hình 2-5-3: Phép gộp.....	15
Hình 2-5-4: Phép kết nối đầy đủ với phương pháp flattening .....	16
Hình 2-5-5: hàm softmax.....	17
Hình 2-6-1: bài toán minh hoạ phương pháp gradient decent. ....	19
Hình 2-6-2: Giá trị đầu tiên của phương pháp gradient decent. ....	19
Hình 2-6-3: Điểm tối ưu của phương pháp gradient decent. ....	20
Hình 2-7-1: Hình minh hoạ vấn đề không khớp và quá khớp. ....	21
Hình 2-7-2: Ví dụ minh hoạ việc lựa chọn các giá trị learning rate khác nhau. ....	22
Hình 2-7-3: Ví dụ minh hoạ một hàm huấn luyện đã bị quá khớp .....	23
Hình 2-8-1: Ảnh minh hoạ cho phương pháp gradient checking .....	25
Hình 2-8-2: Minh hoạ tính toán chỉ số IoU.....	26
Hình 3-2-1: Dữ liệu cần thu thập trong quá trình thực hiện luận văn.....	30
Hình 3-2-2:Ảnh minh sau khi đã đổi tên .....	32

Hình 3-2-3: giao diện ứng dụng labeling .....	32
Hình 3-2-4: Ảnh minh họa của tập dữ liệu sau khi labelling.....	33
Hình 3-2-5: Nội dung file txt gồm: Classes, tọa độ (x,y), chiều rộng w, ..	33
chiều dài h. ....	34
Hình 3-2-6: Ảnh minh họa đối với những dữ liệu không chứa đối tượng.	34
Hình 3-2-7: Giao diện trình biên dịch ipynb trên Google colab.....	35
Hình 3-2-8: Cài đặt để được sử dụng GPU của Google Colab.....	35
Hình 3-2-9: Kết nối Google Drive và Google Colab.....	36
Hình 3-2-10: Ảnh minh họa quá trình tải và cấu hình một số thư viện trong darknet.....	36
Hình 3-2-11: Mã hoá tất cả các dữ liệu huấn luyện vào thư mục obj.....	36
Hình 3-2-12: Ảnh minh họa quá trình tải sao chép kiến trúc mạng yolov4.cfg.....	36
Hình 3-2-13: Tạo file obj.data gồm số đối tượng cần nhận diện và số lớp	37
Hình 3-2-14: Ảnh minh họa quá trình tạo tập tin train.txt và test.txt .....	37
Hình 3-2-15: Ảnh minh họa tải tập tin yolov4.conv.137 – bộ pre-training của yolov4 .....	37
Hình 3-2-16: Huấn luyện với bộ dữ liệu pre-training .....	37
Hình 3-2-17: Huấn luyện YOLO từ phiên bản mới nhất của trọng số 37 .	37
Hình 3-2-18: Ảnh minh họa đồ thị learning Curve.....	37
Hình 3-3-1: Ảnh minh họa kết quả huấn luyện với YOLOv4-tiny.....	38
Hình 4-1-1: Kết quả phương pháp 1 phát hiện thiết bị bảo hộ lao động ...	43
Hình 4-1-2: Ảnh minh họa đầu ra mạng OPENPOSE.....	45
Hình 4-1-3: Sơ đồ mô hình nhận dạng 3 lớp YOLOv4 kết hợp mạng Openpose .....	46

Hình 4-1-4: Kết quả sau khi qua mạng OPENPOSE xác định 4 vị trí mũi, cổ, cổ tay trái, cổ tay phải.....	47
Hình 4-1-5: Kết quả thực hiện theo phương pháp 2 sau khi qua mạng YOLOv4.....	48
Hình 4-1-6: Kết quả nhận dạng theo phương pháp 2.....	49
Hình 4-2-1: Sơ đồ giải thuật phát hiện thiết bị bảo hộ lao động cá nhân. .	48
Hình 4-2-2: Ảnh minh họa biểu tượng OpenCV .....	49
Hình 4-2-3: Thư viện hỗ trợ đồ họa được phát triển bởi Intel - OpenVINO.	50
Hình 4-2-4: Biểu đồ learning Curve kết quả huấn luyện YOLOv4-tiny đối tượng là người .....	51
Hình 4-2-5: Kết quả dự đoán sau khi qua mạng YOLOv4-tiny được huấn luyện phát hiện người .....	53
Hình 4-2-6: Biểu đồ learning Curve kết quả huấn luyện YOLOv4-tiny_3l gồm 6 đối tượng .....	54
Hình 4-2-7: Kết quả dự đoán của bộ phát hiện đối tượng là thiết bị bảo hộ .....	56
Hình 4-2-8: Ảnh minh họa một hộp giới hạn.....	57
Hình 4-2-9: Kết quả mô hình phát hiện thiết bị bảo hộ lao động .....	59
Hình 5-1-1: Kết quả thuật toán phát hiện đối tượng đang giữ mà không mang thiết bị bảo hộ.....	61
Hình 5-1-2: Kết quả thuật toán phát hiện đối tượng nhiều góc độ khi đứng.....	61
Hình 5-1-3: Kết quả thuật toán phát hiện đối tượng nhiều góc độ khi ngồi.....	62
Hình 5-1-4: Kết quả thuật toán phát hiện đối tượng trong khoảng cách 3-10m.....	62
Hình 7-1-1: Đối tượng là người lao động .....	69
Hình 7-1-2: Đối tượng đội nón bảo hộ .....	69

Hình 7-1-3: Đối tượng không đội nón bảo hộ .....	70
Hình 7-1-4: Thuật toán nhận diện người lao động mang áo bảo hộ .....	70
Hình 7-1-5: Thuật toán phát hiện không mang áo bảo hộ .....	71
Hình 7-1-6: Thuật toán phát hiện mang găng tay .....	71
Hình 7-1-7: Thuật toán phát hiện không mang găng tay .....	72

## **DOANH MỤC BẢNG**

Bảng 2-7-1: Định nghĩa về TP, FP, TN, FN.....	26
Bảng 4-1-1: Các khớp xương thuật toán Openpose có thể phát hiện.....	44
Bảng 5-2-1: Bảng kết quả đánh giá TP, TN, FP, FN cho 7 đối tượng.....	65

## DOANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Tên tiếng anh	Ý nghĩa
YOLO	You only look once	Thuật toán nhận dạng vật thể có phương pháp hoàn toàn khác so với các thuật toán nhận dạng khác với tốc độ vượt trội so với các thuật toán ra đời trước nó.
LSTM	Long short term memory	Mô hình mạng bộ nhớ ngắn hạn hướng dài hạn
FPS	Frame per second	Tốc độ xử lý của thuật toán trong 1 giây
mAP	Mean Average Precision	độ chính xác trung bình thường được sử dụng để đo độ chính xác của mạng có sử dụng hộp giới hạn
IoU	Intersection of Union	Chỉ số đánh giá mạng sử dụng hộp giới hạn, giao giữa hộp giới hạn gốc và hộp giới hạn dự đoán
R	Recall	Chỉ số đánh giá mạng so được sử dụng trong F1 Score
P	Precision	Chỉ số đánh giá mạng so được sử dụng trong F1 Score

# **CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI.**

Nội dung chương một nhằm giới thiệu về tính cấp thiết của đề tài luận văn. Qua đó luận văn muốn giới thiệu nhu cầu thực tiễn của đề tài, mục tiêu, phạm vi nghiên cứu và các bước thực hiện đề tài. Chương này có liên quan đến nội dung chương 3,4,5.

## **1.1 Đặt vấn đề.**

Theo báo cáo của Cục an toàn lao động, Bộ lao động – Thương binh và Xã hội, Việt Nam công khai trên trang an toàn lao động [1] thì:

- Trong năm 2017 Việt Nam đã xảy ra 8.956 vụ tai nạn lao động (TNLĐ) làm 9.173 người bị nạn (bao gồm cả khu vực có quan hệ lao động và khu vực người lao động làm việc không theo hợp đồng lao động). Trong đó: số vụ TNLĐ chết người là 898 vụ, số vụ TNLĐ có hai người bị nạn trở lên là 101 vụ; số người chết là 928 người, số người bị thương nặng là 1.915 người, nạn nhân là lao động nữ là 2.727 người.

- Đến năm 2018 trên toàn quốc đã xảy ra 7.997 vụ tai nạn lao động (TNLĐ) làm 8.229 người bị nạn (bao gồm cả khu vực có quan hệ lao động và khu vực người lao động làm việc không theo hợp đồng lao động). Trong đó: số vụ TNLĐ chết người: 972 vụ (tăng 8,24% so với năm 2017); số vụ TNLĐ có hai người bị nạn trở lên: 112 vụ (tăng 10,89% so với năm 2017); số người chết vì TNLĐ là 1.039 người (tăng 11,96% so với năm 2017); số người bị thương nặng: 1.939 người (tăng 1,25% so với năm 2017); nạn nhân là lao động nữ: 2.667 người (giảm 2,2% so với năm 2017).

- Theo thống kê mới nhất trong năm 2019 trên toàn quốc đã xảy ra 8.150 vụ tai nạn lao động (TNLĐ) làm 8.327 người bị nạn (bao gồm cả khu vực có quan hệ lao động và khu vực người lao động làm việc không theo hợp đồng lao động). Trong đó: số vụ TNLĐ chết người: 927 vụ (giảm 4,63% so với năm 2018); số vụ TNLĐ



có hai người bị nạn trở lên: 146 vụ (tăng 3,03% so với năm 2018); số người chết vì TNLD: 979 người (giảm 5,77% so với năm 2018); số người bị thương nặng: 1.892 người (giảm 2,42% so với năm 2018); nạn nhân là lao động nữ: 2.771 người (tăng 3,9% so với năm 2018); .

Từ đó ta thấy rằng vấn đề an toàn lao động vẫn đang xảy ra thường xuyên và gây hậu quả nghiêm trọng ở nước ta. Do đó trong luận văn này em muốn sử dụng các thuật toán trong lĩnh vực học sâu để đề xuất một mô hình phát hiện thiết bị bảo hộ lao động cá nhân nhằm góp một phần nhỏ công sức giảm số vụ tai nạn do an toàn lao động trong tương lai.

Nội dung của luận văn gồm 7 chương:

Chương 1: Tổng quan về đề tài

Chương 2: Cơ sở lý thuyết thuật toán học máy và học sâu

Chương 3: Cơ sở lý thuyết mô hình mạng phát hiện đối tượng YOLO và huấn luyện mô hình trên tập dữ liệu

Chương 4: Nhận dạng thiết bị bảo hộ lao động cá nhân

Chương 5: Kết quả thực hiện và đánh giá mô hình

Chương 6: Kết luận và hướng phát triển

## **1.2 Mục tiêu của luận văn.**

Luận văn này tập trung vào xây dựng một mô hình phát hiện thiết bị bảo hộ lao động cá nhân. Gồm các phần sau:

- Tìm hiểu về thuật toán học máy và học sâu
- Thu thập dữ liệu thực tế thực hiện trong đề tài
- Tìm hiểu thuật toán YOLOv4 và sử dụng để phát hiện thiết bị bảo hộ lao động cá nhân
- Xây dựng mô hình phát hiện thiết bị bảo hộ lao động cá nhân.
- Đánh giá mô hình, kết luận, viết cuốn luận văn

## **1.3 Đối tượng và phạm vi nghiên cứu**

### *1.3.1 Đối tượng*

Luận văn tập trung nghiên cứu vào các phương pháp phát hiện thiết bị bảo hộ lao động cá nhân nhằm ứng dụng thực tiễn trong cuộc sống. Các đối tượng mà luận văn phát hiện gồm người lao động có hay không mang mũ bảo hộ, người lao động có hay không mang áo bảo hộ và có hay không mang găng tay bảo hộ.

### *1.3.2 Phạm vi nghiên cứu*

Luận văn nghiên cứu, thiết kế một mô hình phát hiện thiết bị bảo hộ lao động cá nhân bằng camera và được xử lý bằng máy tính DELL – Inspiron 13. Do tính chất phức tạp của môi trường thực tế nên phạm vi hoạt động của luận văn chỉ tập trung vào nghiên cứu trong nhà để từ đó phát triển để thực hiện trong thực tế.

## **1.4 Các đóng góp của luận văn**

- Tìm hiểu lý thuyết về cách thức hoạt động của một số giải thuật sử dụng trong học máy và học sâu.
- Tìm hiểu lý thuyết về thuật toán nhận dạng đối tượng YOLO
- Huấn luyện bộ dữ liệu thu thập được với mạng YOLOv4 giải quyết một vấn đề thực tế
- Thiết kế một mô hình phát hiện thiết bị bảo hộ lao động cá nhân.
- Đánh giá mô hình, ưu nhược điểm của mô hình.
- Đưa ra các phương pháp cải tiến và kết luận.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VỀ HỌC MÁY VÀ HỌC SÂU.

Trong chương này, luận văn giới thiệu về những phương pháp và cách giải quyết các bài toán cơ bản trong lĩnh vực học máy (machine learning) và học sâu (deep learning), từ đó hình dung ra một mô hình đơn giản để giải quyết các bài toán phức tạp hơn.

### 2.1 Định nghĩa và bài toán của học máy

#### 2.1.1 Định nghĩa.

Học máy (Machine learning) theo định nghĩa của wikipedia, “*Machine learning is the subfield of computer science that “gives computers the ability to learn without being explicitly programmed”*”, ý nghĩa là máy tính được lập trình để có thể tự học nhiều bài toán mà không cần phải lập trình chi tiết để máy có thể học bài toán đó.

Học máy là một thuật ngữ rộng để chỉ hành động chúng ta dạy máy tính cải thiện một nhiệm vụ mà nó đang thực hiện. Khả năng cơ bản nhất của học máy là sử dụng thuật toán để phân tích những thông tin có sẵn, học hỏi từ nó rồi đưa ra quyết định hoặc dự đoán về một thứ gì đó có liên quan. Thay vì tạo ra một phần mềm với những hành động, hướng dẫn chi tiết để thực hiện một nhiệm vụ cụ thể, máy tính được “huấn luyện” bằng cách sử dụng lượng dữ liệu và các thuật toán để học cách thực hiện nhiệm vụ.

#### 2.1.2 Bài toán học có giám sát và học không giám sát.

Thông thường các thuật toán học máy được chia thành hai kỹ thuật học chính là:

- Học có giám sát (Supervised Learning): là những thuật toán học mà ta đã biết trước đáp án ví dụ như phân loại email gửi đến là spam hay không spam. Thì ta đã biết trước đáp án là spam hoặc không spam

- Học không giám sát (Unsupervised Learning): không nói đến thuật toán phân loại mà nói đến các thuật toán nhóm dữ liệu. Ta không biết nó là nhóm dữ liệu gì nhưng ta có thể tự động tìm ra cấu trúc trong dữ liệu từ việc tự động gom nhóm các cá thể từ những loại giống nhau mà ta không biết trước. Bởi vì chúng ta không đưa ra trong thuật toán câu trả lời chính xác cho các mẫu dữ liệu trong bộ dữ liệu.

## 2.2 Hai bài toán cơ bản của học máy.

Có hai loại bài toán kinh điển trong học máy là:

- Hồi quy
- Phân loại

### 2.2.1 bài toán hồi quy (regression).

Bài toán hồi quy là một trong những bài toán cơ bản nhất của học máy (machine learning). Nội dung bài toán là tìm được một phương trình từ những dữ liệu thu thập được (feature) để dự đoán kết quả của của đối tượng. Ví dụ như dự đoán giá nhà tại thành phố từ các dữ liệu thu thập được là diện tích đất, số phòng, vị trí, giá ở quá khứ,...

Phương trình của bài toán hồi quy có dạng như sau:

$$h_{\theta} = a_{i,j}x + b_{i,j} \quad (2.1)$$

$h_{\theta}$ : là kết quả dự đoán được của phương trình hồi quy

$a_{i,j}$ : là trọng số của vector X tương ứng với số chiều i, số bậc j

$x$ : là ma trận số feature của vector  $x = [x_1, x_2, x_3, x_4, \dots]$ , hoặc là số bậc của x trong không gian hai chiều.

$b_{i,j}$ : là chỉ số bias của phương trình với với trọng số a.

bài toán cơ bản của hồi quy là hồi quy tuyến tính (linear regression)

Bài toán hồi quy tuyến tính:

Để hiểu về bài toán này ta sẽ đi qua một ví dụ sau, tại một trung tâm môi giới bất động sản người ta thu thập dữ liệu gồm diện tích của căn nhà cần bán, số phòng trong nhà, giá bán của căn nhà tại thời điểm  $t$ . Tương ứng với các feature là  $x_1, x_2, x_3$ , từ đó trung tâm này muốn dự đoán giá nhà hiện tại  $h_\theta$  bằng công thức hồi quy sau.

$$h_\theta = a_i x + b_i \quad (2.2)$$

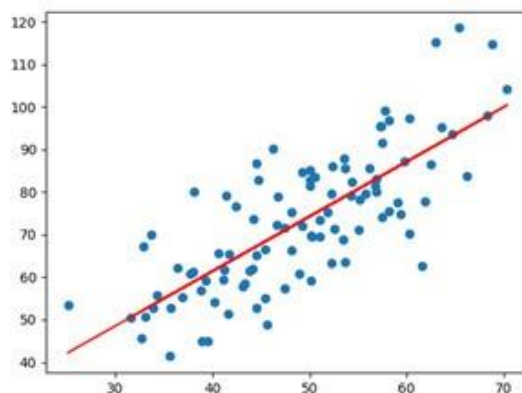
$h_\theta$ : là giá nhà dự đoán ở hiện tại

$a_i$ : là trọng số ứng với  $x_i$

$x$ : là diện tích căn nhà, số phòng, giá trong quá khứ tương ứng với các feature  $x_1, x_2, x_3$

$b_i$ : là chỉ số bias của phương trình với với trọng số  $a_i$

Đồ thị của phương trình hồi quy trong không gian hai chiều sẽ có dạng tuyến tính.



Hình 2-2-1: Ví dụ về hồi quy tuyến tính trong không gian 2D.

### 2.2.2 Bài toán phân loại.

Phân loại là bài toán liên kết đầu vào đã cho như ảnh, âm thanh để phân loại thành các lớp thích hợp hoặc không thích hợp. Các lớp được biểu diễn trong một vector nhị phân  $C = [c_1, c_2, c_3, \dots]$  trong đó  $c_i = 1$  đại diện cho đầu vào thuộc về lớp

thứ  $i$ . Chúng ta chỉ xem xét trường hợp đầu vào chỉ có thể thuộc về một lớp, vì vậy các thành phần  $c_{j \neq i}$  phải bằng 0.

Đại diện cơ bản của bài toán phân loại là bài toán hồi quy logistic (logistic regression).

Bài toán hồi quy logistic:

Chúng ta bắt đầu với trường hợp chỉ có một lớp mục tiêu  $c$ , trong đó  $c$  là xác suất đầu vào thuộc về lớp đã cho. Giá trị gần với 1 hơn thì thể hiện độ tin cậy cao rằng đầu vào thuộc về lớp đã cho, và ngược lại, giá trị gần 0 thể hiện một độ tin cậy thấp.

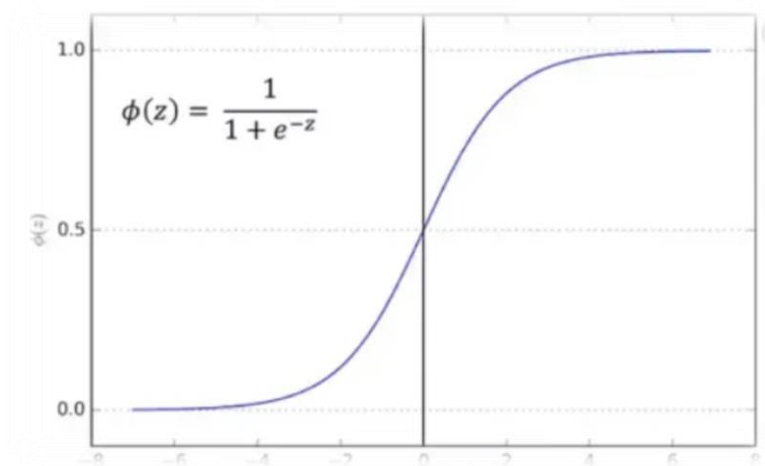
Logistic regression khá giống với hồi quy tuyến tính. Sự khác nhau là đầu ra được giới hạn trong khoảng từ 0 đến 1 (nếu không thì đó không phải là xác suất). Đặt  $z$  là phụ thuộc tuyến tính từ đầu vào  $x$  bởi:

$$z = W^T x + b \quad (2.3)$$

Để giới hạn giá trị đầu ra giữa 0 và 1 người ta sử dụng hàm kích hoạt sigmod.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.4)$$

Điều này làm cho hàm sigmoid khả thi cho việc mô hình hóa một xác suất.



