HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN 1

ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

ĐỀ TÀI:
PHÁT HIỆN ĐỐI TƯỢNG TRONG CAMERA AN NINH DỰA TRÊN HỌC SÂU

Giảng viên hướng dẫn: TS. NGUYỄN NGỌC ĐIỆP
Sinh viên thực hiện: PHẠM VĂN TRÌNH
Lớp: D14CQAT03 – B
Khóa: 2014 – 2019
Hệ: ĐẠI HỌC CHÍNH QUY

Hà Nội, 2018

LỜI CẢM ƠN

Lời đầu tiên em xin gửi lời biết ơn đến cha mẹ, gia đình, những người đã luôn bên em, tạo điều kiện và ủng hộ tinh thần và vật chất để em có thể đi qua 4 năm đại học.

Em xin gửi lời cảm ơn sâu sắc đến Thầy giáo TS.Nguyễn Ngọc Điệp, người thầy đã tận tình chỉ bảo, định hướng và hướng dẫn em trong suốt quá trình học tập và thực hiện đồ án này, đồng thời cũng giúp em hình dung một cách sơ lược về nghiên cứu khoa học.

Em xin cảm ơn đến các thầy, cô trong Khoa Công Nghệ Thông Tin 1 và đặc biệt là các thầy cô trong bộ môn An Toàn Thông Tin, đã dìu dắt động viên em trong suốt 4 năm theo học dưới mái trường.

Em cũng xin cảm ơn đến Phòng Digital Factory của Tập Đoàn Golden Gate đã tạo mọi điều kiện về thời gian để em có thể hoàn thành đồ án của mình.

Với trình độ hiểu biết còn nhiều hạn chế của bản thân, nên trong đồ án không tránh khỏi nhiều thiếu sót. Em mong nhận được những sự góp ý của các Thầy, các Cô, để đồ án của em được hoàn thiện hơn.

Xin chân thành cảm ơn !

Hà Nội , tháng 12 năm 2018

Sinh Viên

Phạm Văn Trình

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM
(Của người hướng dẫn)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm: ..... (bằng chữ: .....)

Đồng ý/Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?

……., ngày … tháng … năm 2018

CÁN BỘ - GIẢNG VIÊN HƯỚNG DẪN

(Ký, họ tên)

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM
(Của người phản biện)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm: ..... (bằng chữ: .....)

Đồng ý/Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?

……., ngày … tháng … năm 2018

CÁN BỘ - GIẢNG VIÊN PHẢN BIỆN

(Ký, họ tên)

MỤC LỤC

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH VẼ

CÁC TỪ VIẾT TẮT

TT	Ký hiệu viết tắt	Tên viết đầy đủ và giải nghĩa
1	YOLO	You Only Look Once Mô hình phát hiện YOLO
2	DPM	Deformable Part Model Mô hình phát hiện DPM
3	IoU	Intersection over Union
4	RPN	Region Proposal Network Mạng đề xuất vùng
5	mAP	mean Average Precision Độ chính xác trung bình
6	HSV	Hue Saturation Value Không gian màu HSV
7	CNN	Convolutional Neural Networks Mạng nơ-ron tích chập
8	ILSVRC	In Large Scale Visual Recognition Challenge Cuộc thi nhận dạng hình ảnh quy mô lớn
9	RoI	Region of Interest
10	FFT	Fast Fourier Transform Biến đổi Fourier nhanh
11	BGR	Blue Green Red Không gian màu BGR
12	RGB	Red Green Blue Không gian màu RGB
13	VGG	Visual Geometry Group Mô hình phân loại hình ảnh
14	VOC	Visual Object Classes Cuộc thi phân loại hình ảnh
15	FPS	Frame Per Sencond Số khung hình trên một giây
16	R-CNN	Region-Convolutional Neural Networks Mô hình phát hiện R-CNN
17	Fast R-CNN	Fast Region-Convolutional Neural Networks Mô hình phát hiện Fast R-CNN
18	Faster R-CNN	Faster Region-Convolutional Neural Networks Mô hình phát hiện Faster R-CNN

LỜI MỞ ĐẦU

Đất nước chúng ta đang bước vào kỷ nguyên của công nghệ. Cuộc cách mạng 4.0 đang được diễn ra từng ngày để nhằm giúp hỗ trợ người dân có thể có cuộc sống tốt hơn. Trong đó vấn đề thị giác máy tính đang là những chủ đề nóng trong cuộc cách mạng này. Thị giác máy tính là một hướng đi mới được nghiên cứu và phát triển trong lĩnh vực công nghệ thông tin trong những năm gần đây. Việc tiếp cận và nghiên cứu các lĩnh vực liên quan trong thị giác máy tính là một việc làm có ý nghĩa khoa học và thực tiễn.

Chính vì thế thị giác máy tính đóng một vai trò cực kì quan trọng trong việc phát triển và hỗ trợ con người trong cuộc sống.

Ý thức được những lợi ích mà thị giác máy tính mang lại vì thế nội dung đồ án là "Phát hiện đối tượng trong camera an ninh dựa trên học sâu" để từ đó có thể giải quyết phần nào bài toán về phát hiện đối tượng đang rất được quan tâm hiện nay.

Trong phạm vi đồ án sẽ trình bày một hệ thống phát hiện đối tượng trong video ở quy mô nhỏ. Có thể phát hiện được các đối tượng trong video dựa vào thuật toán YOLO, một trong những thuật toán có độ chính xác cao và đặc biệt có thời gian phát hiện nhanh. Mục tiêu của của hệ thống là có thể phát hiện nhanh nhất đối tượng có trong video.

Với mục tiêu này, nên đồ án sẽ được chia thành:

Chương 1: Giới thiệu về bài toán phát hiện đối tượng, giới hạn phạm vi của đề tài. Nêu những phương pháp tiêu biểu đã được sử dụng trước đây, nêu ra sơ bộ nội dung của từng phương pháp.

Chương 2: Tổng quan về YOLO và các bước mà YOLO dùng để có thể phát hiện được đối tượng.

Chương 3: Xây dựng chương trình và kết quả đạt được. Nhận xét và phân tích kết quả thu được. Hướng phát triển tiếp theo.

CHƯƠNG 1 : GIỚI THIỆU

1.1. Giới thiệu về bài toán phát hiện đối tượng

Con người khi nhìn một hình ảnh sẽ biết được những đối tượng nào trong hình ảnh "Bức ảnh chụp ở đâu ? Có những gì trong bức ảnh". Thị giác của con người rất nhanh nhạy và chính xác, nó cho phép con người có thể tự lái xe ô tô hay làm những công việc thường ngày.

Phát hiện đối tượng là một trong những công việc thiết yếu trong hệ thống thị giác máy tính. Cụ thể trong hệ thống xe tự lái thì nhiệm vụ phát hiện người đi bộ, phương tiện giao thông, biển báo giao thông có thể giúp chiếc xe tự lái có thể hoạt động an toàn như khi con người vận hành. Tuy nhiên phát hiện đối tượng là một trong những công việc khó khăn và thách thức, bởi các đối tượng trong hình ảnh có thể bị ảnh hưởng bởi nhiều yếu tố như ánh sáng, quy mô và hoạt động.

Nhận thấy trong thực tế một thực trạng đó là trong mỗi tòa nhà chung cư, chúng ta sẽ tính trung bình mỗi tòa chung cư sẽ có 20 tầng và đi kèm với đó là hơn 20 camera an ninh ghi hình 24/7. Kéo theo là phần cứng để lưu trữ những file video mà những camera an ninh này ghi lại được là rất lớn. Và khi muốn kiểm tra và xem lại những đoạn ghi hình thì người sử dụng cần phải bỏ nhiều công sức để tua những đoạn ghi hình không có giá trị sử dụng. Trước vấn đề đặt ra này, đồ án sẽ giới thiệu chương trình giúp phát hiện đối tượng và cắt bớt những đoạn video dư thừa.

1.2. Giới hạn phạm vi

Bài toán phát hiện đối tượng là một bài toán khá rộng. Nhưng giới hạn của đề tài này gồm những phạm vi sau.

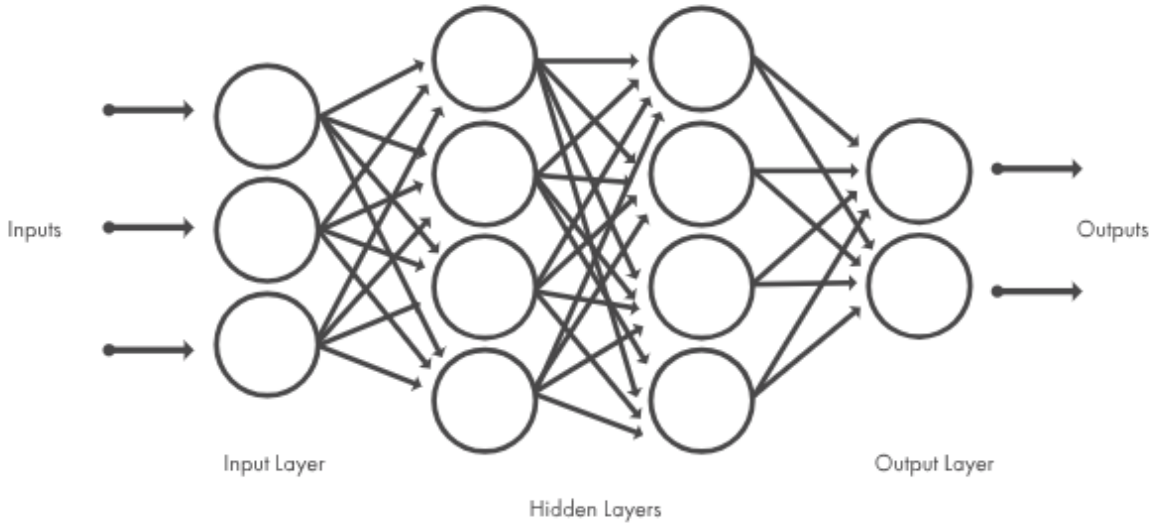
- *Phạm vi địa lý*: Hệ thống sẽ sử dụng để phát hiện tại một chung cư sử dụng camera an ninh.
- *Phạm vi đối tượng*: Hệ thống sẽ phát hiện đối tượng là người và các đồ vật nguy hiểm như dao, kéo.
- *Phạm vi thời gian*: Thực hiện trong khoảng thời gian camera an ninh hoạt động.
- *Phạm vi đầu vào*: Video thu được từ camera an ninh.

1.3. Giới thiệu về học sâu

Học sâu hay còn được gọi là "Deep Learning" là một phạm trù nhỏ của học máy "Machine Learning". Học sâu tập trung giải quyết các vấn đề liên quan đến mạng nơ-

ron nhân tạo, nhằm nâng cấp các công nghệ như nhận diện giọng nói, thị giác máy tính và xử lý ngôn ngữ tự nhiên. Cụ thể trong đồ án này phương pháp học sâu sẽ được ứng dụng trong thị giác máy tính.

Trong [8], học sâu là một lĩnh vực chuyên sâu của học máy và nó giải quyết các vấn đề thực tế bằng cách khai thác các mạng nơ-ron nhân tạo. Nó mô phỏng việc đưa ra các quyết định của con người. Học sâu sử dụng một mạng lưới nơ-ron để bắt chước trí thông minh của động vật. Đa phần các mô hình học sâu sử dụng kiến trúc mạng nơ-ron.



