TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỂN THÔNG

──────── \* ───────

ĐỒ ÁN

**TỐT NGHIỆP ĐẠI HỌC**

NGÀNH KỸ THUẬT ĐO VÀ TIN HỌC CÔNG NGHIỆ

**PHÁT HIỆN NGƯỜI TRONG VIDEO SỬ DỤNG MẠNG NEURON TÍCH CHẬP**

Sinh viên thực hiện : **Nguyễn Đức Trung**

Lớp KSCLC THCN – K58

Giáo viên hướng dẫn: **TS.** **Trần Thị Thanh Hải**

HÀ NỘI 6-2018

# PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Nguyễn Đức Trung

Điện thoại liên lạc 0868869253 Email: nguyenductrung95@gmail.com

Lớp: KSCLC Tin học công nghiệp K58 Hệ đào tạo: KSCLC-TN-TT

Đồ án tốt nghiệp được thực hiện tại:

Viện nghiên cứu quốc tế MICA – Trường Đại học Bách Khoa Hà Nội

Thời gian làm ĐATN: Từ ngày ... đến ...

2. Mục đích nội dung của ĐATN

Nghiên cứu và thử nghiệm kỹ thuật học sâu cho bài toán nhận dạng hoạt động của người từ video

3. Các nhiệm vụ cụ thể của ĐATN

- Tìm hiểu bài toán phát hiện của người từ ảnh và các hướng giải quyết.

- Tìm hiểu và thử nghiệm kỹ thuật mạng neuron tích chập.

- Đánh giá độ chính xác, ưu nhược điểm của kỹ thuật nghiên cứu.

4. Lời cam đoan của sinh viên:

Tôi – *Nguyễn Đức Trung* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Trần Thị Thanh Hải*.

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng năm*  Tác giả ĐATN  *Nguyễn Đức Trung* |

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng năm*  Giáo viên hướng dẫn  *TS. Trần Thị Thanh Hải* |

# TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Trong ĐATN này, em tập trung giải quyết bài toán phát hiện người trong một video hay một chuỗi ảnh đầu vào. Đây là một bài toán có tính ứng dụng thực tiễn cao, làm tiền đề cho việc nhận dạng hoạt động của người. Có 2 hướng tiếp cận chính để phát hiện người: sử dụng đặc trưng được trích xuất bằng tay hoặc đặc trưng được trích xuất dựa trên kỹ thuật học sâu. Mỗi phương pháp đều có những ưu, nhược điểm riêng. Phương pháp mà em lựa chọn nghiên cứu và thử nghiệm trong ĐATN này là sử dụng đặc trưng được trích xuất dựa trên kỹ thuật học sâu. Có nhiều kiến trúc học sâu đã đạt kết quả tốt cho việc giải bài toán nhận dạng hoạt động trong thời gian gần đây. Trong các kỹ thuật đó, em tìm hiểu YOLO trong [3] là một phương pháp đã được đánh giá thực nghiệm là cho kết quả tốt với tốc độ xử lý cao trên một số CSDL dùng chung của cộng đồng nghiên cứu. Trong ĐATN của mình, em sử dụng bộ dữ liệu về hoạt động của người của MICA gồm các video thuộc 21 lớp hoạt động của 50 người dưới 7 góc nhìn khác nhau. Độ chính xác phân lớp đạt được là 95%. Kết quả này vẫn có thể cải thiện thêm nhờ việc chuẩn bị dữ liệu một cách chính xác hơn hay bằng cách kết hợp nhiều nguồn thông tin khác nhau.

# ABSTRACT OF THESIS

# LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn chân thành đến cô Trần Thị Thanh Hải vì đã hướng dẫn ân cần, tận tình, và kiên nhẫn với em và tạo động lực cho em trong khoảng thời gian thực hiện đồ án tốt nghiệp này.

Em cũng xin chân thành cảm ơn Viện Nghiên cứu Quốc tế MICA đã tạo cho em một môi trường thuận lợi để học tập và nghiên cứu.

Em cũng xin gửi lời cảm ơn đến tất cả các thầy cô trường Đại học Bách Khoa Hà Nội trong suốt thời gian học trên giảng đường đã truyền đạt cho em những kiến thức cần thiết.

Em cũng xin gửi lời cảm ơn đến tất cả các bạn cùng lớp đã đồng hành cùng em trong suốt thời gian học và làm việc, đã giúp đỡ động viên em rất nhiều.

Em xin gửi lời cảm ơn đến gia đình đã luôn quan tâm, ủng hộ hết lòng về vật chất và tinh thần trong suốt thời gian qua.

Do thời gian và kiến thức có hạn nên không tránh khỏi những thiếu sót nhất định. Em rất mong nhận được sự đóng góp quý báu của thầy cô và các bạn.

Cuối cùng, em xin gửi lời chúc sức khỏe, hạnh phúc tới thầy cô, gia đình và bạn bè.

*Hà Nội, ngày tháng năm 2017*

Sinh viên

# MỤC LỤC

[PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP 2](#_Toc516428930)

[TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP 3](#_Toc516428931)

[ABSTRACT OF THESIS 4](#_Toc516428932)

[LỜI CẢM ƠN 5](#_Toc516428933)

[MỤC LỤC 6](#_Toc516428934)

[DANH MỤC HÌNH ẢNH 8](#_Toc516428935)

[DANH MỤC BẢNG 9](#_Toc516428936)

[DANH MỤC TỪ VIẾT TẮT VÀ THUẬT NGỮ 10](#_Toc516428937)

[MỞ ĐẦU 11](#_Toc516428938)

[CHƯƠNG 1: TỔNG QUAN VỀ PHÁT HIỆN NGƯỜI 12](#_Toc516428939)

[1.1. Một số khái niệm 12](#_Toc516428940)

[1.2. Một số nghiên cứu liên quan về nhận dạng hoạt động của người 12](#_Toc516428941)

[1.2.1. Hướng tiếp cận biểu diễn hoạt động dựa trên đặc trưng trích chọn được thiết kế bằng tay (hand-crafted features) 12](#_Toc516428942)

[1.2.2. Hướng tiếp cận biểu diễn hoạt động dựa trên kỹ thuật học sâu 13](#_Toc516428943)

[1.2.3. Nhận xét chung 13](#_Toc516428944)

[1.3. Phạm vi của ĐATN và đề xuất phương hướng triển khai 14](#_Toc516428945)

[CHƯƠNG 2: NGHIÊN CỨU GIẢI THUẬT PHÁT HIỆN ĐỐI TƯỢNG SỬ DỤNG MẠNG YOLO 16](#_Toc516428946)

[2.1. Khái niệm chung về mạng neuron tích chập 16](#_Toc516428947)

[2.1.1. Mạng nơ ron 16](#_Toc516428948)

[2.1.2. Mạng nơ ron tích chập 17](#_Toc516428949)

[2.2. Kiến trúc mạng Yolo 20](#_Toc516428950)

[2.2.1 Mô hình kiến trúc mạng 20](#_Toc516428951)

[2.2.2 Cách thức huấn luyện và đánh giá mạng. 21](#_Toc516428952)

[2.2.4 Một số kết quả đã đạt được với bài toán phát hiện đối tượng nói chung. 21](#_Toc516428953)

[2.3 Kết luận chương 22](#_Toc516428954)

[CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ THỬ NGHIỆM MÔ ĐUN PHÁT HIỆN NGƯỜI 23](#_Toc516428955)

[3.1. Cài đặt và huấn luyện mạng 23](#_Toc516428956)

[3.1.1.Môi trường lập trình và thư viện 23](#_Toc516428957)

[3.2. Dữ liệu và thước đo đánh giá 24](#_Toc516428958)

[3.3. Kết quả đánh giá thực nghiệm 27](#_Toc516428959)

[3.3.1 Yolo nguyên bản 27](#_Toc516428960)

[3.3.2 Yolo đã huấn luyện 28](#_Toc516428961)

[CHƯƠNG 4: KẾT LUẬN 29](#_Toc516428962)

[1.1. Kết quả đạt được 29](#_Toc516428963)

[1.2. Những điểm còn hạn chế 29](#_Toc516428964)

[1.3. Hướng phát triển 29](#_Toc516428965)

[TÀI LIỆU THAM KHẢO 30](#_Toc516428966)

# DANH MỤC HÌNH ẢNH

Hình 1: Sơ đồ tổng quát sử dụng đặc trưng thiết kế bằng tay cho phát hiện người 12

Hình 2: Minh họa mạng nơ ron của người (từ Rob Fergus) 17

Hình 3: Mạng nơ ron nhiều tầng 17

Hình 4 Kết nối giữa các tầng trong mạng nơ ron truyền thống 18

Hình 5 Kết nối giữa các tầng trong mạng nơ ron tích chập 19

Hình 6: Các bước cơ bản trong mạng neuron tích chập 19

Hình 7 Kiến trúc mạng YOLO đầy đủ với 24 lớp convolution và 2 lớp liên kết đầy đủ 23

Hình 8 Phân bố số lượng mẫu cho huấn luyện và kiểm kiểm trử trên từng Kinect 27

Hình 9 Sử dụng LabelImg để ground truth cho dữ liệu 29

# DANH MỤC BẢNG

Bảng 1 So sánh độ chính xác và tốc độ so với các phương pháp phát hiện đối tượng khác 24

Bảng 2. Kết quả đạt được khi thử nghiệm với yolo nguyên bản 32

Bảng 3 Kết quả dạt được khi thử nghiệm với yolo đã huấn luyện 32

Bảng 4 Kết quả đạt được khi thử nghiệm với yolo đã huấn luyện với từng kinect 32

# DANH MỤC TỪ VIẾT TẮT VÀ THUẬT NGỮ

# MỞ ĐẦU

Ngày nay với sự phát triển của khoa học kỹ thuật, việc sử dụng công nghệ thông tin trong đời sống hàng ngày đã trở nên phổ biến do tính hiệu quả cao và khả năng cải thiện đời sống của con người. Trong đó, các ứng dụng công nghệ thông tin trong giám sát an ninh, hệ thống cảnh báo an toàn, hệ thống theo dõi hành vi bất thường ở các môi trường cần theo dõi an ninh như văn phòng, ngân hàng, sân bay ngày càng phổ biến. Trong các hệ thống đó thì bài toán phát hiện người là bài toán căn bản, có vai trò quan trọng và làm tiền đề cho các chức năng xử lý nâng cao khác trong hệ thống. Vì vậy, nâng cao độ chính xác cũng như tốc độ tính toán của bài toán phát hiện người trở thành thách thức không nhỏ cho các nhà khoa học trên thế giới.