Hình 2-2-2: Hàm Sigmod

## 2.3 Định nghĩa về học sâu

### 2.3.1 Định nghĩa về học sâu

Học sâu (Deep learning) theo định nghĩa của wikipedia, “*Deep learning (also known as deep structured learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised.*”, ý nghĩa là học sâu là một lĩnh vực của học máy giải quyết vấn đề dựa trên mạng thần kinh nhân tạo. Học máy cũng có thể chia thành học có giám sát, học không giám sát hoặc cả hai.

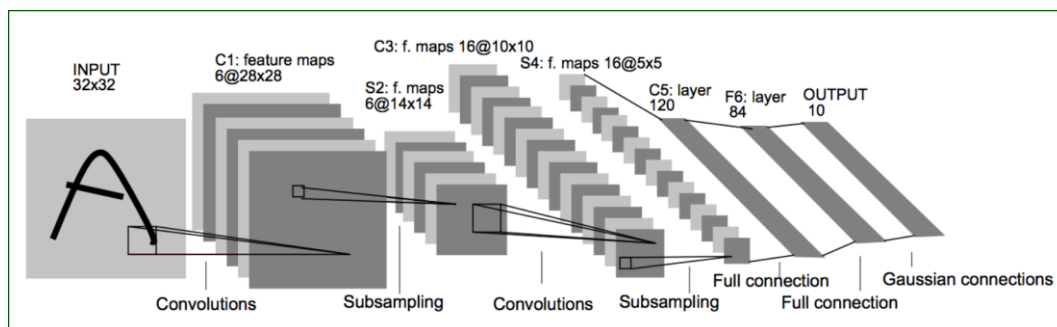
### 2.3.2 Một số kiểu mạng thông dụng trong học sâu

Đưa vào thiết kế mạng trong mô hình học sâu chúng ta chia thành nhiều kiểu mạng khác nhau như mạng truyền thông, mạng inception, mạng LSTM,... Việc nắm bắt về khái niệm của những kiểu mạng sẽ trợ giúp chúng ta trong việc xây dựng hoặc học hỏi một kiến trúc mạng mới một cách dễ dàng.

#### 2.3.2.1 Mạng truyền thông - mạng truyền thẳng.

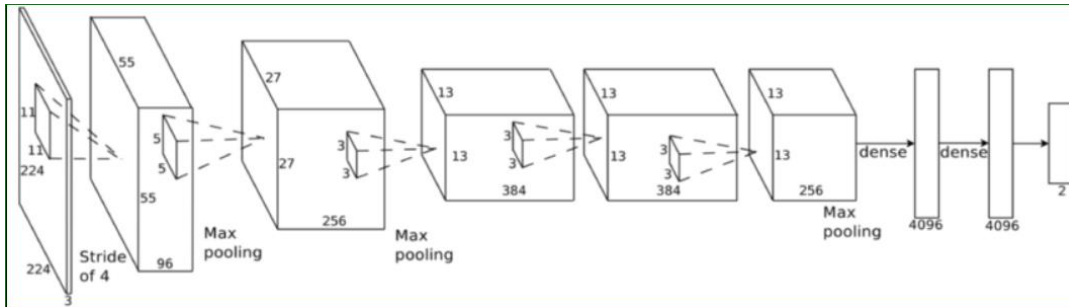
Là kiểu mô hình mạng mà lớp trước là đầu vào của lớp sau tạo thành một chuỗi các lớp đi về phía trước.

Một số ví dụ của mạng này như LeNet-5, Alexnet, VGG-16.



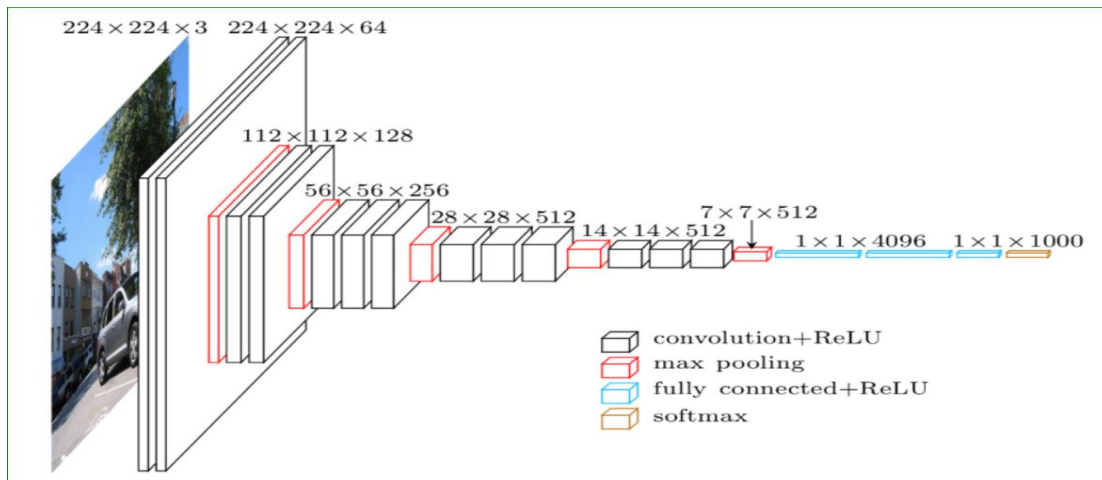
Hình 2-3-1 Kiến trúc mạng LeNet-5

Được tạo ra bởi Yann Lecun năm 1998, để nhận diện chữ viết tay



Hình 2-3-2: Kiến trúc mạng Alexnet

Được phát triển bởi Alex Krizhevsky năm 2012 trong cuộc thi ImageNet 2012.



Hình 2-3-3: Kiến trúc mạng VGG-16

Được phát triển năm 2014, là một biến thể sâu hơn nhưng lại đơn giản hơn so với kiến trúc convolution thường thấy ở CNN.

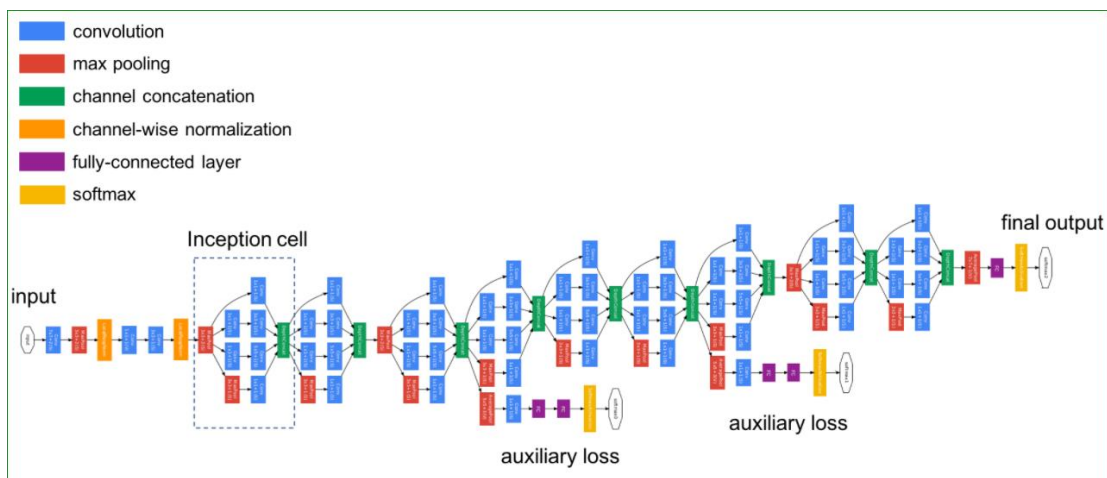
### 2.3.2.2 Mạng Inception (GoogLeNet).

Vào năm 2014, các nhà nghiên cứu của google đã đưa ra mạng Inception -> tham dự cuộc thi ImageNet 2014.



Mô hình này khá đặc biệt, không hoàn toàn là các tầng layer nối tiếp gói đầu lên nhau như các mạng trên. Mạng gồm các đơn vị gọi là “inception cell” - thực hiện phép tích chập (convolution) một giá trị đầu vào (input) với nhiều bộ lọc (filter) khác nhau rồi tổng hợp lại, theo nhiều nhánh (branch).

Để tiết kiệm tính toán, các phép tích chập (convolution) kích thước 1x1 được sử dụng để giảm chiều sâu kênh của giá trị đầu vào (input channel). Với mỗi 1 cell, sử dụng các bộ lọc (filter) kích thước 1x1, 3x3, 5x5 để trích xuất các đặc trưng từ đầu vào.



Hình 2-3-4: Kiến trúc mạng inception

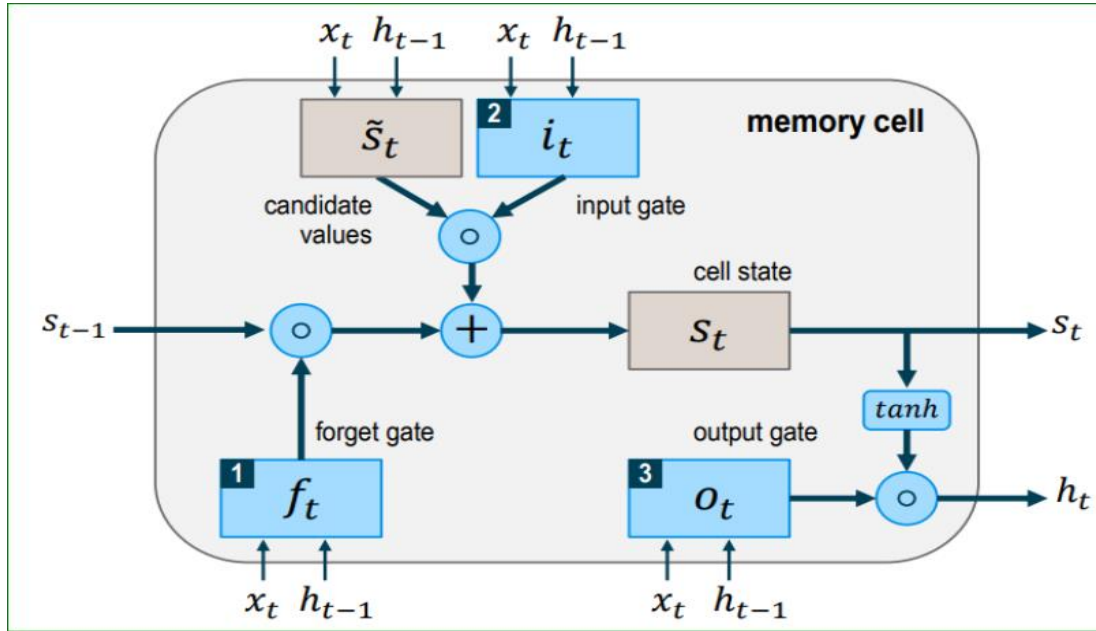
Công bố bởi các kỹ sư Google vào năm 2014. Mạng gồm các cell nối tiếp với nhau mỗi cell gọi là một inception cell như hình vẽ 2-3.

#### 2.3.2.4 Mạng Long short term memory (LSTM)

Mạng LSTM là một phiên bản mở rộng của mạng RNN (Recurrent Neural Network), được đề xuất vào năm 1997 bởi Sepp Hochreiter và Jurgen Schmidhuber. Cũng giống như RNN, LSTM sinh ra để xử lý những bài toán dạng chuỗi nhưng bởi vì RNN không thể ghi nhớ các thông tin từ những bước ở xa dẫn tới các phần tử đầu tiên của chuỗi không ảnh hưởng nhiều đánh kết quả tính toán cho chuỗi đầu ra của mạng. Thay vì đó kiến trúc LSTM có thể liên kết giữa các chuỗi đầu vào với

các chuỗi đầu ra ở xa nó nhằm tạo ảnh hưởng nhất định đến đầu ra từ các chuỗi đầu vào.

Một số ứng dụng của LSTM là xử lý ngôn ngữ như chuỗi ký tự, chuỗi văn bản, chữ viết tay, âm thanh,...



Hình 2-3-5: Sơ đồ biểu diễn kiến trúc bên trong của một tế bào LSTM.

Mạng LSTM có thể bao gồm nhiều tế bào LSTM (LSTM memory cell) liên kết với nhau. Ý tưởng của LSTM là bổ xung thêm trạng thái bên trong tế bào (cell internal state)  $s_t$  và 3 cổng sàng lọc các thông tin đầu vào và đầu ra cho tế bào bao gồm forget gate  $f_t$ , input gate  $i_t$  và output gate  $o_t$ . Tại mỗi bước thời gian  $t$ , các cổng đều lần lượt nhận các giá trị đầu vào là  $x_t$  (đại diện cho phần tử trong chuỗi đầu vào) và giá trị  $h_{t-1}$  có được từ đầu của memory cell của bước trước đó  $t-1$ . Các cổng đều đóng vai trò có nhiệm vụ sàng lọc thông tin với mỗi mục đích khác nhau.

- Forget gate: có nhiệm vụ loại bỏ những thông tin không cần thiết nhận được khỏi cell internal state.
- Input gate: Có nhiệm vụ chọn lọc những thông tin cần thiết nào được thêm vào cell internal state.

- Output gate: Có nhiệm vụ xác định nhưng thông tin nào từ cell internal state được sử dụng như đầu ra.

Trước khi trình bày các phương pháp mô tả hoạt động của LSTM. Ta thống nhất quy ước các ký hiệu sau đây:

- $x_t$  là vector đầu vào tại mỗi bước thời gian  $t$ .
- $W_{f,x}$ ,  $W_{f,h}$ ,  $W_{s,x}$ ,  $W_{s,h}$ ,  $W_{i,x}$ ,  $W_{i,h}$ ,  $W_{o,x}$ ,  $W_{o,h}$  là các ma trận trọng số trong mỗi tế bào LSTM.
- $b_f$ ,  $b_s$ ,  $b_i$ ,  $b_o$  là các vector bias.
- $f_t$ ,  $i_t$ ,  $o_t$  lần lượt là các giá trị kích hoạt lần lượt cho các cổng forget gate, input gate và output gate tương ứng.
- $s_t$ ,  $\delta$  lần lượt là các vector đại diện cho cell internal state và candidate value
- $h_t$  là giá trị đầu ra của tế bào LSTM.

Trong quá trình lan truyền xuôi (forward pass), cell internal state  $s_t$  và giá trị đầu ra  $h_t$  được tính như sau:

- Ở bước đầu tiên, tế bào LSTM quyết định những thông tin nào cần được loại bỏ từ cell internal state ở bước thời gian trước đó  $s_{t-1}$ . Activation value  $f_t$  của forget gate tại bước thời gian  $t$  được tính dựa trên giá trị đầu vào hiện tại  $x_t$ , giá trị đầu ra  $h_{t-1}$  từ tế bào LSTM ở bước trước đó và bias  $b_f$  của forget gate. Hàm sigmoid function biến đổi tất cả activation value về miền có giá trị trong khoảng từ 0 (hoàn toàn quên) và 1 (hoàn toàn nhớ).

$$f_t = \sigma(W_{f,x}x_t + W_{f,h}h_{t-1} + b_f) \quad (2.5)$$

- Ở bước thứ 2, tế bào LSTM quyết định những thông tin nào cần được thêm vào cell internal state  $s_t$ . Bước này bao gồm hai quá trình tính toán đối với  $f_t$  và  $s_t$ . Candidate value  $s_t$  biểu diễn những thông tin tìm năng cần được thông vào cell internal state được tính như sau:

$$s_t = \tanh(W_{s,x}x_t + W_{s,h}h_{t-1} + b_s) \quad (2.6)$$

- Activation value  $i_t$  của input gate theo sau đó cũng được tính như sau:

$$i_t = \tanh(W_{i,x}x_t + W_{i,h}h_{t-1} + b_i) \quad (2.7)$$

- Ở bước thứ ba, giá trị mới của cell internal state  $s_t$  được tính theo kết quả tính toán thu được từ các bước với phép tích chập theo từng phần tử:

$$s_t = f_t * s_{t-1} + i_t * \delta_t \quad (2.8)$$

- Ở bước cuối cùng, giá trị đầu ra  $h_t$  của tế bào LSTM được tính toán được theo hai phương trình sau:

$$o_t = \sigma(W_{o,x}x_t + W_{o,h}h_{t-1} + b_o) \quad (2.9)$$

$$h_t = o_t * \tanh(s_t) \quad (2.10)$$

## 2.4 Các quy ước ký hiệu cơ bản của một mạng thần kinh (neural network).

Để bắt đầu phân tích các thuật toán của học sâu ta trước hết làm quen với một vài quy ước cơ bản trong mạng thần kinh.

### 2.4.1. Lớp (Layer):

Một mô hình mạng gồm ba lớp cơ bản là lớp đầu ở phía đầu của mạng (input layer), lớp đầu ra ở cuối mạng (output layer) và các lớp ở giữa gọi là lớp ẩn (hidden layer).

- Số lượng lớp (ký hiệu  $L$ ) trong một mạng thần kinh được tính bằng số lớp ẩn cộng với 1
- Đối với mạng truyền thẳng đầu ra của lớp này là đầu vào của lớp tiếp theo nó

### 2.4.2. Đơn vị (Units):

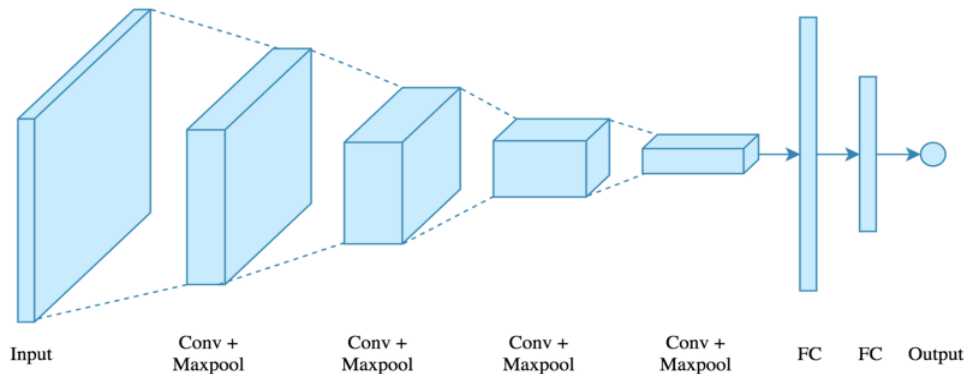
- Một nút hình tròn trong 1 lớp được gọi là 1 đơn vị
- Đầu vào của các lớp ẩn được ký hiệu là  $z$ , đầu ra của mỗi đơn vị thường được ký hiệu là  $a$
- Đầu ra của đơn vị thứ  $i$  trong lớp thứ  $j$  được ký hiệu là  $a_i^{(j)}$

### 2.4.3. Trọng số và sai số:

- $W^{(l)}$  là ma trận trọng số của một mạng thần kinh, nó thể hiện kết nối từ lớp (j-1) đến layer (j)
- Các sai số của lớp thứ (j) được ký hiệu là  $b^{(j)}$

## 2.5 Một số phép toán cơ bản trong mạng thần kinh nhân tạo

Để mô tả về một số phép toán cơ bản trong mạng thần kinh ta quan sát hình 2-5-1. Trong mạng thần kinh trên ta có thể thấy được quy trình từ một đặc tính đầu sau, sau khi qua một tập những phép toán cơ bản sẽ thu được kết quả đầu ra là giá trị cần xác định của bài toán, thường là bộ phân lớp.

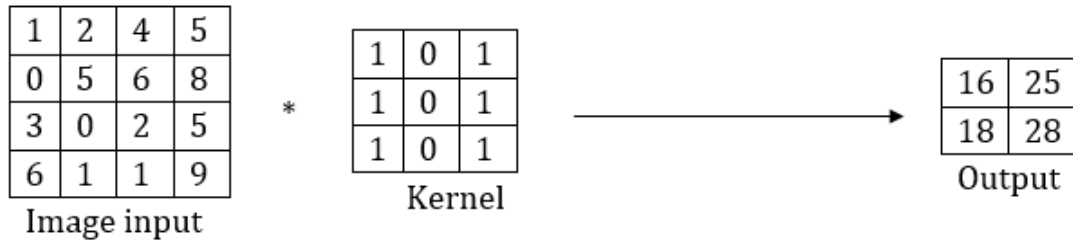


Hình 2-5-1: Cấu trúc của một mạng thần kinh điển hình.