Hình 1-1: Mô hình kiến trúc mạng nơ-ron sử dụng trong học sâu

Những nơ-ron trong mạng nơ-ron được nhóm thành 3 layer khác nhau là Input Layer, Hidden Layer và Output Layer

- Input Layer: Nhận các input data.
- Hidden Layers: Thực hiện các tính toán toán học dựa trên outputs đề ra. Một trong những lưu ý khi xây dựng mô hình dựa trên học sâu là việc quyết định số lượng các hidden layers và số lượng nơ-ron cho mỗi layer.
- Output Layer: Thực hiện tổng hợp các kết quả từ hidden layers và sau đó đưa ra kết quả của mô hình.

Việc khó nhất của học sâu đó là việc huấn luyện mô hình. Tại vì nếu muốn kết quả đầu ra có độ chính xác cao thì cần phải có bộ dữ liệu huấn luyện lớn và phần cứng ổn định.

Học sâu được ứng dụng để đáp ứng yêu cầu của con người trong đời sống. Cụ thể nó được ứng dụng trong thị giác máy tính về việc phát hiện đối tượng và nhận dạng

đối tượng, ứng dụng trong nhận diện giọng nói để tăng cường một lớp bảo mật cho người dùng hoặc để sử dụng giọng nói cho việc điều khiển máy móc. Đối với ứng dụng sử dụng học sâu trong việc xử lý ngôn ngữ tự nhiên bao gồm phân tích cú pháp, phân tích ngữ nghĩa, phân tích hình thái…của văn bản.

Theo [9], ưu điểm của học sâu là

- Có hiệu quả tương đối tốt về việc giải quyết các vấn đề trong các lĩnh vực như xử lý ảnh, xử lý ngôn ngữ, trò chơi…
- Thời gian huấn luyện tương đối dài nhưng tốn ít thời gian khi vận hành.

Nhược điểm của học sâu là

- Yêu cầu lượng dữ liệu rất lớn.
- Mất nhiều thời gian và thiết bị phần cứng để thực hiện huấn luyện.
- Học sâu tích hợp nhiều kiến thức từ các lĩnh vực khác nhau như thống kê, lập trình…nên khiến cho việc hiểu tường tận lý thuyết của nó là điều khó khăn

Từ những ưu điểm và nhược điểm được nêu ở trên và dựa theo yêu cầu của bài toán là phát hiện đối tượng trong trong video, nên đồ án quyết định thực hiện dựa trên học sâu.

1.4. Các mô hình phát hiện đối tượng đã được phát triển

1.4.1. Sự phát triển của phát hiện đối tượng trong hình ảnh

Theo [4] trước năm 2012 các phương pháp phát hiện đối tượng dựa vào học sâu thì đa phần theo mô hình " đặc trưng – trích – cộng – phân loại ". Trước tiên mọi người cần phải xác định đặc điểm cụ thể cho một loại đối tượng để đại diện chính xác cho loại đối tượng ấy. Ví dụ như khi cần xác định "con mèo" trong ảnh, thì cần xác định đúng dấu hiệu của con mèo, đó có thể là có bốn chân, lông màu đen hay có hai cái tai. Sau khi trích xuất đủ các đặc điểm từ tập dữ liệu huấn luyện thì đối tượng có thể được biểu diễn bằng một vector được sử dụng để huấn luyện, và cũng có thể được sử dụng để thực hiện nhiệm vụ phát hiện đối tượng trong thời gian thử nghiệm. Nếu muốn xây dựng một hệ thống phát hiện nhiều loại đối tượng thì cần phải cân nhắc nhiều hơn trong việc chọn đặc điểm chung để phù hợp với các loại đối tượng khác nhau. Vì việc định nghĩa các đặc trưng thường rất khó khăn và phức tạp khiến cho mô hình khó mở rộng khi một loại

đối tượng mới được thêm vào danh sách cần phát hiện. Bên cạnh đó độ chính xác của phát hiện cũng không đạt được yêu cầu đặt ra.

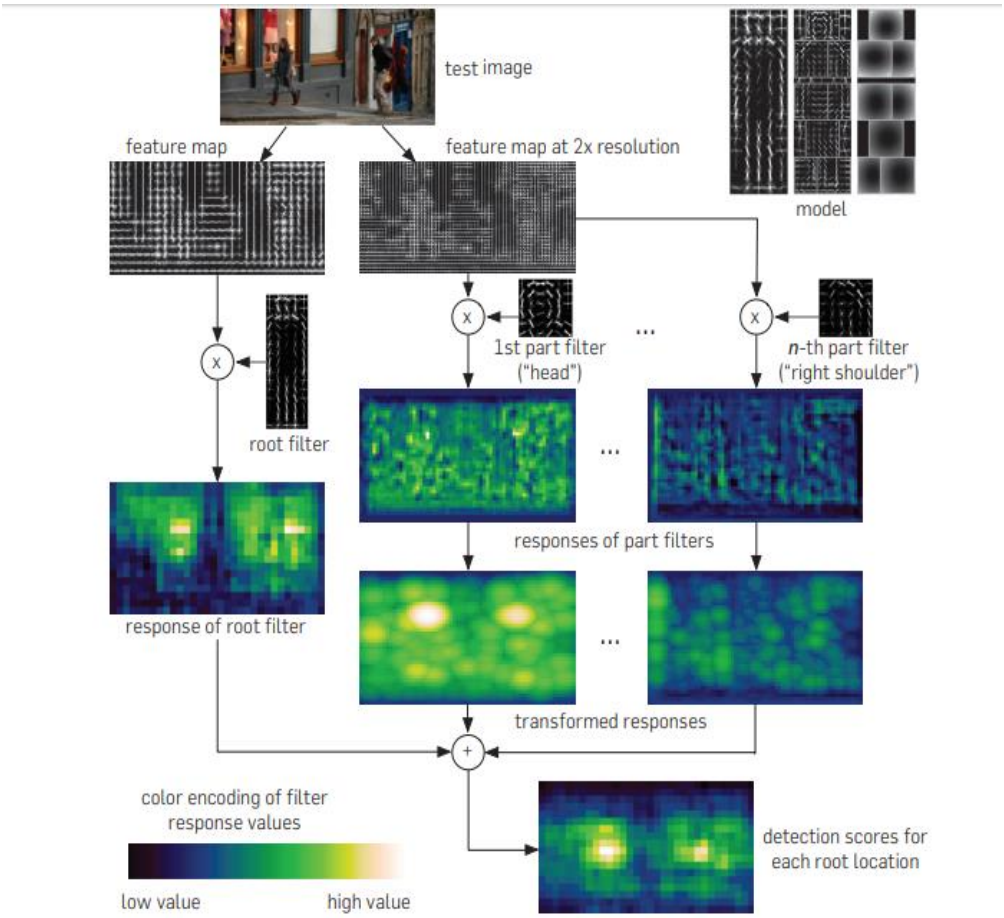
Theo [7] trong cuộc thi In Large Scale Visual Recognition Challenge 2012 (ILSVRC, 2012), mô hình dựa trên CNN của Krizhevsky đã vượt trên tất cả các mô hình khác. Cấu trúc của một mô hình tương tự như CNN của Krizhevsky cũng được trình bày vào những năm 1990, nhưng sức mạnh của nó khi đó đã không được thể hiện hết do thiếu đi các hình ảnh huấn luyện và phần cứng chưa đáp ứng được. Trong ILSVRC(2012) một tập con của tập dữ liệu ImageNet được sử dụng để phân loại 1,2 triệu hình ảnh vào 1000 loại. Với đầy đủ hình ảnh dùng để huấn luyện và GPU mạnh mẽ nên đã chứng tỏ khả năng mạnh mẽ của CNN trong phân loại hình ảnh. Ưu điểm chính của CNN so với các phương pháp truyền thống là khả năng xây dựng các bộ lọc đặc trưng trong quá trình huấn luyện. Từ đó trở đi CNN trở thành công cụ chính cho nhiệm vụ phân loại hình ảnh. Và hiện nay đã có kết quả tốt trong các nhiệm vụ liên quan đến phân loại hình ảnh. Kể từ khi CNN chứng minh lợi thế rất lớn của nó trong việc phân loại hình ảnh thì mọi người bắt đầu đưa ra những biến thể dựa trên mô hình CNN. Công thức đơn giản cho mô hình phân loại – cộng – phân loại là đính kèm một nhóm các lớp được kết nối đầy đủ vào mô hình CNN hiện tại. Rõ ràng các phương pháp dựa trên CNN làm rất tốt trong nhiệm vụ phân loại – cộng – phân loại.

Gần đây một mô hình phát hiện đối tượng mang tên YOLO đã được Joseph cùng các đồng sự giới thiệu. Mô hình có thể từ hình ảnh đầu vào sẽ tạo ra một tensor đại diện cho điểm số của lớp đối tượng và vị trí của đối tượng. Các hình ảnh đầu vào chỉ cần đi qua mạng một lần và kết quả được đưa ra. YOLO đã đạt được độ chính xác trên 50% trong việc phát hiện thời gian thực trên bộ dữ liệu VOC 2007 và 2012 và khiến nó trở thành lựa chọn tốt cho bất cứ bài toán phát hiện đối tượng theo thời gian thực nào.

Sau đây đồ án sẽ khảo sát các phương pháp phát hiện đối tượng để đưa ra phương pháp sử dụng trong đề tài này.

1.4.2. Mô hình DPM

Mô hình DPM [2] được viết tắt của Deformable Parts Model. Mô hình DPM sử dụng phương pháp cửa sổ trượt, bộ phân loại đối tượng được chạy ở các vị trí cách đều nhau trên toàn bộ hình ảnh. DPM sử dụng một quá trình phân tách để trích xuất các tính năng tĩnh và phân loại các vùng dự đoán các hộp giới hạn theo các vùng có điểm số cao



Hình 1-2: Quy trình tính điểm để phát hiện đối tượng trong DPM

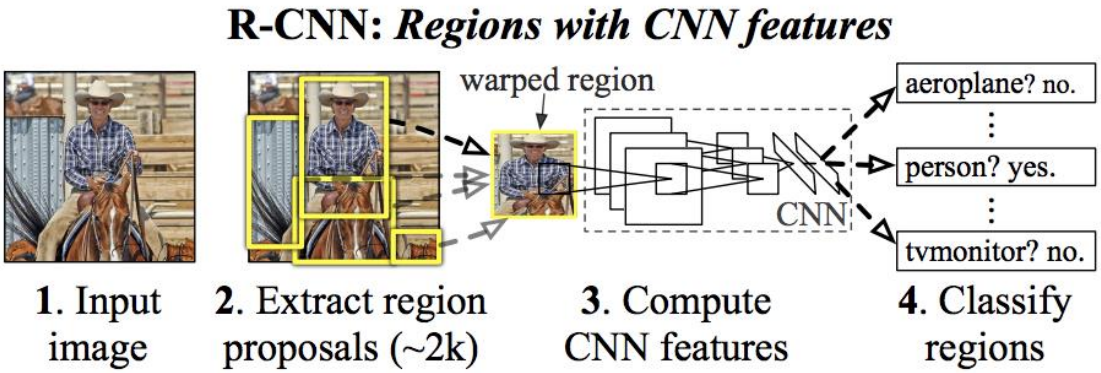
Bảng 1-1: Điểm số AP với các biến thể của DPM trong VOC 2007 (The Fastest Deformable Part Model for Object Detection – 2014 IEEE)

	Plane	**Person**	**Car**	**Boat**	**Table**
DPM	29.2	41.9	54.9	16.5	14.4
Branch-Bound (DPM)	24.1	40.0	53.6	9.1	9.1
Cascale(DPM)	27.6	41.8	55.0	16.6	14.4
FFT(DPM)	30.1	40.9	54.8	15.0	18.1
Coarse-to-fine(DPM)	27.9	30.7	48.3	16.1	21.4
Proposed Method	27.1	41.5	54.1	16.1	14.4

Bảng 1-1 thể hiện điểm số AP đối với các biến thể của DPM trong tập dữ liệu VOC 2007 với các đối tượng phát hiện là Plane, Person, Car, Boat và Table.