Trong khuôn khổ của ĐATN này, em tập trung nghiên cứu một phương pháp học sâu để giải quyết bài toán phát hiện người trong video. Các kỹ thuật học sâu đã được chứng minh là rất hiệu quả trong các bài toán nhận dạng, phân lớp đối tượng trong ảnh tĩnh. Thời gian gần đây, các kỹ thuật này cũng đang dần được nghiên cứu cho bài toán nhận dạng hoạt động từ video. Mỗi kiến trúc mạng cũng như cách thiết kế dữ liệu sẽ cho hiệu quả khác nhau trên cùng CSDL dùng chung. Trong ĐATN này, em nghiên cứu tìm hiểu kỹ thuật sử dụng mạng nơron tích chập theo kiến trúc YOLO được đề xuất bởi Joseph Redmon và các đồng nghiệp trong [1]. Đây là kiến trúc sử dụng trên bài toán phát hiện đối tượng cho độ chính xác cao và tốc độ nhanh so với các phương pháp sử dụng mạng nơ ron tích chập khác khi đánh giá trên các bộ dữ liệu COCO, VOC. Trong [1] các tác giả hướng tới bài toán phát hiện đối tượng nói chung, do vậy em đã tập trung vào nghiên cứu và thử nghiệm kiến trúc YOLO để tối ưu cho bài toán phát hiện người. Đề tài này em thực hiện tại phòng Computer Vision, Viện MICA dưới sự hướng dẫn của TS. Trần Thị Thanh Hải. Trong đồ án tốt nghiệp này, em sẽ trình bày theo 4 chương:

* Chương 1: Tổng quan về phát hiện người.
* Chương 2: Nghiên cứu giải thuật phát hiện đối tượng sử dụng mạng YOLO.
* Chương 3: Triển khai và đánh giá thử nghiệm mô đun phát hiện người sử dụng mạng YOLO.
* Chương 4: Kết luận.

# CHƯƠNG 1: TỔNG QUAN VỀ PHÁT HIỆN NGƯỜI

## Một số khái niệm

Phát hiện người là bài toán xác định vùng có xác suất cao chứa người trên ảnh. Đây là bài toán cơ bản và quan trọng trong lĩnh vực thị giác máy tính vì nó là tiền đề cho các bài toán nhận dạng người, nhận dạng hoạt động của người, theo dõi người trong ảnh. Chính vì vậy phát hiện người là bài toán có tính ứng dụng cao. Ứng dụng của nó rất đa dạng, bao gồm hệ thống giám sát an ninh, hệ thống cảnh báo an toàn, hệ thống theo dõi hành vi bất thường hay trong các hệ thống tương tác giữa người-máy.

Tuy nhiên, phát hiện người là một bài toán thách thức bởi nhiều yếu tố ảnh hưởng đến độ chính xác như:

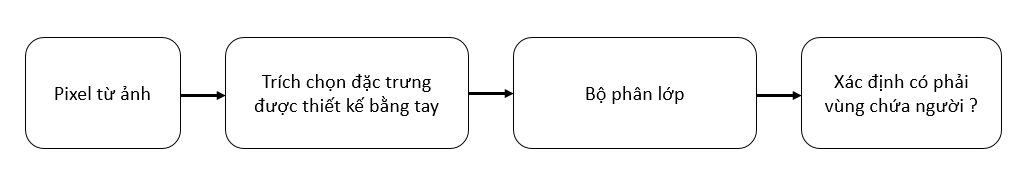
* Người thường bị che khuất trong ảnh do các vật thể khác
* Điều kiện ánh sáng ảnh hưởng đến chất lượng ảnh đầu vào.
* Sự thay đổi góc nhìn của camera cũng như sự đa dạng trong hình trạng, diện mạo của người.

## Một số nghiên cứu liên quan về phát hiện người

### Hướng tiếp cận biểu diễn hoạt động dựa trên đặc trưng trích chọn được thiết kế bằng tay (hand-crafted features)

Đặc trưng thiết kế bằng tay (handcrafted feature) là các đặc trưng được thiết kế từ trước, nhằm đưa ra cấu trúc đặc trưng mới phù hợp nhất với từng đối tượng cần phát hiện. Nhờ vậy mà các mô hình cải thiện được độ chính xác của mình. Đây là công việc đòi hỏi sự sáng tạo và thời gian của các nhà khoa học dữ liệu.

Các đặc trưng giúp cho việc chuyển đổi dữ liệu thô ban đầu thành tập các thuộc tính giúp biểu diễn dữ liệu tốt hơn, giúp tương thích với từng mô hình dự đoán cụ thể, cũng như cải thiện độ chính xác của mô hình hiện tại.



Hình 1: Sơ đồ tổng quát sử dụng đặc trưng thiết kế bằng tay cho phát hiện người

Hình 1 minh họa hệ thống tổng quát sử dụng đặc trưng được thiết kế bằng tay cho bài toán phát hiện người.

* Thông thường các bài toán phát hiện sẽ sử dụng một cửa sổ có kích thước thay đổi quét trên ảnh, các pixel từ các cửa sổ này chính là đầu vào của hệ thống phát hiện.
* Khối trích chọn đặc trưng bằng tay: Nhận đầu vào là các pixel và đưa ra các đặc trưng cho ảnh đó. Đặc trưng ở khối này được chọn từ trước tùy theo yêu cầu bài toán và đặc điểm của từng dataset.
* Khối phân lớp: Là một bộ phân lớp được huấn luyện từ trước với đầu vào là các đặc trưng được xuất ra từ khối trích chọn đặc trưng và quyết định cửa sổ quét từ đầu vào có phải vùng chứa đối tượng cần phát hiện hay không.

Hiện nay, trên thế giới đã có rất nhiều đặc trưng được thiết kế để giải quyết bài toán phát hiện người. Trong [2] hai tác giả Dalal và Trigg đề xuất sử dụng đặc trưng HOG để biểu diễn trích xuất đặc trưng và bộ phân lớp SVM. Mở rộng ý tưởng đó Hong Liu [3] đã sử dụng đặc trưng HOG trên bộ phân lớp Adaboost-Cascade. Một số nghiên cứu khác [4] lại sử dụng ảnh độ sâu thu được từ camera kinect để xác định vùng chứa người. Trong khi đó [5] đi theo một hướng khác là sử dụng phương pháp trừ nền để xác định vùng chứa người.

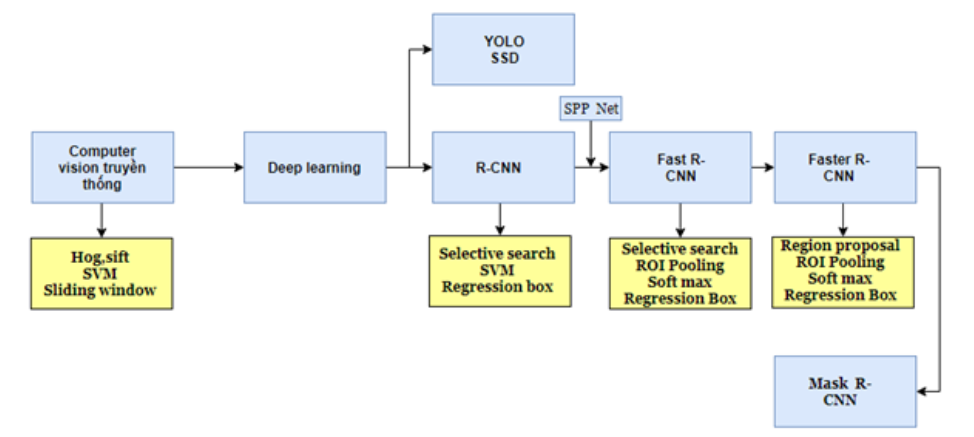
### Hướng tiếp cận biểu diễn hoạt động dựa trên kỹ thuật học sâu.

Tuy nhiên, với tốc độ phát triển của công nghệ hiện nay , dữ liệu của chúng ta ngày càng nhiều, các tình huống phải xử lý càng ngày càng phức tạp hơn. Trong khi đó các đặc trưng được thiết kế bằng tay chỉ lấy được những đặc trưng nổi. Chính vì thế ta cần những đặc trưng sâu hơn, thể hiện được mối liên hệ giữa các yếu tố trong ảnh. Đó chính là các kĩ thuật sử dụng deep learning

Kỹ thuật học sâu (Deep learning) là một thuật toán học máy được xây dựng dựa trên một số ý tưởng mô phỏng hệ thống não bộ của con người. Nó biểu diễn dữ liệu thông qua nhiều tầng từ cụ thể đến trừu tượng qua đó trích rút được các đặc trưng có ý nghĩa trong nhận dạng đối tượng ảnh. Khác với đặc trưng được thiết kế bằng tay, các đặc trưng không cần phải thiết kể một cách thủ công mà sẽ được học một cách tự động thông qua các bộ dữ liệu huấn luyện.