### 2.5.1 Phép tính tích chập trong mạng thần kinh

Phép tích chập (convolution): là phép toán thực hiện nhân chập từng phần tử bên trong ma trận kernel với ma trận ảnh đầu vào.

Ví dụ: ta có một ma trận ảnh đầu vào có kích thước 4x4 và một ma trận kernel có kích thước 3x3. Qua phép tích chập ta sẽ được kết quả như sau:



Hình 2-5-1: Một mô hình tính tích chập

Gọi ma trận ảnh đầu ra (kết quả của phép tích chập của ma trận đầu vào X) là A, ta có:

$$A_{11} = 1 \times 1 + 2 \times 0 + 4 \times 1 + 0 \times 1 + 5 \times 0 + 6 \times 1 + 3 \times 1 + 0 \times 0 + 2 \times 1 = 16$$

$$A_{12} = 2 \times 1 + 4 \times 0 + 5 \times 1 + 5 \times 1 + 6 \times 0 + 8 \times 1 + 0 \times 1 + 2 \times 0 + 5 \times 1 = 25$$

$$A_{21} = 0 \times 1 + 5 \times 0 + 6 \times 1 + 3 \times 1 + 0 \times 0 + 2 \times 1 + 6 \times 1 + 1 \times 0 + 1 \times 1 = 18$$

$$A_{22} = 5 \times 1 + 6 \times 0 + 8 \times 1 + 0 \times 1 + 2 \times 0 + 5 \times 1 + 1 \times 1 + 1 \times 0 + 9 \times 1 = 28$$

Như vậy ta có một ma trận A có kích thước nhỏ hơn X, tuy nhiên ta muốn ma trận A có kích thước bằng với X thì có hướng giải quyết là ta sẽ thêm giá trị 0 ở viền ngoài của ma trận X bằng phép đệm 0

### 2.5.2 Phép đệm 0

Phép đệm 0 (Zero – Padding): một đường viền phụ được thêm vào xung quanh ma trận đầu vào làm tăng kích thước của nó, dẫn đến tăng kích thước của ma trận đầu ra tránh mất mát thông tin.

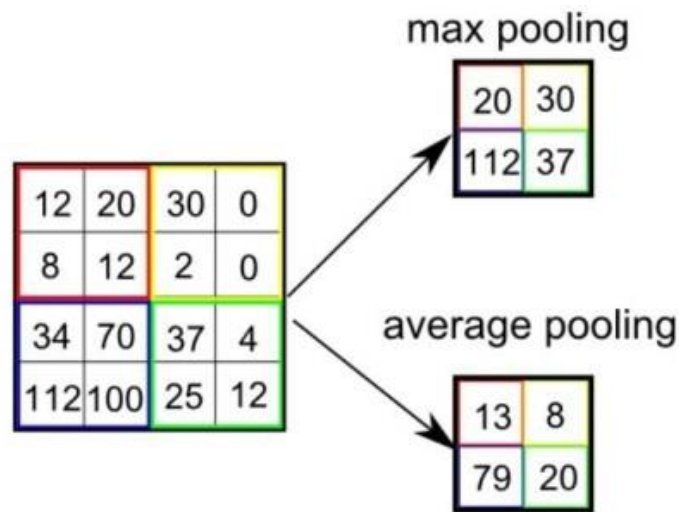
0	0	0	0	0	0	0
0						0
0						0
0						0
0						0
0						0
0	0	0	0	0	0	0

Hình 2-5-2: Phép đệm 0

### 2.5.3 Phép gộp (pooling)

Trong phần trước ta đã tìm hiểu về phép mở rộng ma trận là phép đệm 0. Trong phần tiếp theo này ta sẽ tìm hiểu về phép tính thu nhỏ ma trận gọi là phép gộp.

Phép đệm là một phương thức làm co ảnh nhưng vẫn giữ được đặc trưng của ảnh. Lớp đệm thường được sử dụng ngay sau lớp tích chập để đơn giản hóa thông tin đầu ra, giảm bớt số lượng nút ở đầu cuối



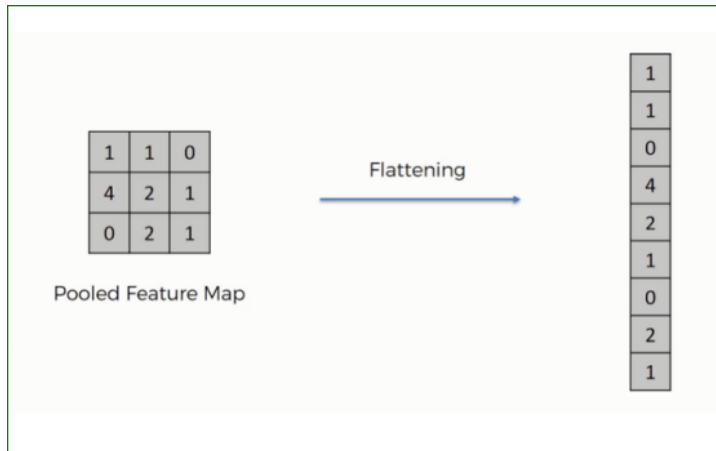
Hình 2-5-3: Ảnh minh họa phép gộp

Để thực hiện phép đệm ta chia một ma trận thành các ma trận vuông nhỏ hơn.

Sau đó có hai gộp là lấy giá trị lớn nhất trong ma trận và lấy giá trị trung bình trong ma trận

### 2.5.4 Lớp kết nối đầy đủ

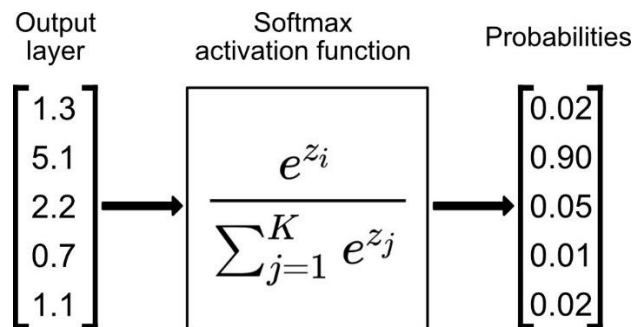
Lớp kết nối đầy đủ (fulling connected): Sau khi ảnh được truyền qua nhiều lớp tích chập thì mô hình đã học được tương đối các đặc điểm của ảnh thì đầu ra của lớp cuối cùng, kích thước  $H \times W \times D$ , sẽ được chuyển về 1 vector kích thước  $(H \times W \times D)$ .



Hình 2-5-4: Phép kết nối đầy đủ với phương pháp flattening

Thường sau lớp kết nối đầy đủ người ta thường dùng hàm softmax để xác định ra được đối tượng có xác suất cao nhất.

### 2.5.5 Hàm softmax



Hình 2-5-5: hàm softmax

Từ hình 2-5-5 ta thấy hàm softmax có chức năng hướng tăng xác suất của giá trị cao nhất ở đầu vào gần về 1, còn xác suất của các giá trị nhỏ hơn gần về 0.

Hàm softmax thường được sử dụng để phân lớp đối tượng.

## 2.6 Phương pháp giải bài toán mạng thần kinh cơ bản.



### 2.6.1 Hàm mất mát (loss function)

Để giải bài toán mạng thần kinh cơ bản ta sử dụng một hàm gọi là *loss function*  $\xi$  đôi khi người ta gọi là *cost function*. Đây không phải là tên gọi của một hàm mà là tên dành riêng cho hàm tối ưu để giải bài toán mạng thần kinh nhân tạo.

Đối với bài toán hồi quy tuyến tính ta sử dụng hàm mất mát là hàm mean squared error (MSE):

$$\xi = \frac{1}{N} \sum_{i=1}^N \|t_i - y_i\|^2 \quad (2.11)$$

Đối với bài toán hồi quy logistic ta sử dụng hàm mất mát là hàm cross-entropy:

$$\xi = \frac{1}{m} \sum_{i=1}^m y^{(i)} \log h^{(i)} + (1 - y^{(i)}) \log(1 - h^{(i)}) \quad (2.12)$$

Đối với bài toán SVM (support vector machine) người ta sử dụng hàm hinge loss:

$$\xi = \max(0, 1 - z) \quad (2.13)$$

Việc giải bài toán mạng thần kinh cũng tương tự việc tìm nghiệm tối ưu cho bài toán hàm mất mát  $\xi$ .

### 2.6.2 Giải thuật lan truyền ngược (Back-propagation).

Lan truyền ngược sai số (backward propagation of errors): là phương pháp để huấn luyện mạng thần kinh kết hợp với một thuật toán tối ưu như gradient descent để tính hàm tổn thất. Nhằm để cập nhật trọng số và hàm mất mát.

Giải thuật này là vòng lặp của hai quá trình:

- Lan truyền về trước (forward propagation): từ giá trị đầu vào tính lần lượt các giá trị phía trước.
- Lan truyền ngược (backward propagation): tính sai số ở đầu ra của của mạng rồi lan truyền ngược để cập nhật lại các trọng số phía trước.

Sau hai bước trên giá trị hàm mất mát  $\xi$  và trọng số  $\theta_j^{(i)}$  sẽ được cập nhật lại

Thuật toán sẽ kết thúc khi hàm mất mát tại vị trí cực trị toàn cục hoặc cực trị địa phương.

Giá trị trọng số sau khi cập nhật:

$$\theta^{(k+1)} = \theta^{(k)} - n\delta\xi(\theta^{(k)}) \quad (2.14)$$

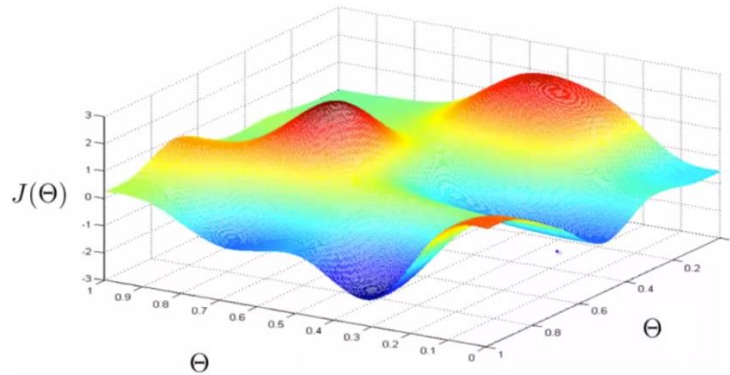
$\delta\xi(\theta^{(k)})$  : là đạo hàm của hàm loss function tại vị trí trọng số lớp thứ k-1.

$\theta^{(k+1)}$  : là giá trị trọng số lớp thứ k sau khi cập nhật.

$\theta^{(k)}$  : là giá trị trọng số lớp thứ k-1 trước khi cập nhật.

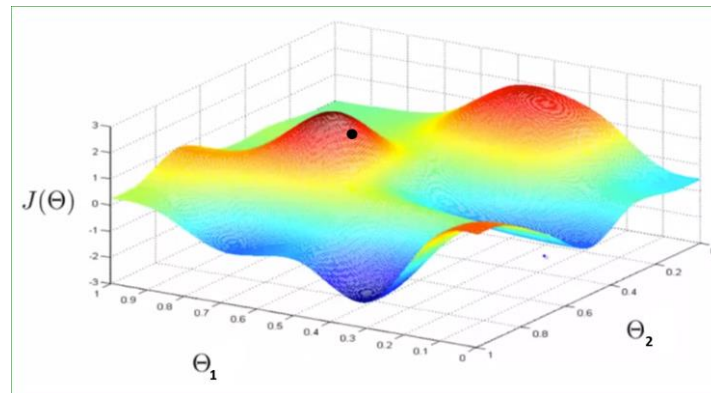
n: là chỉ số learning rate của bài toán quyết định tốc độ hội tụ của bài toán

Ví dụ sau thể hiện giá trị hội tụ của hàm mất mát sau n bước lặp



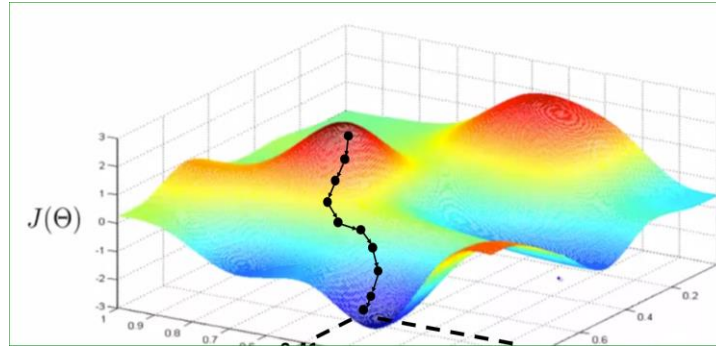
Hình 2-6-1: bài toán minh họa phương pháp gradient decent.

Sau vòng lặp đầu tiên vị trí  $\xi$  là hình 2-6-2



Hình 2-6-2: Giá trị đầu tiên của phương pháp gradient decent.

Vòng lặp sẽ tiếp tục cho đến khi  $\xi$  đạt giá trị cực trị.



Hình 2-6-3: Điểm tối ưu của phương pháp gradient decent.

### 2.6.3 Các thuật toán tối ưu nâng cao.

Ngoài thuật toán tối ưu gradient descent ra thì người ta còn sử dụng một số thuật toán tối ưu nâng cao khác như: Conjugate gradient, BFGS, L-BFGS.

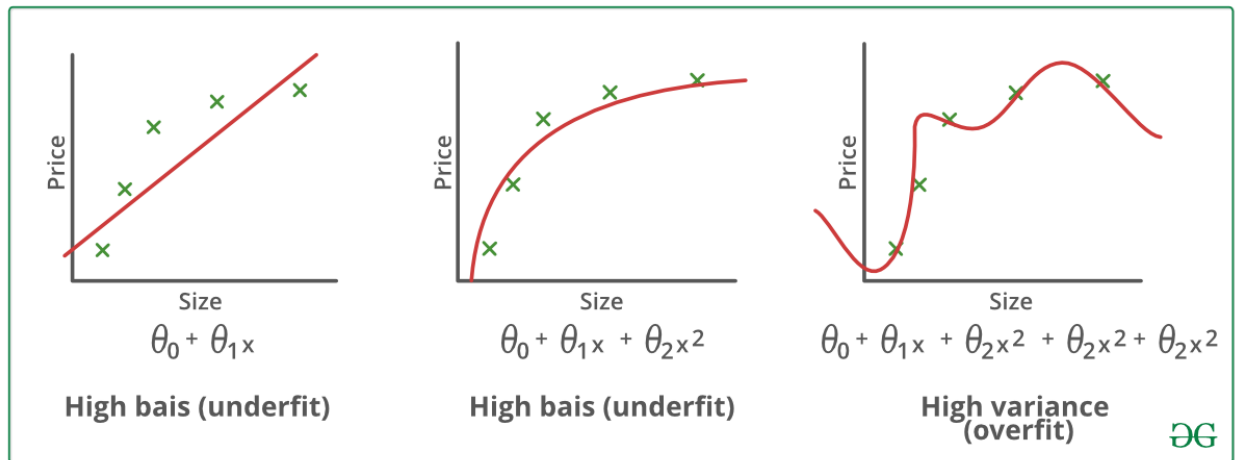
Ưu điểm của những thuật toán này là không cần quan tâm giá trị learning rate vì thuật toán có giải thuật thông minh để chọn ra giá trị learning tối ưu qua mỗi vòng lặp, thường nhanh hơn thuật toán gradient descent.

Nhược điểm là: thuật toán này quá phức tạp

Còn một ưu điểm nữa là các thuật toán này đã có sẵn trong thư viện của Matlab, Octave, python ta chỉ cần sử dụng mà không cần hiểu bên trong nó là gì. Nhưng ta phải đánh giá được trong những thư viện ta tìm kiếm được thì cái nào là tốt, cái nào là không tốt.

## 2.7 Vấn đề quá khớp (Overfitting) và không khớp (underfitting) và phương pháp giải quyết

### 2.7.1 Vấn đề quá khớp và không khớp



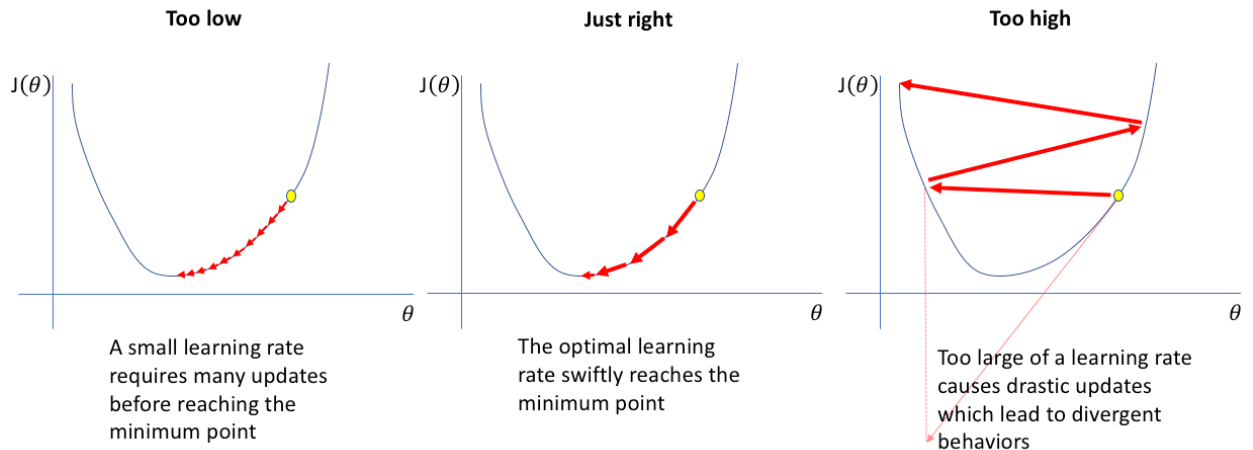
Hình 2-7-1: Hình minh họa vấn đề không khớp và quá khớp.

Nếu ta tìm được một hàm mất mát  $\xi$  với các hệ số  $\theta^{(1)}, \theta^{(2)}, \theta^{(3)}, \dots$  quá hợp với tập giữ liệu huấn luyện thì sẽ dẫn tới hiện tượng quá khớp bên phải. Đối với hiện tượng này dữ liệu thì dữ liệu sẽ không tốt khi ta thử với tập thử

Nếu các đặc tính của ta quá ít hay hàm mất mát  $\xi$  của chúng ta có bậc thấp hơn quá nhiều so với dữ liệu ta cần phân loại thì sẽ xảy ra hiện tượng không khớp (hay bias) bên trái thì dự đoán của ta về dữ liệu ra không còn chính xác.

## 2.7.2 Giải pháp

### 2.7.2.1 Chọn learning rate hợp lý.



Hình 2-7-2: Ví dụ minh họa việc lựa chọn các giá trị learning rate khác nhau.

Nếu ta chọn chỉ số learning rate quá nhỏ thì tốc độ hội tụ của thuật toán sẽ rất chậm như hình bên trái. Nếu ta chọn giá trị learning rate quá cao thì tốc độ hội tụ của thuật toán sẽ rất nhanh dẫn đến ló và thuật toán hoạt động không đúng hình bên phải.

Để chọn ra giá trị learning rate hợp lý:

- Người ta thường sử dụng một vòng lặp giá trị learning rate này từ 0.001-1 với mỗi bước là gấp 3 lần tức 0.001, 0.003, 0.009. Sau đó vẽ đồ thị learning rate và chọn giá trị lớn nhất hoặc gần với giá trị lớn nhất.

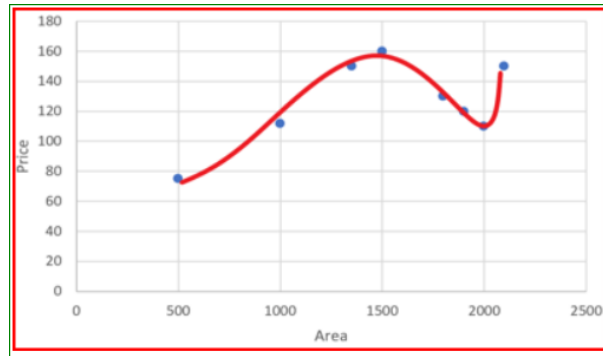
### 2.7.2.2 Chọn chỉ số regularization hợp lý.

Để tránh hiện tượng quá khớp thì có một số cách đó là giảm đặc trưng của dữ liệu (loại bỏ thủ công hoặc drop out). Ngoài ra người ta không mong muốn loại bỏ những đặc trưng này vì biết đâu được nó có một ý nghĩa nào đó trong quá trình thử hoặc huấn luyện một bộ phân loại khác. Để thực hiện nó người ta đưa ra một thuật toán là regularization.

