ĐẠI HỌC QUỐC GIA TP HCM

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

NGUYỄN DUY KHÁNH

PHƯƠNG PHÁP PHÁT HIỆN ĐỐI TƯỢNG KHÓ TRONG ẢNH

Chuyên ngành: Khoa học Máy tính

Mã số: 62.48.01.01

TÓM TẮT LUẬN ÁN TIẾN SĨ KHOA HỌC MÁY TÍNH

NGƯỜI HƯỚNG DẪN KHOA HỌC

PGS.TS. DƯƠNG ANH ĐỨC

PGS.TS. LÊ ĐÌNH DUY

TP HỒ CHÍ MINH – Năm 2020

Công trình được hoàn thành tại: Trường Đại học Công nghệ Thông tin – Đại học Quốc gia TP. Hồ Chí Minh.

Người hướng dẫn khoa học:
Hướng dẫn 1: PGS.TS. Dương Anh Đức
Hướng dẫn 2: PGS.TS. Lê Đình Duy

Phản biện 1: PGS.TS. Nguyễn Thanh Bình
Phản biện 2: TS. Ngô Quốc Việt

Luận án sẽ/đã được bảo vệ trước
Hội đồng chấm luận án cấp Trường tại : Trường Đại học Công nghệ Thông tin – Đại học Quốc gia TP. Hồ Chí Minh
vào lúc 14 giờ ngày 24 tháng 02 năm 2021. Có thể tìm hiểu luận án tại:
- Thư viện Quốc gia Việt Nam
- Thư viện Trường Đại học Công nghệ Thông tin

Mục lục

MỞ ĐẦU

Dẫn nhập

Phát hiện đối tượng là một trong những bài toán quan trọng của thị giác máy tính với các ứng dụng trải rộng trong nhiều lĩnh vực khác nhau như: công nghệ robot (*robotics*), xử lý ảnh y khoa, các hệ thống giám sát, hệ thống tương tác người-máy, giao thông thông minh. Trong công nghệ robot, phát hiện đối tượng hỗ trợ việc định vị cũng như nhận dạng các đối tượng nhờ đó robot có thể tương tác chính xác với các đối tượng trong thực tế. Trong lĩnh vực xử lý ảnh y khoa, các ảnh chụp (như X quang) có thể được xử lý tự động để phát hiện các vùng bất thường (ví dụ như vùng chứa khối ung thư). Đối với các hệ thống giám sát, phát hiện đối tượng hỗ trợ khả năng phát hiện người, phương tiện, vật thể được ghi hình thông qua hệ thống camera. Dữ liệu này sẽ tiếp tục xử lý để phục vụ các chức năng nâng cao. Trong các hệ thống tương tác người-máy, vị trí khuôn mặt người hoặc cánh tay sẽ được xác định thông qua các thuật toán phát hiện đối tượng, sau đó được nhận dạng, phân tích để xác định các chỉ thị cho máy. Trong giao thông thông minh, phát hiện đối tượng là một trong những thành phần quan trọng trong các xe tự hành, nhằm trang bị khả năng nhận biết các vật cản một cách tự động.

Trong những năm gần đây, các phương pháp phát hiện đối tượng đã phát triển mạnh mẽ, đặc biệt đạt được bước cải tiến lớn về cả độ chính xác và tốc độ xử lý. Rất nhiều công trình nghiên cứu được đề xuất, từ việc sử dụng các loại đặc trưng tự thiết kế như Haar-like [10], HOG [11], DPM [12] trong các phương pháp truyền thống đến việc sử dụng các kỹ thuật hiện đại dựa trên mạng học sâu như R-CNN [13], Fast R-CNN [14], Faster R-CNN [6], Mask R-CNN [15], YOLO [16], SSD [17], Retinanet [5]. Một số công trình đã khảo sát và phân tích chi tiết các cách tiếp cận cho bài toán đã được công bố gần đây, bao gồm các công trình đáng chú ý như [3, 18, 19] (2019), [20, 21, 4] (2018). Để đạt hiệu quả

cao, các bộ phát hiện đối tượng phải hoạt động tốt trước nhiều thách thức đã được định nghĩa rõ ràng như các thay đổi về ánh sáng môi trường, hình dáng của đối tượng, ảnh nền nhập nhằng, độ phân giải thấp, đối tượng bị che khuất, có nhiều kích thước, góc độ, hay sự đa dạng trong nội bộ lớp đối tượng.