Có rất nhiều công trình liên quan đến bài toán phát hiện người nói riêng cũng như phát hiện đối tượng nói chung sử dụng kỹ thuật học sâu. Trong đó hai hướng tiếp cận chính theo hướng này là:

* Tìm những region proposal có nhiều khả năng chứa object nhất sử dụng selective search hay RPN (R-CNN, Fast R-CNN, Faster R-CNN [10]).
* Chia image ra thành những grid cell SxS . Mỗi cell được coi như region proposal giúp giảm thời gian và chi phí tính toán thay vì sử dụng trực tiếp image (SSD [7], YOLO [1]).



Phương pháp thứ nhất sử dụng selective search xuất phát từ R-CNN và đã có nhiều cải tiến, phương pháp mới nhất theo hướng tiếp cận này là Faster R-CNN. Ban đầu, R-CNN (viết tắt của “Region-based Convolutional Neural Networks”) có ý tưởng chính là sử dụng selective search tìm các region of interest (ROI) trên ảnh có xác suất cao chứa đối tượng cần phát hiện. Sau đó dùng CNN để lấy feature từ các region này để classifier và regression box. Mỗi ảnh sẽ tìm khoảng 2000 ROI. Nhược điểm của phương pháp này là training rất lâu vì cần sử dụng CNN tính đặc trưng cho 2000 ROI trên mỗi ảnh.

Fast R-CNN cải thiện được các nhược điểm của R-CNN bằng sử dụng CNN để lấy feature map một lần thay vì dùng riêng cho mỗi region proposal. Sau đó những feature này sẽ được đưa qua một Fully connection layer để classifier và regression bounding box.

Faster R-CNN là phiên bản mới nhất của họ R-CNN. Phương pháp này cải tiến Fast R-CNN bằng cách sử dụng region proposal network. Region proposal network sử dụng 1 cửa số 3x3 trượt trên ảnh, tại mỗi vị trí ta sử dụng các anchor box và tìm xác suất chứa đối tượng và tọa độ của anchor box cuối cùng sử dụng maximum suppression để loại bỏ bớt những box không có nhiều khả năng chứa đối tượng. Các box này được sử dụng tương tự như các box tìm được bằng phương pháp selective search.

Phương pháp thứ hai (SSD và YOLO) có một cách tiếp cận hoàn toàn khác là chỉ sử dụng duy nhất một neural network cho toàn bộ ảnh. Hình ảnh đầu vào sẽ được chia thành các ô lưới (grid cell), và dự đoán các bounding box và xác suất phân loại cho mỗi grid cell. Các bounding box này được đánh trọng số theo xác suất đã dự đoán. Họ phương pháp này có ưu điểm là tính toán trên toàn bộ ảnh để tìm đồng thời đối tượng và vị trí đối tương. Do chỉ cần tính toán một mạng duy nhất, thay vì hàng nghìn như họ R-CNN nên vậy tốc độ của YOLO, SSD nhanh hơn rõ rệt so với họ R-CNN.

### Nhận xét chung

Hướng biểu diễn dựa trên đặc trưng được trích xuất bằng tay cho kết quả rất tốt trên các tập dữ liệu nhỏ và có một số đặc điểm nhất định. Tuy vậy trên thực tế, dữ liệu được thu nhận được ở các điều kiện, góc nhìn khác nhau và các tư thế của người rất đa dạng. Không chỉ vậy, người trong ảnh còn bị che khuất bởi các vật thể phía trước ... Ngoài ra, phương pháp này có tốc độ tính toán khá chậm do quá trình xử lý phức tạp tùy vào loại đặc trưng sử dụng. Do đó, đối với cơ sở dữ liệu lớn thì việc sử dụng phương pháp này là khá khó khăn.

Kỹ thuật học sâu có thể cải thiện phần nào các thách thức này. Khác với đặc trưng trích xuất bằng tay, kỹ thuật học sâu không phụ thuộc vào sự đa dạng của bộ dữ liệu. Kỹ thuật này có thể xây dựng bộ trích xuất đặc trưng dựa trên việc học cách biểu diễn chúng. Tuy nhiên, kỹ thuật này đòi hỏi một lượng dữ liệu đủ lớn để đạt độ chính xác cho mô hình và dữ liệu phải đủ tốt và bao quát toàn bộ các tình huống thực tế. Hơn nữa, việc tìm ra các tham số và kiến trúc phù hợp cần rất nhiều thời gian để thử nghiệm.

## Phạm vi của ĐATN và đề xuất phương hướng triển khai

Trong các ứng dụng thực tế của bài toán phát hiện người thì bài toán nhận dạng hoạt động người đa góc nhìn là một bài toán có tính ứng dụng cao. Nó giải quyết các bài toán như giám sát an ninh trong tòa nhà, theo dõi hoạt động bất thường của bệnh nhân trong bệnh viện để đưa ra cảnh báo kịp thời,... Đặc trưng của bài toán này là môi trường hoạt động của hệ thống là trong phòng, các camera được gắn cố định theo nhiều góc nhìn khác nhau, điều kiện ánh sáng trong phòng ổn định. Hệ thống thường tập trung vào một người nhưng trong ảnh lại có nhiều đối tượng khác và hoạt động của người rất đa dạng: đứng, ngồi, nằm, ngã. Sự đa dạng và phức tạp trong các hoạt động của người khiến bài toán trở nên thách thức hơn với các nhà khoa học.

Yêu cầu của bài toán nhận dạng hoạt động người đa góc nhìn đặt ra nhiệm vụ cho bài toán phát hiện người là cần phát hiện được người trong khung hình với độ chính xác cao và thời gian tính toán nhanh. Có như vậy mới đảm bảo được chất lượng đầu vào cho các lớp nhận dạng hoạt động phía sau.

Qua tìm hiểu các hướng nghiên cứu trước đó, em nhận thấy để đáp ứng được yêu cầu phát hiện người trong khung hình với rất nhiều hoạt động khác nhau thì hướng giải quyết sử dụng các kĩ thuật học sâu là phương án phù hợp và có nhiều tiềm năng trong tương lai. Hơn nữa, với sự phát triển nhanh chóng của khoa học và kỹ thuật, máy tính, siêu máy tính với card đồ họa cho phép giải các bài toán dữ liệu lớn. Nhờ đó việc thời gian tính toán được giảm đi hàng chục lần so với ban đầu.

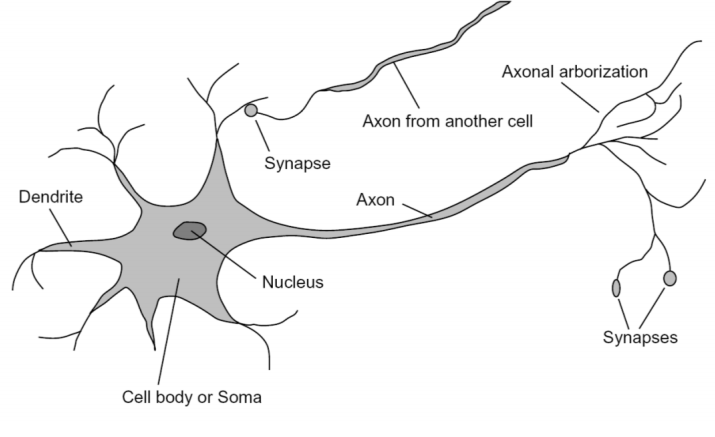
Trong ĐATN này, em lựa chọn kỹ thuật sử dụng kỹ thuật học sâu sử dụng mạng nơ ron tích chập theo kiến trúc YOLO được trình bày trong [1] để tìm hiểu và thử nghiệm. Đây là một kỹ thuật đã được đánh giá thử nghiệm với nhiều bộ dữ liệu và đã cho kết quả khá cao với tốc độ được đánh giá là cao nhất trong các phương pháp sử dụng kỹ thuật học sâu hiện tại. Trong bài toán này em sử dụng bộ cơ sở dữ liệu đa góc nhìn về hoạt động của người từ MICA cho đánh giá và thử nghiệm hiệu năng thuật toán.

# CHƯƠNG 2: NGHIÊN CỨU GIẢI THUẬT PHÁT HIỆN ĐỐI TƯỢNG SỬ DỤNG MẠNG YOLO

## 2.1. Khái niệm chung về mạng neuron tích chập

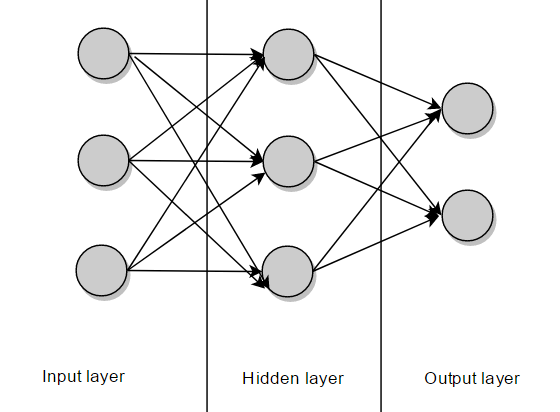
### 2.1.1. Mạng nơ ron

Mạng nơ ron nhân tạo được thiết kế nhằm mô phỏng mạng neuron của bộ não người (Hình 4). Mạng neuron nhân tạo được cấu thành từ một tập các phần tử xử lý đơn giản được kết nối với nhau. Mỗi phần tử xử lý này chỉ có thể thực hiện được một thao tác tính toán nhỏ, nhưng một mạng lưới các phần tử như vậy có một khả năng tính toán lớn hơn rất nhiều. Phần tử tính toán cơ bản của mạng nơ ron là một perceptron hay một nơ ron.