1.4.3. Mô hình R-CNN

Hệ thống phát hiện đối tượng R-CNN [10] dựa trên học sâu sử dụng bao gồm ba module. Module đầu tiên tạo ra các đề xuất vùng, những vùng đề xuất này xác định những đối tượng tiềm năng. Module hai là mạng chuyển đổi trích xuất một vector đặc trưng có độ dài cố định từ mỗi vùng. Module ba là tập hợp các SVM tuyến tính theo lớp cụ thể.

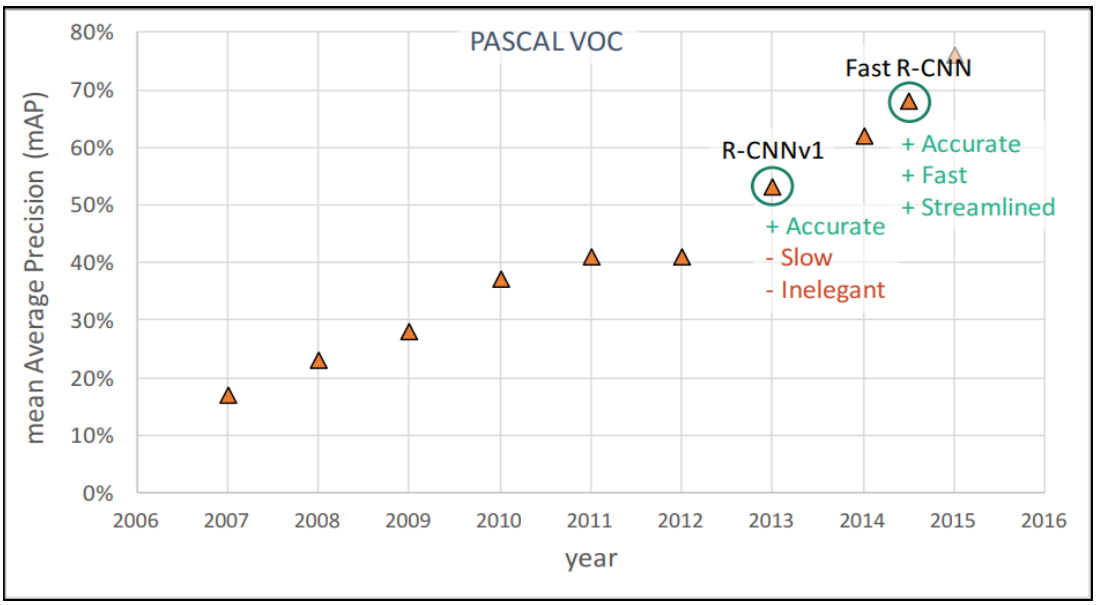


Hình 1-3: Tổng quát hệ thống phát hiện đối tượng

Hệ thống phát hiện đối tượng bao gồm (1) Hình ảnh đầy vào (2) Trích xuất khoảng 2000 đề xuất vùng (3) Tính toán các đề xuất cho mỗi vùng đề xuất sử dụng mạng nơ-ron tích chập (CNN) sau đó (4) phân loại từng khu vực bằng cách sử dụng lớp SVM tuyến tính. R-CNN đạt được độ chính xác trung bình (mAP) là 53,7% trên PASCAL VOC 2010 còn các mô hình DPM đạt được mAP là 33,4%. Trên bộ dữ liệu phát hiện ILSVRC 2013 thì mAP của R-CNN đạt được 31,4% đó là một cải tiến lớn so với OverFeat có kết quả tốt nhất trước đó là 24,3%.

1.4.4. Mô hình Fast R-CNN

Mô hình Fast R-CNN là mô hình dựa trên học sâu và được xây dựng dựa trên R-CNN. Khác nhau cơ bản giữa hai mô hình là trong mô hình Fast R – CNN có thêm một box regression layer để có thể cải thiện ví trí của đối tượng trong hình ảnh và theo [3] lớp gộp RoI (RoI pooling layer) để gộp các CNN feature cho từng đề xuất vùng.



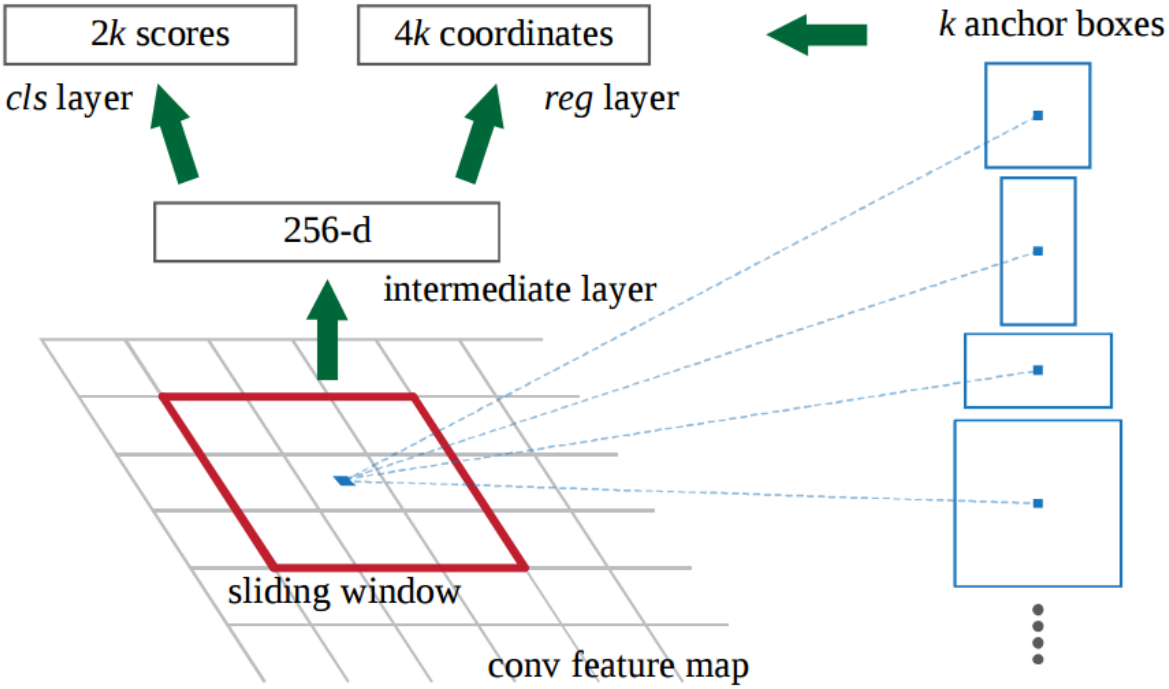
Hình 1-4: Độ chính xác trung bình của các mô hình theo từng năm

(Hình ảnh lấy từ ICCV 15 - International Confenrence on Computer Vision)

Trên hình 1-4 có thể thấy, từ năm 2007 đến năm 2011 mAP đang tăng theo từng năm nhưng từ năm 2011 đến năm 2012 thì bị chững lại. Nhưng từ năm 2013 có sự xuất hiện của CNN đã tạo ra những tiến triển lớn trong chỉ số mAP và đến mô hình Fast R-CNN đã đạt gần 70% mAP với độ chính xác cao và tốc độ phát hiện tương đối nhanh so với các mô hình trước đây.

1.4.5. Mô hình Faster R-CNN

Theo [5] khi mà cả hai R – CNN và Fast R – CNN đều sử dụng tìm kiếm chọn lọc để tìm ra các vùng đề xuất. Tìm kiếm chọn lọc là một quá trình chậm và mất thời gian gây ảnh hưởng đến hiệu suất của mô hình. Do đó Shaoqing Ren và các cộng sự đã đưa ra mô hình Faster R-CNN dựa trên học sâu. Nó có một mạng nơ-ron đề xuất vùng, thay thế cho thuật toán tìm kiếm chọn lọc vùng và nó mang tên RPN.



Hình 1-5: Mô tả quá trình hoạt động của RPN

Bảng 1-2: So sánh về thời gian chạy, tốc độ, và Độ chính xác trung bình của 3 loại R-CNN, Fast R-CNN, Faster R-CNN

	R-CNN	**Fast R-CNN**	**Faster R-CNN**
Test time per image	50 giây	2 giây	0.2 giây
Speedup	1x	25x	250x
mAP (VOC 2007)	66.0	66.9	66.9

Bảng 1-2 so sánh về thời gian chạy, độ chính xác trung bình và tốc độ phát hiện của 3 mô hình phát hiện đối tượng được sử dụng trong tập dữ liệu VOC 2007.

1.4.6. Mô hình YOLO

YOLO là một mô hình phát hiện đối tượng trong ảnh hoặc video dựa trên học sâu được biết đến gần đây. Nó tương tự như các biến thể của R-CNN là sử dụng mạng nơ-ron để phát hiện vị trí của đối tượng và phân loại chúng. Nó được biết đến với tốc độ phát hiện nhanh vì thế nó được sử dụng nhiều trong các mô hình xe tự lái, hay những hệ thống cần xử lý với tốc độ thời gian thực. Và độ chính xác của YOLO cũng ở mức tương đối cao vì nó ít gặp phải những lỗi như các mô hình khác là lỗi Background, do YOLO được thực hiện bằng nguyên lý quét tổng quát trên toàn bộ hình ảnh trong cùng một lúc. Lỗi Background là lỗi do các mô hình phát hiện chỉ tập trung vào một phần của hình ảnh mà không quan tâm đến những phần xung quanh và gây ra việc dự đoán thiếu chính xác.

Đến thời điểm hiện tại YOLO đã trải qua 3 phiên bản, nên nó càng được hoàn thiện về mặt thiết kế và cho ra kết quả tương đối chính xác.

1.5. Mục tiêu của hệ thống phát hiện đối tượng

Mục tiêu của đồ án này là việc phát hiện nhanh đối tượng, cụ thể là người và đồ vật nguy hiểm trong video mà camera an ninh thu được. Sau đó sẽ tạo ra một video output mà những đoạn video không có chứa đối tượng sẽ bị xóa và chỉ giữ lại những đoạn video có đối tượng xuất hiện. Nó giúp làm giảm đi đáng kể bộ nhớ để lưu trữ video ban đầu, thay vào đó chỉ cần lưu lại đoạn video chứa đối tượng, cùng với đó chương trình sẽ đưa ra một tệp txt mang tên Care.txt, nó lưu lại ở khung hình bao nhiêu có tọa độ đối tượng xuất hiện.

Yêu cầu của mô hình phát hiện đối tượng trong camera an ninh là phải có tốc độ phát hiện nhanh và độ chính xác cao. Vì các camera an ninh hoạt động với thời gian là 24/7 nên nó đòi hỏi mô hình được sử dụng phải có tốc độ xử lý thời gian thực, tức là trên 45 FPS. Nhưng ở các mô hình phát hiện mà chúng ta tìm hiểu ở trên thì không đạt được yêu cầu đặt ra này. Cụ thể với mô hình phát hiện Fast R-CNN là 0.5 FPS, Faster R-CNN ZF là 18 FPS, 30 Hz DPM là 30 FPS.

