

DẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
DẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA KHOA HỌC MÁY TÍNH

BÁO CÁO KẾT QUẢ NGHIÊN CỨU

**HỆ THỐNG PHÁT HIỆN  
THIẾT BỊ TRONG PHÒNG  
HỌC THÔNG MINH THỜI  
GIAN THỰC**

GIẢNG VIÊN HƯỚNG DẪN:  
CN. TRẦN DOANH THUYỀN

CHỦ NHIỆM ĐỀ TÀI:  
NGUYỄN VIỆT NHẬT

4, 2023

# Mục lục

<b>Mục lục</b>	<b>i</b>
<b>Danh sách hình vẽ</b>	<b>iii</b>
<b>Nomenclature</b>	<b>iv</b>
<b>1 Tổng Quan</b>	<b>1</b>
1.1 Giới thiệu đề tài . . . . .	1
1.2 Tính cấp thiết . . . . .	3
1.3 Thách thức . . . . .	3
1.4 Ý tưởng khoa học . . . . .	4
1.5 Mục tiêu hướng tới . . . . .	4
<b>2 Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan</b>	<b>5</b>
2.1 Tổng quan về bài toán nhận diện vật thể . . . . .	5
2.2 Faster R-CNN . . . . .	7
2.2.1 Kiến trúc chung . . . . .	7
2.2.2 R-CNN (2014) . . . . .	8
2.2.3 Fast R-CNN (2015) . . . . .	10
2.2.4 Faster R-CNN (2016) . . . . .	12
2.3 YOLOv7 (2022) . . . . .	19
2.3.1 Kiến trúc . . . . .	20
2.3.2 Trainable bag-of-freebies . . . . .	22
<b>3 Thực Nghiệm Và Kết Quả</b>	<b>23</b>

## **MỤC LỤC**

---

3.1	Cài đặt thực nghiệm . . . . .	23
3.2	Kết quả thực nghiệm . . . . .	23
<b>4</b>	<b>Kết luận và hướng phát triển</b>	<b>24</b>
	<b>Phụ lục A</b>	<b>25</b>
	<b>Phụ lục B</b>	<b>26</b>
	<b>References</b>	<b>27</b>

# Danh sách hình vẽ

1.1	Các bước xử lý cơ bản của hệ thống . . . . .	2
2.1	Nhận diện các vật thể trong một căn phòng . . . . .	6
2.2	Object detection pipeline . . . . .	8
2.3	Hình ảnh minh họa về thuật toán Selective search . . . . .	9
2.4	Hình ảnh minh họa kiến trúc R-CNN (được trích xuất từ bài báo gốc [6]). Mô hình (1) đọc ảnh đầu vào, (2) trích xuất khoảng 2000 region proposal, (3) tính toán các đặc trưng trên từng đề xuất bằng CNN, cuối cùng (4) phân lớp từng khu vực sử dụng SVMs. . . . .	10
2.5	Kiến trúc của Fast R-CNN . . . . .	11
2.6	Hình ảnh minh họa RoI Pooling . . . . .	11
2.7	Kiến trúc của Faster R-CNN . . . . .	13
2.8	Hình ảnh minh họa kiến trúc RPN (ảnh trích xuất từ bài báo Ren et al. [16]) . . . . .	14
2.9	So sánh hiệu suất của Yolov7 và các real-time detector khác (Ảnh trích xuất từ bài báo [21]) . . . . .	20
2.10	Hình ảnh minh họa kiến trúc One-stage detector . . . . .	20

# Nomenclature

Aspect ratio	Tỷ lệ cạnh $w/h$
Bounding box	Hình chữ nhật được vẽ bao quanh đối tượng nhằm xác định đối tượng.
Ground-truth box	Tập các tọa độ của một đối tượng trong ảnh, được gán nhãn một cách thủ công hoặc nhận được từ một nguồn tin cậy
mAP	mean Average Precision
Pipeline	Tập hợp các bước xử lý liên tiếp nhận đầu vào là dữ liệu (ảnh, âm thanh, các trường dữ liệu) và trả ra kết quả dự đoán ở output.
Region proposal	Vùng đề xuất, là những vùng mà có khả năng chứa đối tượng ở bên trong nó.
RoI	Region of Interest
Scale	Độ phóng đại so với khung hình gốc

# Chương 1

## Tổng Quan

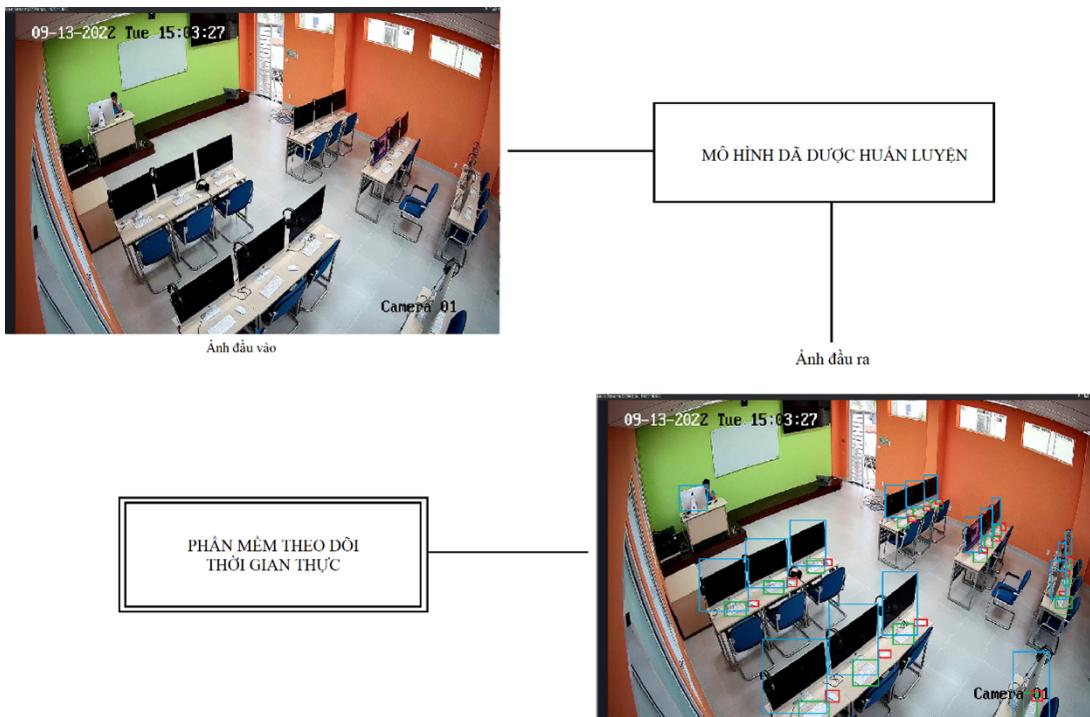
### 1.1 Giới thiệu đề tài

Với việc khánh thành phòng học thông minh, một không gian học mở dành cho sinh viên UIT với các trang thiết bị tiên tiến. Tuy nhiên, đi cùng với việc chất lượng giảng dạy được nâng cao, nhu cầu bảo vệ tài sản công đã trở thành một vấn đề cực kỳ quan trọng và được đặt lên hàng đầu. Để đảm bảo an ninh và tránh mất tài sản công, nhóm nghiên cứu đã quyết định thực hiện đề tài “Hệ thống phát hiện các thiết bị trong phòng học thông minh thời gian thực”.

Bài toán nghiên cứu có đầu vào là hình ảnh (image) được trích xuất từ 2 camera giám sát thuộc phòng học thông minh của trường, đầu ra là hình ảnh (image) đã được xử lý để phát hiện các trang thiết bị (màn hình, chuột, bàn phím) trong thời gian thực. Đây là bài toán có tính cấp thiết và ứng dụng cao, với mục đích đặt ra nhằm bảo vệ tài sản công của nhà Trường, tránh việc thiệt hại về các trang thiết bị. Ngoài ra, đề tài còn có thể mở rộng ứng dụng trong nhiều địa điểm thuộc nhà Trường, hay phạm vi bên ngoài nhà Trường.