## LUẬN VĂN TỐT NGHIỆP

Để hiểu về thuật toán này ta sẽ qua một ví dụ:

Đây là đồ thị hàm dự đoán của chúng ta sau khi đã huấn luyện và dễ thấy nó bị quá khớp.



Hình 2-7-3: Ví dụ minh họa một hàm huấn luyện đã bị quá khớp

Hàm giả thuyết của chúng ta là một hàm đa thức bậc 4 và ta mong muốn nó là đa thức bậc 2

$$f(x_i) = h_{\theta}x = \theta_0 + \theta_1x_1 + \theta_2x_2^2 + \theta_3x_3^3 + \theta_4x_4^4$$

$$\rightarrow f(x_i) = h_{\theta}x = \theta_0 + \theta_1x_1 + \theta_2x_2^2$$

Sau đây là phương pháp thực hiện

Giải xử bài toán có hàm mất mát  $\xi$

$$\xi = \sum_{i=1}^N \|y_i - h_{\theta}(x_i)\|^2$$

$$\text{Và } h_{\theta}x = \theta_0 + \theta_1x_1 + \theta_2x_2^2 + \theta_3x_3^3 + \theta_4x_4^4$$

Sau đó thêm một hệ số  $\lambda$  để phương trình trở thành:

$$\xi = \sum_{i=1}^N \|y_i - h_{\theta}(x_i)\|^2 + \lambda \sum_{i=1}^n \theta_i^2$$

Dễ thấy khi hệ số regularization( $\lambda$ ) lớn thì để cực tiểu hàm mất mát thì hệ số  $\theta_1, \theta_2, \theta_3, \theta_4$  sẽ giảm (theo quy ước thì người ta thường không thêm hệ số regularization vào  $\theta_0$ ) và khi đó hàm giả thuyết  $f(x_i)$  sẽ không còn là hàm đa thức bậc 4 nữa

$$h_{\theta}x = \theta_0$$

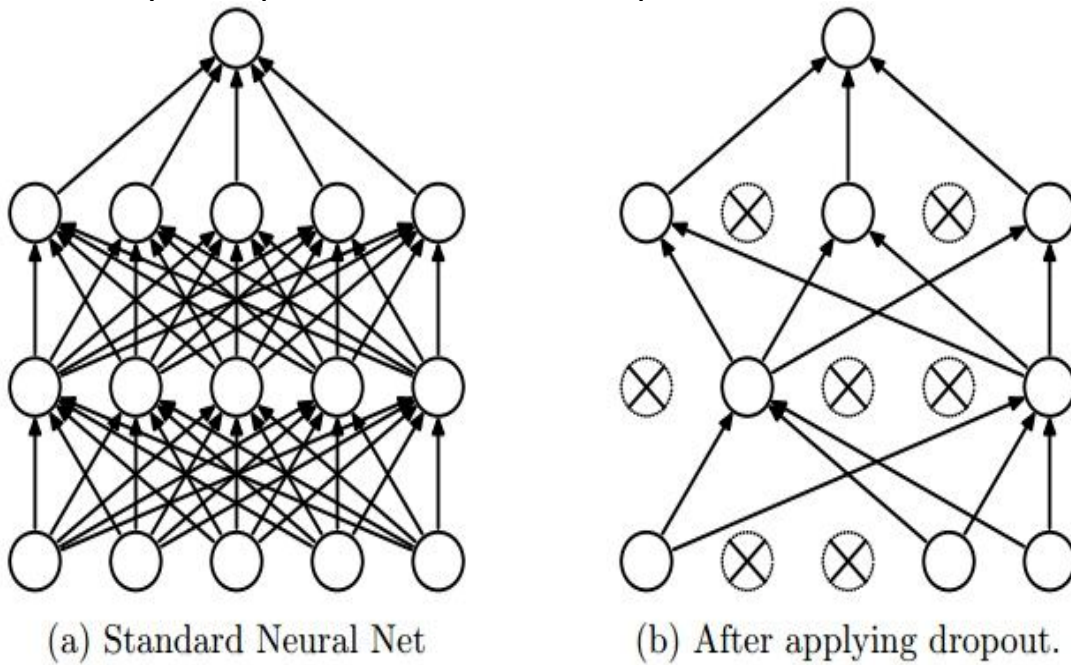
Và lựa chọn một giá trị regularization hợp lý thì hàm dự đoán của ta sẽ có kết quả chính xác hơn.

### 2.7.2.3 Drop-out

Kỹ thuật dropout là việc chúng ta sẽ bỏ qua một vài đơn vị trong suốt quá trình huấn luyện mô hình, những đơn vị bị bỏ qua được lựa chọn ngẫu nhiên.

Ở đây, chúng ta hiểu bỏ qua là đơn vị đó sẽ:

- Không tham gia và đóng góp vào quá trình huấn luyện (lan truyền tiến và lan truyền ngược).
- Về mặt kỹ thuật, tại mỗi giai đoạn huấn luyện, mỗi node có xác suất bị bỏ qua là  $1-p$  và xác suất được chọn là  $p$

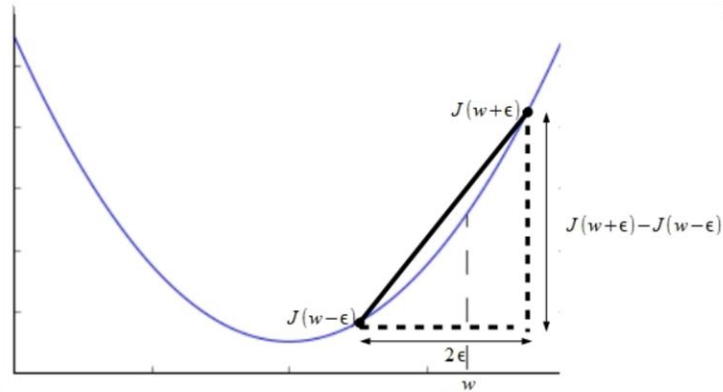


Hình 2-7-4: Ảnh minh họa một quá trình drop-out.

## 2.8 Một số phương pháp kiểm tra độ chính xác của mô hình mạng

### 2.8.1. Gradient checking.

Trong quá trình thực hiện thuật toán gradient descent thì làm thế nào ta kiểm tra được chương trình của chúng ta tính chính xác gradient tại  $\theta$  (đạo hàm của hàm cost function  $\xi$  theo  $\theta$ ).



Hình 2-8-1: Ảnh minh hoạ cho phương pháp gradient checking

Từ đồ thị của gradient ta có công thức:

$$\frac{\partial}{\partial \theta} \xi(\theta) \approx \frac{\xi(\theta + \varsigma) - \xi(\theta - \varsigma)}{2\varsigma} \quad (2.15)$$

Thuật toán chạy đúng khi vế trái xấp xỉ vế phải.

### 2.8.2 Đánh giá bằng chỉ số IoU (intersection of union)

Đối với nhiều thuật toán phát hiện vật thể sử dụng mạng RCNN, Faster RCNN, Mask RCNN, YOLO,... chúng ta sẽ vẽ được một hộp giới hạn (bounding box) xung quanh vật thể chúng ta muốn phát hiện trong một bức ảnh.

IoU được biết đến như một thức đo đánh giá khả năng dự đoán của bất kỳ thuật toán nào có đầu ra là hộp giới hạn (bounding box).

Để áp dụng IoU để đánh giá khả năng dự đoán của một thuật toán phát hiện đối tượng chúng ta cần:



## LUẬN VĂN TỐT NGHIỆP

Hộp giới hạn gốc (ground-truth bounding boxes): là hộp giới hạn mà chính tay ta tự gán nhãn cho đối tượng tham khảo chương 4

Hộp giới hạn dự đoán (predicted bounding boxes): là hộp giới hạn mà mô hình được sử dụng dự đoán được.

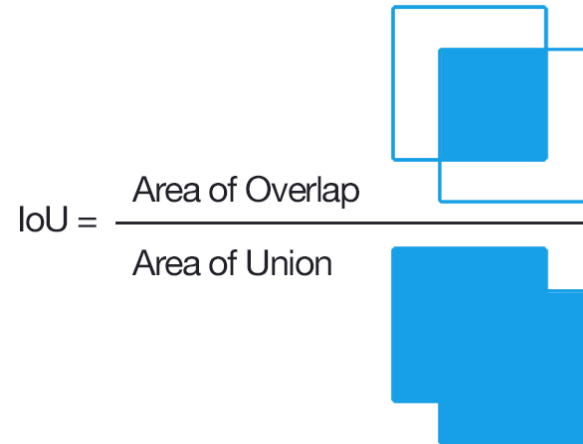
Công thức tính chỉ số IoU theo [3]:

$$IOU = \frac{A \cap B}{A \cup B} \quad (2.16)$$

Với A: là hộp giới hạn gốc, đối với thuật toán YOLO thì A gồm (x,y) tọa độ trung tâm hộp giới hạn; (w,h) chiều rộng và chiều cao của hộp giới hạn

Với B: là hộp giới hạn dự đoán, đối với thuật toán YOLO thì A gồm (x,y) tọa độ trung tâm hộp giới hạn; (w,h) chiều rộng và chiều cao của hộp giới hạn.

Thông thường chỉ số IOU > 0.5 thì mô hình được xem như là dự đoán tốt



Hình 2-8-2: Minh hoạ tính toán chỉ số IoU

### 2.8.3 Đánh giá bằng chỉ số F1 score.

Để hiểu hơn về Fscore ta sẽ tìm hiểu một số khái niệm

True Positive (TP)	Thuật toán phát hiện được đối tượng đúng phân lớp
False Positive (FP)	Thuật toán phát hiện đối tượng nhưng sai phân lớp
True Negative (TN)	Thuật toán không phát hiện được đối tượng và không có đối tượng trong khung hình

False Negative (FN)	Thuật toán không phát hiện được đối tượng nhưng có đối tượng trong khung hình
---------------------	-------------------------------------------------------------------------------

Bảng 1: Định nghĩa về TP, FP, TN, FN

+ Giá trị dự đoán (Precision)

$$P = \frac{\text{True positive}}{\text{predicted positive}} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}} \quad (2.17)$$

+ Giá trị gọi lại (Recall)

$$R = \frac{\text{True positive}}{\text{Actualy positive}} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}} \quad (2.18)$$

+ F1 Score (Fcore)

$$\text{F1 Score} = \frac{2 * P * R}{(P + R)} \quad (2.19)$$

Giá trị F1 Score hoàn hảo sẽ bằng 1 và nếu phải lựa chọn giữa các giá trị đánh giá F1 Score chúng ta nên chọn chỉ số F1 Score có giá trị cao nhất.

## 2.9 Kết luận chương 2

Trong chương 2 luận văn đã nêu ra một số khái niệm cơ bản về học máy và học sâu. Cũng như đưa ra hướng giải quyết cho một bài toán học cơ bản từ đó có thể hình dung những bài toán lớn hơn. Trong chương 2 cũng trình bày một số cách đánh giá mô hình mạng sẽ được đánh giá tại chương 5.

## CHƯƠNG 3. MÔ HÌNH MẠNG YOLOv4 VÀ QUY TRÌNH HUẤN LUYỆN MÔ HÌNH MẠNG

Cách đây bốn năm vào tháng 9 năm 2016, YOLO [2] phiên bản đầu tiên khi được công bố với tốc độ xử lý 45 frame per second (FPS) xử lý ảnh với thời gian thực đã gây ra tiếng vang lớn và áp đảo các thuật toán nhận diện đối tượng cùng thời như Deformable parts models (DPM), R-CNN. Cùng năm vào tháng 12 năm 2016 YOLO lại ra phiên bản hai với một số thay đổi về kiến trúc dẫn đến một sự đánh đổi nhỏ về tốc độ (từ 45 FPS – xuống 40 FPS) YOLOv2 đã tăng độ chính xác từ 63.4 mAP lên 78.6 mAP so với YOLOv1 đánh giá trên tập VOC2007-VOC2012, thêm vào đó YOLO đã nhận diện được trên 9000 đối tượng với cái tên khác là YOLO9000 [3]. Đến tháng 4 năm 2018 YOLOv3 đã ra đời với một vài thay đổi từ mạng Darknet-19 đến mạng Darknet-53 đã tăng độ chính xác của mạng từ 21.6-33.0 AP [4]. Trong năm nay 2020 vào tháng 4 YOLO đã phát hành phiên bản thứ tư với nhận định là tăng 10% AP độ chính xác và 12% FPS so với YOLOv3 [5]. Cũng trong năm nay chúng ta cũng nghe về YOLOv5 sẽ được phát hành vào cuối năm. Rõ ràng công nghệ đang phát triển với tốc độ nhanh chóng từng ngày và trong luận văn lần này em sử dụng mạng YOLOv4 phiên bản công khai mới nhất hiện nay để thực hiện đề tài.

### 3.1 Lý thuyết thuật toán YOLOv4 (you look only one).

Trước khi tìm hiểu về thuật toán YOLOv4 chúng ta sẽ đi tìm hiểu về phiên bản đầu tiên của thuật toán này là thuật toán YOLO

Thuật toán YOLO (you look only once): đây là một thuật toán phát hiện đối tượng với tốc độ nhanh 45 FPS chạy trên thời gian thực và 155 FPS với phiên bản Fast YOLO. Để làm được điều đó YOLO đã có bước đột phá về phương thức khi thay vì quét lần lượt trong ảnh theo phương pháp truyền thống thì YOLO chia một ảnh các lưới (grid), mỗi lưới có chức năng dự đoán năm hộp (box). Mỗi hộp sẽ có

chức năng phát hiện một đối tượng. Các đối tượng chồng lên nhau sẽ được xử lý bằng thuật toán non-maximal suppression (NMS), chỉ giữ lại những hộp có xác suất cao nhất.

YOLO được công bố vào năm 2016, hiện nay đã có năm phiên bản được công bố chính thức là YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv5.

Tại thời điểm thực hiện đề tài phiên bản YOLOv5 còn chưa chính thức phát hành nên phiên bản mới nhất YOLOv4 sẽ được lựa chọn để thực hiện mô hình của luận văn.

Một vài lý do luận văn sử dụng thuật toán YOLOv4 để sử dụng:

- Đề tài luận văn được lấy cảm hứng từ bài báo [7], sử dụng mạng YOLOv3 để phát hiện thiết bị bảo hộ lao động là mũ và áo bảo hộ. Trong đề tài này luận văn cải tiến mạng được sử dụng là YOLOv4 và tăng thêm đối tượng cần phát hiện là găng tay bảo hộ.
- Phiên bản: YOLOv4 ra đời với ba phiên bản chính là YOLOv4, YOLOv4-tiny với 2 lớp mạng YOLO ở cuối, YOLOv4-tiny\_3L với 3 lớp mạng YOLO ở cuối. Các phiên bản YOLOv4-tiny có thể thực hiện trên các máy tính không có đồ họa với tốc độ xử lý thực.
- Về tốc độ và độ chính xác: theo [5] thì YOLOv4 tăng 10% độ chính xác và 12% tốc độ so với YOLOv3.
- Huấn luyện mạng: thư viện Darknet[6] hỗ trợ các phương pháp huấn luyện mạng YOLOv4 trên google colab được thực hiện thuận tiện và phổ biến.

### **3.2 Huấn luyện thuật toán YOLOv4 nhận diện thiết bị bảo hộ lao động.**

YOLOv4-tiny gồm 2 mạng YOLO ở cuối cùng dùng để phát hiện những đối tượng vừa và lớn so với khung hình.

YOLOv4-tiny\_3l gồm 3 mạng YOLO ở cuối cùng dùng để phát hiện cả 3 đối tượng khác nhau là lớn, vừa và nhỏ.

Tuỳ theo đối tượng cần nhận dạng mà luận văn sẽ sử dụng từng mạng khác nhau, và luận văn sẽ sử dụng cả hai phiên bản thu nhỏ của YOLOv4 trong đề tài lần này.

Để huấn luyện mạng YOLOv4 cho ứng dụng nhận diện đối tượng ta qua hai quá trình lớn:

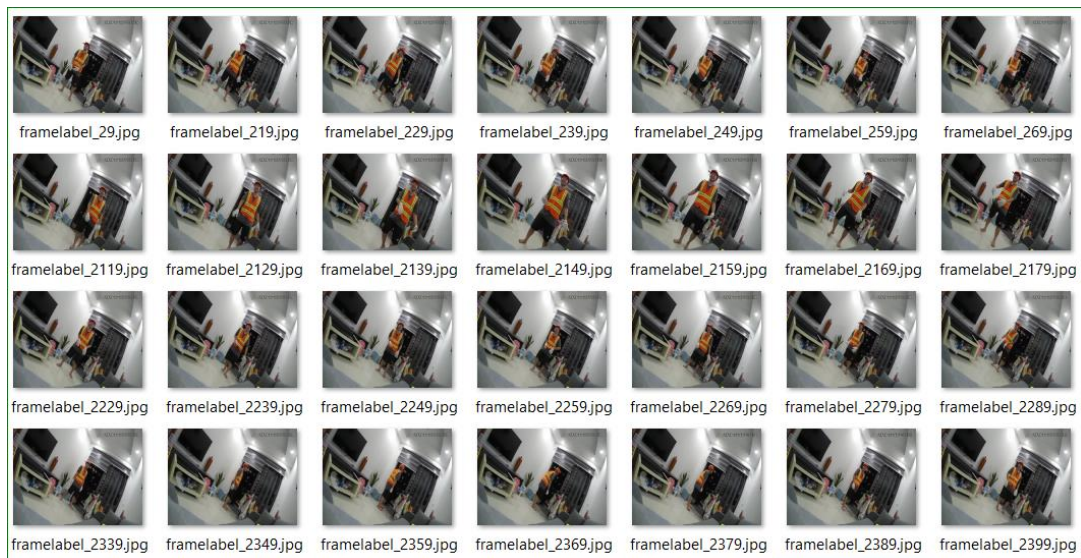
- Thu thập dữ liệu và xử lý dữ liệu
- Huấn luyện mạng YOLOv4 trên google colab

### 3.2.1 Thu thập dữ liệu và xử lý dữ liệu

#### 3.2.1.1 Thu thập dữ liệu

Dữ liệu mà luận văn sử dụng chủ yếu là dữ liệu tự thu thập được và phát triển trong quá trình kiểm thử để tìm ra bộ dữ liệu phù hợp nhất cho đề tài luận văn

Dữ liệu chủ yếu của luận văn là các video được quay từ điện thoại di động, webcam và camera. Trong đó để tăng thêm tập dữ liệu luận văn đã sử dụng dữ liệu được lấy từ internet nhưng với tỉ lệ đảm bảo là 80:20



Hình 3-2-1: Dữ liệu cần thu thập trong quá trình thực hiện luận văn

Trong hình 3-2-1 dữ liệu được quay bằng camera sau đó sẽ được cắt ra thành nhiều khung ảnh riêng biệt. Đối với các dữ liệu này luận văn sẽ thêm một bước là thay đổi tên nhằm giúp tập dữ liệu trở nên rõ ràng hơn.