Nhưng với yêu cầu của bài toán này thì mô hình YOLO đã đáp ứng những yêu cầu mà bài toán phát hiện đối tượng trong camera an ninh đưa ra với các lý do sau. Đầu tiên YOLO là một mã nguồn mở hoàn toàn miễn phí, chúng ta có thể cài đặt hoặc lập trình thêm những tính năng khác dựa vào yêu cầu đặt ra. Thứ hai [1] YOLO có tốc độ là 45 FPS và độ chính xác trung bình là 63.4 mAP, hiện nay có thêm một biến thể YOLO mang tên Fast YOLO có tốc độ là 155 FPS và độ chính xác trung bình là 52.7 mAP. Thứ ba là về cách thiết kế của YOLO giúp nó tránh gặp phải loại lỗi Backgroup, vì loại lỗi Backgroup này sẽ khiến cho mô hình phát hiện không nhận ra đối tượng có kích thước lớn hoặc nằm ở vùng biên của khung hình.

1.6. Kết chương

Chương 1 đã giới thiệu khái quát về học sâu, các mô hình phát hiện được sử dụng trước đây, yêu cầu và mục tiêu của hệ thống phát hiện đối tượng dựa trên học sâu. Chương 2 sẽ tiếp nối chương 1, nhưng đi sâu hơn về mô hình YOLO giúp phát hiện đối tượng trong camera an ninh.

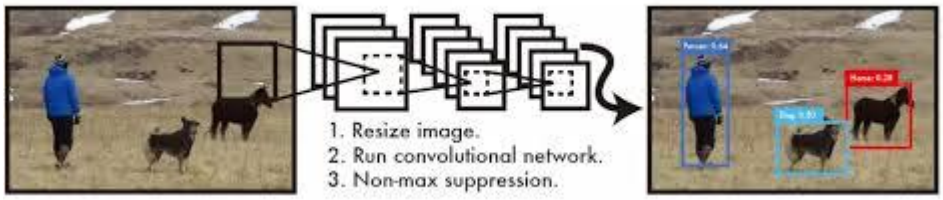
CHƯƠNG 2 : PHƯƠNG PHÁP PHÁT HIỆN ĐỐI TƯỢNG TRONG CAMERA DỰA TRÊN YOLO

2.1. Tổng quát về YOLO

YOLO là một mô hình để phát hiện đối tượng. Nhiệm vụ phát hiện đối tượng bao gồm xác định vị trí của đối tượng đó trên hình ảnh và phân loại đối tượng đó. Các phương pháp trước đây từng làm điều này như là R – CNN và các biến thể của nó. Nó khiến cho hệ thống chạy chậm và khó để tối ưu, bởi vì mỗi phần riêng lẽ phải được huấn luyện riêng biệt. Với YOLO nó hoạt động với mạng nơ-ron duy nhất.

Mô hình này tương tự như mô hình DPM (DPM Deformable Parts Model) sử dụng phương pháp cửa sổ trượt và bộ phân loại được chạy ở các vị trí cách đều nhau trên toàn bộ hình ảnh.

YOLO định vị đối tượng phát hiện dưới dạng hồi quy đơn lẻ, nghĩa là trực tiếp từ Pixel ảnh đến tọa độ hộp giới hạn và xác suất lớp. Sử dụng mô hình YOLO tại một hình ảnh để có thể dự đoán những đối tượng nào xuất hiện trong hình ảnh và chúng xuất hiện ở vị trí nào trong hình ảnh.



Hình 2-1: Hệ thống phát hiện YOLO

Xử lý hình ảnh với YOLO tương đối dễ dàng. Hệ thống của YOLO gồm (1) thay đổi kích thước ảnh đầu vào là 448 x 448 (2) chạy một mạng chuyển đổi đơn trên hình ảnh (3) Đưa ra kết quả dựa trên độ tin cậy của mô hình.

Mô hình YOLO rất đơn giản. Hình 2-1 thể hiện một mạng chuyển đổi đơn đồng thời dự đoán nhiều hộp giới hạn và xác suất lớp. YOLO huấn luyện hình ảnh đầy đủ và trực tiếp tối ưu hiệu suất phát hiện. Mô hình YOLO này có một số lợi ích so với phương pháp học máy truyền thống để phát hiện đối tượng.

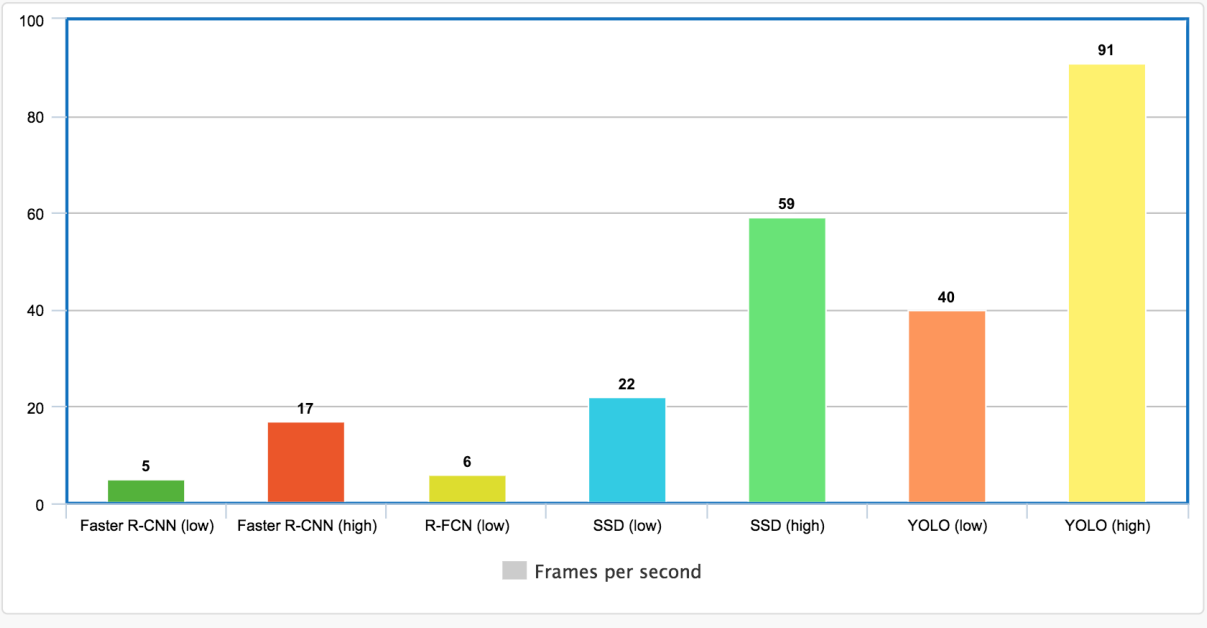
Trong [1] có nêu những điểm mạnh để chúng ta có thể nhìn nhận sơ lược trước về mô hình YOLO, cũng như hiểu sơ qua về sự khác biệt của YOLO với các phương pháp phát hiện đối tượng trước đây.

Thứ nhất YOLO có tốc độ phát hiện đối tượng nhanh. Vì phát hiện khung hình là quá trình hồi quy, nên không cần một quá trình làm việc phức tạp. Chỉ cần chạy mạng nơ-ron trong một ảnh tại thời điểm kiểm tra để dự đoán phát hiện. YOLO chạy ở tốc độ 45 khung hình/giây còn phiên bản Fast YOLO khoảng 150 khung hình/giây. Điều này có nghĩa rằng có thể xử lý video trực tuyến trong thời gian thực với thời gian chờ là dưới 25 mili giây. Hơn nữa YOLO đạt được gấp đôi độ chính xác trung bình của các hệ thống thời gian thực khác tổng lại chia trung bình.

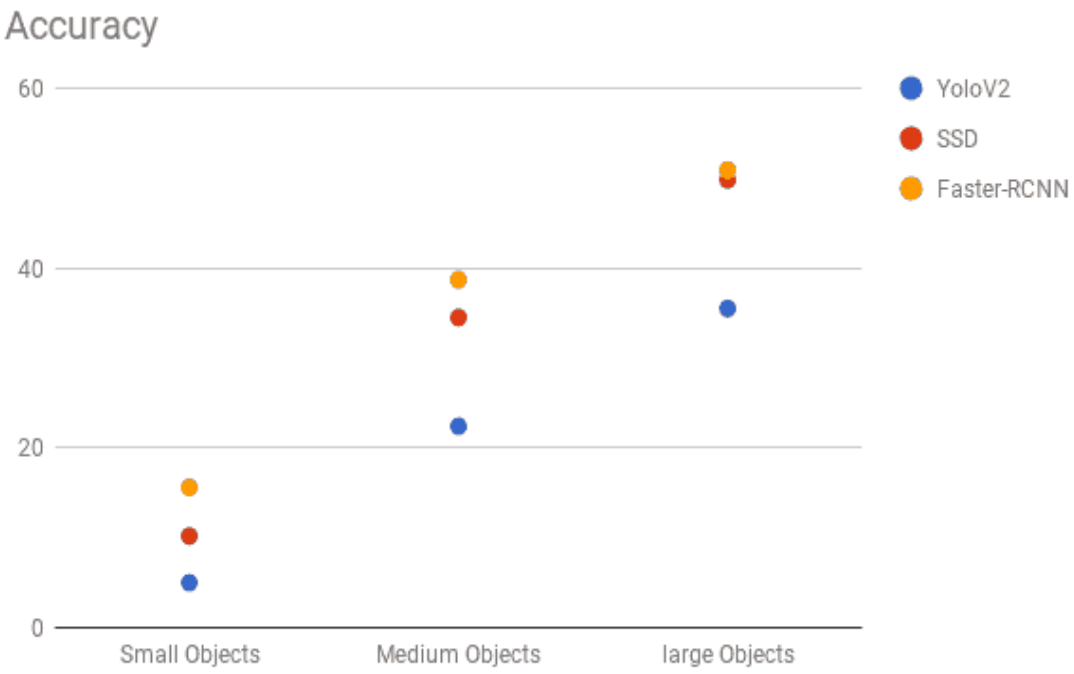
Thứ hai YOLO có độ khái quát cao về hình ảnh khi đưa ra dự đoán. Không như các mô hình sử dụng cửa sổ trượt và các kỹ thuật dựa trên đề xuất khu vực thì YOLO quét toàn bộ hình ảnh trong suốt thời gian huấn luyện và thời gian kiểm tra. Fast R-CNN là một phương pháp phát hiện hàng đầu, nó có một lỗi là nhầm lẫn các mảng nền trong một hình ảnh bởi vì nó không thể quét khu vực ảnh một cách rộng hơn. YOLO mắc phải ít hơn một phần hai số lỗi Background so với Fast R-CNN.

Thứ ba YOLO "học" đặc điểm chung của các đối tượng. Khi được huấn luyện với ảnh tự nhiên (ảnh gốc) và ảnh nghệ thuật thì YOLO hoạt động tốt hơn so với hai phương pháp DPM và Fast R-CNN với tỷ lệ chính xác cao hơn. Vì YOLO có khả năng khái quát cao nên ít bị lỗi khi được áp dụng với đầu vào input không đúng yêu cầu đặt ra.