Bằng việc xem xét kết quả từ các mô hình phát hiện đối tượng hiện có, chúng tôi nhận thấy có nhiều **đối tượng khó** thường bị bỏ qua hoặc dự đoán sai. Nguyên nhân chính đó là do quá trình huấn luyện với việc tối thiểu hàm mất mát trên toàn bộ tập dữ liệu khiến mô hình bị lệch về phía các đối tượng dễ (thông thường có số lượng mẫu vượt trội). Việc phát hiện thành công các đối tượng khó này sẽ hứa hẹn nâng cao hiệu suất cho các mô hình phát hiện đối tượng. Do vậy, trong luận án này chúng tôi tập trung vào việc đề xuất các phương pháp để phát hiện các đối tượng khó, nhằm cải tiến các mô hình phát hiện đối tượng hiện có.

Mục tiêu và nội dung thực hiện của luận án

Mục tiêu: luận án tập trung vào việc nghiên cứu và đề xuất các phương pháp phát hiện đối tượng khó trong ảnh.

Luận án đề ra các nội dung cụ thể như sau:

1. Khảo sát các hướng nghiên cứu gần đây trên bài toán phát hiện đối tượng trong ảnh.

   a. Các hướng tiếp cận truyền thống sử dụng đặc trưng tự thiết kế (*handcrafted features*).

   b. Các hướng tiếp cận hiện đại dựa trên mạng học sâu.

2. Nghiên cứu việc phát hiện các đối tượng khó để cải tiến cho phương pháp phát hiện đối tượng dựa trên mạng học sâu.

   a. Phát hiện đối tượng khó trên tập đối tượng bị bỏ sót bởi các phương pháp học sâu.

   b. Phát triển phương pháp phát sinh dữ liệu nhân tạo (tập trung vào các đối tượng khó) nhằm tăng cường hiệu quả cho việc phát hiện các đối tượng khó trong tập ảnh thực nghiệm.

Đối tượng và phạm vi nghiên cứu

- Đối tượng:

  a. Các phương pháp phát hiện đối tượng truyền thống.

  b. Các phương pháp phát hiện đối tượng dựa trên mạng học sâu.

  c. Các tập dữ liệu cho các bài toán phát hiện đối tượng: phát hiện các đối tượng tham gia giao thông (KITTI), phát hiện các đối tượng phổ biến (PASCAL VOC, COCO).

- Phạm vi: phạm vi của luận án này được giới hạn trên các tập dữ liệu ảnh tĩnh cho bài toán phát hiện đối tượng. Trong đó các thực nghiệm được tiến hành trên các tập dữ liệu được cộng đồng nghiên cứu thừa nhận và sử dụng liên quan đến bài toán phát hiện đối tượng trên ảnh tĩnh.

Các đóng góp chính của luận án

1. Đề xuất phương pháp phát hiện lại đối tượng khó trên tập đối tượng bỏ sót sử dụng mạng học sâu ở hai giai đoạn. Nội dung của phương pháp này đã được công bố ở tạp chí JVCI-2019 [CT.1].

2. Đề xuất phương pháp phát sinh dữ liệu nhân tạo nhằm tăng cường hiệu quả cho việc phát hiện các đối tượng khó. Nội dung của phương pháp này đã được công bố ở tạp chí MTAP-2019 [CT.2].

3. Phát triển thuật toán phát hiện đối tượng cho các bài toán liên quan: phát hiện đối tượng chính trong ảnh (*Salient Object Detection*). Nội dung của phương pháp này đã được công bố ở tạp chí IEEE TIP-2019 [CT.3].