### 3.2.1.2. Xử lý dữ liệu.

Sau khi thu thập dữ liệu thì nên qua một bước xử lý để có được một bộ dữ liệu tốt cho quá trình huấn luyện

Có ba bước cho quá trình này là:

- Cắt khung ảnh từ video
- Đổi tên
- Dán nhãn cho ảnh

#### *Bước 1: Lấy khung ảnh từ video*

Bình thường thì video chạy với 30 FPS tức là có 30 ảnh mỗi 1 giây, chúng ta có thể sử dụng hàm VideoCapture(), Read() và imwrite() trong thư viện OpenCv để lấy ảnh từ frame.

```
cam = cv2.VideoCapture("demo3.mp4")
```

```
while (ret == True):
```

```
    ret, frame = cam.read()
```

```
    cv2.imwrite(name, frame)
```

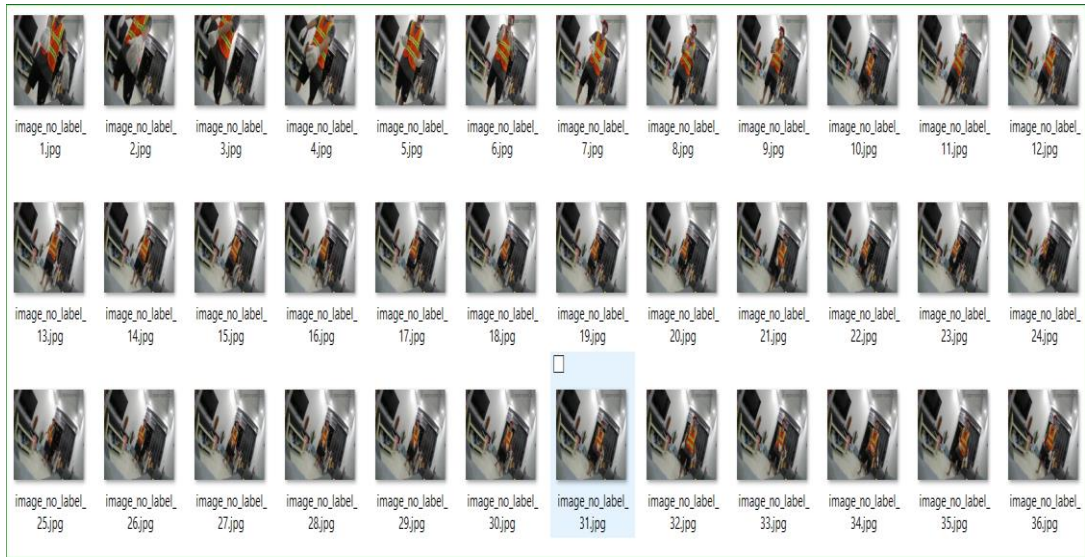
#### *Bước 2: Đổi tên ảnh*

Trong quá trình thu thập dữ liệu chúng ta có thể có một lượng lớn dữ liệu từ nhiều nguồn khác nhau và có tên không đồng đều. Vì vậy cần thay đổi tên của bộ dữ liệu giúp quản lý dữ liệu dễ dàng hơn, để làm được điều đó ta sử dụng hai thư viện chính là OpenCv và glob

```
for file in glob.glob(r"F:\folder_path/*.jpg"):
```

```
    image = cv2.imread(file, 1)
```

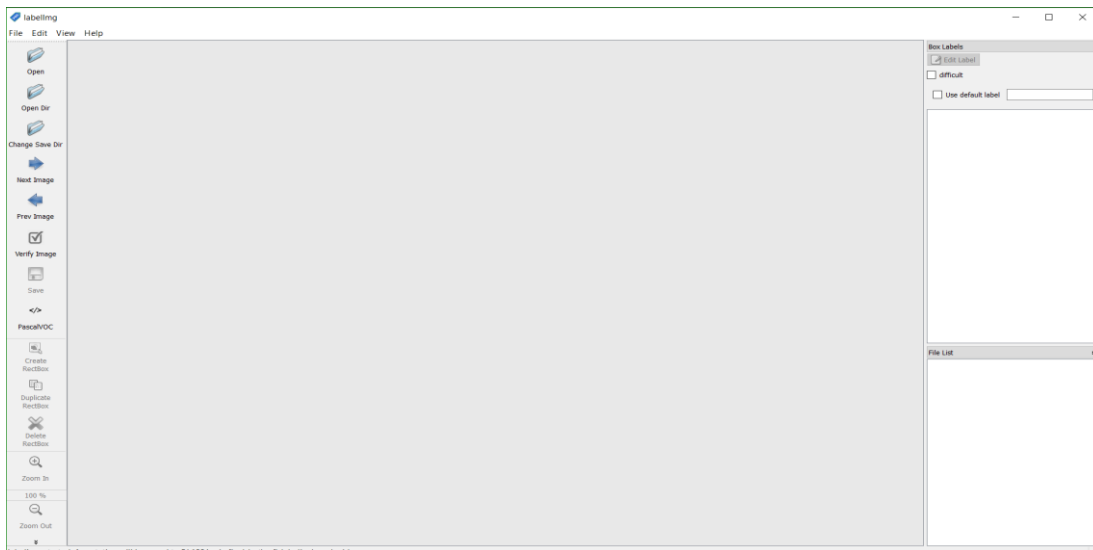
```
    cv2.imwrite(Folder_save/name_{}.jpg'.format(a), image)
```



Hình 3-2-2: Ảnh minh sau khi đã đổi tên

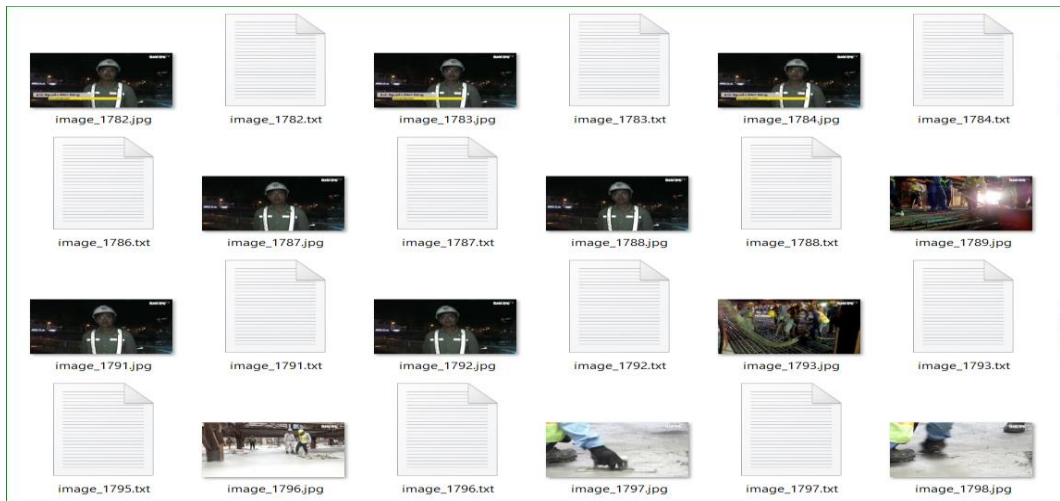
### Bước 3: Dán nhãn cho ảnh

Sau khi xong bước 1 và 2 chúng ta đã có tập dữ liệu phục vụ cho quá trình huấn luyện ở bước tiếp theo này luận văn sẽ gián nhãn cho từng ảnh với một phần mềm hỗ trợ labelling.

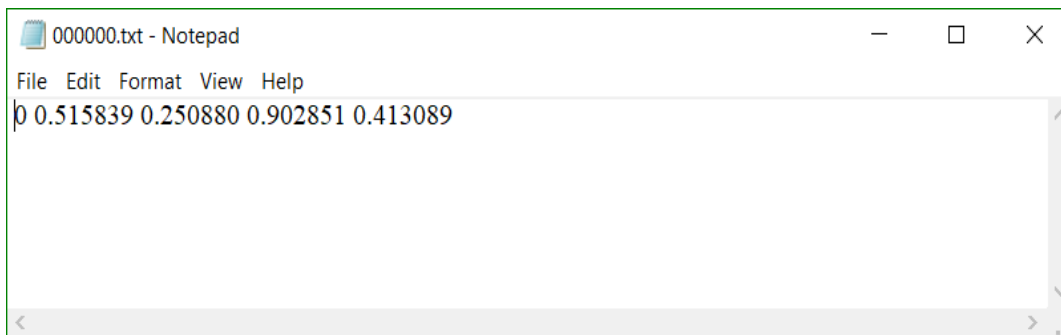


Hình 3-2-3: giao diện ứng dụng labeling

Dữ liệu sau khi labelling sẽ có dạng như hình 4-2-3 gồm một file jpg chứa ảnh có đối tượng và một file text chứa nhãn và vị trí của đối tượng trong ảnh.



Hình 3-2-4: Ảnh minh họa của tập dữ liệu sau khi labelling



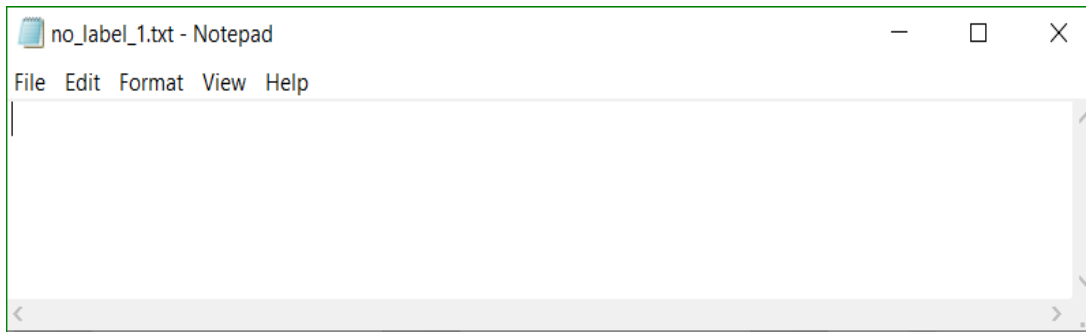
Hình 3-2-5: Nội dung file txt gồm: Classes, tọa độ (x,y), chiều rộng w, chiều dài h.

\* Một lưu ý: Mô hình mạng thần kinh có khả năng học được các đối tượng đã được dán nhãn và những lớp không được dán nhãn nên chúng ta cần một tập dữ liệu không chứa đối tượng nhận dạng, điều này sẽ giúp giảm xác suất nhận dạng sai của mạng

Để tạo đối tượng không gián nhãn chúng ta sử dụng hàm glob để truy xuất ảnh trong thư mục và sử dụng lệnh open,close file hỗ trợ bởi thư viện python.

```
for file in glob.glob(r"F:\100.Lesis_final_last\sefl-labeling\name/*.jpg") :  
    f = open("text/image_no_label_{}.txt".format(a),"w+")  
    f.close()
```





Hình 3-2-6: Ảnh minh họa đối với những dữ liệu không chứa đối tượng

Sau bước này ta đã có toàn bộ dữ liệu huấn luyện, từ đó đóng gói vào file zip và gửi lên google driver để thực hiện huấn luyện.

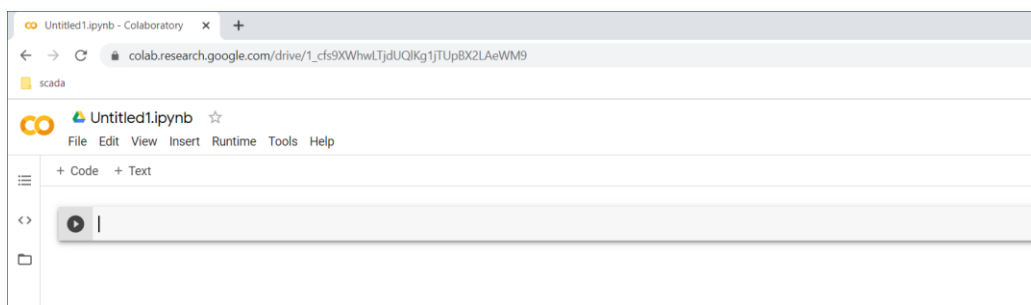
### 3.2.2 Huấn luyện mạng trên google colab.

Sau các bước chuẩn bị dữ liệu và thông số huấn luyện, chúng ta tiến hành huấn luyện trên Google Colab.

Google Colab là một môi trường gồm nhiều máy tính ảo được hỗ trợ bởi google cho phép người sử dụng miễn phí đồ họa như Tesla T4. Với thời gian mỗi tài khoản chỉ có thể sử dụng 12/24 giờ liên tục.

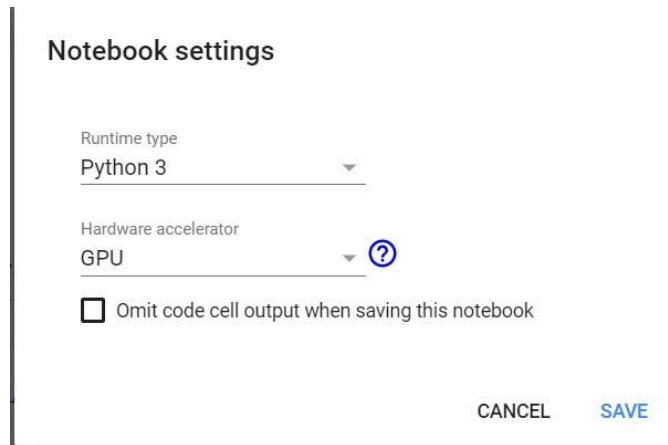
Để huấn luyện mạng YOLOv4 trên colab luận văn chia thành 10 bước cơ bản sau:

Bước 1: Truy cập vào url: “<https://colab.research.google.com>”. Đăng nhập tài khoản Google, sau đó chọn New Python 3 notebook. Một notebook sẽ được tạo ra có giao diện như hình 3-2-7.



Hình 3-2-7: Giao diện trình biên dịch ipynb trên Google colab.

Bước 2: Để sử dụng GPU của Google Colab, chúng ta cần thực hiện việc thay đổi runtime type. Chọn Edit ☐ Notebook settings ☐ Hardware accelerator ☐ GPU ☐ Save.



Hình 3-2-8: Cài đặt để được sử dụng GPU của Google Colab

Bước 3: cần kết nối Google Colab với Google Drive để truy cập vào dữ liệu huấn luyện đã được tải lên Google Drive. Sử dụng đoạn code dưới đây truy cập vào Google Drive từ Google Colab.

```
] # Linking between virtual computer and your drive
from google.colab import drive
drive.mount('/content/gdrive')
!ln -s /content/gdrive/My\ Drive/ /mydrive
!ls /mydrive
```

Hình 3-2-9: Kết nối Google Drive và Google Colab.

Bước 4: Tải thư viện darknet từ github và cấu hình các thư viện như OPENCV, GPU, CUDNN trên mạng máy ảo.

```
# Clone library darknet with github
!git clone https://github.com/AlexeyAB/darknet

# change makefile to have GPU and OPENCV enabled
%cd darknet
!sed -i 's/OPENCV=0/OPENCV=1/' Makefile
!sed -i 's/GPU=0/GPU=1/' Makefile
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile
!sed -i 's/CUDNN_HALF=0/CUDNN_HALF=1/' Makefile
!make

[ ] !/usr/local/cuda/bin/nvcc --version
```

Hình 3-2-10: Ảnh minh hoạt quá trình tải và cấu hình một số thư viện trong darknet.

Bước 5: Mã hoá tất cả các dữ liệu huấn luyện vào thư mục obj

```
[ ] # unzip file database.zip from your drive to folder data
!unzip /mydrive/thesis_safewear_YOLOv4/database_big.zip -d data/obj
```

Hình 3-2-11: Mã hoá tất cả các dữ liệu huấn luyện vào thư mục obj

Bước 6: Sao chép kiến trúc mạng đã qua cấu hình

```
# upload the custom .cfg back to cloud VM from Google Drive
!cp /mydrive/thesis_safewear_YOLOv4/yolov4-obj.cfg ./cfg
```

Hình 3-2-12: Ảnh minh hoạ quá trình tải sao chép kiến trúc mạng yolov4.cfg

Bước 7: Tạo file obj.names và obj.data gồm tên các đối tượng nhận dạng và số phân lớp của chúng.

```
# create file obj.name anad obj.data in folder data
!echo -e 'hard_hat \ngloves \nsafe_vest' > data/obj.names
!echo -e 'classes= 3\ntrain = data/train.txt\nvalid = data/test.txt\nnames = data/obj.names\nbackup = /mydrive/thesis_safewear_YOLOv4' > data/obj.data
!mkdir data/obj
```

Hình 3-2-13: Tạo file obj.data gồm số đối tượng cần nhận diện và số lớp

Bước 8: Tạo hai tập tin train.txt và test.txt là số đối tượng được huấn luyện và số đối tượng được đánh giá dựa vào tập kiểm thử, thông thường tỉ lệ là 80/20 hoặc 100/100.

```
# upload the generate_train.py and generate_test.py script to cloud VM from Google Drive
!cp /mydrive/luanvan_project_v3/generate_train.py ./
!cp /mydrive/luanvan_project_v3/generate_test.py ./

!mkdir "data/test"

!python generate_train.py
!python generate_test.py
```

Hình 3-2-14: Ảnh minh họa quá trình tạo tập tin train.txt và test.txt

Bước 9: Tải file pre-training về máy ảo

Tùy từng mô hình mạng mà ta có từng bộ pre-training khác nhau đối với YOLOv4 là yolov4.conv.137, đối với YOLOv4-tiny và YOLOv4-tiny\_3L là yolov4.conv.29

```
[ ] # download architecture yolov4.conv.137
!wget https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.conv.137
```

Hình 3-2-15: Ảnh minh họa tải tập tin yolov4.conv.137 – bộ pre-training của yolov4

Bước 10: bắt đầu quá trình huấn luyện

```
!./darknet detector train data/obj.data cfg/yolov4-obj.cfg yolov4.conv.137 -dont_show -map
```

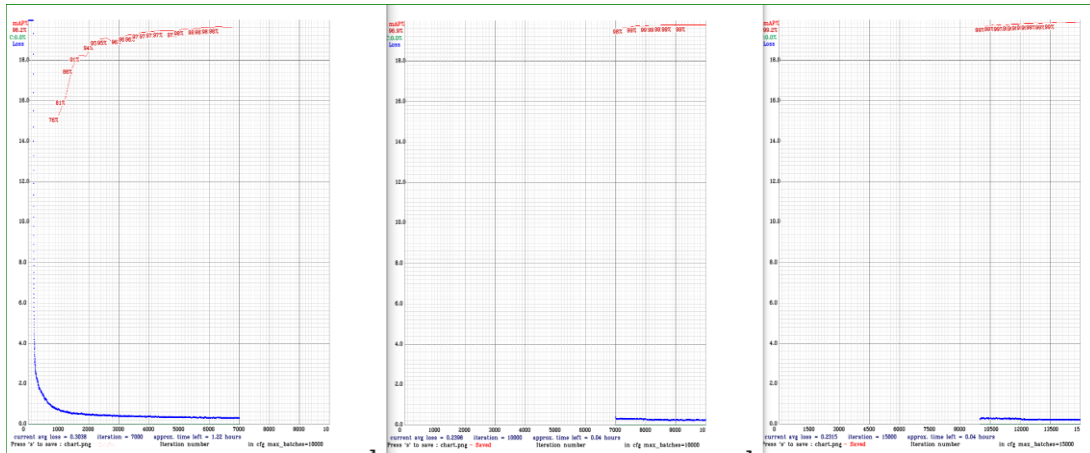
Hình 3-2-16: Huấn luyện với bộ dữ liệu pre-training

Dòng mã trên được sử dụng để huấn luyện mạng YOLO khi ta mới bắt đầu. Nhưng vì việc sử dụng máy ảo trên google colab là không ổn định nên ta có thể huấn luyện từ bộ trọng số cuối cùng đã huấn luyện với dòng mã bên dưới

```
!./darknet detector train data/obj.data cfg/yolov4-obj.cfg /mydrive/thesis_safewear_YOLOv4//yolov4-obj_last.weights -dont_show -map
```

Hình 3-2-17: Huấn luyện YOLO từ phiên bản mới nhất của trọng số

Sau khi đã huấn luyện ta sử dụng hàm `imshow('chart.png')` để xem đồ thị learning curve.



Hình 3-2-18: Ảnh minh họa đồ thị learning Curve

Trong biểu đồ trên dòng màu xanh thể hiện hàm sai số đang hội tụ về không đã thể hiện trong chương 2, ý nghĩa thuật toán đã hội tụ tại một điểm. Dòng màu đỏ là phép đánh giá trên tập kiểm thử test.txt, trong hình Darknet sử dụng phương pháp mAP, trong luận văn này sinh viên sẽ thực hiện việc đánh giá theo phương pháp khác ít phức tạp hơn.

Một chú ý khi sử dụng Google Colab:

Google Colab sẽ tự động ngắt kết nối nếu người dùng không có hoạt động trong 30 phút. Nên sau khi bắt đầu huấn luyện mô hình, chúng ta cần thêm đoạn mã sau Google Colab để tự động kết nối sau mỗi 30 phút.

```
function ClickConnect(){ console.log("Working");  
document.querySelector("colab-toolbar-button#connect").click() }  
setInterval(ClickConnect,60000)
```

### 3.2.3 Cấu hình tập tin cấu trúc mạng YOLOv4

Về cơ bản việc cấu hình tập tin cấu trúc mạng trong yolov4.cfg, yolov4-tiny.cfg, yolov4-tiny\_3l.cfg là như nhau. Vì phạm vi của đề tài chưa có khả năng thay đổi kiến trúc mạng.

Learning rate = 0.001

Batch = 64 : đây là số ảnh load vào RAM mỗi lần huấn luyện

Subdivisions = 16

width=416 là kích thước đầu vào của ảnh -chiều rộng

height=416 là kích thước đầu vào của ảnh -chiều ngang

burn\_in = 1000: số lượng steps ban đầu được giữ sao cho learning rate rất bé.

Giá trị này tăng từ 0 đến learning rate. Sau đó learning rate được giữ ổn định

Max-batches = 2000\*classes là số bước lặp đối đa của mô hình huấn luyện. Để có giá trị tốt thì số này nên >2000\*classes

Steps = 80%-90%\*max-batches: là các vị trí mà learning rate sẽ giảm dần, lúc này thuật toán đã hội tụ nên ta không cần learning rate lớn

- Filter = (classes + 5)\*3 tại các lớp convolution trước lớp yolo, hàm kích hoạt tại các lớp này là linear.