Tuy nhiên, bài toán phát hiện các thiết bị trong phòng học thông minh không phải là một bài toán đơn giản. Với yêu cầu phát hiện các thiết bị trong thời gian thực, đòi hỏi hệ thống phải có khả năng phát hiện các thiết bị một cách nhanh chóng và chính xác, từ đó giúp người quản lý phòng học có thể có những

## 1. Tổng Quan



**Hình 1.1: Các bước xử lý cơ bản của hệ thống**

biện pháp kịp thời để bảo vệ tài sản công. Đề tài nghiên cứu còn gấp phai vấn đề khoảng cách từ 2 camera giám sát được đặt ở trên cao, dẫn đến việc khoảng cách đến các thiết bị có thể khác nhau, khiến hình ảnh của chúng bị mờ hay có thể bị che khuất bởi các thiết bị khác. Bên cạnh đó, việc sử dụng 2 camera giám sát làm tăng thêm tầm nhìn nhưng lại phát sinh thêm bài toán phải đồng bộ giữa chúng để tránh nhận diện một thiết bị đến 2 lần. Không những thế, phòng học thông minh chỉ vừa được khánh thành cách đây không quá lâu, nên bài toán còn chịu sự hạn chế về mặt dữ liệu.

Hiện nay, có khá nhiều phương pháp đối với bài toán phát hiện vật thể nhưng chưa thực sự áp dụng cho một lĩnh vực cụ thể như đề tài, nên việc lựa chọn phương pháp phù hợp và tối ưu nhất cho đề tài nghiên cứu cũng là một thách thức lớn.

### **1.2 Tính cấp thiết**

Đề tài nghiên cứu khoa học này là cấp thiết vì những đặc điểm sau:

1. Tính ứng dụng cao: Hỗ trợ bảo vệ trang thiết bị của nhà Trường, tránh những mất mát xuất phát từ ý thức của một số thành viên trong quá trình sử dụng tài sản công.
2. Thiếu hụt trong nghiên cứu trước đây: Mặc dù đã có nghiên cứu về nhận diện vật thể, tuy nhiên, vẫn chưa có bất kỳ mô hình nào để bảo vệ cụ thể các trang thiết bị trong phòng máy (máy tính, chuột, bàn phím). Điều này cho thấy sự cấp thiết của đề tài này để giải quyết vấn đề thiếu hụt này.
3. Thiếu ứng dụng minh họa: Hiện tại chưa có ứng dụng minh họa nào cho đề tài này. Việc tạo ra một ứng dụng minh họa sẽ giúp cho người sử dụng dễ dàng hiểu được cách thức hoạt động của hệ thống bảo vệ trang thiết bị.

Tóm lại, đề tài nghiên cứu khoa học này cấp thiết vì nó giúp bảo vệ trang thiết bị của nhà trường, giải quyết thiếu hụt trong nghiên cứu trước đây và cần có ứng dụng minh họa để giúp người dùng dễ dàng hiểu được hệ thống bảo vệ trang thiết bị.

### **1.3 Thách thức**

Đề tài phải đối diện với một số thách thức đáng kể, bao gồm:

1. Xây dựng hệ thống trong thời gian thực yêu cầu cả sự suy diễn nhanh chóng trong khi vẫn duy trì được độ chính xác gốc của mô hình.
2. Vấn đề đồng bộ hóa giữa 2 camera do khác biệt về vị trí đặt và khoảng cách đến các thiết bị.
3. Dữ liệu đầu vào vẫn còn hạn chế về số lượng lẫn chất lượng.

### **1.4 Ý tưởng khoa học**

Trong lĩnh vực Computer Vision, bài toán nhận diện vật thể đóng vai trò rất quan trọng và đang nhận được nhiều sự quan tâm. Các mô hình nhận diện vật thể đang được nghiên cứu và phát triển liên tục để có thể đáp ứng được nhu cầu xã hội.

Hiện nay, các mô hình nhận diện vật thể thường được các nhóm nghiên cứu tiếp cận thông qua thuật toán YOLO – một state-of-the-art trong lĩnh vực Computer Vision, cụ thể hơn là bài toán nhận diện vật thể (Object detection). Tuy nhiên, bên cạnh YOLO, vẫn còn một hướng tiếp cận phổ biến thông qua các thuật toán two-stages, sử dụng mạng neural tích chập, mà tiêu biểu của trong số đó là Faster R-CNN. Do đó, việc tìm hiểu và đánh giá các phương pháp tiếp cận đóng vai trò rất lớn trong việc xây dựng hệ thống.

### **1.5 Mục tiêu hướng tới**

Trong công trình nghiên cứu này, các mục tiêu đề ra bao gồm:

1. Tìm hiểu tổng quan về bài toán nhận diện vật thể (object detection).
2. Dánh giá một số phương pháp tiên tiến có thể giải quyết đề tài nhận diện các thiết bị trong phòng học thông minh.
3. Xây dựng tập dữ liệu cho đề tài và đánh giá các phương pháp đã tìm hiểu trên tập dữ liệu đó.
4. Sử dụng phương pháp đã thực hiện để xây dựng ứng dụng minh họa.

Theo đó, phạm vi nghiên cứu của đề tài như sau:

1. Dánh giá một số phương pháp:
  - Thuật toán Two-stages mà tiêu biểu là Faster R-CNN.
  - Thuật toán One-stage với YOLOv7.
2. Xây dựng tập dữ liệu: Thu thập dữ liệu qua 2 camera giám sát thuộc phòng học thông minh.

## Chương 2

# Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan

Chương này tập trung vào việc tìm hiểu tổng quan bài toán mô tả hình ảnh và các phương pháp được áp dụng trên các tập dữ liệu lớn như Faster R-CNN [16] và YOLOv7 [21].

### 2.1 Tổng quan về bài toán nhận diện vật thể

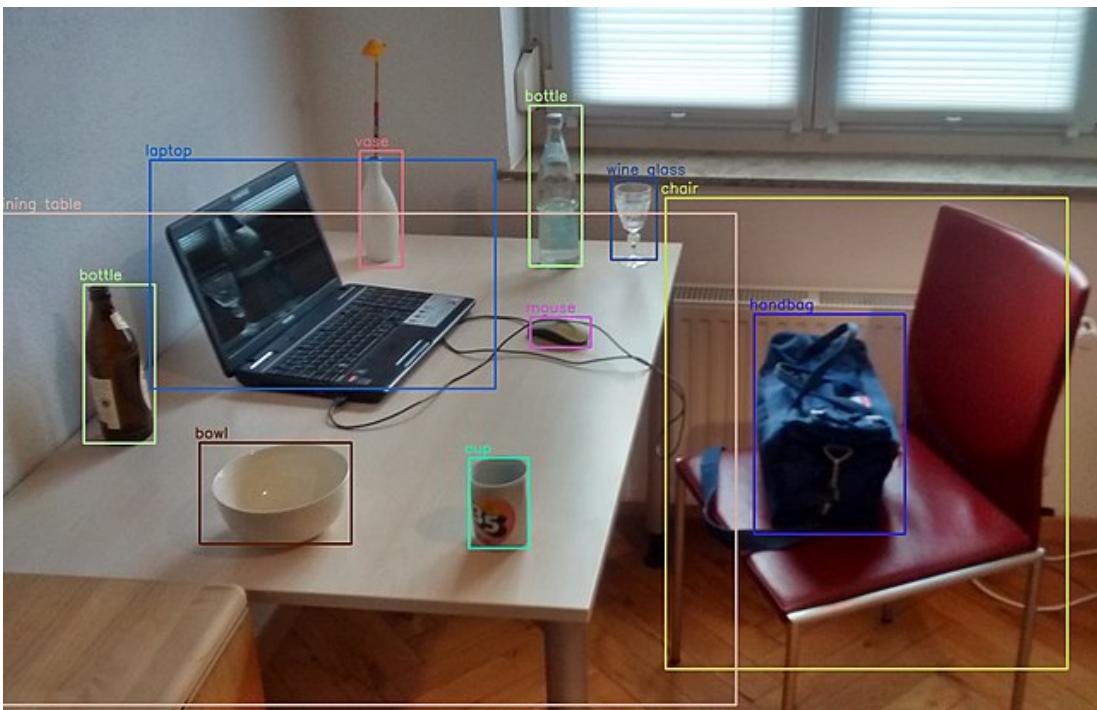
Bài toán nhận diện vật thể (object detection) là một trong những bài toán quan trọng của lĩnh vực Computer Vision. Bài toán này yêu cầu máy tính phải nhận dạng các đối tượng (thuộc về một lớp cụ thể như con người, xe, máy tính...) trong ảnh hoặc video, đồng thời xác định vị trí của những đối tượng này trong bức ảnh (video) đó. Mục tiêu của bài toán nhận diện vật thể chính là mô phỏng khả năng thị giác của con người vào máy tính.