Hình 2: Minh họa mạng nơ ron của người (từ Rob Fergus)

Một nơ ron mô phỏng quá trình tính toán của bộ não con người. Dữ liệu được đưa tới các nơ ron thông qua các Dendrite vào Nucleus để tính toán. Tín hiệu ra được xuất ra ở dây Axon. Các nơ ron được liên kết với nhau thông qua các dây Synapse. Mạng nơ ron bao gồm rất nhiều phần tử như vậy liên kết với nhau.



Hình 3: Mạng nơ ron nhiều tầng

Hình 5 minh họa một mạng nơ ron truyền thẳng với 3 loại node sau:

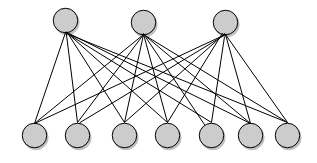
* Input nodes (Node đầu vào): Chứa dữ liệu đầu vào từ bên ngoài và đưa trực tiếp vào các Hidden nodes.
* Hidden nodes (Node ẩn): Nó không chứa kết nối trực tiếp đến dữ liệu từ bên ngoài. Nó thực hiện tính toán các dữ liệu nhận được từ các input nodes, thực hiện tính toán và đưa ra các output nodes. Tập hợp các node ẩn trong mạng tạo thành tầng ẩn. Một mạng nơ ron truyền thẳng có thể có hoặc không có tầng ẩn.
* Output nodes (Node đầu ra): Có nhiệm vụ tính toán và đưa dữ liệu từ trong mạng ra bên ngoài.

### 2.1.2. Mạng nơ ron tích chập

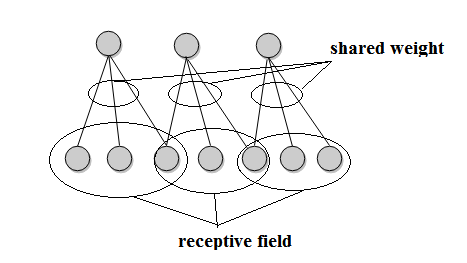
Mạng nơ ron tích chập là một mạng neuron nhân tạo với các toán tử tích chập. Nó có khả năng học một lượng lớn các dữ liệu trong khoảng thời gian ngắn hơn nhiều so với mạng nơ ron thông thường. Lý do là nó sử dụng ít trọng số hơn trong khi độ chính xác chỉ kém hơn một phần nhỏ so với kiến trúc truyền thống. Mô hình này sử dụng trong [5] và đã đạt kết quả khá tốt trong bài toán phân loại ảnh

Trong mạng nơ ron truyền thống, các node ở các tầng phía sau sẽ liên kết với toàn bộ các node ở layer phía dưới thông qua một tập các trọng số. Với mỗi nơ ron khác nhau, chúng ta cần một tập trọng số hoàn toàn độc lập để liên kết với các nơ ron ở tầng trước đó.

Điểm khác biệt của mạng nơ ron tích chập so với mạng nơ ron truyền thống đó là trong liên kết giữa 2 tầng liên tiếp nhau việc các node ở các tầng phía sau chỉ liên kết với một bộ phận các node ở tầng phía trước đó gọi là receptive field thông qua một tập các trọng số. Hơn nữa tập trọng số này là như nhau đối với mỗi nơ ron ở tầng sau. Do đó số lượng tham số cần training ít hơn trong khi vẫn giữ được lượng thông tin cần thiết.



Hình 4 Kết nối giữa các tầng trong mạng nơ ron truyền thống



Hình 5 Kết nối giữa các tầng trong mạng nơ ron tích chập

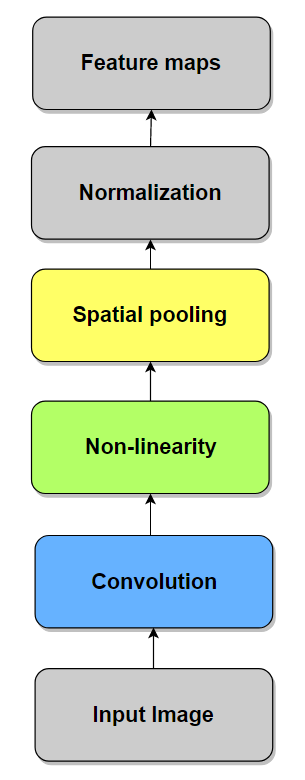
**Một mạng nơ ron tích chập thường được thực hiện thông qua các bước sau:**

- Convolutional layer

- Pooling layer

- Non-linearity layer

- Fully-connected layer



Hình 6: Các bước cơ bản trong mạng neuron tích chập

**Convolutional layer**

Trong mô hình CNN, các layer sẽ kết nối với nhau thông qua phép toán tích chập. Mỗi nơ ron tiếp theo là kết quả của việc áp dụng một filter lên một vùng nơ ron trước đó. Mỗi layer như vậy có thể có từ vài trăm đến vài nghìn filter như vậy. Hình dưới đây mô tả một phép toán tích chập trong CNN. Kết quả của phép toán này là activation map hay feature map. Ta có thể điều chỉnh kích cỡ của các feature map bằng các tham số:

* Depth - Độ sâu: là số lượng filter ta sử dụng trong tầng convolution. Ví dụ tầng convolution có đầu vào là một ảnh có kích thước 32x32 sử dụng 3 filter thì đầu ra sẽ có độ sâu là 3.
* Stride: Là bước dịch chuyển của bộ lọc trên đầu vào. Ví dụ stride là 1 thì ta sẽ dịch chuyển bộ lọc liên tiếp trên ảnh đầu vào.
* Zero-padding: Tăng kích thước của đầu vào bằng cách bao quanh nó bằng các giá trị 0.

**Non Linearity**

**ReLU (Rectified Linear Unit)** là toán tử được sử dụng sau mỗi phép toán tích chập có tác dụng thay các giá trị âm trong feature map đầu ra thành các giá trị 0. Mục đích của phép toán này là đưa vào các giá trị phi tuyến vì hầu hết các giá trị dữ liệu trong thực tế là phi tuyến.

Ngoài ra chúng ta có thể sử dụng toán tử phi tuyến khác như tanh hay sigmoid, nhưng trong hầu hết các trường hợp toán tử ReLU hiệu quả hơn.

**Tầng Pooling**

**Pooling** là phép toán lấy mẫu để giảm số chiều của các feature map nhưng vẫn giữ được các thông tin quan trọng nhất. Giống như tầng convolution, tầng pooling cũng sử dụng một filter đặt trên một phần của feature map đầu vào và di chuyển filter đó trên toàn bộ vị trí của feature map đầu vào. Có nhiều loại phép toán pooling được sử dụng:

* Max pooling: Lấy giá trị lớn nhất
* Average pooling: Lấy giá trị trung bình
* Sum pooling: Lấy tổng các giá trị

**Tầng liên kết đầy đủ**

Là một mạng nơ ron nhiều tầng (Multi Layer Perceptron) sử dụng hàm kích hoạt softmax ở đầu ra (có thể sử dụng bộ phân lớp khác như SVM). “Liên kết đầy đủ” nghĩa là mỗi nơ ron ở tầng trước sẽ được liên kết với toàn bộ nơ ron ở tầng phía sau. Mục tiêu của tầng liên kết đầy đủ là sử dụng các đặc trưng ở mức cao đã được phân tách bởi các tầng convolution và pooling ở phía trước để đưa vào các lớp của bộ dữ liệu.

Kiến trúc phổ biến của một mạng nơ ron tích chập là xếp chồng một số khuôn mẫu gồm các tầng như trên hình vẽ. Sau đó các đặc trưng sẽ được chuyển vào các một số tầng liên kết đầy đủ. Tầng cuối cùng sẽ đánh giá điểm cho từng lớp của bộ dữ liệu. Điểm này đánh giá cho khả năng đầu vào sẽ thuộc vào lớp đó. Kiến trúc phổ biến ở dưới dạng biểu thức như sau:

INPUT -> [[CONV->RELU]\*N->POOL?]\*M -> [FC->RELU]\*K -> FC

Trong đó dấu “\*” thể hiện sự lặp lại, còn POOL? chỉ sự tùy chọn. Thông thường ta sử dụng: 

## 2.2. Kiến trúc mạng YOLO

### 2.2.1 Tổng quan về YOLO

YOLO thống nhất toàn bộ các thành phần riêng biệt trong object detecion vào một neural network duy nhất. Nó sử dụng các feature từ toàn bức ảnh để predict mỗi bounding box với tất cả các class. Bức ảnh được chia thành lưới có kích thước S x S, nếu tâm của object nằm trong một grid cell, grid cell đó sẽ chịu trách nhiệm phát hiện ra object này.