YOLO vẫn còn kém so với các hệ thống phát hiện đối tượng hiện nay ở tính chính xác. Trong khi đó YOLO có thể nhanh chóng xác định các đối tượng trong hình ảnh mà nó phát hiện và YOLO gặp khó khăn để có thể xác định chính xác một số đối tượng, đặc biệt là các đối tượng có kích thước nhỏ.



Hình 2-2: Tỷ lệ tốc độ phát hiện giữa Faster RCNN, SSD, YOLO



Hình 2-3: Độ chính xác của YOLO, SSD, Faster RCNN thông qua 3 loại đối tượng [4].

Phát hiện đối tượng là một vấn đề cốt lõi trong thị giác máy tính. Các hệ thống phát hiện thường bắt đầu bằng cách trích xuất một tập hợp các đặc trưng mạnh mẽ từ các hình ảnh đầu vào (như Haar, SIFT, HOG, convolutional features). Sau đó các bộ phân loại được sử dụng để xác định đối tượng trong không gian đặc trưng cụ thể. Các bộ phân loại này chạy theo kiểu cửa sổ trượt trên toàn bộ hình ảnh. Dưới đây là bảng so

sánh YOLO với một số mô hình phát hiện hàng đầu để làm nổi bật điểm tương đồng và sự khác biệt chính đối với các mô hình phát hiện này.

So sánh mô hình YOLO với 5 mô hình phát hiện có thành tích tốt đó chính là DPM, R – CNN, DeepMutilbox, OverFeat, MultilGrasp [1].

Bảng 2-1: Các hệ thống thời gian thực trên PASCAL VOC 2007

	Train	**mAP**	**FPS**
Real-Time Detectors			
100Hz DPM	2007	16.0	100
30Hz DPM	2007	26.1	30
Fast YOLO	2007+2012	52.7	155
YOLO	2007+2012	63.4	45
Less Than Real-Time			
Fastest DPM	2007	30.4	15
R-CNN Minus R	2007	53.5	6
Fast R-CNN	2007+2012	70.0	0.5
Faster R-CNN VGG-16	2007+2012	73.2	7
Faster R-CNN ZF	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

So sánh độ chính xác trung bình và tốc độ phát hiện của một số mô hình phát hiện. Các mô hình được huấn luyện theo bộ dữ liệu VOC năm 2007 và 2012. Fast YOLO là mô hình phát hiện đối tượng nhanh nhất trên bộ dữ liệu, đối với bộ dữ liệu VOC PASCAL thì mô hình YOLO vẫn có độ chính xác gấp đôi so với bất kì mô hình phát hiện trong thời gian thực nào khác. YOLO có độ chính xác cao hơn 10 mAP so với Fast YOLO.

YOLO thống nhất những thành phần riêng biệt của mô hình phát hiện đối tượng tạo thành một mạng nơ-ron đơn. Mạng lưới của YOLO sử dụng những đặc trưng và thông số từ toàn bộ hình ảnh để dự đoán toàn bộ hộp giới hạn của hình ảnh. Nó cũng dự đoán toàn bộ hộp giới hạn trên tất cả các lớp cho một hình ảnh cùng một lúc. Điều này giải thích cho lý do tại sao kiến trúc của YOLO thể hiện tính chất tổng quát về hình ảnh đầu vào và các đối tượng trong ảnh. Thiết kế của YOLO cho phép huấn luyện đầu cuối với tốc độ thời gian thực trong khi vẫn duy trì độ chính xác trung bình cao.

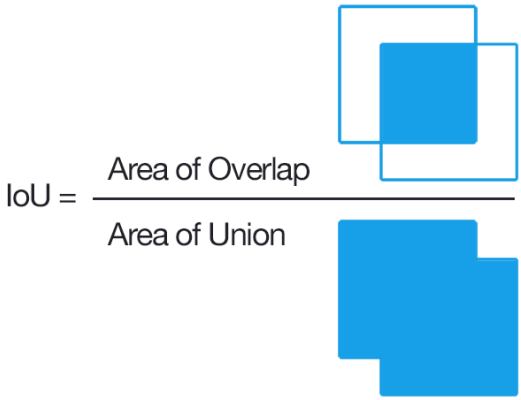
Hệ thống chia hình ảnh đầu vào thành SxS lưới. Nếu trung tâm của một đối tượng rơi vào ô lưới nào thì ô lưới đó chịu trách nhiệm phát hiện đối tượng đó.

Thông thường tùy vào việc cấu hình thì mô hình YOLO thường chia hình ảnh đầu vào thành 13 x 13 ô, hoặc 26 x 26 ô, hoặc 52 x 52 ô.

Hình 2-4: Hình ảnh đầu vào được chia thành 13 x 13 khối

Mỗi ô lưới dự đoán B hộp giới hạn và điểm tin cậy của các hộp đó. Điểm tin cậy phản ánh mức độ tin cậy của mô hình đối với những hộp giới hạn chứa các đối tượng. Định nghĩa độ tin cậy là

$\Pr(Object) * IOU_{pred}^{truth}$. Nếu không có đối tượng nào tồn tại trong ô đó thì điểm tin cậy phải bằng 0. Nếu không YOLO muốn điểm tin cậy phải bằng với điểm giao IOU , IOU là tỷ lệ giữa hộp giới hạn và độ chính xác việc có đối tượng xuất hiện trong hộp giới hạn đó hay không.



Hình 2-5: Thể hiện tính toán tỷ lệ IOU

Tỷ lệ IOU là phép chia giữa phần giao nhau và phần tổng hợp của 2 hộp giới hạn, đó là hộp chứa đối tượng chính xác và hộp dự đoán đối tượng đó.

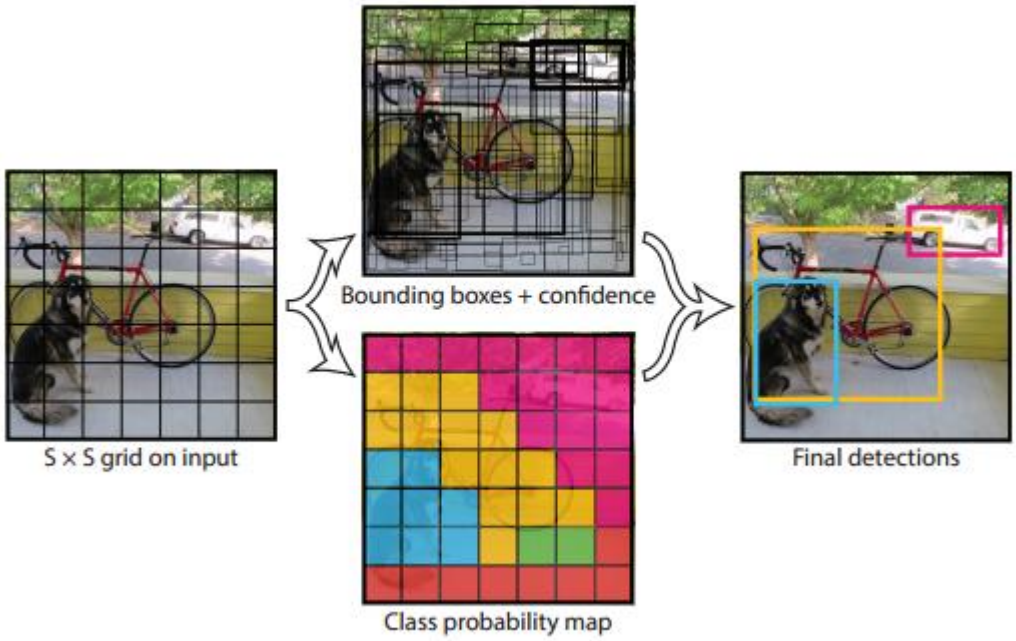
Mỗi hộp giới hạn bao gồm 5 tham số : x,y,w,h và độ tin cậy. Các tọa độ (x,y) đại diện cho tâm của hộp giới hạn. Chiều rộng và chiều cao được dự đoán tương đối so với toàn bộ hình ảnh. Cuối cùng dự đoán độ tin cậy thể hiện vào chỉ số IOU.

Mỗi ô lưới cũng dự đoán C xác suất lớp có điều kiện là $\Pr(Class_i|Object)$ .Các xác suất này được điều chỉnh trên các ô lưới có chứa một tùy chỉnh. Chúng ta chỉ dự đoán một bộ phận xác suất lớp cho mỗi ô lưới, bất kể số lượng của hộp giới hạn trong ô lưới.

Tại thời gian kiểm tra. Nhân xác suất lớp có điều kiện và dự đoán độ tin cậy của từng hộp riêng biệt.

$$\Pr(Class_i|Object) * \Pr(Object) * IOU_{Pred}^{Truth} = \Pr(Class_i) * IOU_{Pred}^{Truth}$$

Nó cho thấy điểm số tin cậy cụ thể của từng ô. Những điểm số này mã hóa cả xác suất của lớp đó xuất hiện trong hộp. Và hộp giới hạn phù hợp với đối tượng như thế nào.



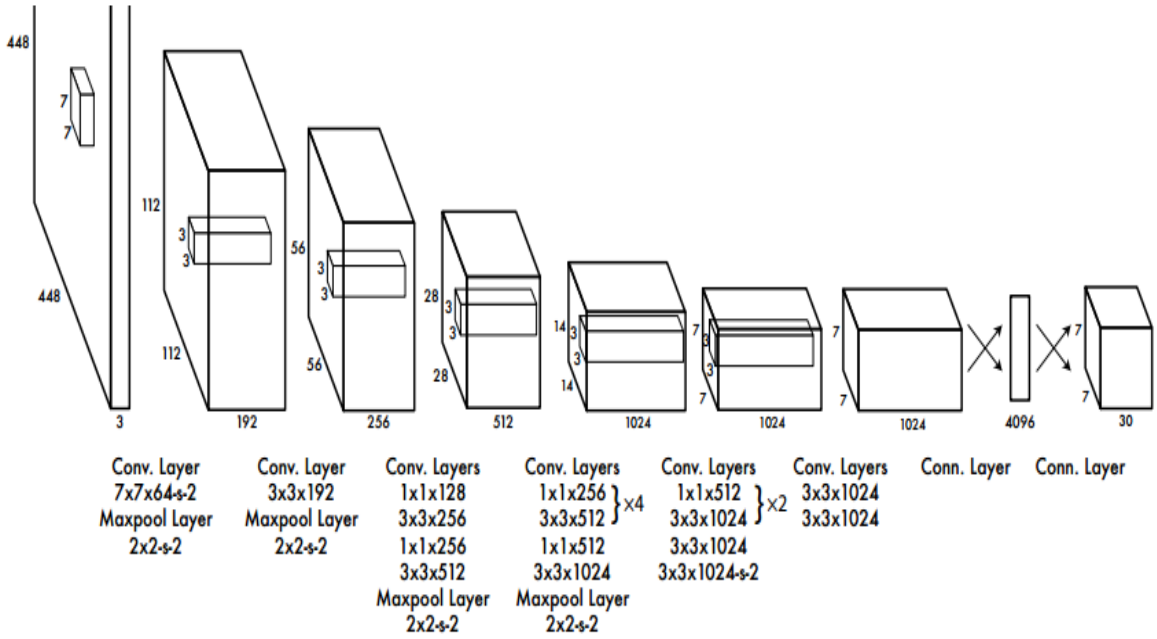
Hình 2-6: Mô hình phát hiện YOLO

Phát hiện mô hình hệ thống dưới dạng hồi quy. Nó chia ảnh thành SxS lưới và cho mỗi ô lưới dự đoán hộp giới hạn B. Độ tin cậy cho các hộp, xác suất lớp C và những dự đoán này được mã hóa dưới dạng S x S x (B * 5 x C) tensor

Để đánh giá YOLO [1] trên PASCAL VOC sử dụng S=7 , B = 2, VOC có 20 lớp nhãn nên C = 20. Dự đoán cuối cùng là 7 x 7 x 30 tensor.