Trong khi số lượng các loại vật thể trong một bức ảnh không nhiều, nhận diện vật thể được đánh giá là bài toán có tính thách thức cao bởi số lượng lớn các khả năng về vị trí cũng như kích thước của các đối tượng. Do đó, bài toán nhận diện vật thể vẫn luôn là một lĩnh vực cần được nhiều sự nghiên cứu.

Hiện nay, có rất nhiều phương pháp để giải quyết bài toán nhận diện vật thể,

## 2. Cơ Sở Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan

tùy vào tính chất và độ phức tạp của bài toán mà mỗi phương pháp có ưu, nhược điểm khác nhau. Bao gồm các phương pháp truyền thống dựa trên sự phân tích và rút trích đặc trưng như Haar Cascades, Viola-Jones... và các phương pháp hiện đại sử dụng thuật toán Deep Learning như SSD, Faster R-CNN, YOLO... Thông thường, các thuật toán Deep Learning cho ra kết quả tốt hơn so với các phương pháp cổ điển, đặc biệt là khi xử lý các vật thể phức tạp và có nhiều biến thể khác nhau.



Hình 2.1: Nhận diện các vật thể trong một căn phòng

Các phương pháp sử dụng Deep Learning cho bài toán nhận diện vật thể có thể được chia thành hai loại chính: two-stages và one-stage [3]. Trong two-stage, mô hình sẽ đưa ra các vùng tiềm năng chứa đối tượng trước khi xác định đối tượng. Ví dụ như phương pháp Faster R-CNN sẽ sử dụng Region Proposal Network (RPN) để đưa ra các vùng tiềm năng trước khi đưa vào mô hình để xác định đối tượng. Ngược lại, trong one-stage, mô hình sẽ trực tiếp dự đoán các bounding box và loại đối tượng tương ứng trong ảnh. Tiêu biểu của one-stage là phương pháp YOLO (You Only Look Once) sẽ trực tiếp dự đoán bounding box và loại đối tượng trong ảnh.

## **2. Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan**

---

Tuy nhiên, bài toán nhận diện vật thể vẫn còn rất nhiều thách thức, đặc biệt là khi đối tượng có kích thước nhỏ, có nhiều biến thể hoặc ở nhiều vị trí khác nhau trên ảnh. Để giải quyết những thách thức này, bài toán vẫn cần được nghiên cứu và đưa ra những giải pháp mới tối ưu hơn.

Tóm lại, bài toán nhận diện vật thể là một bài toán quan trọng trong lĩnh vực Computer Vision. Có nhiều phương pháp để giải quyết bài toán này, từ các phương pháp cổ điển đến các phương pháp sử dụng Deep Learning. Tuy nhiên, bài toán vẫn còn nhiều thách thức và cần nghiên cứu thêm để giải quyết.

### **2.2 Faster R-CNN**

Faster R-CNN (Region-based Convolutional Neural Network) là một mô hình deep learning được phát triển để giải quyết bài toán Object Detection - phát hiện và nhận diện vật thể. Với tốc độ và sự chính xác của mình, mô hình Faster R-CNN đã trở thành một trong những giải pháp tiên tiến nhất hiện nay trong lĩnh vực này. Để thực sự hiểu về Faster R-CNN, trước tiên chúng ta cần tìm hiểu ngắn gọn về kiến trúc chung và các tiền thân của nó, đó là R-CNN và Fast R-CNN.

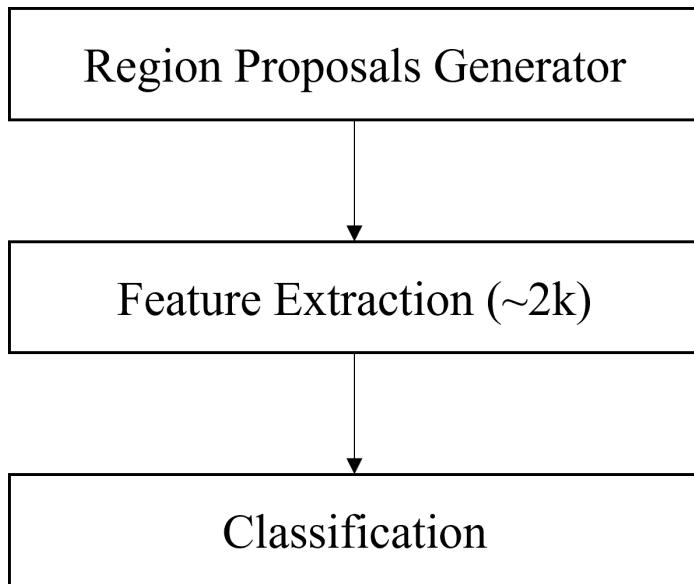
#### **2.2.1 Kiến trúc chung**

Các mô hình họ R-CNN tương tự như các kỹ thuật nhận diện vật thể truyền thống, được thực hiện thông qua 3 bước chính: Tạo các vùng đề xuất hình ảnh (Region proposals generator), trích lọc đặc trưng (Feature extraction) và phân loại đối tượng (Classification).

Bước đầu tiên của quy trình có tác dụng tạo và trích xuất các vùng có thể chứa vật thể được bao bởi các bounding box. Các region proposal là các vùng có khả năng chứa các đối tượng hoặc hình ảnh ở bên trong đó. Số lượng các region proposal này thường rất lớn (khoảng 2 ngàn hoặc hơn). Một số thuật toán tạo ra các region proposal có thể kể đến là Selective Search và EdgeBoxes.

Với mỗi region proposal, một vector đặc trưng có độ dài cố định được trích xuất bằng cách sử dụng các mô tả đặc trưng hình ảnh khác nhau, ví dụ như

## 2. Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan



**Hình 2.2: Object detection pipeline**

Histogram of Oriented Gradients (HOG). Vector đặc trưng này rất quan trọng đối với thành công của mô hình nhận diện vật thể. Do đó, một vector đặc trưng cần phải mô tả đủ đối tượng ngay cả khi đối tượng bị thay đổi về tỉ lệ hay ở một vị trí khác.

Các vector đặc trưng sau đó được sử dụng để gán từng region proposal vào lớp nền hoặc một trong các lớp đối tượng. Khi số lượng lớp tăng lên, độ phức tạp của việc xây dựng mô hình nhằm phân biệt giữa các đối tượng này cũng tăng lên. Một thuật toán phổ biến được dùng để tách các đối tượng khác nhau vào đúng lớp là Support Vector Machine (SVM).

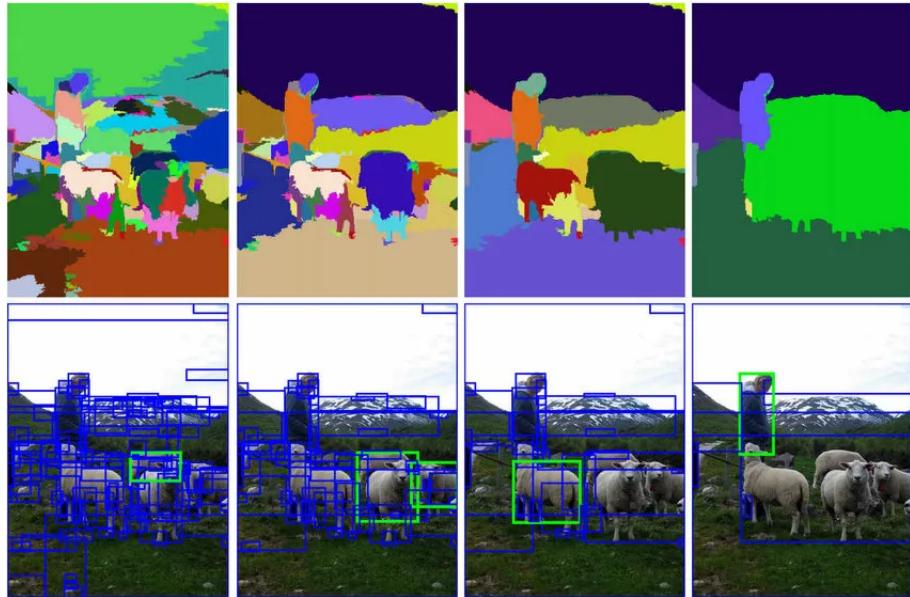
### 2.2.2 R-CNN (2014)

R-CNN (Region-based Convolutional Neural Networks) là một mô hình mạng neural sử dụng cho bài toán nhận dạng vật thể trong hình ảnh [6]. Được giới thiệu lần đầu tiên vào năm 2014 bởi Ross Girshick, Jeff Donahue, Trevor Darrell và Jitendra Malik, R-CNN đã tạo ra một bước đột phá trong việc giải quyết bài toán nhận dạng vật thể trong hình ảnh.