Bố cục của luận án

Luận án được bố cục gồm các chương mục như sau: **Mở đầu:** Giới thiệu tóm tắt về động cơ, mục tiêu, nội dung nghiên cứu và các đóng góp chính của luận án; **Chương 1**: Giới thiệu về bài toán phát hiện đối tượng; **Chương 2**: Trình bày về cơ sở lý thuyết và các hướng tiếp cận cho bài toán; **Chương 3**: Trình bày các phương pháp phát hiện đối tượng khó được đề xuất [CT1, CT2]; **Chương 4**: Trình bày các thử nghiệm và kết quả đạt được; **Chương 5**: Trình bày phương

pháp áp dụng kết quả của phát hiện đối tượng cho bài toán phát hiện đối tượng chính trong ảnh [CT3]; **Chương 6**: Thảo luận về ưu nhược điểm của các phương pháp đề xuất và hướng phát triển.

Chương 1
GIỚI THIỆU BÀI TOÁN

1.1 Giới thiệu bài toán Phát hiện đối tượng tổng quát

1.1.1 Định nghĩa và mục tiêu giải quyết của bài toán

Trong thị giác máy tính, phát hiện đối tượng là một trong những bài toán thu hút được khá nhiều sự quan tâm. Phát hiện đối tượng thông thường được định nghĩa là bài toán xác định vị trí của tất cả các thể hiện (*instances*) của một số loại đối tượng được cho trước (ví dụ như "máy bay", "con người", "xe hơi", ...) trong ảnh. Phát hiện đối tượng tập trung đồng thời vào hai mục tiêu: xác định vị trí cụ thể của đối tượng trong ảnh và xác định tên loại mà đối tượng đó thuộc về. Vị trí của đối tượng có thể được chỉ ra dưới dạng khung bao đối tượng (hình chữ nhật), hoặc danh sách các điểm ảnh thuộc về đối tượng đó.

1.2 Đối tượng khó và các thách thức trong việc phát hiện

Qua việc xem xét kết quả của các phương pháp tân tiến (đã đề cập ở phần trên) trong việc phát hiện đối tượng, chúng tôi thấy rằng có một số lượng đáng kể các đối tượng không được phát hiện chính xác. Những đối tượng này có thể thuộc vào một trong số các trường hợp dưới đây:

- Các đối tượng bị dự đoán sai nhãn.
- Các đối tượng bị bỏ sót, không được phát hiện.
- Các đối tượng được phát hiện nhưng có giá trị độ tin cậy thấp dưới ngưỡng cần thỏa mãn.

(a) Ánh sáng thay đổi (b) Hình dáng biến đổi (c) Nhiều kích thước

(d) Bị che khuất (e) Ảnh nền nhập nhằng (f) Độ phân giải thấp

(g) Sự đa dạng trong mỗi lớp đối tượng

(h) Sự tương đồng của các lớp đối tượng khác nhau

Hình 1.1: Ảnh minh họa các thách thức của bài toán Phát hiện đối tượng [4].

Chúng tôi gọi những đối tượng này là *đối tượng khó*. Nguyên nhân phát sinh và các thách thức trong việc phát hiện các đối tượng này được phân loại như dưới đây:

1.2.1 Vấn đề thay đổi hình dáng của đối tượng

Một số loại đối tượng điển hình như con người, động vật có thể có nhiều tư thế dẫn đến hình dáng khác nhau trong quá trình hoạt động (xem minh họa ở Hình 1.1b). Đây là một thách thức truyền thống đã được đặt ra và giải quyết ở nhiều nghiên cứu, điển hình là [12], trong đó các phương pháp cần xây dựng một cơ chế biểu diễn đối tượng bền vững trước sự thay đổi về vị trí tương đối của các

bộ phận đối tượng. Tuy nhiên, trong một số trường hợp ngẫu nhiên, đối tượng được ghi nhận với tư thế đặc biệt (ví dụ như đối tượng "con ngựa" có thể đứng thẳng lên bằng hai chân sau). Những trường hợp đặc biệt này làm phát sinh các đối tượng với hình dáng đặc biệt và hiếm thấy, đồng thời với số lượng mẫu hiếm, dẫn đến các bộ phát hiện không được huấn luyện tốt cho các trường hợp này. Những trường hợp này cũng được xem là đối tượng khó.