2.2. Thiết kế mạng lưới dùng trong mô hình YOLO

Các kiến trúc mạng của YOLO lấy cảm hứng từ mô hình GoogleNet để phân loại hình ảnh. Mạng của YOLO có 24 lớp chuyển đổi và tiếp theo là 2 lớp được kết nối hoàn toàn. Thay vì các module khởi động được sử dụng bởi GoogLeNet thì mô hình chỉ đơn giản sử dụng 1x1 lớp giảm và tiếp theo là 3x3 lớp chuyển đổi tương tự Lin và đồng sự đã sử dụng . Mạng đầy đủ được trình bày ở hình 2-7.



Hình 2-7: Kiến trúc YOLO

Trong [1] đưa ra mạng phát hiện của YOLO gồm 24 lớp chuyển đổi và tiếp theo là 2 lớp được kết nối hoàn toàn. Luân phiên 1x1 lớp co giãn làm giảm feature map từ các lớp trước. Giả định rằng các lớp chuyển đổi trong nhiệm vụ phân loại ImageNet ở độ phân giải (hình ảnh đầu vào 224x224) và sau đó tăng gấp đôi độ phân giải để phát hiện.

Name	Filters	Output Dimension
Conv 1	7 x 7 x 64, stride=2	224 x 224 x 64
Max Pool 1	2 x 2, stride=2	112 x 112 x 64
Conv 2	3 x 3 x 192	112 x 112 x 192
Max Pool 2	2 x 2, stride=2	56 x 56 x 192
Conv 3	1 x 1 x 128	56 x 56 x 128
Conv 4	3 x 3 x 256	56 x 56 x 256
Conv 5	1 x 1 x 256	56 x 56 x 256
Conv 6	1 x 1 x 512	56 x 56 x 512
Max Pool 3	2 x 2, stride=2	28 x 28 x 512
Conv 7	1 x 1 x 256	28 x 28 x 256
Conv 8	3 x 3 x 512	28 x 28 x 512
Conv 9	1 x 1 x 256	28 x 28 x 256
Conv 10	3 x 3 x 512	28 x 28 x 512
Conv 11	1 x 1 x 256	28 x 28 x 256
Conv 12	3 x 3 x 512	28 x 28 x 512
Conv 13	1 x 1 x 256	28 x 28 x 256
Conv 14	3 x 3 x 512	28 x 28 x 512
Conv 15	1 x 1 x 512	28 x 28 x 512
Conv 16	3 x 3 x 1024	28 x 28 x 1024
Max Pool 4	2 x 2, stride=2	14 x 14 x 1024
Conv 17	1 x 1 x 512	14 x 14 x 512
Conv 18	3 x 3 x 1024	14 x 14 x 1024
Conv 19	1 x 1 x 512	14 x 14 x 512
Conv 20	3 x 3 x 1024	14 x 14 x 1024
Conv 21	3 x 3 x 1024	14 x 14 x 1024
Conv 22	3 x 3 x 1024, stride=2	7 x 7 x 1024
Conv 23	3 x 3 x 1024	7 x 7 x 1024
Conv 24	3 x 3 x 1024	7 x 7 x 1024
FC 1	-	4096
FC 2	-	7 x 7 x 30 (1470)

Hình 2-8: Thể hiện mạng nơ-ron đơn của YOLO khi chạy hệ thống

Kết quả cuối cùng của mạng là 7x7x30 tensor

Để hiểu rõ hơn về các lớp layer mà YOLO sử dụng, dưới đây sẽ giải thích các thông số trong hình 2-8.

Đầu vào là hình ảnh có chiều cao là 448, chiều rộng 448 và 3 kênh màu là RGB, BGR, HSV.

YOLO sẽ sử dụng các loại layer là Conv, Max – Pool, FC. Trong đó :

- Layer Conv đóng vai trò là lớp chuyển đổi chứa feature map.
- Layer Max – Pooling đóng vai trò đơn giản thông tin đầu ra và làm giảm số neuron.
- Layer Full Connect đóng vai trò là nơi phát hiện đối tượng.

Layer Conv 1 có Filter là 7x7x64, stride=2. Chiều rộng và chiều cao được giảm một nửa vì stride = 2 đầu ra là 224x224x64. Sau đó nó đi qua Max Pool 1 sử dụng 2x2 stride = 2 thì được đầu ra có kích thước là 112x112x64. Sau đó sử dụng tiếp theo sau là các lớp Conv và Max-pool sau cùng ta được đầu ra là 7x7x1024. Sau đó layer FC1 (lớp

full kết nối 1) có 4096 lớp, lấy từ 4 lớp Conv cuối cùng có 1024 lớp sau đó nhân 4 thành 4096 lớp. Lớp FC2 (lớp full kết nối 2) 7x7x30 = 1470 tensor

2.3. Huấn Luyện

Trong phần này đồ án sẽ đi tìm hiểu về việc huấn luyện sử dụng trong mô hình YOLO. Sử dụng framework Darknet cho tất cả thời gian huấn luyện và kiểm tra.

Sau thời gian huấn luyện và thời gian kiểm tra sẽ chuyển đổi mô hình để thực hiện phát hiện đối tượng trong hệ thống thực tế. Thêm 2 lớp chuyển đổi và kết nối với các mạng giả định có thể cải thiện hiệu suất của mô hình. Theo như ví dụ thêm 4 lớp chuyển đổi và 2 lớp kết nối hoàn toàn với trọng số khởi tạo ngẫu nhiên. Hoạt động phát hiện thường yêu cầu thông tin trực quan chi tiết để YOLO tăng độ phân giải của mạng từ 224 x 224 lên 448 x 448.

Lớp cuối cùng của YOLO dự đoán cả hai " xác suất lớp" và "hộp giới hạn". Chuẩn hóa chiều rộng và chiều cao của hộp giới hạn theo chiều rộng và chiều cao của ảnh, sao cho chúng nằm trong khoảng 0 đến 1. Tham số hộp giới hạn x và y là tọa độ của vị trí ô lưới cụ thể để chúng cũng nằm trong khoảng 0 đến 1.

YOLO dự đoán nhiều hộp giới hạn trên mỗi ô lưới. Vào thời gian huấn luyện thì việc dự đoán của hộp giới hạn liên quan với từng đối tượng. Một hộp giới hạn phải "chịu trách nhiệm" để dự đoán một đối tượng, dựa trên dự đoán có chỉ số IOU cao nhất hiện tại. Điều này dẫn đến sự thay đổi giữa các yếu tố dự đoán hộp giới hạn. Mỗi yếu tố dự đoán sẽ có kết quả tốt hơn khi dự đoán được kích thước và tỷ lệ của khung hình hoặc các lớp đối tượng.

Trong suốt quá trình huấn luyện, tối ưu hóa hàm loss theo [1] như sau:

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left[(x_i-\hat{y}_i)^2+\left(x_j-\hat{y}_j\right)^2\right] \quad (2\text{-}1)$$

$$+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left[\left(\sqrt{w_i}-\sqrt{\widehat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right]$$

$$+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left(C_i-\hat{C}_i\right)^2$$

$$+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{noobj}\left(C_i-\hat{C}_i\right)^2$$

$$+\sum_{i=0}^{S^2}\mathbb{1}_{i}^{obj}\sum_{c\in classes}\left(p_i(c)-\hat{p}_i(c)\right)^2$$

Trong đó ta có $\mathbb{1}_i^{obj}$ biểu thị nếu đối tượng xuất hiện trong ô thứ i và $\mathbb{1}_{ij}^{obj}$ biểu thị dự đoán của hộp giới hạn thứ j trong ô thứ i phải chịu trách nhiệm cho dự đoán đó. Chúng ta sẽ tìm hiểu về hàm Loss, một trong những thành phần quan trọng tạo nên sự khác biệt của tốc độ và độ chính xác của YOLO so với các mô hình khác.

Phần đầu tiên của Hàm Loss

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left[(x_i-\hat{y}_i)^2+\left(x_j-\hat{y}_j\right)^2\right] \quad (2\text{-}2)$$

Phương trình này tính toán sự mất mát liên quan đến vị trí hộp giới hạn dự đoán tham số (x,y). Chỉ cần xem λ như là một hằng số.

Tiếp tục chuyển sang phần thứ hai

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left[\left(\sqrt{w_i}-\sqrt{\widehat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right] \quad (2\text{-}3)$$

Đây là Loss liên quan đến chiều rộng và chiều cao của hộp giới hạn. Phần này trông giống như phần đầu tiên ngoại trừ căn bậc hai. Độ lệch nhỏ trong các hộp giới hạn nhỏ thì nguy hiểm hơn trong hộp giới hạn lớn, vì thế dự đoán căn bậc hai của chiều rộng và chiều cao thay vì sử dụng chiều rộng và chiều cao trực tiếp của hộp giới hạn.

Tiếp theo đến phần thứ ba.

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left(C_i-\hat{C}_i\right)^2+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{noobj}\left(C_i-\hat{C}_i\right)^2 \quad (2\text{-}4)$$

Ở đây sẽ tính toán của hàm loss liên quan đến điểm số tin cậy cho mỗi hộp giới hạn. C là điển số tin cậy và $\hat{C}$ là giao điểm trên liên kết của hộp giới hạn với chân lý. $\mathbb{1}_i^{obj}$ được hiểu nếu có đối tượng nằm trong ô , và $\mathbb{1}_i^{noobj}$ cho trường hợp ngược lại.

Điều này là cần thiết cho việc tăng tính ổn định của mô hình, ta đặt hai thông số λ_{coord} và λ_{noobj} để thực hiện. Đặt $\lambda_{coord}=5$ và $\lambda_{noobj}=0.5$

Phần cuối cùng của hàm Loss chính là phân loại Loss

$$\sum_{i=0}^{S^2}\mathbb{1}_i^{obj}\sum_{c\in classes}\left(p_i(c)-\hat{p}_i(c)\right)^2 \quad (2\text{-}5)$$

Nó thể hiện một lỗi bình phương để phân loại ngoại trừ cụm $\mathbb{1}_i^{obj}$. Điều này được sử dụng vì không sửa lỗi phân loại khi không có đối tượng nào xuất hiện trong ô.

Theo như tài liệu gốc [1] hướng dẫn để huấn luyện yolo, thì các tác giả sẽ làm như sau:

Epochs sẽ từ từ nâng tỷ lệ học tập từ 10^{-3} đến 10^{-2}. Nếu như bắt đầu với một tỷ lệ học tập cao thì mô hình sẽ nhanh chóng bị phân ra do độ dốc (gradients) không ổn định. Tiếp tục huấn luyện 10^{-2} cho 75 epochs, sau đó là 10^{-3} cho 30 epochs và cuối cùng là 10^{-4}cho 30 epochs.