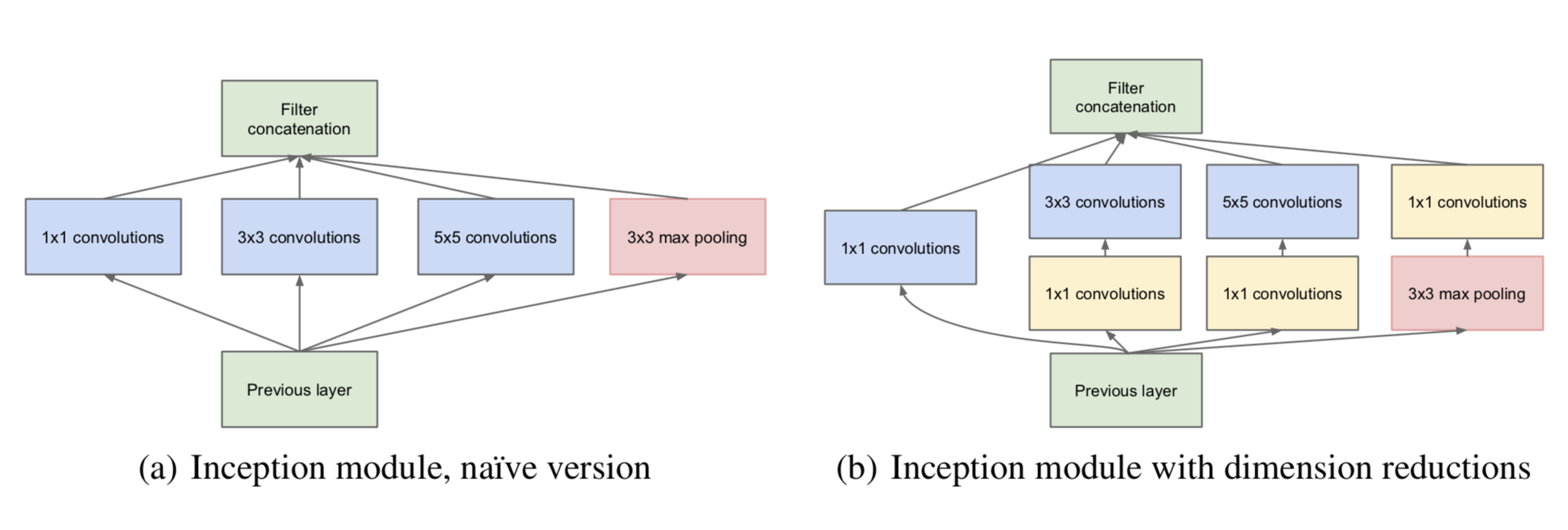
Mỗi grid cell dự đoán B bounding box và điểm số tin cậy (confidence score) tương ứng mỗi box. Confidence score phản ánh độ tin cậy của việc object nằm trong box, đồng thời là độ chính xác, đô khớp của box so với object. Độ tin cậy được định nghĩa bởi công thức:

Với giá trị trong [0, 1] thể hiện khả năng box có chứa object, là giá trị Intersection Over Union, thể hiện độ khớp của prediction box so với ground-truth box. Nếu không tồn tại object nào trong grid cell, các confidence score của nó có giá trị bằng không.

Mỗi bounding box chứa 5 giá trị x, y, w, h và confidence. Trong đó, toạ độ (x, y) thể hiện vị trí tương đối của tâm object so với viền grid cell. w, h là kích thước tương đối của chiều rộng và chiều cao của box so với kích thước toàn bức ảnh. Cuối cùng, confidence thể hiện IOU giữa predicted box và ground truth box. Mỗi grid cell đồng thời cũng dự đoán C xác suất có điều kiện của C class cho trước,. Xác suất này thể hiện xác suất tồn tại object ở grid cell, và chỉ predict một set C xác suất tương ứng mỗi grid cell, không cần quan tâm đến B bounding box.

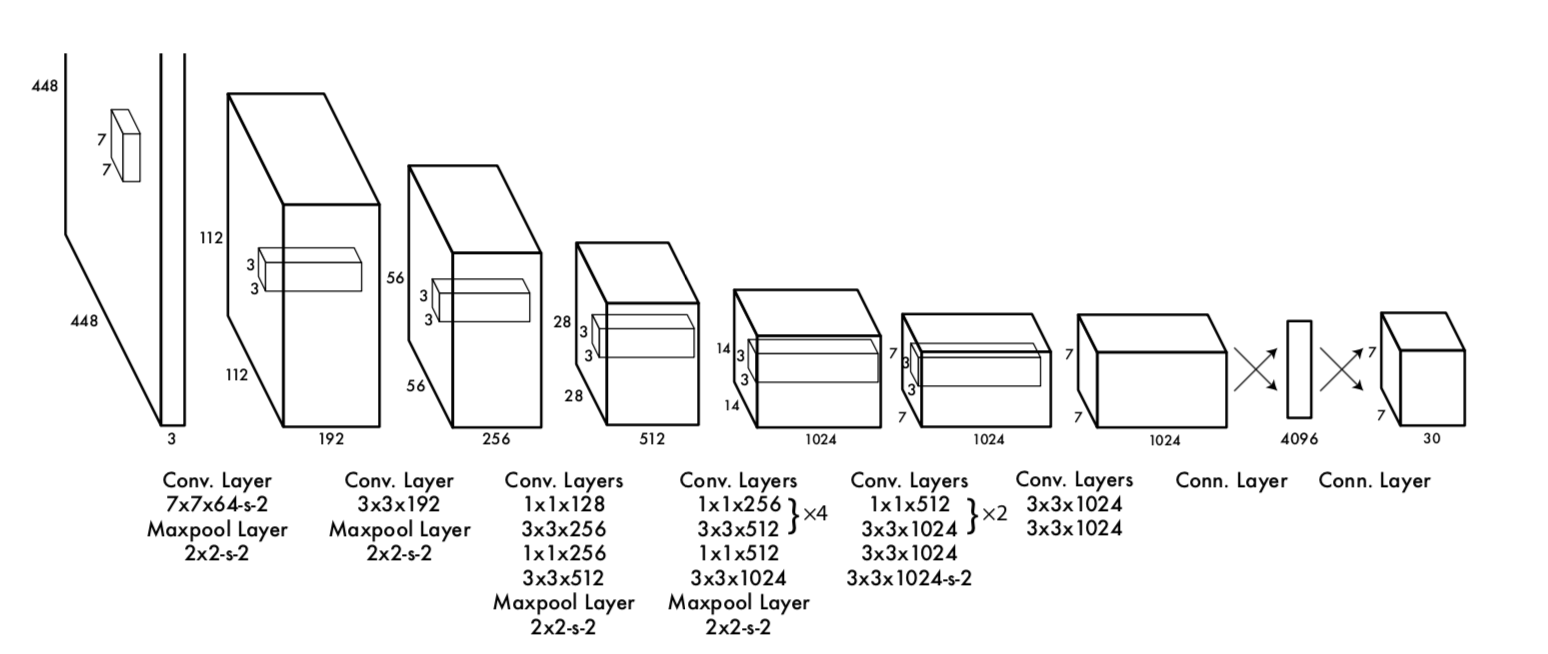
Với S x S grid, mỗi grid cell predict B bounding box (5 giá trị mỗi box) và C xác suất ứng với mỗi class, đầu ra của mạng sẽ là vector có số chiều là S x S (B x 5 + C).

### 2.2.2 Mô hình kiến trúc mạng.

Kiến trúc mạng nơ ron tích chập của YOLO được tham khảo theo kiến trúc mạng GoogLeNet [4]. Ý tưởng của kiến trúc mạng GoogleLeNet là sử dụng inception module, đây là một mạng CNN giúp huấn luyện rộng hơn thay vì thêm nhiều layer hơn dễ xảy ra overfitting và làm tăng số lượng parameter. Đối với mạng CNN truyền thống, mỗi layer sẽ trích xuất các thông tin khác nhau. Việc lựa chọn kích thước của kernel rất quan trọng, ảnh hưởng rất lớn tới hiệu năng của mạng. Thay vì lựa chọn trước kích thước kernel cho mỗi layer, mạng CNN sử dụng kiến trúc GoogleLeNet tính toán đồng thời với nhiều kernel có kích thước khác nhau và kết hợp lại thành đầu ra của layer. Phiên bản đầu tiên của Inception module có 2 dạng là naïve và dimension reduction:

Khác biệt chính giữa 2 dạng này là phiên bản dimension reduction sử dụng conv 1x1 ở mỗi layer để giảm depth và số lượng tham số tính toán. Inception naïve có architect gồm 1x1 conv,3x3 conv, 5x5 conv và 3x3 maxpooling.

Kế thừa ý tưởng này của mạng GoogleLeNet, Yolo sử dụng kiến trúc tương tự Inception nhưng chỉ bao gồm các lớp convolution 1x1 phía trước các lớp convolution 3x3. Ngoài ra yolo còn có một phiên bản rút gọn là Tiny Yolo với 9 lớp convolution thay vì 24 lớp convolution ở phiên bản yolo đầy đủ. Các kiến trúc của yolo đều có 2 tầng liên kết đầy đủ ở cuối. Hình 7 thể hiện kiến trúc mạng yolo đầy đủ với 24 lớp convolution và 2 lớp liên kết đầy đủ.