Bước đầu tiên trong pipeline của R-CNN nhằm sinh ra các region proposal

## 2. Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan

dựa trên thuật toán selective search [19]. Thuật toán selective search tạo ra các vùng phân đoạn phụ (sub-segmentation) từ hình ảnh có thể thuộc về một đối tượng (dựa trên các đặc trưng màu sắc, kích thước, hình dạng hay kết cấu) và lặp lại việc kết hợp các vùng giống nhau để tạo nên các đối tượng.

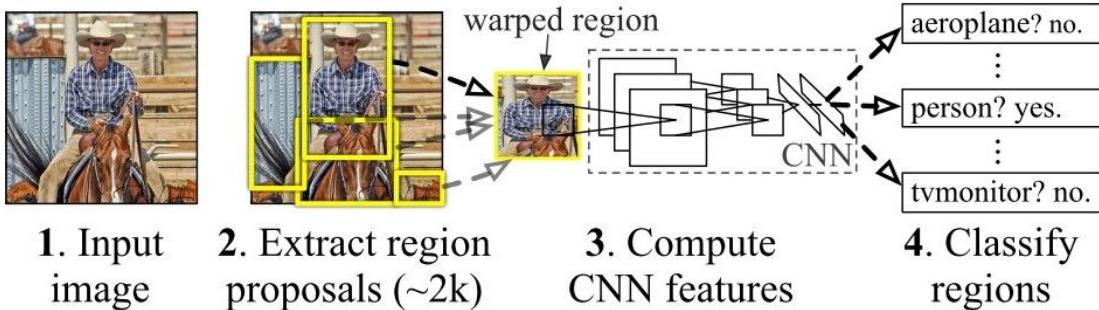


**Hình 2.3: Hình ảnh minh họa về thuật toán Selective search**

Dóng góp chính của R-CNN vào quy trình chung (Hình 2.2) chỉ đơn giản là thực hiện trích xuất đặc trưng dựa trên một mạng CNN học sâu. Các region proposal đã tạo ra sau đó được điều chỉnh về một kích thước cố định (4,096). Cuối cùng, mô hình sử dụng thuật toán pre-trained SVM nhằm phân loại các đối tượng trong các vùng ảnh đó.

R-CNN là một trong những mô hình đầu tiên sử dụng mạng học sâu để nhận diện vật thể trên hình ảnh. Việc kết hợp các phương pháp truyền thống và mạng học sâu là một bước đột phá trong bài toán nhận diện vật thể, mô hình đã cho thấy khả năng phát hiện và nhận dạng vật thể với độ chính xác cao hơn so với phương pháp truyền thống (khoảng 30% so với kết quả tốt nhất trước đó trên PASCAL VOC 2012 [6]). Không những thế, nhờ sự ứng dụng CNN tương đối đơn giản và dễ hiểu, mô hình R-CNN còn cho thấy khả năng mở rộng số lượng các lớp của mình.

## 2. Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan



**Hình 2.4: Hình ảnh minh họa kiến trúc R-CNN** (được trích xuất từ bài báo gốc [6]). Mô hình (1) đọc ảnh đầu vào, (2) trích xuất khoảng 2000 region proposal, (3) tính toán các đặc trưng trên từng đề xuất bằng CNN, cuối cùng (4) phân lớp từng khu vực sử dụng SVMs.

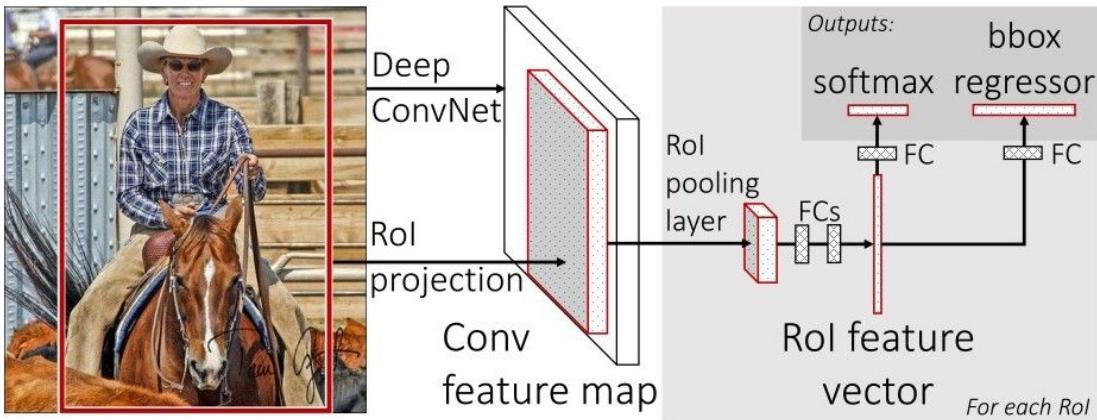
Mặc dù R-CNN đã đem đến nhiều ưu điểm và đột phá trong việc phát hiện và nhận dạng vật thể, nhưng mô hình này cũng tồn tại nhiều hạn chế. Trong đó, đáng nói nhất là tốc độ xử lý chậm, đặc biệt là trong quá trình tạo các region proposals và trích xuất đặc trưng. Mỗi region proposal trong khoảng 2000 vùng được selective search tạo ra phải được đưa vào mạng CNN độc lập với nhau, đồng nghĩa với việc có khoảng 2000 vùng ảnh phải đi qua CNN, do đó quá trình này cần một khoảng thời gian rất lớn, nhận dạng trên tập VGG16 cần 47s / ảnh (GPU) [6], làm cho việc chạy R-CNN trong thời gian thực (real-time) là bất khả thi. Không những thế, mô hình R-CNN cần lưu lại các đặc trưng đã được trích xuất từ mạng CNN để huấn luyện SVMs sau đó, dẫn đến việc mô hình yêu cầu một tài nguyên bộ nhớ đáng kể.

Để giải quyết được những nhược điểm trên, Fast R-CNN [5] đã được ra đời nhằm khắc phục những hạn chế, đồng thời cải thiện các ưu điểm của R-CNN.

### 2.2.3 Fast R-CNN (2015)

Fast R-CNN là một mô hình phát hiện đối tượng trong Computer Vision, được giới thiệu bởi Ross Girshick vào năm 2015 [5]. Fast R-CNN được xây dựng dựa trên mô hình R-CNN trước đó nhằm phân loại hiệu quả các region proposals bằng cách sử dụng mạng CNN. So với R-CNN, Fast R-CNN đã có nhiều cải tiến nhằm cải thiện tốc độ huấn luyện và kiểm tra, trong khi vẫn tăng độ chính xác.

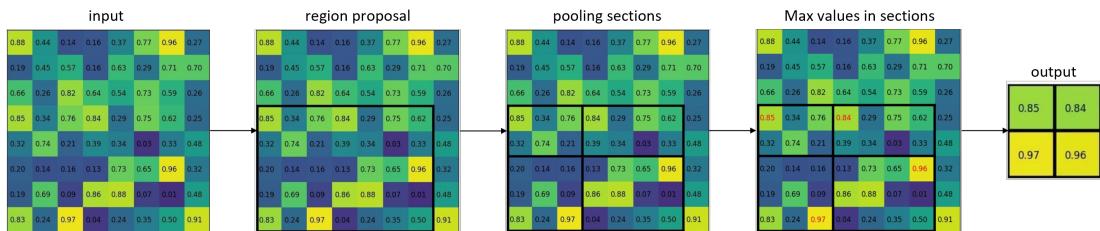
## 2. Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan



Hình 2.5: Kiến trúc của Fast R-CNN

Fast R-CNN huấn luyện trên mạng VGG16 nhanh hơn R-CNN gấp 9 lần, trong khi cho ra kết quả nhanh hơn 213 lần tại thời điểm thử nghiệm (test-time) và đạt được mAP cao hơn trên PASCAL VOC 2012 với mAP khoảng 66% (vs. 62% sử dụng R-CNN).