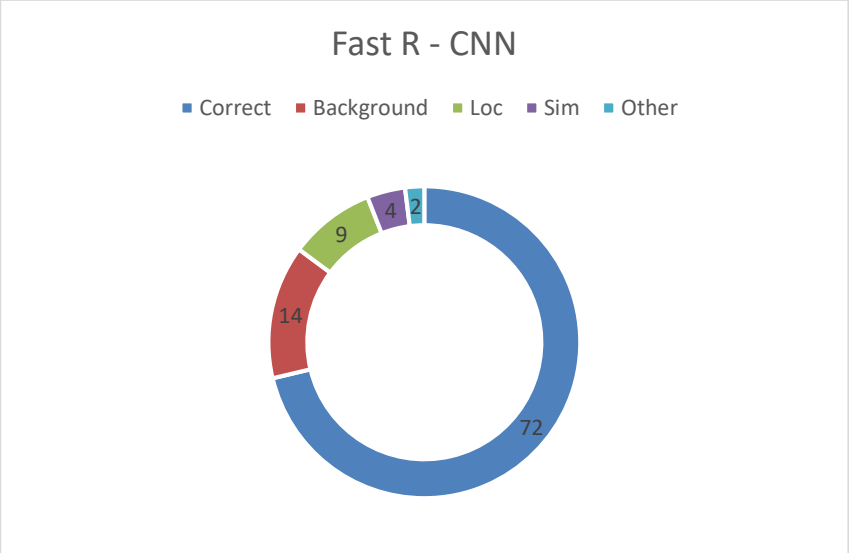
Để tránh overfitting thì sử dụng tăng thêm dữ liệu mở rộng. Để tăng cường dữ liệu cần phải sử dụng việc chia tỷ lệ ngẫu nhiên và bản dịch lên tới 20% kích thước hình ảnh gốc. Mô hình YOLO cũng điều chỉnh ngẫu nhiên độ phơi sáng (exposure) và độ bão hòa của hình ảnh theo hệ số 1:5 trong không gian màu HSV.

2.4. Phân tích các lỗi của mô hình phát hiện đối tượng YOLO

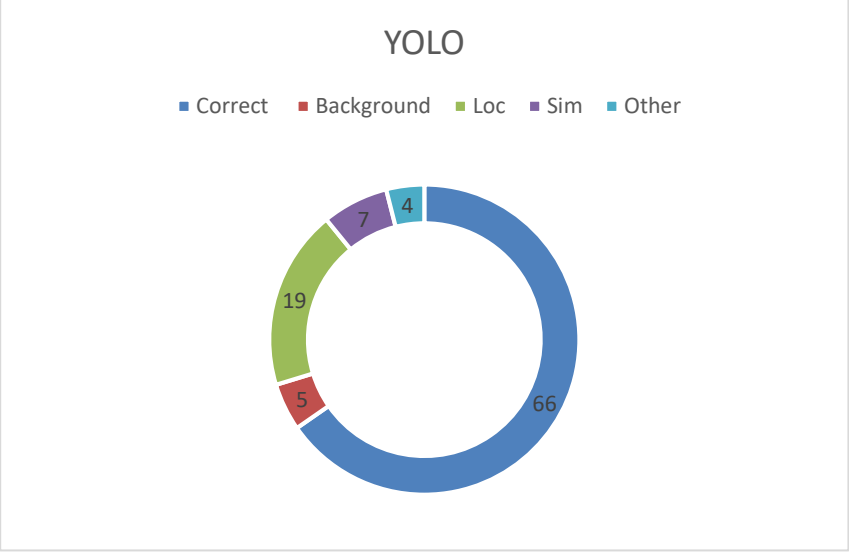
Để nhận thấy sự khác biệt giữa YOLO và các mô hình phát hiện đối tượng hiện tại, chúng ta cùng xem xét chi tiết các loại lỗi trong kết quả trên VOC 2007. So sánh YOLO với Fast RCNN vì Fast RCNN là một trong những máy dò có hiệu suất cao nhất trên PASCAL.

Sử dụng phương pháp và công cụ của Hoiem cùng các đồng sự. Đối với mỗi danh mục tại thời điểm thử nghiệm chúng ta xem xét các dự đoán N hàng đầu cho danh mục đó. Mỗi dự đoán đều đúng hoặc được phân loại dựa trên loại lỗi :

- Chính xác (Correct): đúng lớp và IOU > .5
- Bản địa hóa (Localization): đúng lớp , .1< IOU <.5
- Tương tự (Similar): lớp tương tự , IOU >.1



Hình 2-9: Phân bố các lỗi trong mô hình R – CNN [6]



Hình 2-10: Phân bố các lỗi trong mô hình YOLO

Biểu đồ này cho thấy phân bố các lỗi như lỗi Background, Localization, Similar và các lỗi khác.

YOLO xử lý để tìm vị trí các đối tượng một cách chính xác. Lỗi Localization chiếm nhiều lỗi nhất trong YOLO, hơn tất cả các lỗi khác cộng lại. Fast RCNN giảm ít

lỗi Localization hơn nhưng có nhiều lỗi Background hơn. 13,6% số phát hiện hàng đầu là những phát hiện sai không chứa bất kì đối tượng nào.

2.5. Kết chương

Chương 2 đã tập trung nêu về kiến trúc mạng lưới của mô hình YOLO, các lỗi thường gặp của mô hình và lý thuyết phát hiện và huấn luyện trong YOLO qua đó giúp hiểu sâu hơn về YOLO giúp cho việc cài đặt chương trình ở chương 3.

CHƯƠNG 3 : XÂY DỰNG CHƯƠNG TRÌNH VÀ THỬ NGHIỆM

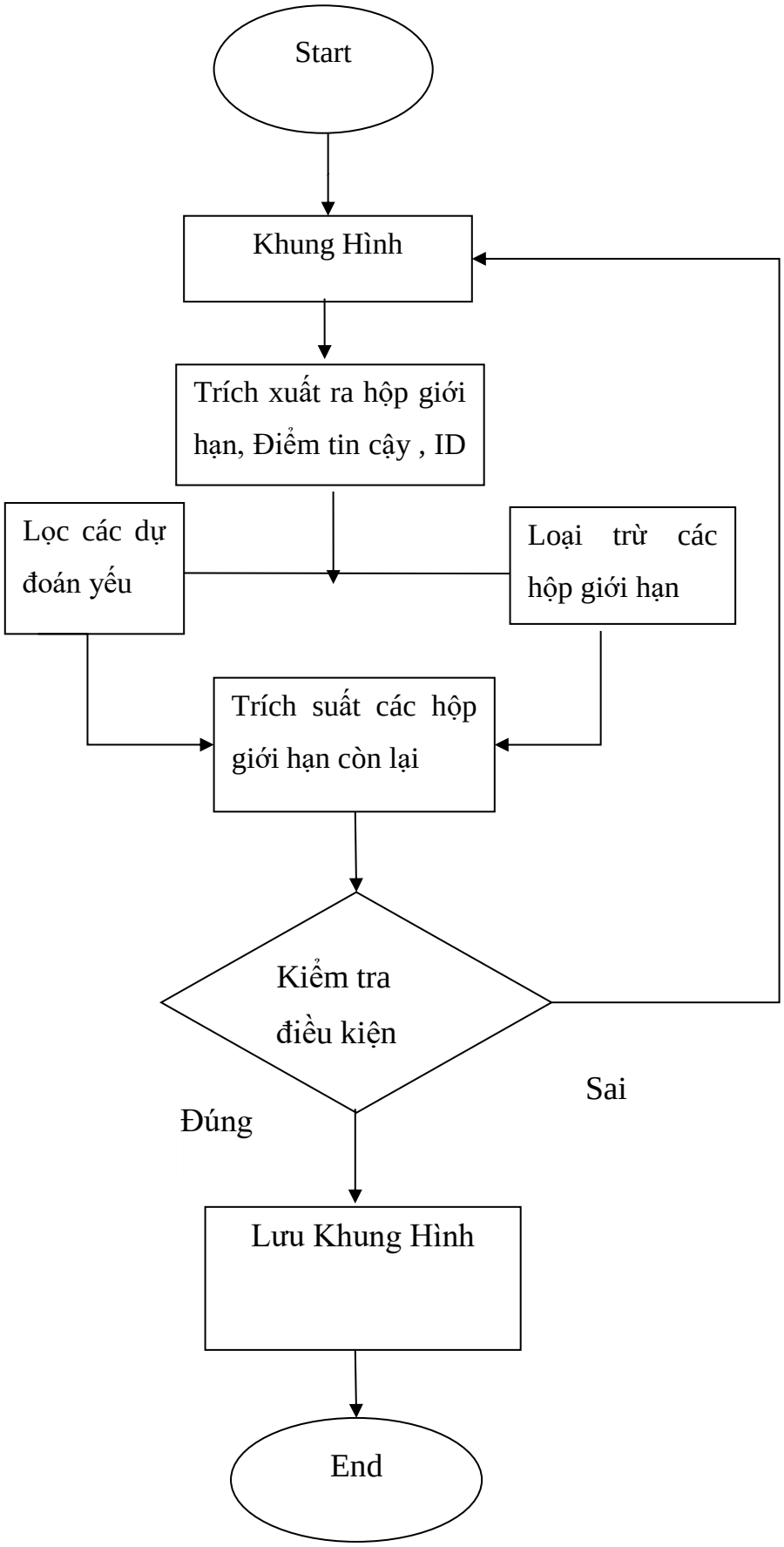
3.1. Xây dựng hệ thống

Nhiệm vụ của hệ thống là phát hiện đối tượng trong camera an ninh, được đặt ở chung cư. Đối tượng cụ thể ở đây là người và đồ vật nguy hiểm. Nên hướng thực hiện của đồ án là việc phát hiện và nhận dạng đối tượng trong camera. Hệ thống được xây dựng dựa trên mô hình phát hiện đối tượng YOLO.

Mô tả về các thành phần của hệ thống bao gồm

- Đầu vào: Đầu vào là video định dạng mp4 thu được từ camera an ninh.
- Đầu ra: Đầu ra của hệ thống là video định dạng avi, sau đó được nén lại với định dạng mp4. Do việc tạo luồng video từ sự hỗ trợ của OpenCV chỉ có thể tạo ra video có định dạng avi. Trong file video đầu ra sẽ là những khung hình chứa đối tượng là người hoặc đồ vật nguy hiểm. Các đối tượng này được kẻ khung với các màu khác nhau và được dán nhãn ở phía trên bên phải của khung.
- Ngôn ngữ lập trình: Python.
- Các thư viện được sử dụng trong file code Python là:
    - Thư viện numpy, nó được dùng để tính toán khoa học trên Python. Vì Numpy hỗ trợ mạnh mẽ việc tính toán các matrix và vector hay nó có các hàm đại số tuyến tính cơ bản nên nó được sử dụng trong các thuật toán học máy.
    - Thư viện argparse, nó được dùng để làm việc với các đối số do người dùng nhập vào trong lập trình Python.
    - Thư viện imutils, nó được sử dụng với các chức năng xử lý hình ảnh cơ bản như dịch, xoay hoặc thay đổi kích thước khung hình, phát hiện ra các đường viền hay đường biên.
    - Thư viện time, nó được dùng để làm việc với thời gian.
    - Thư viện os, cung cấp các hàm làm việc với hệ điều hành.
- Công cụ được sử dụng trong đề tài là OpenCV 4.0.0 và Python 3.7.0
- Hệ điều hành được sử dụng là Win 10 Pro 64 Bit.