Hình 7 Kiến trúc mạng YOLO đầy đủ với 24 lớp convolution và 2 lớp liên kết đầy đủ

### 2.2.3 Cách thức huấn luyện và đánh giá mạng.

Để so sánh hiệu quả của YOLO với các phương pháp khác, các tác giả đã sử dụng cơ sở dữ liệu PASCAL VOC [5] đây là một bộ cơ sở dữ liệu cho phát hiện đối tượng lớn, được nhiều bài báo sử dụng để đánh giá. Nó bao gồm cơ sở dữ liệu được dán nhãn của 20 lớp đối tượng khác nhau. Bộ cơ sở dữ liệu VOC dùng cho đánh giá là phiên bản 2007 và 2012. Trong đó

* Phiên bản 2007 bao gồm: 9,963 ảnh với 24,640 đối tượng được dán nhãn của 20 lớp đối tượng khác nhau.
* Phiên bản 2012 bao gồm: 11,530 ảnh với 27,450 đối tượng được dán nhãn của 20 lớp đối tượng giống với phiên bản 2007.

Thư viện Darknet được sử dụng cho toàn bộ quá trình huấn luyện và kiểm thử. Đây là thư viện mạng nơ ron nguồn mở viết trên C và CUDA. Hỗ trợ thư viện OpenCV cho xử lý ảnh.

Để đánh giá mạng, ngoài đánh giá tốc độ xử lý qua FPS thể hiện số hình xử lý được trên giây, YOLO còn sử dụng chỉ số mAP để đánh giá độ chính xác của phát hiện.

### 2.2.4 Một số kết quả đã đạt được với bài toán phát hiện đối tượng nói chung.

Qua so sánh với các phương pháp phát hiện đối tượng, YOLO cho độ chính xác cao so trong khi tốc độ vượt trội so với các phương pháp còn lại.

|  |  |  |  |
| --- | --- | --- | --- |
| Phương pháp | Bộ dữ liệu | mAP | FPS |
| YOLO | VOC 2007+2012 | **63.4** | **45** |
| Fast YOLO | VOC 2007+2012 | **52.7** | **155** |
| Fastest DPM | VOC 2007 | 30.4 | 15 |
| R-CNN | VOC 2007+2012 | 53.5 | 6 |
| Fast R-CNN | VOC 2007+2012 | 70.0 | 0.5 |

Bảng 1 So sánh độ chính xác và tốc độ so với các phương pháp phát hiện đối tượng khác

YOLO áp đặt các ràng buộc về không gian trên những bounding box, mỗi grid cell chỉ có thể predict rất ít bounding box và duy nhất một class. Các ràng buộc này hạn chế khả năng nhận biết số object nằm gần nhau, cũng như đối với các object có kích thước nhỏ.

YOLO sử dụng các feature tương đối thô để predict bounding box, do model sử dụng nhiều lớp downsampling từ ảnh đầu vào. Bởi các hạn chế này của model khi huấn luyện để predict bounding box từ data, dẫn đến YOLO không thực sự tốt trong việc nhận diện các object với tỉ lệ hình khối mới hoặc bất thường so với tập data. YOLOv2 đã khắc phục phần nào vấn đề này, nhưng vẫn thua kém nhiều so với FRCNN.

Ngoài ra, trong quá trình training, loss function không có sự đánh giá riêng biệt giữa error của bounding box kích thước nhỏ so với error của bounding box kích thước lớn. Việc coi chúng như cùng loại và tổng hợp lại làm ảnh hưởng đến độ chính xác toàn cục của mạng. Error nhỏ trên box lớn nhìn chung ít tác hại, nhưng error nhỏ với box rất nhỏ sẽ đặc biệt ảnh hưởng đến giá trị IOU.

## 2.3 Kết luận chương

Do yêu cầu trong bài toàn nhận dạng hoạt động người đa góc nhìn là toán yêu cầu số lượng tính toán cao, do cần xử lý dữ liệu từ nhiều camera đồng thời, không những vậy bài toán nhận dạng hoạt động cũng là bài toán thách thức, cần nhiều tài nguyên xử lý nên giai đoạn phát hiện người trong ảnh cần độ chính xác cao với tốc độ tính toán nhanh. Do vậy em lựa chọn kiến trúc YOLO để huấn luyện và thử nghiệm cho bài toán phát hiện người. Tuy nhiên, các phiên bản YOLO hiện có chủ yếu tập trung vào bài toán phát hiện đối tượng nói chung nên đối với bài toán phát hiện người sẽ cho độ chính xác chưa được như mong muốn của bài toán em cần giải quyết. Do vậy trong đồ án này em sẽ tiến hành chuẩn bị bộ dữ liệu, huấn luyện và thử nghiệm cho bài toán cụ thể là phát hiện người đa góc nhìn.

# CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ THỬ NGHIỆM MÔ ĐUN PHÁT HIỆN NGƯỜI

## 3.1. Cài đặt và huấn luyện mạng

### 3.1.1. Môi trường lập trình và thư viện

Để thực hiện huấn luyện và thử nghiệm YOLO, em đã sử dụng các công cụ sau:

**Framework sử dụng cho mạng nơ ron:** [**Darknet**](https://pjreddie.com/darknet/)

Darknet là một thư viện mã nguồn mở viết bằng C và CUDA sử dụng để triển khai mạng nơ ron tích chập. Darknet đã được các tác giả sử dụng để huấn luyện vào thử nghiệm YOLO nên em lựa chọn thư viện này để triển khải mạng nơ ron nhanh chóng hơn.

**Framework sử dụng cho xử lý ảnh: OpenCV và CUDA:**

OpenCV là một thư viện xử lý ảnh và video mã nguồn mở được viết trên C/C++, đây là thư viện phổ biến nhất cho xử lý ảnh và có nhiều giao diện cho phép sử dụng bằng các ngôn ngữ khác như java, python,... nên em lựa chọn để sử dụng cho phần xử lý ảnh trong đồ án. Trong khi đó CUDA là thư viện cho phép thực hiện tính toán song song trên các thiết bị phần cứng do NVIDIA phát triển. Kiến trúc CUDA hỗ trợ mọi chức năng tính toán thông qua ngôn ngữ C. Để cài đặt YOLO sử dụng GPU cho tính toán cần cái đặt thư viện này.

**Công cụ chuẩn bị dữ liệu:** [**LabelImg**](https://github.com/tzutalin/labelImg)**:**

Đây là công cụ hỗ trợ anotation nhanh chóng và hiệu quả, hỗ trợ các định dạng cho huấn luyện dùng trong các framework phổ biến như XML, PASCAL VOC.

**Ngôn ngữ lập trình Python và C:**

Do Darknet sử dụng ngôn ngữ lập trình C nên em sử dụng ngôn ngữ này cho việc huấn luyện và thử nghiệm. Ngoài ra ngôn ngữ python được em sử dụng cho quá trình chuẩn bị dữ liệu cũng như thống kê kết quả của đồ án.

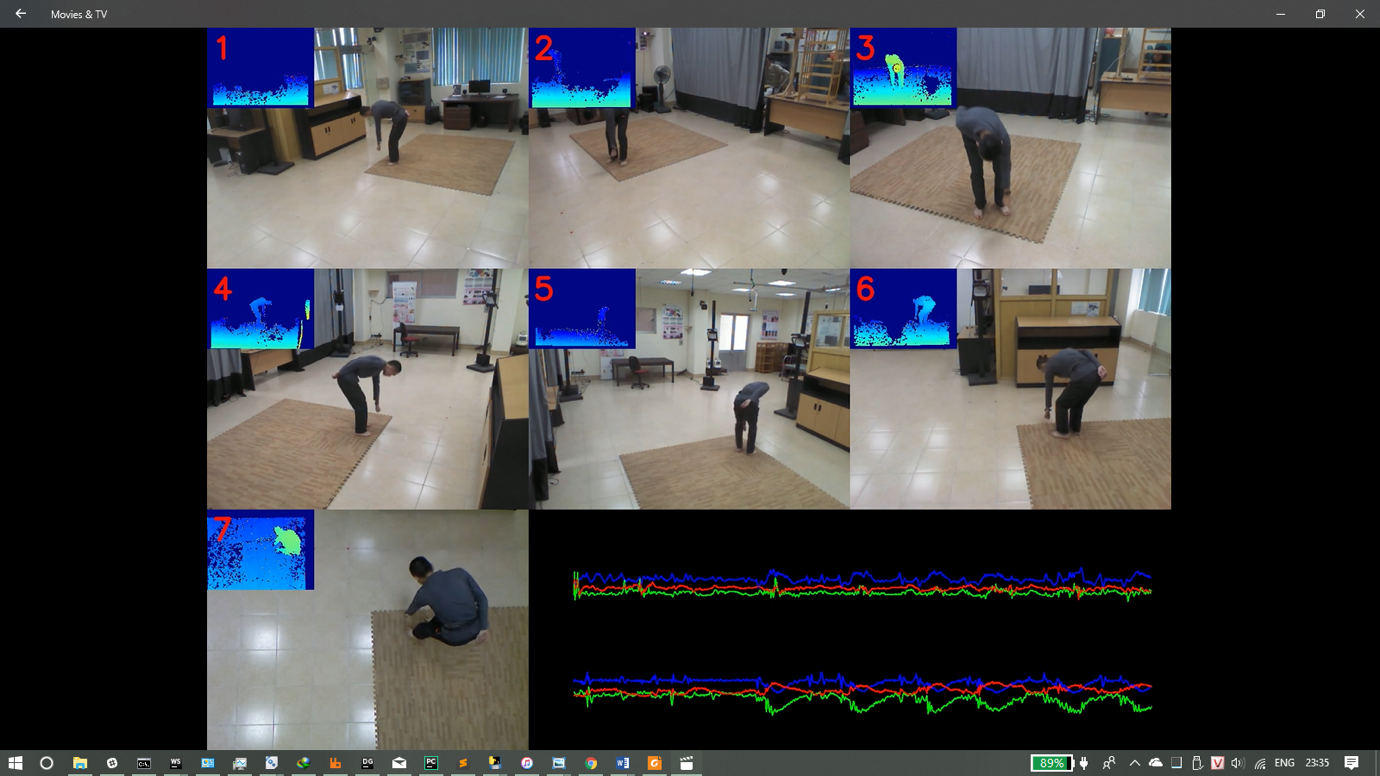
**Cấu hình máy tính sử dụng để huấn luyện và thử nghiệm:** 12 core CPU 2.5GHz, GPU GeForce GTX 970, 4GB Vram.