Hình 2.5 thể hiện cách Fast R-CNN tiếp cận bài toán nhận diện vật thể theo tương tự như R-CNN. Tuy nhiên, thay vì đưa từng region proposals vào CNN, Fast R-CNN đưa cả bức ảnh vào ConvNet nhằm tạo một conv feature map. Từ conv feature map, mô hình xác định các region proposals, sau đó nén chúng thành các hình vuông có kích thước cố định (định nghĩa trước) bằng cách sử dụng ROI Pooling. Các vector đặc trưng được đưa vào Fully connected layer bao gồm 1 lớp softmax dự đoán lớp cho các region proposals và giá trị offset của bounding box thông qua bbox regressor.



Hình 2.6: Hình ảnh minh họa ROI Pooling

Bằng việc tạo conv feature map bằng ConvNet, Fast R-CNN đã tránh được việc phải đưa 2000 region proposals vào mạng CNN với mỗi ảnh. Thay vào đó, mô

## 2. Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan

---

hình chỉ thực hiện trích xuất đặc trưng một lần mỗi ảnh, sau đó tạo ra một conv feature map dùng chung cho input. Từ đó, tăng tốc độ xử lý của Fast R-CNN một cách đáng kể so với R-CNN. Tuy nhiên, việc yêu cầu các region proposal của thuật toán làm giảm tốc độ xử lý so với trường hợp không sử dụng chúng. Do đó, region proposals trở thành một trong những hạn chế, ảnh hưởng đến hiệu suất trong cả mô hình Fast R-CNN và R-CNN.

### 2.2.4 Faster R-CNN (2016)

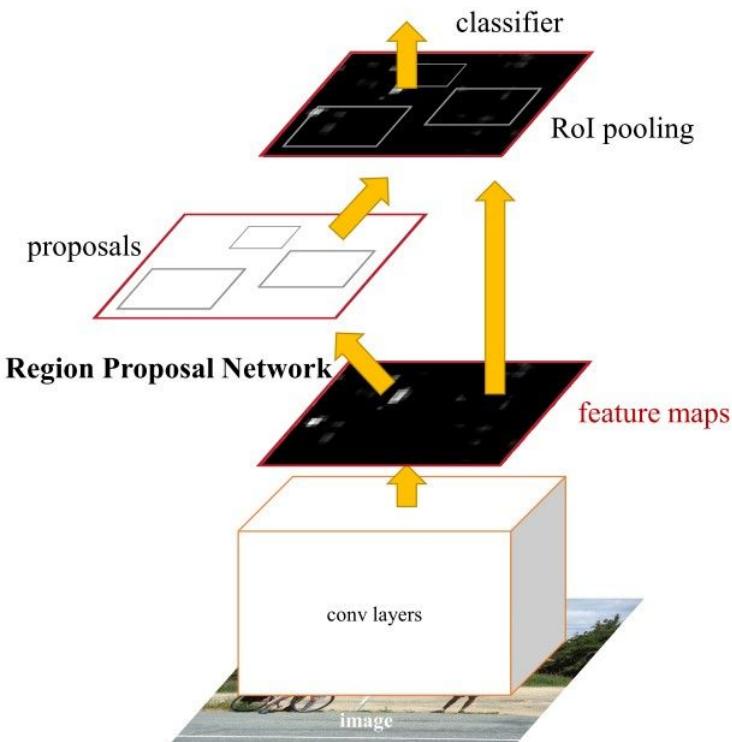
Các mạng nhận diện vật thể dựa trên thuật toán region proposals nhằm giả định vị trí của các đối tượng. Các thuật toán như SPPnet [8] và Fast R-CNN [5] đã giảm thời gian xử lý của các mạng nhận diện này, tuy nhiên vẫn chia sẻ chung một điểm hạn chế trong quá trình tính toán các region proposals. Để khắc phục hạn chế này, vào năm 2015, Shaoqing Ren cùng nhóm nghiên cứu của mình trong bài báo về Faster R-CNN [16] đã giới thiệu *Region Proposal Network* (RPN), qua đó giảm thiểu chi phí cần cho việc tính toán region proposals gần như bằng không. Faster R-CNN tốt hơn hẳn Fast R-CNN cả về tốc độ lẫn độ chính xác.

Những đóng góp chính trong bài báo này [16] bao gồm:

1. Sử dụng **region proposal network (RPN)**, một mạng tích chập toàn phần (Fully convolutional network) sinh ra các vùng đề xuất với các scale và aspect ratio khác nhau. Mạng RPN triển khai thuật ngữ neural network with attention nhằm hướng Fast R-CNN vị trí cần xử lý.
2. Đề xuất một khái niệm mới thay thế cho việc sử dụng **pyramids of images** (nhiều thể hiện có kích thước khác nhau của cùng một ảnh số) và **pyramids of filters** (nhiều filters với kích thước khác nhau), đó là **anchor boxes**. Mỗi vùng đề xuất được ánh xạ đến từng anchor box tương ứng, sau đó nhận diện các vật thể với các scale và aspect ratio khác nhau.
3. Tối ưu hóa xen kẽ, cho phép RPN và Fast R-CNN huấn luyện trên cùng một conv features. Qua đó làm giảm đáng kể thời gian tính toán.

Kiến trúc của Faster R-CNN được thể hiện như hình 2.7, bao gồm 2 module là **RPN** (tạo các region proposals) và **Fast R-CNN** (phát hiện các đối tượng

## 2. Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan



Hình 2.7: Kiến trúc của Faster R-CNN

trong region proposals). Conv feature maps được sử dụng chung cho cả hai module. Mô hình Faster R-CNN thực hiện xử lý theo các bước sau:

- RPN tạo ra các region proposals.
- Với mỗi region proposal trong ảnh, một vector đặc trưng có độ dài cố định được trích xuất sử dụng một lớp ROI Pooling.
- Các vector đặc trưng trên được phân lớp bằng module Fast R-CNN.
- Trả về bounding-boxes kèm theo scores của các đối tượng được nhận diện.

### Region Proposal Network

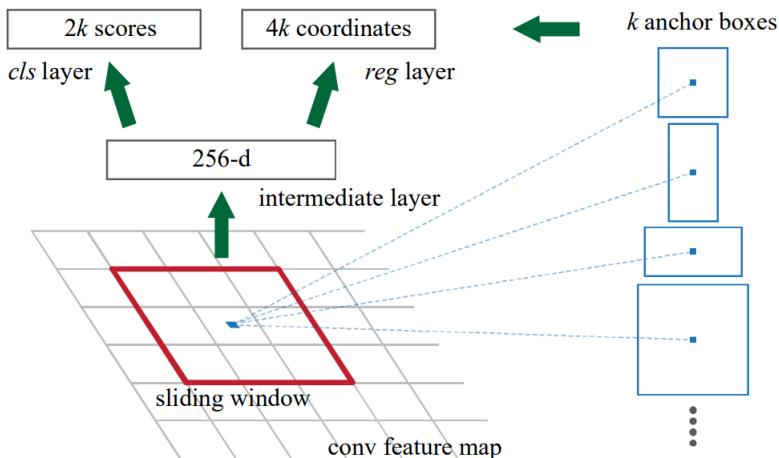
Một mạng RPN (Region Proposal Network) nhận một ảnh (có kích thước bất kỳ) làm đầu vào và đưa ra một tập hợp các đề xuất vật thể được chứa trong hình chữ nhật, mỗi đề xuất được gán một **Object Score**. Quá trình này được mô hình hóa bằng một mạng tích chập toàn phần.

## 2. Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan

Nhằm tạo ra các region proposals, một mạng nhỏ được sử dụng để trượt trên conv feature map (hình 2.8). Mạng này được kết nối đầy đủ với một cửa sổ không gian  $n \times n$  của conv feature map đầu vào. Mỗi cửa sổ trượt được ánh xạ thành một vector có số chiều ít hơn (256-d đối với ZF và 512-d đối với VGG). Một cửa sổ trượt có thể tạo ra rất nhiều region proposals. Tuy nhiên, các region proposals này không được đưa vào các mạng hồi quy và phân lớp ngay lập tức mà được loại bỏ bớt dựa trên "objectness score".

### Translation-Invariant Anchors

Kết quả của quá trình trên, mô hình đưa ra đồng thời  $k$  region proposals, do đó lớp *reg* có  $4k$  đầu ra nhằm mã hóa tọa độ của  $k$  đề xuất. Trong khi lớp *cls* có  $2k$  đầu ra nhằm ước tính xác suất là vật thể hay không phải vật thể với mỗi đề xuất. Mỗi đề xuất được tham số hóa theo một hộp tham chiếu, gọi là *anchor box*. Các anchor được căn chỉnh ở chính giữa tại cửa sổ trượt tương ứng, và bao gồm 2 tham số là *Scale* và *Aspect ratio*. Số lượng scale và aspect ratio thường được chọn bằng 3, cho ra  $k = 9$  anchors (multi-scale anchors) tại mỗi vị trí trượt. Đối với một feature map có kích thước  $W \times H$  ( $\sim 2,400$ ), thuật toán sẽ đưa ra tổng cộng  $WHk$  anchors.