Với nhiệm vụ đưa ra ở trên nên toàn bộ hoạt động của hệ thống được mô tả bằng sơ đồ sau:



Hình 3-1: Sơ đồ quá trình phát hiện đối tượng trong video

3.2. Quá trình chạy hệ thống

Đồ án sử dụng mô hình YOLO, 3 file dùng trong hệ thống phát hiện được lấy từ trang web của https://pjreddie.com/darknet/yolo/ gồm có các file sau coco.names, yolov3.cfg, yolov3.weight.

Hình 3-2 : Hình ở 17:34:03 trong file video "Camera.mp4"

Chúng ta chạy tập lệnh trong cmd:

"*python yolo_video.py --input videos/Camera.mp4 –output output/Camera_output.avi --yolo yolo-coco*"

Ý nghĩa chính của các tham số :

- *yolo_video.py:* file python chạy chương trình.
- *videos/Camera.mp4:* Sử dụng file input là Camera.mp4.
- *output/Camera_output.avi:* Lưu file sau khi chạy chương trình vào thư mục output, với tên là Camera_output.avi.
- *yolo-coco:* Thư mục chứa các file .weight và .cfg phục vụ cho chương trình.

Hình 3-3: Hình được cắt ở 17:34:03 ở file thu được

Care - Notepad

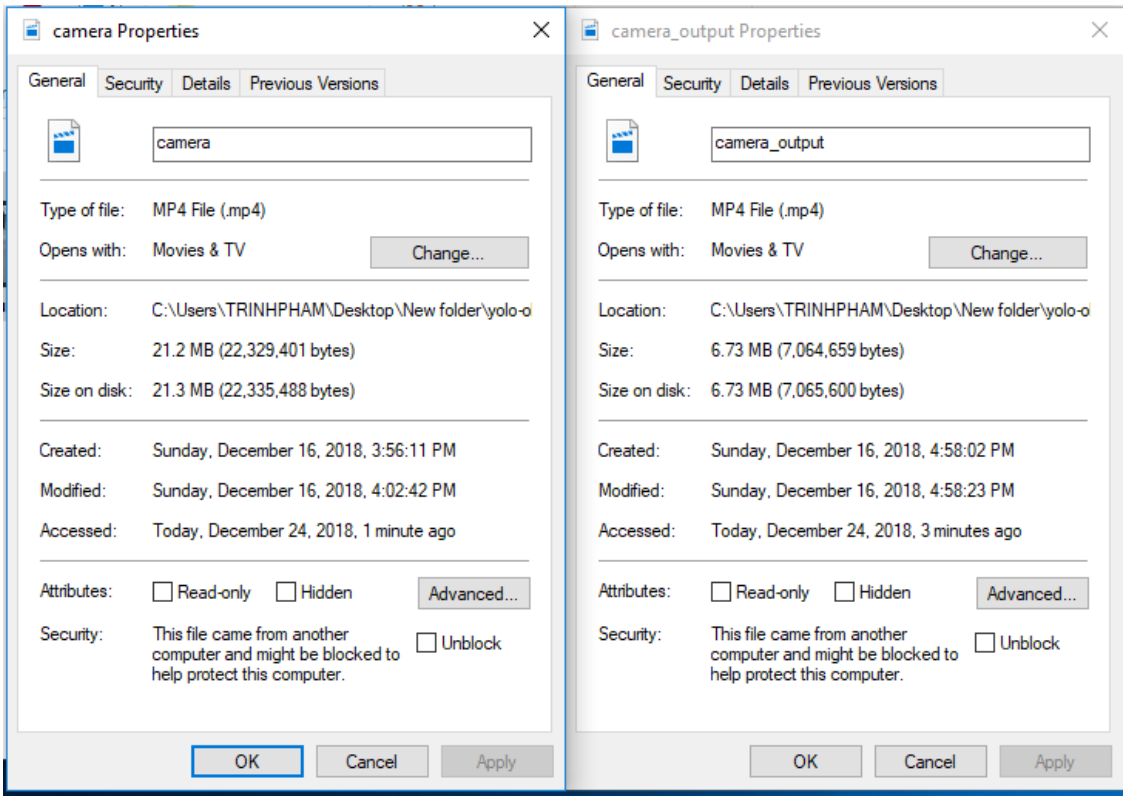
File Edit Format View Help

```
Frame 1 : person 388,2,225,348
Frame 1 : person 27,49,349,261
Frame 1 : knife 353,285,48,60
Frame 2 : person 387,0,225,351
Frame 2 : person 27,49,349,261
Frame 2 : knife 352,283,49,61
Frame 3 : person 387,1,225,349
Frame 3 : person 28,50,349,259
Frame 3 : knife 353,284,49,60
Frame 4 : person 388,2,224,349
Frame 4 : person 30,51,348,257
Frame 4 : knife 353,284,49,60
Frame 5 : person 388,4,224,346
Frame 5 : person 30,52,347,256
Frame 5 : knife 353,284,49,60
Frame 6 : person 389,2,222,348
Frame 6 : person 30,51,345,257
Frame 6 : knife 354,285,48,58
Frame 7 : person 387,3,224,346
Frame 7 : person 30,52,345,258
Frame 7 : knife 354,286,48,57
Frame 8 : person 386,3,225,345
Frame 8 : person 31,52,345,258
Frame 8 : knife 354,285,47,57
Frame 9 : person 386,3,226,344
Frame 9 : person 30,52,347,258
Frame 9 : knife 354,285,47,57
Frame 10 : person 387,1,224,348
Frame 10 : person 30,52,345,257
Frame 10 : knife 354,286,47,58
Frame 11 : person 387,1,224,350
Frame 11 : person 33,51,339,257
Frame 11 : knife 355,285,46,58
Frame 12 : person 389,1,222,349
Frame 12 : person 32,50,341,260
Frame 12 : knife 354,286,45,58
Frame 13 : person 388,2,222,347
```

Hình 3-4: Nội dung file care.txt

File care.txt là file sẽ lưu lại tọa độ của đối tượng "người và đồ vật nguy hiểm", được phát hiện trong khung hình bao nhiêu. Giúp thuận tiện trong việc tra cứu sau này.

3.3. Đánh giá kết quả hệ thống



Hình 3-5: So sánh dung lượng của hai file trước và sau khi demo

Trong file "camera.mp4" ban đầu có độ dài chỉ 60 giây nhưng chúng ta cần 22,335,488 Bytes tương đương 21,3 MB để lưu trữ. Và cụ thể đối tượng cần phải phát hiện chỉ có 2 người đàn ông ở ngoài cửa căn hộ. Nhưng khi chạy chương trình và thu được file output thì chúng ta chỉ cần 7,064,659 bytes, tương đương 6,73 MB để lưu trữ đoạn video output. Như vậy bộ nhớ dùng để lưu trữ file video giảm khoảng 70% và giúp bộ nhớ lưu trữ tiết kiệm 14,6 MB.

Bảng 3-1: Kết quả chạy chương trình với 4 đoạn video khác nhau

	Tổng thời gian của video input	**Tổng thời gian của video output**	**Dung lượng của video input**	**Dung lượng của video output**	**% Dung lượng tiết kiệm được**
Video_1	24 giây	16 giây	21,2MB	4,73MB	77,69%
Video_2	27 giây	17 giây	30,7MB	8,57MB	72,09%
Video_3	19 giây	2 giây	10,4MB	2,34MB	77.54%
Video_4	14 giây	9 giây	2,52MB	1,53MB	39.29%

Trong bảng 3-1 cho chúng ta thấy kết quả sau khi chạy chương trình. Đầu vào là 4 đoạn video có định dạng mà mp4 với độ dài khác nhau. Sau đó chạy chương trình ta thu được những đoạn video chỉ chứa những đối tượng cần phát hiện là người hoặc đồ vật nguy hiểm. Chương trình sẽ giúp làm giảm dung lượng lưu trữ file video từ 39,29% đến 77,69%.

3.4. Kết chương

Chương 3 đã thực hiện việc cài đặt thành công chương trình phát hiện đối tượng trong camera an ninh và cụ thể đối tượng ở đây là người và các đồ vật nguy hiểm. Chương trình giúp xóa đi những khung hình không chứa đối tượng và mang lại việc tiết kiệm phần cứng dùng để lưu trữ.

KẾT LUẬN

❖ Kết quả đạt được

Đồ án đã trình bày một cách tương đối đầy đủ về các mô hình được sử dụng để phát hiện đối tượng. Cụ thể nội dung đồ án đã thực hiện nghiên cứu:

- Tìm hiểu về phát hiện đối tượng sử dụng mô hình YOLO
- Cài đặt chương trình thử nghiệm để rút gọn video và trích xuất ra tọa độ của người và đồ vật nguy hiểm.

❖ Hướng phát triển trong tương lai

- Mô hình YOLO dùng để phát hiện đối tượng cụ thể là người vẫn có độ chính xác trung bình chưa cao. Do vậy mô hình vẫn cần phải nghiên cứu phát triển và khắc phục những nhược điểm của nó.
- Tiến tới sẽ sử dụng mô hình YOLO để giải quyết bài toán phát hiện khuôn mặt con người từ xa, ứng dụng trong lĩnh vực thành phố thông minh, đặt mục tiêu góp phần vào việc quản lý người dân thông qua số hóa.
- Xây dựng hệ thống hoàn chỉnh như mô hình đã trình bày.

TÀI LIỆU THAM KHẢO

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi (2016), *You only look once: Unified, real-time object detection*, *Information Hidding: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016.* University of Washington, Allen Institute for AI, Facebook AI Research, trang 779-788.

[2] Hossein Azizpour, Ivan Laptev (2012), *Object detection using strongly-supervised deformable part models*, *Information Hidding: European Conference on Computer Vision 2012*. Springer, Berlin, Heidelberg, trang 836-849.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun (2014), *Spatial pyramid pooling in deep convolutional networks for visual recognition*, *Information Hidding: European conference on computer vision 2014*. Springer, Cham, China, trang 346-361.

[4] Guangrui Liu, (2017), *Real-time object detection for autonomous driving-based on deep learning*. Diss, BS, Beijing University of Technology, China, 80 trang.

[5] Liliang Zhang, Liang Lin, Xiaodan Liang, Kaiming He (2016), *Is faster R-CNN doing well for pedestrian detection?*, *Information Hidding: European Conference on Computer Vision 2016*. Springer, Cham, China, trang 443-457.

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik (2014), *Rich feature hierarchies for accurate object detection and semantic segmentation*, *Information Hidding*: *Proceedings of the IEEE conference on computer vision and pattern recognition 2014*. Springer, Berlin, Heidelberg, trang 580-587.

[7] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun (2015), *Faster r-cnn: Towards real-time object detection with region proposal networks*, *Information Hidding: Advances in neural information processing systems 2015*. Montreal, Canada, trang 1137-1149.

[8] Tech Insight (2017), http://chungta.vn/tin-tuc/chuyen-gia/phan-biet-ai-machine-learning-va-deep-learning-56854.html, truy cập tháng 12 năm 2018.

[9] Waleed Kadous (2017), https://www.quora.com/What-are-the-advantages-and-disadvantages-of-deep-learning-Can-you-compare-it-with-the-statistical-learning-theory, truy cập tháng 12 năm 2018.

[10] Ross Girshick, Jeff Donahue, Student Member, (2015), *Region-based Convolutional Networks for Accurate Object Detection and Segmentation*, *Information Hidding: IEEE transactions on pattern analysis and machine intelligence 38.1*. USA, trang 142-158