1.2.2 Vấn đề đa dạng về kích thước và tỉ lệ của đối tượng

Đây là vấn đề thường gặp ở các đối tượng khó, phát sinh do sự đa dạng về khoảng cách và góc độ của thiết bị chụp ảnh. Một số đối tượng được lấy mẫu ở khoảng cách và góc độ khác biệt với đa số mẫu khác sẽ dẫn đến kích thước và tỉ lệ các chiều khác biệt (xem minh họa ở Hình 1.1c). Điều này dẫn đến sự thất bại của các bộ phát hiện đối tượng trên những đối tượng này.

Các bộ phát hiện đối tượng tiên tiến hiện nay cũng đã đề xuất một số giải pháp để giải quyết vấn đề phát hiện trên nhiều kích thước và tỉ lệ. Ví du như, R-CNN [13] sử dụng thuật toán đề xuất vùng ứng viên *Selective Search* [38] để dò tìm các đối tượng ở nhiều kích thước và tỉ lệ khác nhau. Một số phương pháp khác như Faster R-CNN [6], SSD [17], YOLO [16] xem vấn đề phát hiện trên nhiều kích thước và tỉ lệ dưới dạng một bài toán hồi qui, trong đó kích thước và tỉ lệ của đối tượng được dự đoán thông qua các tầng mạng nơ-ron. Các giải pháp này giúp bộ phát hiện đối tượng có thể phát hiện được đối tượng với các kích thước và tỉ lệ khác nhau. Tuy nhiên, chúng đều tồn tại các yếu điểm dẫn đến khả năng phát hiện đối tượng khó bị hạn chế. *Selective Search* sử dụng phân đoạn màu, vì vậy dễ bị ảnh hưởng bởi điều kiện ngoại cảnh. Trong khi đó chiến lược hồi qui của Faster RCNN, SSD, YOLO phụ thuộc vào cách chia các ô với kích thước và tỉ lệ mặc định (được gọi là "neo" - *anchor*).

1.2.3 Vấn đề đa dạng cao trong nội bộ lớp đối tượng

Vấn đề đa dạng trong nội bộ lớp (*intra-class variation*) cũng là một nguyên nhân khác tạo ra các đối tượng khó. Mỗi lớp đối tượng có thể có nhiều kiểu dáng, hình dạng, màu sắc khác nhau. Ví dụ như lớp đối tượng "Ghế" sẽ có nhiều chủng loại (xem minh họa ở Hình 1.1g). Các đối tượng khó là có thể nằm trong các chủng loại với số lượng mẫu hiếm, dẫn đến quá trình huấn luyện sẽ không đủ dữ liệu để định hướng bộ phát hiện đối tượng dò tìm chính xác các đối tượng này.

Đối mặt với vấn đề đa dạng cao trong nội bộ lớp, phương pháp giải quyết điển hình là khai thác các tập con (*sub-category*) dựa trên mạng học sâu [40]. Phương pháp này được xây dựng dựa trên bộ phát hiện đối tượng Faster R-CNN [6], trong đó bổ sung một lớp mạng cho RPN nhằm vào việc phân loại tập con và sau đó kết hợp thông tin về tập con này vào mạng RCNN kế tiếp để phân lớp chính xác. Tuy nhiên, để hoạt động hiệu quả thì các chủng loại cần có số lượng đủ lớn để có thể dễ dàng huấn luyện cho lớp mạng phân loại tập con. Đối với các chủng loại có số lượng ít (các đối tượng " khó") thì việc phân nhóm sẽ mang tính thách thức cao hơn.