Hình 2.8: Hình ảnh minh họa kiến trúc RPN (ảnh trích xuất từ bài báo Ren et al. [16])

Việc sử dụng các anchor boxes làm cho mô hình có thể có được scale-invariant (thuộc tính mô hình không thay đổi khi kích thước của vật thể thay

## 2. Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan

đổi) và translation-invariant (thuộc tính mô hình không thay đổi khi vị trí của vật thể thay đổi), trong khi chỉ yêu cầu 1 ảnh số với 1 kích thước duy nhất. Điều này giúp cho mô hình tránh phải sử dụng nhiều ảnh số hay nhiều filters. Các multi-scale anchors là điều kiện thiết yếu giúp chia sẻ các đặc trưng giữa RPN và mạng Fast R-CNN, qua đó giảm thiểu số lượng tham số và thời gian xử lý.

### **Object Score**

Với mỗi anchor, lớp *cls* trả về một vector có độ dài 2 thể hiện việc là vật thể hoặc không chứa vật thể (background) của anchor đó. Nếu vector trả về có giá trị [1, 0], thì anchor đang xét được phân loại là background. Ngược lại, nếu vector trả về có giá trị [0, 1], anchor đó có chứa vật thể.

Để huấn luyện mạng RPNs, mỗi anchors được gán một nhãn nhị phân (là vật thể hoặc không là vật thể). Các nhãn positive hoặc negative được gán cho anchors dựa trên một độ đo gọi là Intersection-over-Union (IoU).

IoU đo lường tỉ lệ giữa diện tích phần giao giữa 2 vùng quan tâm (region of interest, RoIs) và diện tích phần hợp của chúng. Giá trị của IoU nằm trong khoảng từ 0.0 đến 1.0. Khi 2 RoIs không giao nhau, IoU nhận giá trị là 0.0. Ngược lại, IoU bằng 1.0 khi 2 RoIs hoàn toàn trùng khớp với nhau. Cụ thể, trong quá trình huấn luyện RPNs, IoU được tính giữa các anchors và các ground-truth box.

Dựa trên IoU giữa anchor box và ground-truth box, RPN có thể quyết định objectness score cho từng anchor box theo các quy tắc sau: (i) Anchor box có IoU lớn hơn 0.7 với bất kỳ ground-truth box nào có objectness score là positive, (ii) nếu không anchor nào có IoU lớn hơn 0.7, nhãn positive được gán cho anchor(s) có IoU cao nhất với một ground-truth box bất kỳ, (iii) một anchor có objectness score là negative nếu IoU trên tất cả các ground-truth box nhỏ hơn 0.3 và (iv) các anchor có objectness score không phải là positive hay negative không ảnh hưởng đến tính khách quan của quá trình huấn luyện.

## 2. Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan

---

Objectness score có thể được tổng quát bằng công thức sau:

$$Objectness\_score(IoU) = \begin{cases} positive, & IoU > 0.7 \\ negative, & IoU < 0.3 \\ just\_ignore, & otherwise \end{cases} \quad (2.1)$$

With special case (ii)

### Loss Function

Dựa vào các định nghĩa đã được nêu ở trên, ta cần cực tiểu hóa một hàm mục tiêu dựa trên hàm mất mát đa nhiệm (multi-task loss) trong Fast R-CNN [5]. Hàm loss được định nghĩa như sau:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (2.2)$$

Trong công thức 2.2,  $i$  đại diện cho chỉ số của một anchor trong mini-batch. Kí hiệu  $p_i$  đại diện cho objectness score của anchor  $i$ , trong khi  $p_i^*$  thể hiện nhãn đúng (ground-truth label) cho anchor đó,  $p_i^*$  bằng 1 nếu anchor chứa đối tượng và bằng 0 nếu anchor là nền (không chứa đối tượng). Các kí hiệu  $t_i$  và  $t_i^*$  là lần lượt thể hiện vector tọa độ của bounding-box dự đoán và ground-truth tương ứng cho anchor  $i$ . Hàm mất mát phân lớp  $L_{cls}$  được tính bằng hàm log dựa trên 2 lớp (đối tượng và không phải đối tượng). Đối với hàm loss cho regression,  $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$ , trong đó  $R$  là hàm loss robust (smooth  $L_1$ ) được định nghĩa trong mô hình Fast R-CNN [5]. Hàm regression loss chỉ được kích hoạt đối với các anchor được dự đoán là có chứa vật thể ( $p_i = 1$ ) và bị vô hiệu hóa nếu không phải positive anchor ( $p_i = 0$ ). Các lớp  $cls$  và  $reg$  cho ra  $\{p_i\}$  và  $\{t_i\}$  tương ứng. Các giá trị này sau đó được chuẩn hóa với  $N_{cls}$  và  $N_{reg}$ . Hằng số  $\lambda$  được sử dụng để cân bằng ảnh hưởng của 2 thành phần loss trong tổng loss.

## 2. Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan

---

Đối với lớp *reg*, 4 tọa độ được tham số như sau [6]:

$$\begin{aligned} t_x &= (x - x_a)/w_a, & t_y &= (y - y_a)/h_a, & t_w &= \log(w/w_a), & t_h &= \log(h/h_a), \\ t_x^* &= (x^* - x_a)/w_a, & t_y^* &= (y^* - y_a)/h_a, & t_w^* &= \log(w^*/w_a), & t_h^* &= \log(h^*/h_a) \end{aligned}$$

Trong đó,  $x, y, w, h$  lần lượt là 2 tọa độ tâm của hộp, chiều rộng và chiều cao. Các tham số  $x, x_a, x^*$  lần lượt thuộc về predicted box, anchor box và ground-truth box (tương tự với  $y, w, h$ ). Công thức này có thể được sử dụng như là quá trình bounding-box regression từ một anchor box đến ground-truth box gần đó.

### Feature Sharing

Cả 2 module trong kiến trúc của Faster R-CNN, bao gồm RPN và Fast R-CNN, là các mạng độc lập với nhau. Do được huấn luyện độc lập với nhau, chúng tinh chỉnh lớp conv theo các cách khác nhau. Tuy nhiên, trong kiến trúc Faster R-CNN sử dụng một mạng hợp nhất, cho phép RPN và Fast R-CNN được huấn luyện một cách đồng thời.

Ý tưởng cốt lõi là xây dựng một mạng cho phép chia sẻ feature maps giữa hai mạng, thay vì học hai mạng tách biệt. Các feature maps chỉ cần tính toán một lần duy nhất, nhưng phải được dùng trong cả hai. Các kỹ thuật giải quyết vấn đề này có thể được gọi là **feature sharing** hay **layer sharing**. Nhờ có anchor box, việc chia sẻ các đặc trưng hay các tầng giữa hai modules trong Faster R-CNN là hoàn toàn khả thi.

Trong nghiên cứu này [5], nhóm tác giả chỉ ra rằng cách giải quyết vấn đề trên không đơn giản chỉ định nghĩa một mạng duy nhất bao gồm cả RPN và Fast R-CNN, sau đó tối ưu hóa chúng đồng thời với back-propagation. Lí do là quá trình huấn luyện Fast R-CNN phụ thuộc vào các object proposals cố định, làm cho việc hội tụ của việc học Fast R-CNN trong khi thay đổi cơ chế đề xuất đối tượng bằng RPN trở nên không rõ ràng. Trong khi việc tối ưu hóa đồng thời này là một câu hỏi thú vị trong tương lai, nhóm tác giả đã phát triển một thuật toán huấn luyện 4 bước nhằm chia sẻ các đặc trưng trong quá trình học. Thuật toán này được thiết kế để chia sẻ feature maps giữa RPN và Fast R-CNN thông qua việc tối ưu xen kẽ giữa hai phần của mạng.