## 3.2. Dữ liệu và thước đo đánh giá

### 3.2.1. Bộ cơ sở dữ liệu

Bộ cơ sở dữ liệu MICA được em sử dụng để đánh giá mô hình đề xuất. Bộ dữ liệu MICA bao gồm các video được ghi hình trong điều kiện trong phòng thí nghiệm, các camera được gắn cố định và ghi hình theo 7 góc khác nhau theo dõi hoạt động của người. Đây là bộ dữ liệu multiview lớn với số lượng người tham gia thu thập dữ liệu là 50 người, với 21 hành động khác nhau được sử dụng để đánh giá, thử nghiệm với các bài toàn phát hiện, nhận dạng hành động của người. Dữ liệu được thu thập bởi camera Kinect với độ phân giải 640x480, bao gồm dữ liệu ảnh màu RGB, dữ liệu ảnh độ sâu và dữ liệu cảm biến gia tốc gắn trên người.

Dưới đây là hình minh họa bộ cơ sở dữ liệu MICA thể hiện các dữ liệu ảnh màu, ảnh độ sâu và dữ liệu gia tốc.



Hình 1: Các dữ liệu trong bộ dữ liệu MICA

Để phục vụ bài toán phát hiện người, em đã trích xuất một phần trong bộ cơ sở dữ liệu MICA để phục vụ huấn luyện và thử nghiệm. Dữ liệu này bao gồm các ảnh màu RGB có độ phân giải 640x480 trích xuất từ các clip chứa ảnh màu.

Bộ dữ liệu này được trích xuất từ clip của 15 người/50 người trong bộ cơ sở dữ liệu, bao quát 21 hành động của người bao gồm:

1: Đi bộ theo các hướng

2: Chạy chậm

3: Nhảy tại chỗ

4: Cử động tay và đầu gối

5: Cúi xuống nhặt đồ bằng tay trái

6: Cúi xuống nhặt đồ bằng tay phải

7: Đi loạng choạng

8: Ngã về phía trước

9: Ngã về phía sau

10: Ngã về bên trái

11: Ngã về bên phải

12: Bò

13: Ngồi lên ghế rồi đứng lên

14: Di chuyển ghế sang vị trí khác

15: Ngã về bên trái khi đang ngồi trên ghế

16: Ngã về bên phải khi đang ngồi trên ghế

17: Ngồi lên giường rồi đứng lên

18: Nằm lên giường rồi ngồi dậy

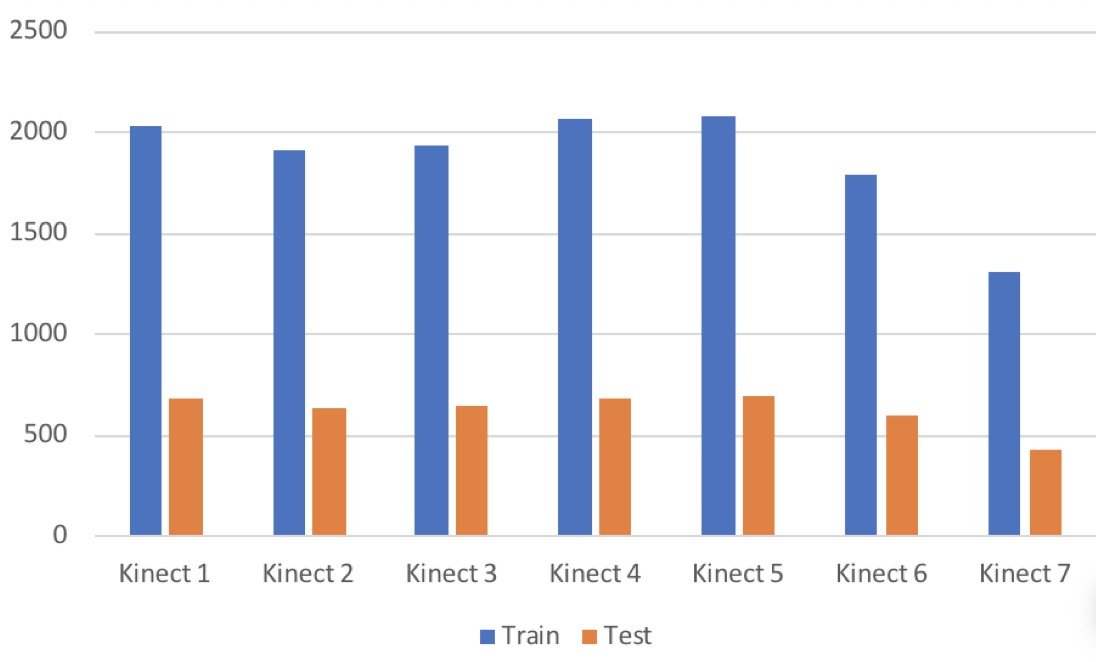
19: Ngã xuống về bên trái khi đang nằm trên giường

20: Nằm ngược lại, ngã xuống về bên phải khi đang nằm trên giường

Bộ dữ liệu sau đó được ground truth bằng tay và được chia làm 2 phần cho giai đoạn huấn luyện và kiểm thử phương pháp trong đó.

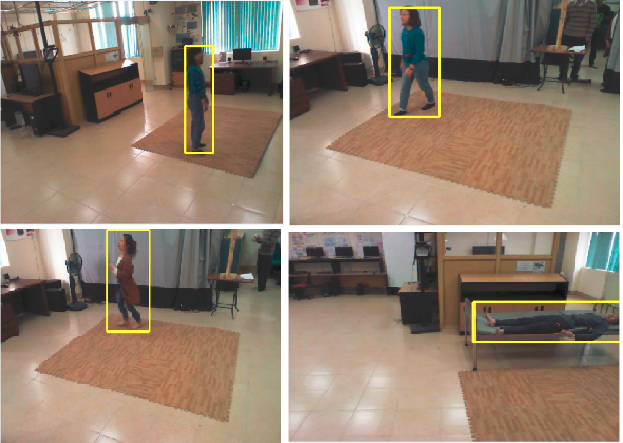
Số lượng ảnh cho huấn luyện: **13165**

Số lượng ảnh cho kiểm thử: **4386**



Hình 8 Phân bố số lượng mẫu cho huấn luyện và kiểm kiểm trử trên từng Kinect

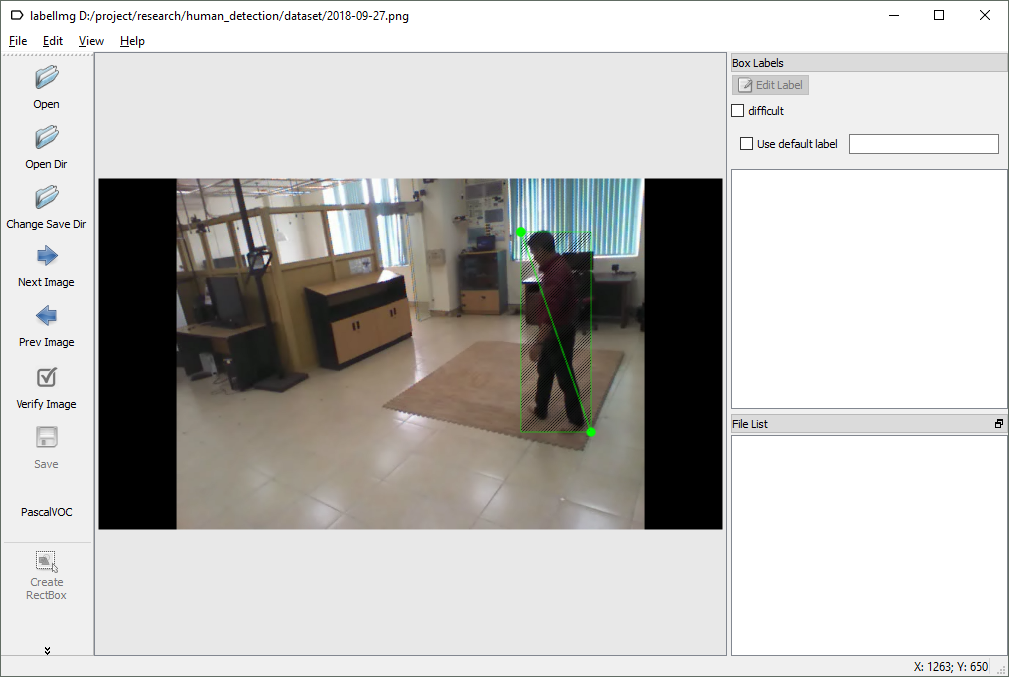
Dưới đây là một số hình ảnh minh họa dữ liệu trong bộ dữ liệu sử dụng trong đồ án này:



Hình 9. Minh họa cơ sở dữ liệu cho huấn luyện và thử nghiệm (bounding box màu vàng thể hiện ground truth của dữ liệu)

### 3.3.2 Chuẩn bị dữ liệu

Do dữ liệu trong bộ cơ sở dữ liệu MICA chưa được gán nhãn cho huấn luyện và kiểm thử nên cần thực hiện ground truth. Để thực hiện ground truth, em sử dụng công cụ [**LabelImg**](https://github.com/tzutalin/labelImg). Để tải công cụ ta truy cập trang https://tzutalin.github.io/labelImg/ và lựa chọn phiên bản phù hợp với hệ điều hành sử dụng.