1.2.4 Vấn đề tương đồng giữa các lớp đối tượng

Bên cạnh sự đa dạng trong mỗi lớp đối tượng như đã trình bày ở mục trước, các lớp đối tượng lại tồn tại những trường hợp có sự tương đồng về hình hạng khá lớn (xem minh họa ở Hình 1.1h). Vấn đề tương đồng này (*inter-class variation*) làm bộ phát hiện đối tượng có thể gán nhãn nhầm lẫn giữa các lớp đối tượng với nhau. Vấn đề này được quan tâm nhiều hơn trong bài toán về nhận dạng ảnh, trong đó các nghiên cứu đề xuất việc lựa chọn các bộ phận mang tính phân biệt cao của đối tượng để rút trích đặc trưng [41]. Các đối tượng khó cũng thường là những đối tượng bị gán nhãn sai do sự tương đồng về hình dạng giữa các lớp đối tượng.

1.2.5 Các nguyên nhân từ điều kiện môi trường

Ngoài ra các đối tượng khó cũng có thể được phát sinh do các thách thức trong quá trình lấy mẫu dưới các điều kiện môi trường ngẫu nhiên. Một số điều kiện phức tạp có thể được liệt kê như dưới đây:

- Các điều kiện môi trường như ánh sáng, thời tiết phức tạp trong quá trình lấy mẫu cũng dẫn đến sự khác biệt về màu sắc, cạnh, hay vân ảnh của đối tượng (xem minh họa ở Hình 1.1a). Các đối tượng khó có thể phát sinh trong các điều kiện ánh sáng kém (như ngược sáng, mức sáng thấp, chói sáng), thời tiết phức tạp (sương mù, mưa) từ đó làm biến đối các đặc trưng về màu sắc, cạnh, hay vân ảnh của đối tượng.

- Đối tượng bị che khuất: đối tượng thường bị che khuất bởi ảnh nền (các đối tượng không quan tâm) hoặc bởi chính các đối tượng khác. Việc bị che

khuất dẫn đến đối tượng được ghi nhận với hình dạng khác thường (kèm với mất mát các đặc trưng) và gây ra các thách thức cho quá trình phát hiện cũng như xác định vị trí chính xác (xem minh họa ở Hình 1.1d). Các khả năng bị che khuất rất đa dạng, dẫn đến hình dạng trong ảnh của đối tượng khó có thể dự đoán trước. Một số trường hợp có thể phát sinh ra các đối tượng khó với hình dạng ít thấy và mang ít đặc trưng để nhận dạng.

- Ảnh nền nhập nhằng, khó tách biệt giữa đối tượng và ảnh nền sẽ dẫn tới việc xác định vị trí của đối tượng khó khăn (xem minh họa ở Hình 1.1e). Các ảnh nền có mức độ tương đồng cao với màu sắc, vân ảnh của đối tượng sẽ gây nhiễu đến quá trình rút trích đặc trưng của đối tượng, từ đó có thể gây nhầm lẫn đối tượng là ảnh nền và phát hiện sót. Các đối tượng khó có thể được phát sinh trong điều kiện trên.

- Đối tượng có kích thước nhỏ (được lấy mẫu từ khoảng các xa với máy ảnh hoặc độ phân giải ảnh thấp). Các đặc trưng thường không được rút trích tốt trên các đối tượng có kích thước nhỏ, từ đó dẫn đến bị bỏ sót đối tượng hoặc việc xác định vị trí của đối tượng không chính xác. Đây cũng là một trong những nguyên nhân tạo ra các đối tượng khó (xem minh họa ở Hình 1.1f).