## 2. Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan

### Huấn luyện Faster R-CNN

Bước đầu tiên, mạng RPN được huấn luyện với kiến trúc đã đề cập ở trên nhằm tạo ra các region proposals. Các trọng số ở lớp conv chia sẻ của mạng này được khởi tạo với mô hình pre-trained trên ImageNet và được điều chỉnh lại nhằm tối ưu cho bài toán đề xuất đối tượng, trong khi các trọng số khác được khởi tạo một cách ngẫu nhiên. Ở bước thứ hai, một mạng nhận diện bằng Fast R-CNN được huấn luyện độc lập sử dụng các vùng đề xuất được tạo bởi RPN trong bước đầu tiên. Mạng nhận diện này cũng được khởi tạo bằng mô hình đã được huấn luyện trên ImageNet. Sau đó, ở bước thứ ba, mạng nhận diện này được sử dụng để khởi tạo lại việc huấn luyện RPN. Tuy nhiên, tác giả chỉ điều chỉnh lại các tầng riêng biệt của mạng RPN và giữ nguyên các tầng tích chập được chia sẻ giữa hai mạng FPN và Fast R-CNN. Cuối cùng, ở bước thứ tư, bằng việc giữ nguyên tầng tích chập chia sẻ giữa hai mạng và chỉ tinh chỉnh tầng kết nối đầy đủ (fully connected) của Fast R-CNN, cả hai mạng RPN và Fast R-CNN chia sẻ cùng một tầng conv và hình thành một mạng thống nhất.

Phương pháp huấn luyện để chia sẻ các tầng tích chập giữa hai mạng RPN và Fast R-CNN được nhóm tác giả trình bày còn được gọi là **Alternating Training**, ngoài ra, vẫn còn hai phương pháp khác, đó là **Approximate Joint Training** và **Non-Approximate Joint Training**.

### Kết luận

Faster R-CNN là một mô hình nhận diện vật thể được đánh giá cao về độ chính xác và tốc độ xử lý. Với sự kết hợp giữa RPN và Fast R-CNN, Faster R-CNN đã giải quyết được vấn đề tốc độ của các mô hình trước đó, đồng thời giảm thiểu số lượng các region proposals bằng cách thiết lập các điểm Anchor trong RPN. Các kết quả thực nghiệm cho thấy Faster R-CNN có độ chính xác cao hơn so với R-CNN và Fast R-CNN. Với mạng ZF [13], Faster R-CNN đạt độ chính xác mAP là 59.9% trên tập test PASCAL VOC 2007, cao hơn 5% so với Fast R-CNN. Trong khi sử dụng mạng VGG-16 [17], Faster R-CNN vẫn giữ được tỷ lệ khung hình bằng 5fps trên GPU và đạt được kết quả với 73.2% mAP trên PASCAL VOC 2007 và 70.4% mAP vào năm 2012 trong khi sử dụng 300 đề xuất cho mỗi ảnh. Ngoài ra, Faster R-CNN còn có khả năng học được các đặc trưng chung của

## **2. Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan**

---

các vật thể trong quá trình huấn luyện, cho phép áp dụng mô hình trên nhiều bộ dữ liệu khác nhau mà không cần tối ưu lại các siêu tham số của mô hình.

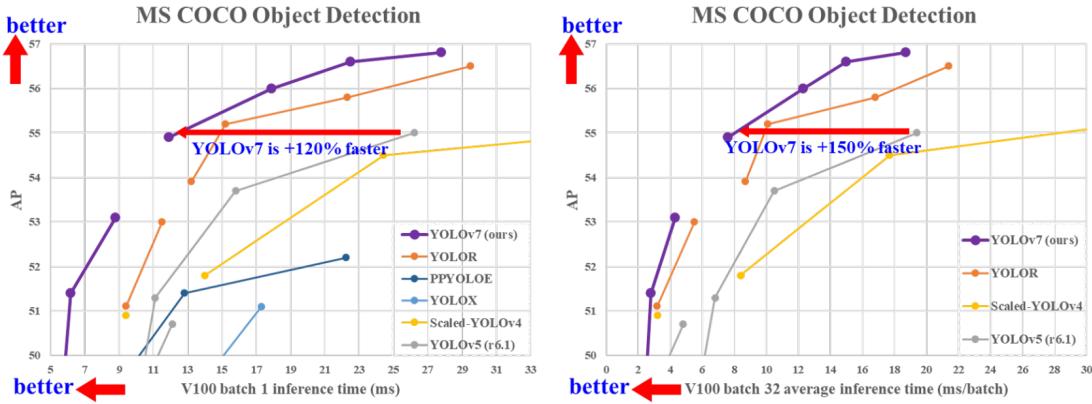
Tuy nhiên, Faster R-CNN vẫn tồn tại một số nhược điểm. Việc sử dụng RPN để tạo ra các region proposals vẫn làm tăng thêm độ phức tạp tính toán của mô hình. Lí do là vì tất cả các anchors trong một mini-batch đều phải được trích xuất từ một bức ảnh, làm cho mạng mất nhiều thời gian hơn để có thể hội tụ. Các thực nghiệm cho thấy thời gian xử lý của Faster R-CNN trên một ảnh là khoảng 198ms (với VGG-16), tuy nhiên, con số này có thể tăng lên nếu kích thước ảnh và số lượng vật thể tăng lên. Tuy nhiên, với sự phát triển của các mô hình mạng neural và các thuật toán cải thiện, Faster R-CNN vẫn là một trong những giải pháp được sử dụng nhiều cho các bài toán nhận diện vật thể hiện nay.

### **2.3 YOLOv7 (2022)**

Được giới thiệu không lâu sau YOLOv6 bởi Wang et al, YOLOv7 [21] là một thuật toán phát hiện đối tượng tiên tiến được thiết kế để nhận dạng và định vị các đối tượng trong hình ảnh và video trong thời gian thực. Đây là phiên bản mở rộng của họ phát hiện đối tượng YOLO (You Only Look Once) phổ biến. Khác với các thuật toán phát hiện đối tượng truyền thống sử dụng nhiều giai đoạn (multi-stage), YOLO là một thuật toán one-stage, nghĩa là chỉ sử dụng một mạng neural để thực hiện đồng thời phát hiện, phân loại và xác định vị trí của đối tượng trực tiếp trên ảnh đầu vào. Việc sử dụng một mạng neural duy nhất giúp YOLO hoạt động nhanh hơn và đồng thời cũng giảm thiểu được sự mất mát thông tin giữa các giai đoạn trong quá trình phát hiện.

YOLOv7 vượt qua mọi mô hình phát hiện đối tượng và các phiên bản YOLO trước đó trong cả tốc độ và độ chính xác trong khoảng từ 5 FPS đến 160 FPS và có độ chính xác cao nhất với 56.8% AP trong số toàn bộ các mô hình phát hiện đối tượng hiện có, trong khi vẫn giữ được tốc độ 30 FPS hoặc nhiều hơn trên GPU V100. Bên cạnh đó, YOLOv7 yêu cầu phần cứng thấp hơn các mạng neural khác và có thể được huấn luyện nhanh hơn trên các dataset nhỏ mà không cần sử dụng bất kỳ pre-trained nào.

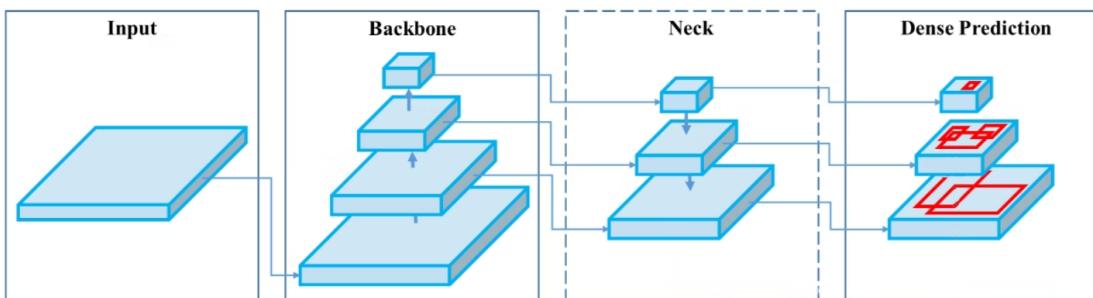
## 2. Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan



**Hình 2.9:** So sánh hiệu suất của Yolov7 và các real-time detector khác (Ảnh trích xuất từ bài báo [21])

### 2.3.1 Kiến trúc

Một trong những cải tiến của YOLOv7 là thay đổi hàm kích hoạt (activation function) từ Leaky ReLU [12] sang Swish [14] - một hàm kích hoạt mới hơn, có độ chính xác cao hơn và thời gian huấn luyện nhanh hơn so với Leaky ReLU. Các module cơ bản khác được tối ưu hóa dựa trên ý tưởng thiết kế của residual connection [9], nhưng kiến trúc cơ bản của không có nhiều sự thay đổi và vẫn bao gồm bốn phần chính, là: Input, Backbone, Neck và Prediction (Head).