Classes = 6 số đối tượng cần huấn luyện

### 3.2.4 Kết quả huấn luyện

Kết quả bộ trọng số sau khi huấn luyện sẽ được cập nhật thường xuyên và lưu lại dưới dạng:

yolov4-last.weights: tập tin chứa trọng số được huấn luyện gần nhất, giá trị sẽ tự động được cập nhật sau 100 vòng lặp, chúng ta có thể sử dụng giá trị này cho lần huấn luyện tiếp theo

yolov4-best.weights: tập tin chứa giá trị trọng số có chỉ số đánh giá mAP (mean Advantage Precision) cao nhất, sẽ được sử dụng trong bước dự đoán

### **3.3 Kết luận chương 3.**

Qua chương 3 luận văn đã giới thiệu về các phiên bản của thuật toán YOLOv4, lý do luận văn chọn thuật toán YOLOv4 để thực hiện đề tài và quá trình huấn luyện thuật toán YOLOv4 trên Google Colab mà luận văn đã thực hiện. Quá trình này sẽ được trình bày lại trong chương 4 để huấn luyện các đối tượng của đề tài nhưng sẽ được lược bỏ phần huấn luyện trên google colab nhằm giảm những chi tiết không cần thiết.

## CHƯƠNG 4. PHÁT HIỆN THIẾT BỊ BẢO HỘ LAO ĐỘNG CÁ NHÂN

Trong chương 3 luận văn đã giới thiệu thuật toán phát hiện đối tượng YOLOv4 cũng như cách thức hoạt động thuật toán này. Trong chương 4 luận văn sẽ kết hợp thuật toán để trình bày phương pháp giải bài toán mà luận văn đề ra là nhận dạng thiết bị bảo hộ lao động cá nhân.

### 4.1 Đánh giá các phương pháp đã thực hiện.

Trong quá trình thực hiện luận văn đã tham khảo bài báo [6,7], từ đó luận văn đã thực hiện 3 phương pháp để giải quyết bài toán nhận diện thiết bị bảo hộ lao động cá nhân. Thông qua những đánh giá khách quan về những phương pháp luận văn tập trung vào phương pháp giải quyết được bài toán của đề tài theo đúng lý thuyết.

Các phương pháp đã thực hiện:

*Phương pháp 1:* Sử dụng thuật toán YOLOv4 được huấn luyện với 3 đối tượng là mũ bảo hộ, áo bảo hộ và găng tay bảo hộ.

Ý tưởng đơn giản của phương pháp này là huấn luyện mạng YOLOv4-tiny với 3 đối tượng là mũ, áo bảo hộ, nón bảo hộ.

Sau đó cho ảnh đầu vào qua mô hình mạng YOLOv4 sẽ ra kết quả như sau:





Hình 4-1-1: Kết quả phương pháp 1 phát hiện thiết bị bảo hộ lao động

Nhận xét:

+ Ưu điểm:

- Số đối tượng cần phát hiện là 3 dẫn thu thập dữ liệu và huấn luyện mạng dễ dàng hơn
- Kết quả nhận dạng được đối tượng đã huấn luyện cao
- Chỉ sử dụng một mạng nên tốc độ xử lý cao

+ Nhược điểm:

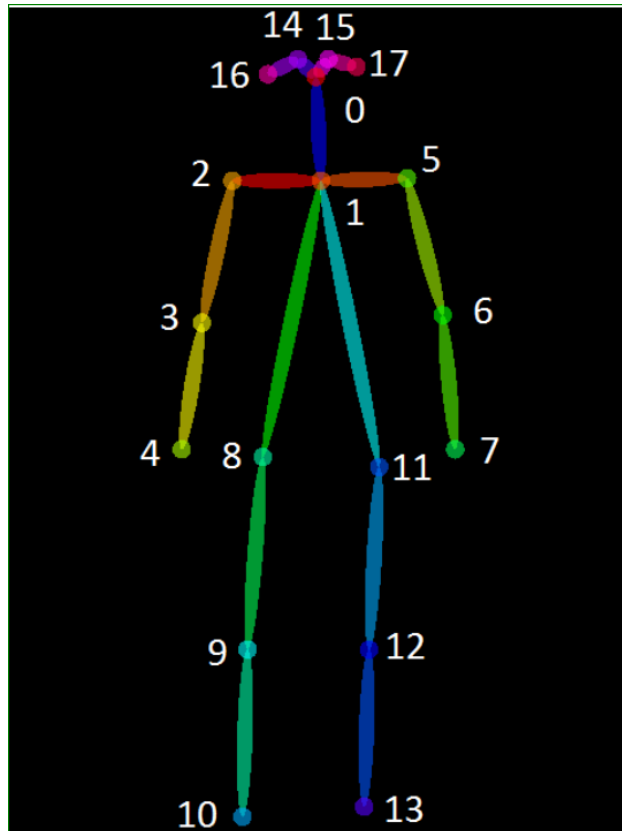
- Nhược điểm quan trọng nhất dẫn đến luận văn không tập trung vào phương pháp này là không trả lời câu hỏi chính xác của luận văn là nhận dạng thiết bị bảo hộ lao động cá nhân với điều kiện là có người lao động đang trang bị.

*Phương pháp 2:* Nhận dạng thiết bị bảo hộ lao động cá nhân sử dụng kết hợp mạng phát hiện khung xương (Openpose) [8] và mô hình YOLOv4 được huấn luyện với 3 đối tượng

Để cải tiến phương pháp 1 đã nêu vì không xác định được người có mang thiết bị bảo hộ lao động nên em đề xuất một mạng xác định khung xương người (Openpose). Các khung xương mô hình mạng có thể phát hiện được thể hiện qua bảng 4-1-1 và hình 4-1-2

Số thứ tự	Vị trí	Số thứ tự	Vị trí
0	Mũi	9	Đầu gối phải
1	Cổ	10	Cổ chân phải
2	Vai phải	11	Hông trái
3	Khuỷ tay phải	12	Đầu gối trái
4	Cổ tay phải	13	Cổ chân trái
5	Vai trái	14	Mắt phải
6	Khuỷ trái	15	Mắt trái
7	Cổ tay trái	16	Tai phải
8	Hông phải	17	Tai trái

Bảng 4-1-1: Các khớp xương thuật toán Openpose có thể phát hiện  
POSE OUTPUT FORMAT (COCO).



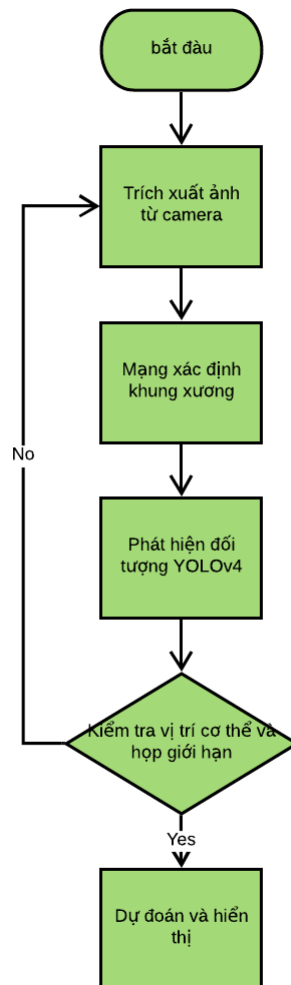
Hình 4-1-2: Ảnh minh họa đầu ra mạng OPENPOSE

Theo hình 4-1-2 thì mạng Openpose có thể phát hiện 18 vị trí trên cơ thể người nhưng chúng ta không cần tất cả, chỉ cần 4 vị trí để đánh giá việc người lao động có đang mặc thiết bị bảo hộ hay không là:

- Nose- 0 : mũi -- đại diện cho vị trí nón bảo hộ.
- Neck-1 : Cổ -- đại diện cho vị trí áo bảo hộ.
- Rwrist- 4: cổ tay phải - đại diện cho găng tay trái.
- Lwrist-7: cổ tay trái - đại diện cho găng tay phải.

Từ những tọa độ vị trí của điểm ta đã phát hiện so sánh với tọa độ của mũ, áo, găng tay ta sẽ có kết quả nhận diện tốt.

4.1.1. Sơ đồ khối phương pháp 2



Hình 4-1-3: Sơ đồ mô hình nhận dạng 3 lớp YOLOv4 kết hợp mạng Openpose

Các bước thực hiện:

Bước 1: Ảnh sẽ lấy từ camera

Bước 2: Đưa ảnh qua mạng openpose nhằm phát hiện toạ độ các điểm như mũi, cổ, cổ tay trái, cổ tay phải.

Bước 3: Ảnh sau khi qua mạng Openpose sẽ được tiếp tục xử lý với mạng YOLOv4 để phát hiện đối tượng cần nhận dạng

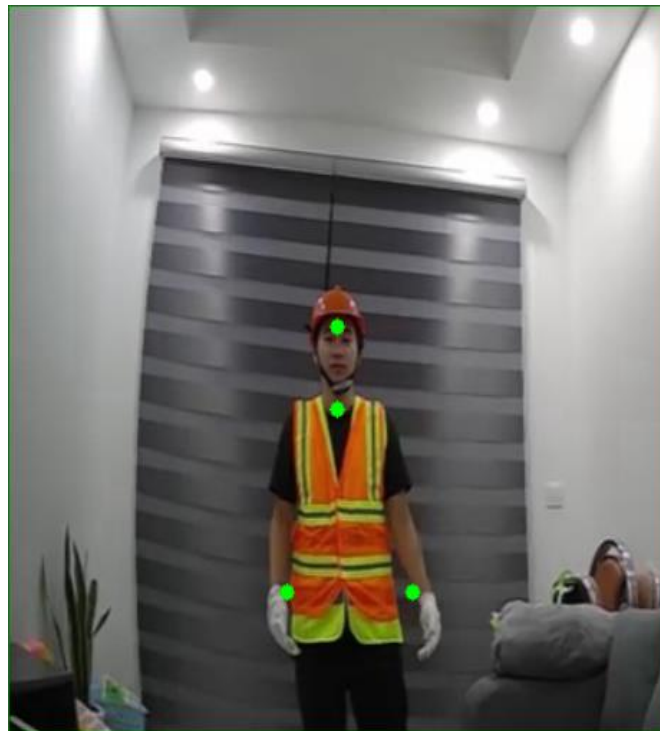
Bước 4: Thuật toán YOLOv4 sẽ xác định đối tượng dưới dạng các hộp giới hạn dưới dạng  $(x,y,w,h)$  gồm vị trí trung tâm tọa độ  $(x,y)$ , chiều rộng và chiều ngang  $(w,h)$

Ta sẽ chuyển các hộp giới hạn này về tọa độ  $(x1,x2,y1,y2)$  và kiểm tra:

- Nếu tọa độ của khung xương là đầu thuộc hộp giới hạn là mũ thì có thể kết luận người lao động đội mũ hoặc không.
- Nếu tọa độ của khung xương là áo thuộc hộp giới hạn là áo bảo hộ thì có thể kết luận người lao động mang áo bảo hộ hay không.
- Nếu tọa độ của khung xương là áo thuộc hộp giới hạn là găng tay bảo hộ thì có thể kết luận người lao động mang găng tay bảo hộ hay không.

#### *4.1.2. Kết quả thực hiện.*

*Kết quả thực hiện khi sử dụng mạng Openpose:*



Hình 4-1-4: Kết quả sau khi qua mạng OPENPOSE xác định 4 vị trí mũi, cổ, cổ tay trái, cổ tay phải.

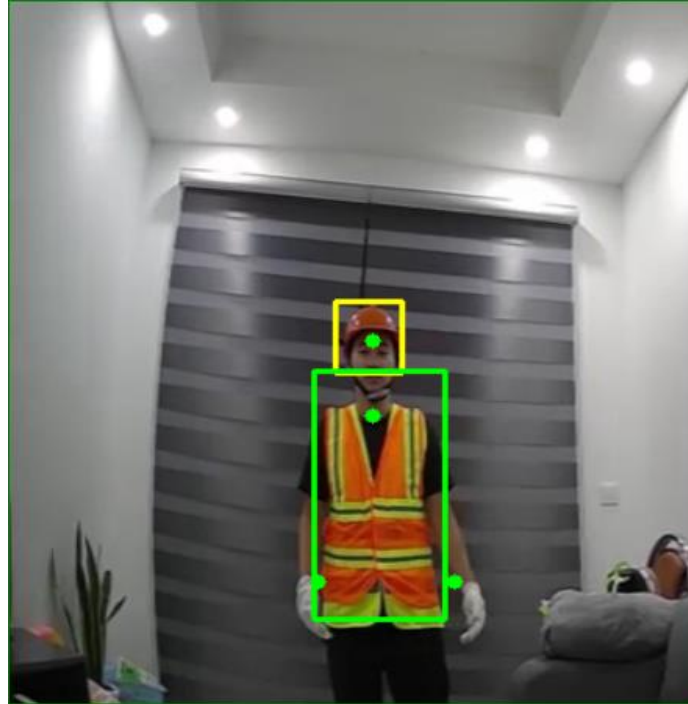
Trong hình 4-1-4 luận văn đã trích xuất được tọa độ 4 vị trí cần thiết để phục vụ cho bước tiếp theo là có các thiết bị bảo hộ lao động tại những vị trí này không.

## LUẬN VĂN TỐT NGHIỆP

Toạ độ 4 điểm trong hình 4-1-4 là:

[(194, 271), (194, 313), (236, 406), (166, 406)]

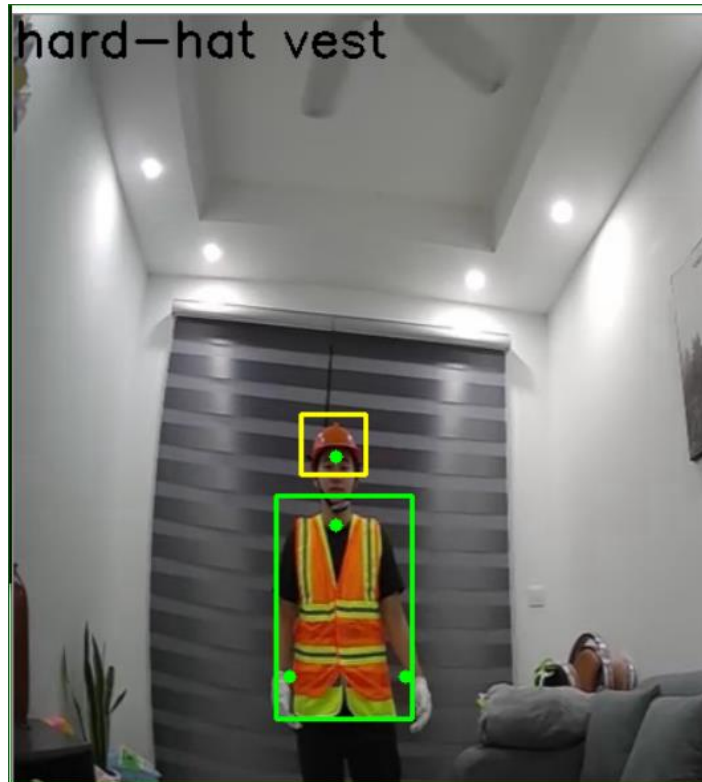
Kết quả thực hiện sau khi qua mạng YOLOv4



Hình 4-1-5: Kết quả thực hiện theo phương pháp 2 sau khi qua mạng YOLOv4

Qua hình 4-1-5 ta thấy được mạng YOLOv4 chỉ phát hiện được hai đối tượng là nón bảo hộ lao động và áo bảo hộ lao động nhưng không thể phát hiện được đối tượng là găng tay bảo hộ lao động.

Sau bước này chúng ta sẽ xác định được hộp giới hạn của áo bảo hộ lao động và nón bảo hộ lao động. Trong bước tiếp theo chúng ta sẽ đưa ra kết luận rằng người lao động có mang theo thiết bị bảo hộ hay không.



Hình 4-1-6: Kết quả thực hiện theo phương pháp 2

Qua kết quả hình 4-1-6 ta kết luận người lao động có mang theo mũ và áo bảo hộ

Đánh giá mô hình nhận dạng

+ Ưu điểm:

Phần nào đó cải thiện được câu trả lời của luận văn là người lao động có đội nón bảo hộ, người lao động có mặc áo bảo hộ

+ Nhược điểm:

Trong quá trình di chuyển rất khó để xác định chính xác vị trí hai cổ tay và việc mạng YOLOv4 luận văn huấn luyện không thể phát hiện được đôi găng tay bảo hộ trong trường hợp này, dẫn đến không thể dự đoán được kết quả.

Các trường hợp đặt mũ trước đầu, đang giữ áo bảo hộ trước ngực hay đang giữ găng tay thuật toán sẽ thể nhận dạng sai.

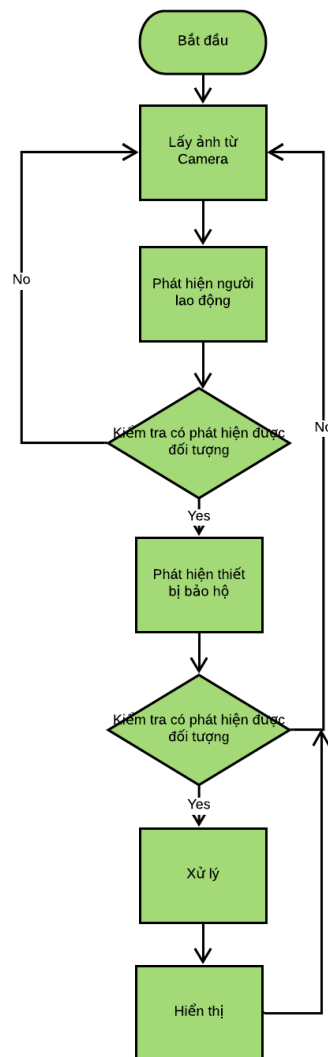
Sai số của quá trình dự đoán là sai số của cả hai mạng YOLOv4-tiny và Openpose dẫn đến khó kiểm soát kết quả dự đoán.

Vì những khuyến điểm khi sử dụng mạng YOLOv4 được huấn luyện với 3 đối tượng (nón, áo, găng tay) không thể giải quyết được do đó luận văn đã xác định lại số đối tượng cần phát hiện và cho ra kết quả tốt hơn.

## 4.2 Phương pháp 3 nhận dạng thiết bị bảo hộ lao động cá nhân với YOLOv4 được huấn luyện với 6 đối tượng

### 4.2.1 Sơ đồ giải thuật theo phương pháp 3

Lưu đồ giải thuật theo phương pháp 3





Hình 4-2-1: Sơ đồ giải thuật phát hiện thiết bị bảo hộ lao động cá nhân.

Thuật toán phát hiện thiết bị bảo hộ lao động sẽ được thực hiện qua 7 bước:

Bước 1: Ảnh sẽ được trích xuất từ camera

Bước 2: Ảnh sau khi trích xuất sẽ được qua mô hình mạng phát hiện người để vẽ hộp giới hạn của đối tượng, và các hộp giới hạn sẽ được lưu lại dưới dạng ma trận  $D(x,y,w,h)$

Bước 3: Để tăng tốc độ xử lý cho mô hình

- Nếu trong khung hình không phát hiện được đối tượng là người thuật toán sẽ quay lại bước 1.
- Nếu phát hiện đối tượng là người thuật toán sẽ tiếp tục đến bước 4.

Bước 4: Ảnh sẽ tiếp tục qua một bộ phát hiện đối tượng với số đối tượng cần phát hiện là 6 bao gồm (đối tượng có mang mũ bảo hộ, đối tượng có mang áo bảo hộ, đối tượng có mang găng tay bảo hộ, đối tượng không mang mũ bảo hộ, đối tượng không mang áo bảo hộ, đối tượng không mang găng tay bảo hộ). Tương tự các hộp giới hạn của bộ phát hiện đối tượng của sẽ được lưu lại dưới dạng ma trận  $T(x,y,w,h)$

Bước 5: Kiểm tra nếu ở bước 4 không xác định được đối tượng cần nhận dạng quay lại bước 1, nếu phát hiện được đối tượng trong  $T$  thuật toán tiếp tục bước 6

Bước 6: Luận văn sẽ chuyển hộp giới hạn  $D(x,y,w,h)$  thành  $D(x1,x2,y1,y2)$  cùng với vị trí  $(x,y)$  của các đối tượng trong  $T$  ta kết luận:

Nếu tọa độ các điểm trong  $T(x,y)$  thuộc về hộp giới hạn  $D(x1,x2,y1,y2)$  thì các đối tượng  $T(x,y)$  được chấp nhận.