1.3 Các xu hướng nghiên cứu

1.3.1 Kết hợp với thông tin ngữ cảnh

Tập trung vào việc giải quyết các đối tượng khó phát hiện, cách tiếp cận dựa trên ngữ cảnh đã được chú ý. Động lực đầu tiên xuất phát từ mối quan hệ giữa phân vùng ngữ nghĩa ảnh (*semantic segmentation/image segmentation*) và phát hiện đối tượng (*object detection*). Thực tế là các kết quả phân đoạn ảnh chứa đựng nhiều thông tin hữu ích cho việc phát hiện đối tượng chính xác hơn. Lấy ý tưởng từ công trình segDPM [42], segDeepM [43] tăng cường độ chính xác của thuật toán phát hiện đối tượng bằng cách sử dụng một tập hợp các vùng ảnh được phân đoạn chính xác kết hợp với mô hình Markov Random Field. [44] khai thác các đặc trưng ngữ cảnh rút trích từ mạng học sâu FCN cho bài toán phát hiện đối tượng thông qua thao tác kết nối (*concatenation*) với vectơ đặc trưng ngữ nghĩa. Tương tự [44], Zagoruyko và các cộng sự [45] sử dụng kết hợp 4 vùng ngữ cảnh khác nhau trong một kiến trúc huấn luyện liền mạch (*end-to-end*). Hướng

tiếp cận khác trong việc khai thác thông tin ngữ cảnh được đề xuất bởi Wu và các cộng sự [46]. Công trình này trình bày ý tưởng sử dụng mô hình And-Or vào việc biểu diễn thông tin ngữ cảnh và thông tin bị che khuất của đối tượng, cụ thể là xe hơi. Gần đây nhất, công trình [47] đề xuất một phương pháp sử dụng Recurrent Neural Networks (RNNs) như một cơ chế để rút trích đặc trưng ngữ cảnh.

1.3.2 Khai thác các đối tượng khó

Khai thác các đối tượng khó (*hard example mining*) cũng là một hướng khác nhằm tăng cường hiệu quả của các bộ phát hiện đối tượng dựa trên CNN. Có hai hướng tiếp cận chính đó là: khai thác mẫu dương khó (*hard positive example mining*) và khai thác mẫu âm khó (*hard negative example mining*). Khai thác mẫu âm khó được thực hiện thông qua các thuật toán bẫy lỗi (*bootstrapping*). Trong các phương pháp phát hiện đối tượng dựa trên CNN như R-CNN [13] và SPPnet [39], phương pháp bẫy lỗi được thực hiện dựa trên các bộ phân lớp SVM. Tuy nhiên, các bộ phát hiện mới nhất như Fast R-CNN [14] và Faster R-CNN [6] chưa sử dụng cơ chế bẫy lỗi này do các khó khăn trong việc tích hợp với kiến trúc mạng. Công trình [48] đã chỉ ra điểm hạn chế này và đồng thời đề xuất một phương pháp bẫy lỗi dựa trên bộ phân lớp rừng đa tầng (*Cascaded Boosted Forest*). Đối với cách tiếp cận khai thác mẫu dương khó, những nỗ lực đầu tiên đến từ các nghiên cứu cho bài toán học đặc trưng (*feature learning*) và phân lớp ảnh (*image classification*) [49, 50], trong đó quá trình huấn luyện sẽ cố gắng khai thác các mẫu dương (*positive*) và mẫu âm (*negative*) khó nhất. Các bộ huấn luyện (*training batches*) được lựa chọn dựa trên các giá trị của hàm mục tiêu (*loss values*). Điều này giúp cho thuật toán Gradient Descent tập trung vào các dữ liệu có liên quan nhất. Gần đây, công trình [51] đã phát triển ý tưởng này cho bộ phát hiện đối tượng bằng cách thêm một lớp mạng có nhiệm vụ khai thác các đối tượng khó (*hard ROI mining layer*) vào mô hình Fast R-CNN.

1.3.3 Xây dựng dữ liệu nhân tạo

Ảnh hưởng của dữ liệu huấn luyện lên các mô hình học sâu cũng là một chủ đề thu hút được khá nhiều quan tâm gần đây. Trong công trình [52], Ross Girshick và các cộng sự đã thu thập 3.5 tỉ ảnh và 17,000 nhãn (*hashtag*) để huấn luyện và nhận được kết quả rất thành công. Tuy nhiên chi phí cho vấn đề gán nhãn

dữ liệu là không nhỏ nên bên cạnh việc sử dụng dữ liệu được gán nhãn bằng tay các nghiên cứu cũng tập trung vào các phương pháp tăng cường dữ liệu (*data augmentation*). Các kỹ thuật cơ bản bao gồm cắt (*cropping*), lật (*flipping*), và biến động màu (*colour jittering*) được sử dụng khá thường xuyên trong nhiều nghiên cứu (Faster R-CNN [6], YOLO [16], SSD [17]).