**Hình 2.10:** Hình ảnh minh họa kiến trúc One-stage detector

#### Input

Trong giai đoạn tiền xử lý, mô hình YOLOv7 áp dụng cả các kỹ thuật tăng cường dữ liệu [1, 18], đồng thời khai thác phương pháp tính toán anchor thích ứng (adaptive anchor) được thiết lập bởi YOLOv5 [7], đảm bảo rằng các hình

## **2. Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan**

---

ảnh đầu vào được scale về tỷ lệ  $640 \times 640$ , đáp ứng yêu cầu về kích thước đầu vào của mạng Backbone.

### **Backbone**

Mạng Backbone đóng vai trò quan trọng nhằm trích xuất các đặc trưng. Mạng Backbone cơ bản của các thuật toán YOLO là Darknet [15], được xây dựng bởi Joseph Redmon vào năm 2016. Các phiên bản YOLO khác nhau tối ưu hóa mạng Darknet này theo các cách khác nhau, phụ thuộc vào kiến trúc của chúng. Mạng Backbone của YOLOv7 được cấu hình bao gồm ba thành phần chính: CBS (kết hợp giữa Convolution, Batch-norm và hàm kích hoạt SiLU), Extended-ELAN (E-ELAN) và MP1. Module E-ELAN cải thiện khả năng học của mạng bằng cách hướng các khối tính toán đặc trưng khác nhau để học được nhiều đặc trưng đa dạng hơn, trong khi vẫn giữ nguyên kiến trúc ELAN ban đầu và đường gradient gốc. Thành phần MaxPool-1 (MP1) bao gồm CBS và MaxPool, được chia thành nhánh trên và nhánh dưới. MaxPool trích xuất thông tin mang giá trị lớn nhất của các vùng địa phương (local) trong khi CBS trích xuất tất cả các thông tin về giá trị của các vùng địa phương, từ đó nâng cao khả năng trích xuất đặc trưng của mạng.

### **Neck (Feature Fusion Zone)**

Lớp hợp nhất đặc trưng (feature fusion) được thiết kế nhằm cung cấp khả năng học tốt hơn các đặc trưng được trích xuất từ mạng Backbone. Các đặc trưng của các mức độ chi tiết khác nhau được học độc lập với nhau tại mạng Backbone và sau đó được hợp nhất tại lớp Neck của mạng, nhằm học được nhiều đặc trưng nhất có thể.

Mạng Neck của YOLOv7 được cấu trúc bằng kiến trúc Feature Pyramid Network (FPN)[10], dựa trên thiết kế PANet [11]. Mạng này bao gồm các CBS, cùng với cấu trúc Spatial Pyramid Pooling and Convolutional Spatial Pyramid Pooling (bao gồm hai mạng là SPPnet và CSPnet) [8, 20], cùng với mạng E-ELAN và MaxPool-2 (MP2).

## 2. Cơ Sở Lý Thuyết Và Các Nghiên Cứu Liên Quan

### Dense Prediction

Áp dụng những ưu điểm của các thuật toán trước, YOLOv7 vẫn giữ lại ba detection heads. Các detection heads này được sử dụng để phát hiện đối tượng, đồng thời trả về xác suất dự đoán (confidence score) và tọa độ khung dự đoán của đối tượng đó.

Mạng dự đoán của YOLOv7 sử dụng kiến trúc Rep, lấy cảm hứng từ RepVGG [4] và giới thiệu thiết kế residual đặc biệt nhằm hỗ trợ quá trình học của mạng. Thiết kế residual có thể đơn giản hóa thành một mạng tích chập trong các dự đoán thực tế, từ đó giảm độ phức tạp của mạng mà không ảnh hưởng đến độ chính xác dự đoán. Các đặc trưng từ tầng Neck được đưa qua mạng Rep để điều chỉnh số kênh ảnh, sau đó được áp dụng một tích chập để dự đoán độ tin cậy, loại vật thể và anchor.

### 2.3.2 Trainable bag-of-freebies

Bag-of-freebies (BoF) là một thuật ngữ được sử dụng trong lĩnh vực Machine Learning và Thị giác Máy tính. BoF bao gồm các kỹ thuật tiền xử lý hoặc tăng cường dữ liệu nhằm tăng cường hiệu suất mà không làm tăng độ phức tạp của mô hình. Trong họ YOLO, BoF lần đầu tiên được đề cập đến trong YOLOv4 [1].

## **Chương 3**

### **Thực Nghiệm VÀ Kết Quả**

**3.1 Cài đặt thực nghiệm**

**3.2 Kết quả thực nghiệm**

## **Chương 4**

### **Kết luận và hướng phát triển**

# **Phụ lục A**

Đây là phụ lục A

## **Phụ lục B**

Đây là phụ lục B

# References

- [1] BOCHKOVSKIY, A., WANG, C.Y. & LIAO, H.Y.M. (2020). Yolov4: Optimal speed and accuracy of object detection. [20](#), [22](#)
- [2] CHENG, B., WEI, Y., SHI, H., FERIS, R., XIONG, J. & HUANG, T. (2018). Revisiting rcnn: On awakening the classification power of faster rcnn. In *Proceedings of the European conference on computer vision (ECCV)*, 453–468.
- [3] DENG, J., XUAN, X., WANG, W., LI, Z., YAO, H. & WANG, Z. (2020). A review of research on object detection based on deep learning. In *Journal of Physics: Conference Series*, vol. 1684, 012028, IOP Publishing. [6](#)
- [4] DING, X., ZHANG, X., MA, N., HAN, J., DING, G. & SUN, J. (2021). Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13733–13742. [22](#)
- [5] GIRSHICK, R. (2015). Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. [10](#), [12](#), [16](#), [17](#)
- [6] GIRSHICK, R., DONAHUE, J., DARRELL, T. & MALIK, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. [iii](#), [8](#), [9](#), [10](#), [17](#)
- [7] GLENN, J. (2022). Yolov5 version 7.0. <https://github.com/ultralytics/yolov5/releases>. [20](#)

---

## REFERENCES

- [8] HE, K., ZHANG, X., REN, S. & SUN, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, **37**, 1904–1916. [12](#), [21](#)
- [9] HE, K., ZHANG, X., REN, S. & SUN, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778. [20](#)
- [10] LIN, T.Y., DOLLÁR, P., GIRSHICK, R., HE, K., HARIHARAN, B. & BELONGIE, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117–2125. [21](#)
- [11] LIU, S., QI, L., QIN, H., SHI, J. & JIA, J. (2018). Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8759–8768. [21](#)
- [12] MAAS, A.L., HANNUN, A.Y., NG, A.Y. *et al.* (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, vol. 30, 3, Atlanta, Georgia, USA. [20](#)
- [13] MATTHEW ZEILER, D. & ROB, F. (2014). Visualizing and understanding convolutional neural networks. *ECCV*. [18](#)
- [14] RAMACHANDRAN, P., ZOPH, B. & LE, Q.V. (2017). Searching for activation functions. *arXiv preprint arXiv:1710.05941*. [20](#)
- [15] REDMON, J., DIVVALA, S., GIRSHICK, R. & FARHADI, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [21](#)
- [16] REN, S., HE, K., GIRSHICK, R. & SUN, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, **28**. [iii](#), [5](#), [12](#), [14](#)
- [17] SIMONYAN, K. & ZISSERMAN, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. [18](#)

## **REFERENCES**

---

- [18] TAN, M. & LE, Q.V. (2019). Mixconv: Mixed depthwise convolutional kernels. *arXiv preprint arXiv:1907.09595*. [20](#)
- [19] UIJLINGS, J.R., VAN DE SANDE, K.E., GEVERS, T. & SMEULDERS, A.W. (2013). Selective search for object recognition. *International journal of computer vision*, **104**, 154–171. [9](#)
- [20] WANG, C.Y., LIAO, H.Y.M., WU, Y.H., CHEN, P.Y., HSIEH, J.W. & YEH, I.H. (2020). CspNet: A new backbone that can enhance learning capability of cnn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 390–391. [21](#)
- [21] WANG, C.Y., BOCHKOVSKIY, A. & LIAO, H.Y.M. (2022). Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. [iii, 5, 19, 20](#)