Nếu tọa độ các điểm trong  $T(x,y)$  không thuộc về hộp giới hạn  $D(x1,x2,y1,y2)$  thì các đối tượng  $T(x,y)$  được loại bỏ.

Bước 7: Hiển thị ra màn hình kết quả dự đoán và quay lại bước 1.

#### 4.2.2 Chương trình nhận dạng thiết bị bảo hộ lao động cá nhân

Để thực hiện mô phỏng cho đề tài, luận văn sử dụng thư viện OpenCV

OpenCV được bắt đầu tại Intel vào năm 1999 bởi Gary Bradsky, và bản phát hành đầu tiên ra mắt vào năm 2000. Vadim Pisarevsky cùng với Gary Bradsky quản lý nhóm OpenCV phần mềm của Intel. Năm 2005, OpenCV được sử dụng trên Stanley, chiếc xe đã giành chiến thắng trong cuộc thi DARPA Grand Challenge năm 2005. Sau đó, sự phát triển tích cực của nó tiếp tục dưới sự hỗ trợ của Willow Garage với Gary Bradsky và Vadim Pisarevsky dẫn đầu dự án. OpenCV hiện hỗ trợ vô số thuật toán liên quan đến thị giác máy tính và học máy và đang mở rộng từng ngày.

OpenCV hỗ trợ nhiều loại ngôn ngữ lập trình như C ++, Python, Java, v.v. và có sẵn trên các nền tảng khác nhau bao gồm Windows, Linux, OS X, Android và iOS. Các giao diện cho hoạt động GPU tốc độ cao dựa trên CUDA và OpenCL cũng đang được phát triển tích cực.

OpenCV-Python là API Python cho OpenCV, kết hợp những phẩm chất tốt nhất của API OpenCV C ++ và ngôn ngữ Python.

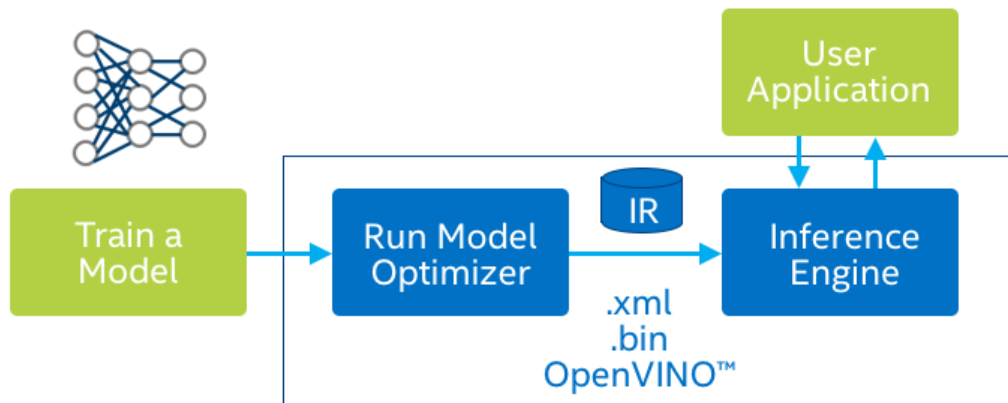
OpenCV-Python là một thư viện liên kết Python được thiết kế để giải quyết các vấn đề về thị giác máy tính và học sâu.



Hình 4-2-2: Ảnh minh họa biểu tượng OpenCV

Một thư viện hỗ trợ khác được luận văn sử dụng là thư viện hỗ trợ OpenVino, đây là thư viện được xây dựng bởi Intel nhằm hỗ trợ chạy các thuật toán về mạng

thần kinh trong các máy không chip đồ hoạ NVIDIA (không hỗ trợ bởi CUDA) như các máy tính chạy chip đồ hoạ Graphis hay Rasperry Pi.



Hình 4-2-3: Thư viện hỗ trợ đồ hoạ được phát triển bởi Intel - OpenVINO

Một số chức năng của OpenVino như:

- Hỗ trợ tăng tốc độ huấn luyện mô hình
- Hỗ trợ tối ưu hoá mô hình với tệp tin định dạng mã nhị phân và mã xml
- Tạo môi trường tăng tốc độ xử lý đối với các chương trình chạy OpenCV

Sau khi đã phân tích các thư viện chính luận văn đã sử dụng, phần tiếp theo luận văn sẽ trình bày vào chi tiết các quá trình trong sơ đồ giải thuật

#### 4.2.2.1 Đầu vào

Đầu vào của thuật toán là ảnh đã được trích xuất từ camera sử dụng thư viện OpenCV.

```
vid = cv2.VideoCapture(0)
```

Sau đó chúng ta sẽ trích xuất ảnh từ camera

```
while True:
```

```
    _, frame = vid.read()
```

4.2.2.2 Tiếp theo ảnh sẽ được đưa qua bộ phát hiện người được huấn luyện bởi mạng YOLOv4-tiny

## LUẬN VĂN TỐT NGHIỆP

**Bộ dữ liệu:** được huấn luyện gồm 1000 ảnh trong đó có 500 ảnh chứa đối tượng là người và 500 ảnh không chứa đối tượng. Đây là tập dữ liệu luận văn đã thu thập được từ thực tế và tự dán nhãn cho dữ liệu.

**Cài đặt thông số cho đối tượng:**

Learning rate = 0.001

Batch = 64

Subdivisions = 16

width=416

height=416

burn\_in = 1000

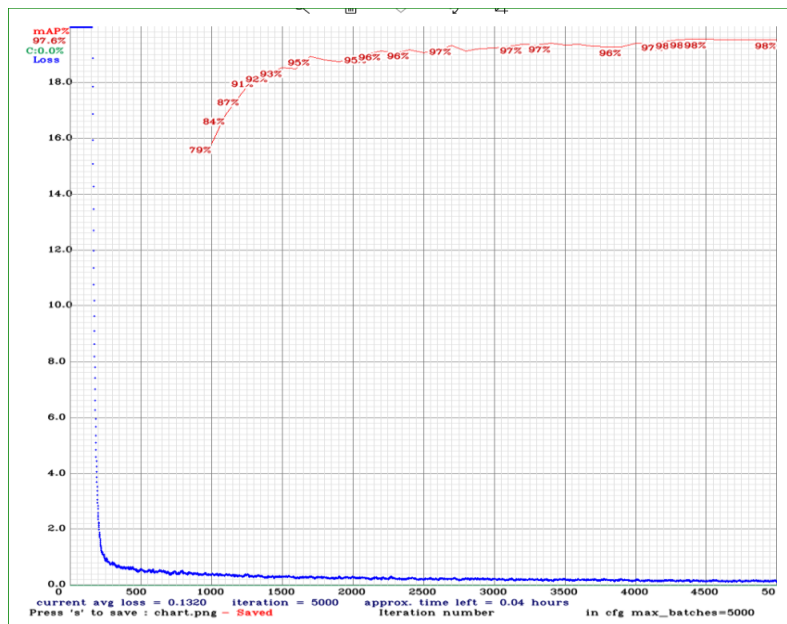
**Max-batches = 5000:** theo công thức đã đề cập trong chương 3 thì số vòng lặp là 2000 vòng nhưng chúng ta có thể tăng số vòng lặp để tăng kết quả hội tụ

Steps = 4000,4500

Filter = 18

Classes = 1

**Kết quả huấn luyện:**



Hình 4-2-4: Biểu đồ learning Curve kết quả huấn luyện YOLOv4-tiny đối tượng là người.

Từ biểu đồ ta thấy mô hình mạng bắt đầu hội tụ tại vòng lặp thứ 2000 và thực sự hội tụ tại vòng lặp 4000. Đúng với lý thuyết luận văn đã đề ra.

Khi huấn luyện với Darknet dòng màu đỏ thể hiện phép đánh giá mAP để đánh giá độ chính xác của tập huấn luyện so với tập thử. Vì tính phức tạp khi lập trình nên luận văn sẽ đánh giá mô hình mạng theo cách khác sẽ trình bày trong chương 5.

### *Kết quả dự đoán:*

Sau khi mạng YOLOv4-tiny được huấn luyện thành công trên Google Colab sẽ xuất ra hai tệp tin thể hiện kiến trúc mạng và trọng số là YOLOv4-tiny.cfg và YOLOv4-tiny.weights.

Luận văn sử dụng thư viện hỗ trợ OpenCv chạy thuật toán Python để chạy chương trình phát hiện đối tượng người.



Hình 4-2-5: Kết quả dự đoán sau khi qua mạng YOLOv4-tiny được huấn luyện phát hiện người

Kết quả dự đoán trong hình 4-2-4 ta thấy mô hình YOLOv4-tiny luận văn huấn luyện đã phát hiện chính xác người trong khung ảnh.

### 4.2.2.3 Kiểm tra

Đối tượng sau bước 4.2.2.2 sẽ được lưu dưới dạng ma trận  $T$ , thuật toán kiểm tra đơn giản như sau:

*if  $T.any()$ :*

*continue*

*else:*

*break*

#### 4.2.2.4 Phát hiện thiết bị bảo hộ lao động

*Bộ dữ liệu:* được huấn luyện gồm 4100 ảnh trong đó có 3000 ảnh chứa 6 đối tượng gồm: cá nhân mang mũ bảo hộ, cá nhân không mang mũ bảo hộ, cá nhân mang áo bảo hộ, cá nhân không mang áo bảo hộ, cá nhân đeo găng tay bảo hộ và cá nhân không đeo găng tay bảo hộ. Ảnh không chứa đối tượng gồm 1100 giúp mô hình có thể nhận dạng những đối tượng không mong muốn và loại bỏ. Đây là tập dữ liệu luận văn đã thu thập được từ thực tế và tự dán nhãn cho dữ liệu.

*Cài đặt thông số cho đối tượng:*

Learning rate = 0.001

Batch = 64

Subdivisions = 16

width=416

height=416

burn\_in = 1000

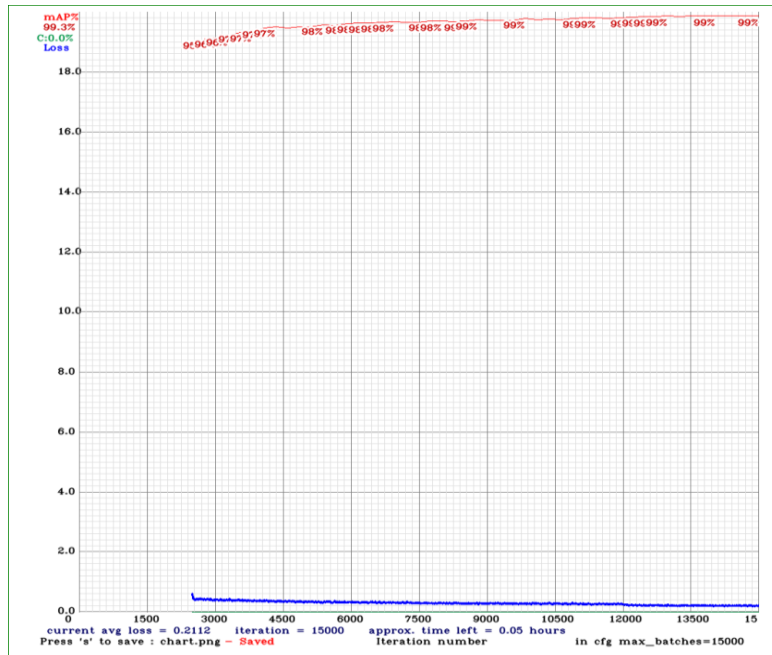
Max-batches = 15000: theo công thức đã đề cập trong chương 3 thì số vòng lặp là 12000 vòng nhưng chúng ta có thể tăng số vòng lặp để tăng kết quả hội tụ của mô hình mạng.

Steps = 12000,13500

Filter = 33

Classes = 6

*Kết quả huấn luyện:*



Hình 4-2-6: Biểu đồ learning Curve kết quả huấn luyện YOLOv4-tiny\_3l gồm 6 đối tượng

Từ biểu đồ hình 4-2-5 ta thấy mô hình mạng bắt đầu hội tụ tại vòng lặp thứ 12000 và thực sự hội tụ tại vòng lặp 13500. Đúng với lý thuyết luận văn đã đề ra.

Khi huấn luyện với Darknet dòng màu đỏ thể hiện phép đánh giá mAP để đánh giá độ chính xác của tập huấn luyện so với tập thử. Vì tính phức tạp khi lập trình nên luận văn sẽ đánh giá mô hình mạng theo cách khác sẽ trình bày trong chương 5.

Kết quả dự đoán:



Hình 4-2-6: Kết quả dự đoán của bộ phát hiện đối tượng là thiết bị bảo hộ

Nhìn vào kết quả dự đoán hình 4-2-6 mô hình mạng YOLOv4-tiny\_3l có thể dự đoán được đối tượng tốt.

#### 4.2.2.5 Kiểm tra

Đối tượng sau bước 4.2.2.4 sẽ được lưu dưới dạng ma trận D, thuật toán kiểm tra đơn giản như sau

```
if D.any():  
    continue  
else:  
    break
```

#### 4.2.2.6 Xử lý

Trong phần 4.2.2.2 và 4.2.2.4 chúng ta đã xác nhận được các hộp giới hạn của tất cả đối tượng cần phát hiện là D và T

Để xác định là người bảo hộ mang thiết bị bảo hộ lao động luận văn sẽ qua các bước:

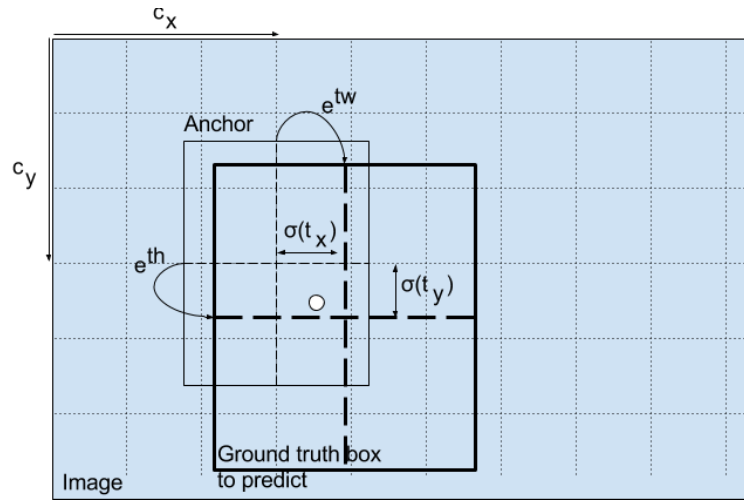


Bước 1: Sau khi qua hai bộ nhận dạng ta sẽ có hộp giới hạn của mạng YOLO phát hiện người sẽ có dạng một mảng D gồm

$$D = (x, y, w, h) \quad (4.1)$$

$x, y$ : tọa độ trung tâm của đối tượng

$w, h$ : chiều ngang và chiều cao của đối tượng cần nhận dạng



Hình 4-2-7: Ảnh minh họa một hộp giới hạn

Mảng  $D(x, y, w, h)$  sẽ được chuyển sang dạng  $D(x1, x2, y1, y2)$  với công thức

$$\begin{aligned} x1 &= x - \frac{w}{2} \\ x2 &= x + \frac{w}{2} \\ y1 &= y - \frac{h}{2} \\ y2 &= y + \frac{h}{2} \end{aligned} \quad (4.2)$$

Bước 2: Ma trận  $T(x, y, w, h)$  sẽ được trích xuất tọa độ  $x, y$  và thực hiện kiểm tra

Một đối tượng  $T$  được xem là thuộc bộ phận cơ thể người khi tọa độ trung tâm của chính nó  $(x, y)$  nằm trong hộp giới hạn của bộ phát hiện người  $D$ .

*if  $x \geq x1$  and  $x \leq x2$  and  $y \geq y1$  and  $y \leq y2$ :*

*return True*

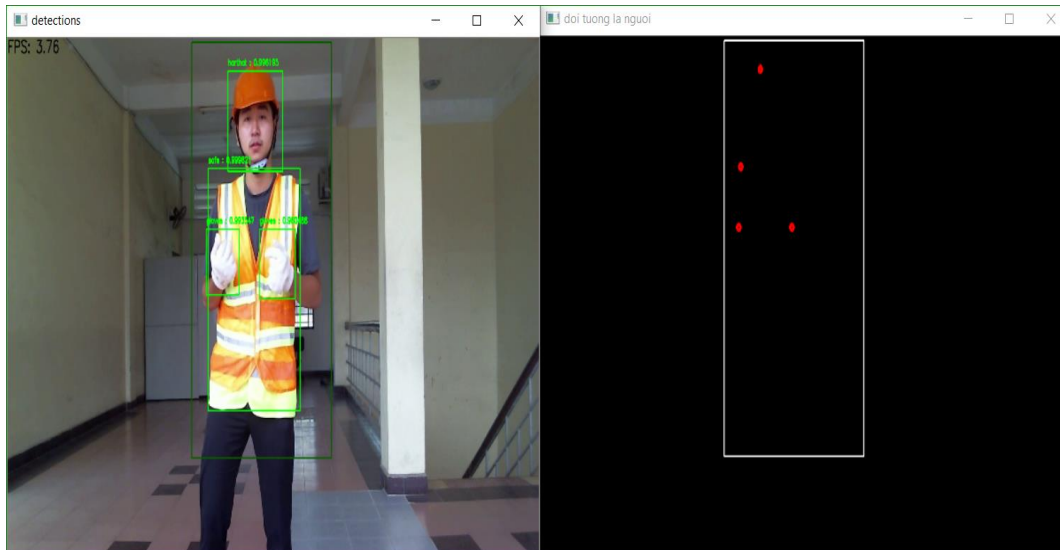
*else:*

*return false*

Nếu True thì đối tượng đang được mang bởi người lao động và hiển thị.

Nếu False đối tượng chỉ là thiết bị bảo hộ và không được hiển thị.

#### 4.2.2.7 Hiển thị kết quả nhận dạng



Hình 4-2-8: Kết quả mô hình phát hiện thiết bị bảo hộ lao động

Hình 4-2-8 thể hiện mô hình mạng YOLOv4-tiny\_3l đã phát hiện được người mang thiết bị bảo hộ lao động, nhưng đôi khi bộ phát hiện thiết bị bảo hộ vẫn còn bị mất một số đối tượng.

### 4.3 Kết thúc chương 4

Trong chương 4 luận văn đã đưa ra đánh giá các phương pháp mà luận văn đã thực hiện gồm sử dụng phát hiện thiết bị bảo hộ lao động sử dụng mạng YOLOv4 được huấn luyện trên 3 đối tượng, một bản nâng cấp của phương pháp này sử dụng mạng Openpose để xác định vị trí khung xương của đối tượng cần phát hiện và cuối cùng là phương pháp mà luận văn đã sử dụng là sử dụng mạng YOLOv4 phát hiện 7 đối tượng. Từ những phương pháp nêu ra luận văn khẳng định phương pháp luận văn đưa ra là đúng và có thể được phát triển hơn.

## CHƯƠNG 5. KẾT QUẢ THỰC HIỆN VÀ ĐÁNH GIÁ MÔ HÌNH

Nội dung của chương 5 sẽ trình bày kết quả thực hiện của phương pháp 3 mà luận văn đề xuất trong chương 4 trong các điều kiện thực hiện khác nhau. Đồng thời đưa ra đánh giá khách quan về kết quả luận văn đã thực hiện được.

### 5.1 Kết quả thực hiện

Thuật toán sau khi được xây dựng hoàn thiện, các thử nghiệm được tiến hành để đánh giá độ chính xác, khả năng nhận diện trong các điều kiện khác nhau của mô hình. Ứng dụng đánh giá được chạy trên máy tính có cấu hình phần cứng như sau:

Tên máy	: DELL - INSPIRON 13 -7359
Tên Chip	: Intel(R) Core™ i7-6500 CPU @ 2.50GHz – 2.60GHz
Card đồ hoạ	: Intel(R) HD Graphics 520 không hỗ trợ với CUDA
Hệ điều hành	: Windows 10 Home
Camera	: Logitech 525

Để đánh giá kết quả thực hiện của thuật toán luận văn sử dụng một không gian rộng rãi thoáng mát là tầng 5 toà C4 trường Đại học Bách Khoa Thành Phố Hồ Chí Minh.