Hình 9 Sử dụng LabelImg để ground truth cho dữ liệu

### 3.2.3 Phương pháp đánh giá

Việc đánh giá mô hình nhận dạng dựa trên các thước đo: Precision, Recall, Accuracy.

* Precision hay positive predictive value: là tỷ lệ của số lượng đầu vào được dự đoán chính xác (true positive) trong số được dự đoán là đúng,
* Recall hay sensitivity: là tỷ lệ của số lượng đầu được dự đoán chính xác (true positive) trong số được dự đoán là đúng.



Hình 3. Minh họa cách tính Recall và Precision

Bên cạnh đó em sử dụng Jaccard Index để đánh giá một phát hiện là đúng khi:

## 3.3 Triển khai mô đun phát hiện người.

### 3.3.1 Cài đặt YOLO

**Cài đặt Darknet**:

1. Để tải Darknet ta chạy lệnh sau trên terminal:

git clone https://github.com/AlexeyAB/darknet.git

1. Compile project:

Tùy chỉnh file Makefile cho phù hợp với môi trường cài đặt thử nghiệm. Các thông số cần chú ý bao gồm:

* GPU=1 để có thể build code với thư viện CUDA, cho phép sử dụng GPU để tăng tốc tính toán.
* OPENCV=1 để build với thư viện OpenCV 3.x/2.4.x – cho phép đọc một số định dạng ảnh đặc biệt hoặc để đọc dữ liệu video từ camera kết nối với hệ thống.

Sau khi tùy chỉnh thông số xong, chạy lệnh make để build code.

**Thiết lập các thông số cho mạng YOLO**

* 1. Thiết lập file config

Để thiết lập thông số về kiến trúc mạng, số lớp, cấu tạo các lớp trong mạng, ta tạo một file person.cfg. Các thông số cần chú ý bao gồm:

* batch, subdivisions để cài đặt số lượng ảnh được sử dụng cho mỗi bước huấn luyện/kiểm thử. Số lượng ảnh được sử dụng sẽ là: batch/subdivisions
* classes là số lớp cần huấn luyện. Ở bài toán phát hiện người thì số lớp là 1.
* filter = (classes + 5)\*5. Trong trường hợp số lớp là 1 thì filter = 30.
* Các thông số khác ta sử dụng giống như file config mẫu của mạng YOLO.
  1. Tạo file person.names chứa tên các lớp đối tượng cần phát hiện. Ở bài toán này ta có 1 lớp là person.
  2. Tạo file person.data có nội dung như bên dưới. Trong đó train và valid là đường dẫn đến file ground truth của bộ dữ liệu. backup là đường dẫn chứa file lưu các trọng số của mạng sau huấn luyện

classes= 1

train = data/train.txt

valid = data/test.txt

names = data/person.names

backup = backup/

**Huấn luyện mạng YOLO cho phát hiện người**

Để quá trình huấn luyện nhanh hơn, ta tải và sử dụng file đã train bằng mạng YOLO trên cơ sở dữ liệu Imagenet bằng câu lệnh sau:

wget <https://pjreddie.com/media/files/darknet53.conv.74>

Bắt đầu huấn luyện bằng các sử dụng câu lệnh:

./darknet detector train train/data/person.data train/data/person.cfg darknet19\_448.conv.23 -dont\_show

### 3.3.2 Kết quả kiểm thử

### Yolo nguyên bản

Trước tiên để đánh giá độ hiểu quả của công việc huấn luyện và thử nghiệm phát hiện người đa hương. Em tiến hành đánh giá YOLO nguyên bản được huấn luyện trên bộ cơ sở dữ liệu COCO trên bộ cơ sở dữ liệu kiểm thử em chuẩn bị.

Kết quả được thể hiện trên bảng 2 cho thấy recall và precision chưa được cao. Nguyên nhân là do trọng số mạng được huấn luyện trên bộ cơ sở dữ liệu COCO chứa nhiều lớp đối tượng khác nhau. Trong khi bộ dữ liệu thử nghiệm chỉ bao gồm đối tượng người và chứa nhiều tư thế phức tạp.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Kinect** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Số phát hiện đúng** | 683 | 631 | 646 | 733 | 698 | 605 | 438 |
| **Tổng số phát hiện** | 976 | 853 | 710 | 852 | 851 | 695 | 859 |
| **Recall** | 0.7 | 0.74 | 0.91 | 0.86 | 0.82 | 0.87 | 0.51 |
| **Precision** | 0.79 | 0.53 | 0.64 | 0.64 | 0.6 | 0.67 | 0.91 |

Bảng 2. Kết quả đạt được khi thử nghiệm với yolo nguyên bản

### Yolo đã huấn luyện

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Kinect** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Số phát hiện đúng** | 683 | 622 | 640 | 728 | 694 | 597 | 435 |
| **Tổng số phát hiện** | 962 | 631 | 646 | 734 | 696 | 605 | 437 |
| **Recall** | 1 | 0.99 | 0.99 | 0.98 | 1 | 0.99 | 0.99 |
| **Precision** | 1 | 0.99 | 0.99 | 0.99 | 0.99 | 0.98 | 0.99 |

Bảng 3 Kết quả dạt được khi thử nghiệm với yolo đã huấn luyện

**Yolo huấn luyện lại với từng kinect**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Kinect** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Số phát hiện đúng** | 683 | 620 | 640 | 727 | 687 | 598 | 437 |
| **Tổng số phát hiện** | 962 | 631 | 646 | 734 | 696 | 605 | 438 |
| **Recall** | 1 | 0.98 | 0.99 | 0.99 | 0.99 | 0.99 | 1 |
| **Precision** | 1 | 0.99 | 1 | 0.99 | 0.99 | 0.98 | 1 |

Bảng 4 Kết quả đạt được khi thử nghiệm với yolo đã huấn luyện với từng kinect

# CHƯƠNG 4: KẾT LUẬN

## Kết quả đạt được

Trong quá trình thực hiện ĐATN, em đã tìm hiểu và làm chủ được một số kỹ thuật trong lĩnh vực thị giác máy tính và học máy, cụ thể là các phương pháp phát hiện đối tượng nói chung và phát hiện người nói riêng, các phương pháp sử dụng deep learning cho bài toán phát hiện đối tượng. Bên cạnh đó em đã tìm hiểu và cài đặt mạng nơron tích chập theo kiến trúc YOLO cho phát hiện người. Sử dụng ngôn ngữ lập trình C/C++ và Python để cài đặt mạng và xử lý dữ liệu. Kết quả thu được đáp ứng được yêu cầu đặt ra từ đầu đó là phát hiện người trên video đa hướng.

## Những điểm còn hạn chế

* Chưa tận dụng hết các dữ liệu từ bộ dữ liệu, dữ liệu độ sâu cũng có thể sử dụng để phát hiện người.
* Chưa đánh giá được tốc độ khi xử lý đồng thời nhiều video đa hướng.
* Công việc mới dừng lại ở mức tìm hiểu, và cài đặt lại mạng YOLO, chưa tùy biến mạng để tối ưu cho bài toán đặt ra.
* Công việc thử nghiệm mới dừng lại ở bộ dữ liệu MICA, cần thử nghiệm thêm trên các bộ dữ liệu khác.

## Hướng phát triển

Từ những điểm đạt được và những hạn chế được nêu ra, hướng phát triển của ĐATN này là cải thiện độ chính xác của việc tính toán bằng cách kết hợp các dữ liệu khác trong khi tính toán như là kết hợp sử dụng dữ liệu độ sâu từ camera Kinect để tăng tốc độ tính toán. Kết hợp sử dụng các phương pháp trích xuất đặc trưng bằng tay để tăng tốc độ trong các trường hợp đối tượng trong hình có điều kiện tốt. Ngoài ra, em sẽ nghiên cứu thêm các phương pháp tùy chỉnh các thông số của mạng YOLO để tối ưu cho bài toán phát hiện người.

# TÀI LIỆU THAM KHẢO

1. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi: *You Only Look Once: Unified, Real-Time Object Detection*
2. Navneet Dalal and Bill Triggs: *Histograms of Oriented Gradients for Human Detection*.
3. Hong Liu, Tao Xu, Xiangdong Wang, and Yueliang Qian: *Related HOG Features for Human Detection Using Cascaded Adaboost and SVM Classifiers*
4. Lu Xia, Chia-Chih Chen and J. K. Aggarwal*: Human Detection Using Depth Information by Kinect*
5. James W. Davis Vinay Sharma: *Robust Background-Subtraction for Person Detection in Thermal Imagery⁄*
6. D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. *Learning spatiotemporal features with 3d convolutional networks.* In ICCV, 2015.
7. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg. SSD: *Single Shot MultiBox Detector*
8. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna. *Rethinking the Inception Architecture for Computer Vision*
9. Jifeng Dai, Yi Li, Kaiming He, Jian Sun. R-FCN: *Object Detection via Region-based Fully Convolutional Networks*
10. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: *Towards Real-Time Object Detection with Region Proposal Networks*
11. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich*. Going deeper with convolutions. CoRR, abs/1409.4842, 2014*
12. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015.
13. M. Lin, Q. Chen, and S. Yan. *Network in network. CoRR,  
    abs/1312.4400, 2013.*