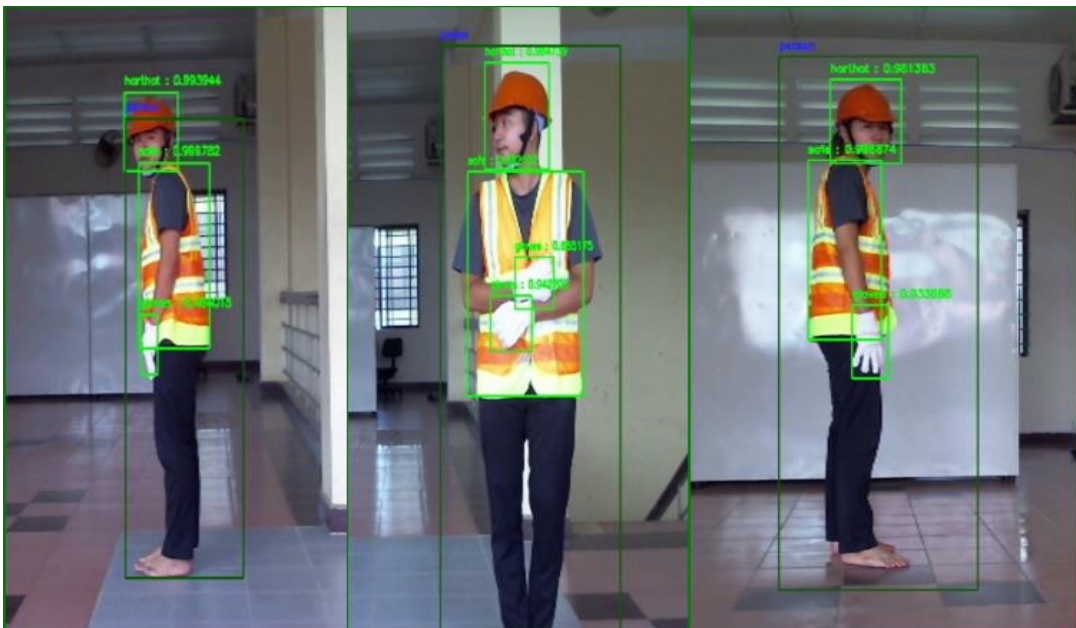
Kết quả nhận diện trong trường hợp đối tượng chỉ cầm nhưng không mặc thiết bị bảo hộ



Hình 5-1-1: Kết quả thuật toán phát hiện đối tượng đang giữ mà không mang thiết bị bảo hộ

Mô hình có thể phát hiện khi đối tượng là người lao động đang giữ thiết bị bảo hộ lao động thay vì mặc. Đây là trường hợp lỗi mắc phải trong phương pháp 1 và 2 mà luận văn đã đề ra. Dẫn đến phương pháp này phần nào là một phiên bản hoàn thiện hơn trong phương pháp 1 và 2.

Kết quả nhận dạng với các góc độ khác nhau



Hình 5-1-2: Kết quả thuật toán phát hiện đối tượng nhiều góc độ khi đứng  
Mô hình được huấn luyện để có thể phát hiện đối tượng ở nhiều tư thế khác nhau và góc độ khác nhau như nhìn phía trước, quay bên trái, quay bên phải.



Hình 5-1-3: Kết quả thuật toán phát hiện đối tượng nhiều góc độ khi ngồi  
Từ hình 5-1-2 và 5-1-3 mô hình thuật toán luận văn đề xuất đã có thể phát hiện được những hành động cơ bản nhất của người lao động, từ nó đáp ứng mục tiêu tối thiểu của yêu cầu luận văn là có thể phát hiện người mang thiết bị bảo hộ lao động.  
Kết quả nhận dạng trong phạm vi từ 3 - 10m



Hình 5-1-4: Kết quả thuật toán phát hiện đối tượng trong khoảng cách 3-10m

Trong hình 5-1-4 ta thấy rằng mô hình phát tốt trong khoảng cách từ 2.5 đến 6m, đặc biệt ở bị trí 6m thuật toán có khả năng phát hiện tốt hơn cả phạm vi 2.5m. Điều này được lý giải là tập dữ liệu huấn luyện hoạt động tốt hơn ở khoảng cách 4-6(m). Đối với khoảng cách 7m trở lên mô hình không có khả năng phát hiện.

Tốc độ trung bình một lần xử lý là 3.61 FPS

## **5.2 Đánh giá.**

Để đánh giá chính xác khả năng phát hiện của toàn bộ mô hình thuật toán luận văn đã đề xuất, chúng ta sẽ đánh giá từng phần riêng biệt về mô hình mạng được huấn luyện và đánh giá về thực nghiệm trong thực tế.

### *5.2.1 Đánh giá mô hình phát hiện đối tượng với thuật toán YOLOv4*

Trong đề tài này luận văn sử dụng hai mô hình mạng tương tự nhau là YOLOv4-tiny và YOLOv4-tiny\_3l nhưng được huấn luyện với hai tập dữ liệu khác nhau và không trùng lên nhau. Do đó luận văn sẽ đánh giá mạng yolov4 trên toàn tập dữ liệu.

Tập dữ liệu luận văn đánh giá gồm 300 ảnh có 7 lớp đối tượng là người, có đội mũ bảo hộ, không đội mũ bảo hộ, có mang áo bảo hộ, không mang áo bảo hộ, có mang găng tay bảo hộ, không mang găng tay bảo hộ.

Có rất nhiều cách đánh giá một mô hình nhận dạng. Tùy vào từng bài toán khác nhau mà chúng ta nên sử dụng các phương pháp đánh giá khác nhau, một số phương pháp đánh giá phổ biến như: accuracy score, confusion matrix, ROC curve, Area Under the Curve, Precision and Recall, F1 Score, Mean Advantage Precision,...

Trong luận văn lần này tôi sử dụng mô hình phát hiện đối tượng YOLOv4, mô hình này thường được đánh giá với hai phương pháp chính là Mean Advantage Precision (mAP) và F1 Score. Do đó luận văn sẽ đánh giá mô hình thông qua một trong hai phương pháp là F1 Score.

Đánh giá bằng chỉ số F1 Score:

## LUẬN VĂN TỐT NGHIỆP

Để đánh giá bằng chỉ số F1 Score luận văn sẽ đưa toàn bộ ảnh cần đánh giá qua mô hình phát hiện đối tượng và chạy với từng đối tượng riêng biệt, kết quả chạy chương trình được thông kê với bảng sau:

	FN	TN	FP	TP
Người	0	0	0	300
Mang mũ	7	52	1	240
Mang găng tay	116	35	0	149
Mang áo	0	5	1	294
Không mang mũ	0	252	0	48
Không mang găng tay	3	274	0	24
Không mang áo	0	294	0	6
Tổng	126	914	2	1059

Bảng 5-2-1: Bảng kết quả đánh giá TP, TN, FP, FN cho 7 đối tượng

Ta có hiểu nội dung của bảng 5-2-1 như sau:

Mô hình phát hiện được đối tượng và phân đúng phân lớp (TP) = 1059

Mô hình phát hiện được đối tượng nhưng phân sai phân lớp (TN) = 2

Mô hình không phát hiện được đối tượng và thực thể không có đối tượng (FP)  
= 914

Mô hình không phát hiện đối tượng nhưng có đối tượng tồn tại trong ảnh (FN)  
= 126

$$R = \frac{TP}{TP + FP} = \frac{1059}{1059 + 2} = 0.998$$

$$P = \frac{TP}{TP + FN} = \frac{1059}{1059 + 126} = 0.894$$

$$F1\ Score = \frac{2 \cdot P \cdot R}{P + R} = \frac{2 \cdot 0.998 \cdot 0.894}{0.998 + 0.894} = 0.943$$

Nhận xét:

- Dựa vào tỉ số F1 Score ta đánh giá được độ chính xác của mô hình mạng là 0.943 tỉ số này là chấp nhận được chứng tỏ tỉ lệ phát hiện chính xác



một đối tượng trong ảnh là 92,86%.

- Tỷ số gọi lại  $R = 0.998$  ý nghĩa là mô hình có tỷ lệ phân lớp sai thấp
- Tỷ số dự đoán  $P = 0.894$  ý nghĩa mô hình có khả năng phát hiện được đối tượng là 0.894, hầu hết sai số đều tập trung vào găng tay.
- Thực tế hầu hết dự đoán sai FN xảy ra đối với đối tượng mang găng tay bảo hộ mà mô hình mạng không thể phát hiện được.

Qua hai phương pháp đánh giá F1 Score luận văn đã trình bày một cách trực quan về độ chính xác của mô hình mạng luận văn đã thực hiện, trong phần tiếp theo luận văn sẽ đánh giá mô hình thực nghiệm.

### *5.2.2 Đánh giá mô hình nhận dạng thiết bị bảo hộ cá nhân trong thực tế.*

Đối với thử nghiệm về khoảng cách, khoảng cách giữa đối tượng với camera được thay đổi từ 1–10(m). Thử nghiệm cho ra kết quả như sau:

- Khoảng cách từ 1-2.5(m): Mô hình không hoạt động tốt khi khoảng cách từ camera đối đối tượng, kết quả nhận dạng chỉ nhận dạng được đối tượng là người.
- Khoảng cách từ 2.5-4(m): Đây là khoảng cách thuật toán phát hiện tốt, nhưng đối với tượng là găng tay thì thuật toán vẫn khó có thể phát hiện được đối tượng chính xác.
- Từ 4-6(m): Thuật toán có thể phát hiện được đối tượng với độ chính xác cao đặc biệt là găng tay.
- Từ 7-10(m): thuật toán không thể phát hiện được đối tượng.

Đối khả năng phát hiện của mô hình mạng:

Mô hình mạng có thể phát hiện tốt hầu hết các đối tượng nhưng đối tượng là có mang găng tay và không mang găng tay mô hình vẫn khó phát hiện chính xác.

Mô hình có thể phát hiện đúng khi đối tượng đang giữ thiết bị bảo hộ hoặc không mang thiết bị bảo hộ.

## **5.3 Kết luận chương 5.**



## *LUẬN VĂN TỐT NGHIỆP*

Trong chương 5 luận văn đã xác định được các trường hợp mà mô hình thuật toán luận văn đề xuất có thể thực hiện. Qua đó cũng bao quát tổng quát được các trường hợp khó giải quyết trong bài toán phát hiện thiết bị bảo hộ lao động cá nhân.

Trong chương này luận văn cũng đã đưa ra được đánh giá thực nghiệm của mô hình thuật toán luận văn trong thực tế, đề xuất cũng như đưa ra được đánh giá về độ chính xác của mạng mà luận văn đã huấn luyện.

## CHƯƠNG.6 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1 Kết luận.

Qua các bước tìm hiểu, nghiên cứu, mô phỏng và xây dựng mô hình mặc dù gặp nhiều khó khăn trong quá trình thực hiện, nhưng đề tài cũng đã đạt được những mục tiêu đã đề ra.

Các mục tiêu đã đạt được:

- Tìm hiểu lý thuyết về cách thức hoạt động của một số giải thuật sử dụng trong học máy và học sâu.
- Tìm hiểu lý thuyết về thuật toán nhận dạng đối tượng YOLO.
- Huấn luyện bộ dữ liệu thu thập được với mạng YOLOv4 giải quyết một vấn đề thực tế.
- Thiết kế một hệ thống phát hiện thiết bị bảo hộ lao động cá nhân.
- Đánh giá mô hình, ưu nhược điểm của mô hình.
- Đưa ra các phương pháp cải tiến và kết luận.

Tuy nhiên luận văn vẫn còn một số điểm hạn chế:

- Luận văn vẫn còn mang tính lý thuyết và chưa có khả năng ứng dụng được trong thực tế.
- Tập dữ liệu được sử dụng trong luận văn chưa đủ tốt dẫn tới kết quả dự đoán sai so với thực tế.
- Chưa thể kiểm thử trong môi trường làm việc công nghiệp thực tế.
- Chưa có khả năng phối hợp nhiều mô hình mạng

### 6.2 Hướng phát triển.

Với những kết quả đạt được hiện tại có thể trở thành nền tảng để xây dựng và phát triển thành công một ứng dụng nhận dạng và cảnh báo người lao động mang

thiết bị bảo hộ trong tương lai có khả năng thích nghi và hoạt động ổn định với sự thay đổi của môi trường.

Một số giải pháp đưa ra để cải thiện:

- Chuẩn bị một tập dữ liệu huấn luyện tốt hơn, phù hợp với môi trường thực tế.
- Sử dụng các mô hình mạnh hơn về theo dõi đối tượng và phát hiện đối tượng nhằm tăng khả năng học của mô hình.
- Sử dụng kết hợp thêm cảm biến đã được nghiên cứu phát triển để tăng khả năng chính xác của bộ nhận dạng.
- Tăng số đối tượng cần nhận diện gồm mũ bảo hộ, áo bảo hộ, găng tay bảo hộ, quần bảo hộ và giày bảo hộ để đáp ứng nhu cầu ứng dụng thực tế.
- Sử dụng máy tính có tốc độ xử lý cao và đồ họa tốt hơn có thể chạy các mô hình mạng lớn với thời gian thực và tăng độ chính xác của quá trình phát hiện.

Ngoài ra luận văn còn đề xuất sử dụng thuật toán theo dõi DeepSort sau bộ phát hiện người, điều này sẽ giúp ổn định hộp giới hạn luôn bao quanh cơ thể người và giúp phát hiện cùng một đối tượng trong ảnh dễ dàng cho sự phát triển của mô hình sau này trong ứng dụng phát hiện và cảnh báo người lao động khi không mang thiết bị bảo hộ.

## PHỤ LỤC A

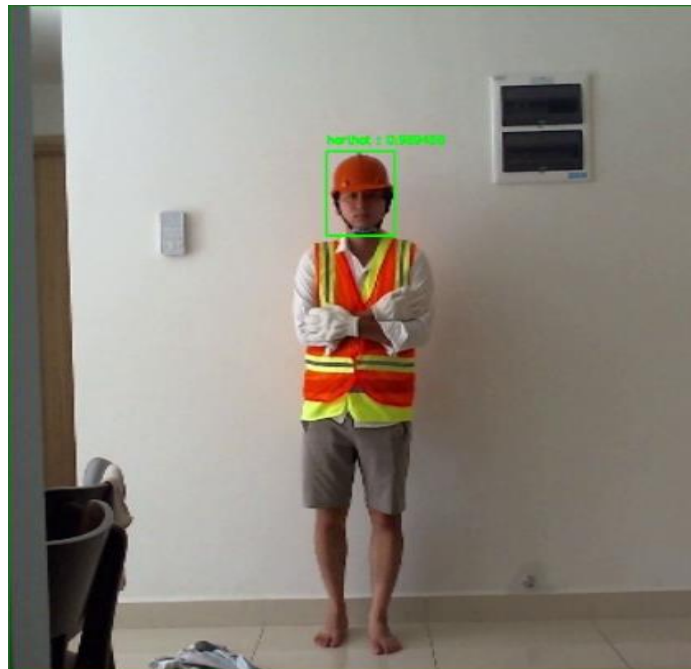
Các đối tượng cần thuật toán phát hiện trong luận văn

\* Người lao động



Hình 7-1-1: Đối tượng là người lao động

\* Người lao động có đội nón bảo hộ lao động



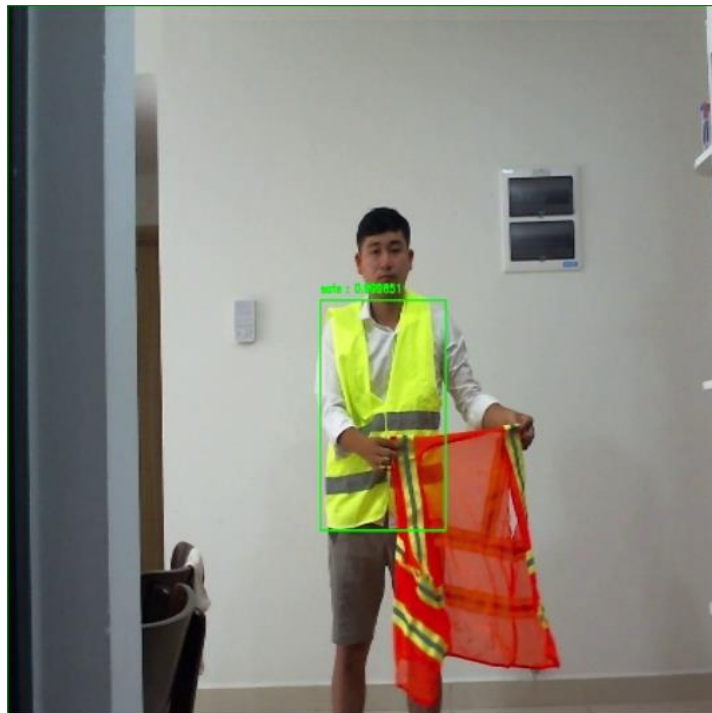
Hình 7-1-2: Đối tượng đội nón bảo hộ

\* Người lao động không đội nón bảo hộ lao động



Hình 7-1-3: Đối tượng không đội nón bảo hộ

\* Người lao động mang áo bảo hộ lao động



Hình 7-1-4: Thuật toán nhận diện người lao động mang áo bảo hộ

\* Người lao động không mang áo bảo hộ lao động



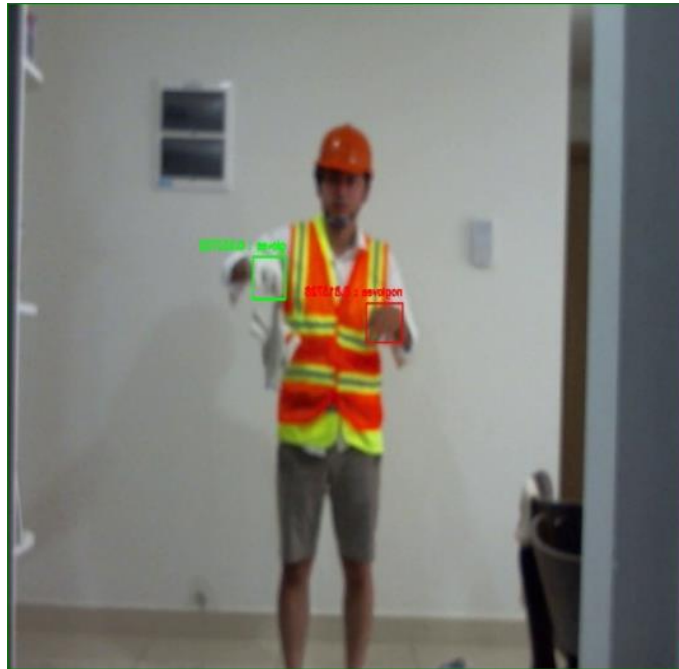
Hình 7-1-5: Thuật toán phát hiện không mang áo bảo hộ

\* Người lao động mang găng tay bảo hộ lao động



Hình 7-1-6: Thuật toán phát hiện mang găng tay

\* Cá nhân không mang găng tay bảo hộ lao động



Hình 7-1-7: Thuật toán phát hiện không mang găng tay

## TÀI LIỆU THAM KHẢO

- [1] Cục An toàn lao động, Bộ lao động – Thương binh và Xã hội, Việt Nam.  
<http://antoanlaodong.gov.vn/>
- [2] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, “*You Look Only Once: Unified, Real-Time Object Detection*”, published in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), USA, NV, June.27-30 ,2016.
- [3] Joseph Redmon, Ali Farhadi, “*YOLO9000: Better, Faster, Stronger*”, published in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), USA, HI, July.21-26 ,2017.
- [4] Joseph Redmon, Ali Farhadi, “*YOLOv3: An Incremental Improvement*”, arXiv preprint arXiv:1804.02767v1, April.8,2018.
- [5] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, “*YOLOv4: Optimal Speed and Accuracy of Object Detection*” arXiv preprint arXiv:2004.10934v1, April.23 ,2020.
- [6] Darknet - open source neural network framework,  
<https://github.com/AlexeyAB/darknet>
- [7] Nipun D. Nath, Amir H.Behzadan, Stephanie G. Paal, “Deep learning for site safety: Real-time detection of personal protective equipment” Automation in Construction 112 (2020) 103085.
- [8] Shi Chen, Kazuyuki Demachi, “*A Vision-Based Approach for Ensuring Proper Use of Personal Protective Equipment (PPE) in Decommissioning of Fukushima Daiichi Nuclear Power Station*”, Applied sciences by MDPI, Japan, Tokyo, July.26, 2020.



[9] Zhe Cao, Gines Hidalgo, Tomas Simon, Shin-En Wei, Yaser Sheikh, “*OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*”, arXiv preprint arXiv:1812.08008v2, May.30, 2019.

[10] V. H. Tiệp, Machine Learning Cơ Bản, Nhà Xuất bản Khoa Học và Kỹ Thuật, 2018.