Bên cạnh đó, ở một góc độ khác, chất lượng của dữ liệu cũng quan trọng không kém so với số lượng dữ liệu. Điều này có nghĩa là việc tăng cường dữ liệu cần đảm bảo được tính hợp lý và tạo ra các mẫu huấn luyện càng gần thực tế càng tốt. Để hiện thực ý tưởng này, một số công trình nghiên cứu đã tập trung vào việc phát sinh các dữ liệu nhân tạo có độ trung thực cao (tạo ra các đối tượng nhân tạo có hình dáng, vị trí trong ảnh giống như thực tế, với ít các dấu hiệu nhân tạo nhất có thể). Kết xuất đồ họa (*rendering*) từ các mô hình 3D là một hướng tiếp cận khá phổ biến để tạo ra các tập dữ liệu nhân tạo, ví dụ như SYNTHIA [53], SceneNet [54], Virtual KITTI [55], SURREAL [56].

1.4 Các vấn đề nghiên cứu trong luận án

Nội dung của luận án bao gồm các vấn đề nghiên cứu sau đây:

1. **Đề xuất phương pháp phát hiện lại đối tượng khó trên tập đối tượng bỏ sót sử dụng mạng học sâu ở hai giai đoạn.**

   Lấy ý tưởng từ các cách tiếp cận khai thác đối tượng khó (*hard example mining*) - như đề cập ở mục 1.3.2, trong luận án này chúng tôi đề xuất một phương pháp đơn giản nhưng có hiệu quả tốt, được đặt tên là YALA (**Y**ou **A**lways **L**ook **A**gain). Phương pháp này tập trung vào việc khai thác các đối tượng khó mà các thuật thoán phát hiện đối tượng hiện đại dựa trên CNN chưa giải quyết tốt. Chúng tôi đề xuất sử dụng mô hình phát hiện đối tượng hai giai đoạn dựa trên mạng học sâu được tăng cường khả năng hoạt động trên đa tỉ lệ nhằm huấn luyện máy tính có thể phát hiện tốt những đối tượng còn bị bỏ sót.

2. **Đề xuất phương pháp phát sinh dữ liệu nhân tạo nhằm tăng cường hiệu quả của các bộ phát hiện đối tượng trên tập các đối tượng khó.**

   Xuất phát từ ý tưởng phát sinh dữ liệu nhân tạo để tăng cường cho việc huấn luyện bộ phát hiện đối tượng - như đề cập ở mục 1.3.3, chúng tôi

đề xuất một phương pháp phát sinh dữ liệu nhân tạo có định hướng. Giả thiết đặt ra là việc phát sinh dữ liệu nhân tạo cần tập trung vào các đối tượng khó và thông thường ít xuất hiện trong dữ liệu. Chúng tôi đề xuất phát sinh dữ liệu nhân tạo phải được thực hiện kết hợp với quá trình khai thác đối tượng khó trong tập dữ liệu. Phương pháp đề xuất được đặt tên là YADA (**Y**ou **A**lways **D**ream **A**gain).

3. **Luận án cũng áp dụng kết quả của mô hình phát hiện đối tượng vào bài toán liên quan: phát hiện đối tượng chính trong ảnh (*Salient Object Detection*).**

   Chúng tôi trích xuất các mặt nạ phân vùng ảnh chạy trên các khung bao từ mô hình phát hiện đối tượng để tính toán thông tin ngữ nghĩa. Khung bao đối tượng có thể thu được từ YALA/YADA hay bất kỳ phương pháp phát hiện đối tượng tân tiến nào khác. Sau đó chúng tôi đề xuất các ánh xạ tường mình và không tường minh sử dụng thông tin ngữ nghĩa để phát hiện đối tượng chính với độ chính xác cao.

Chương 2
CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu

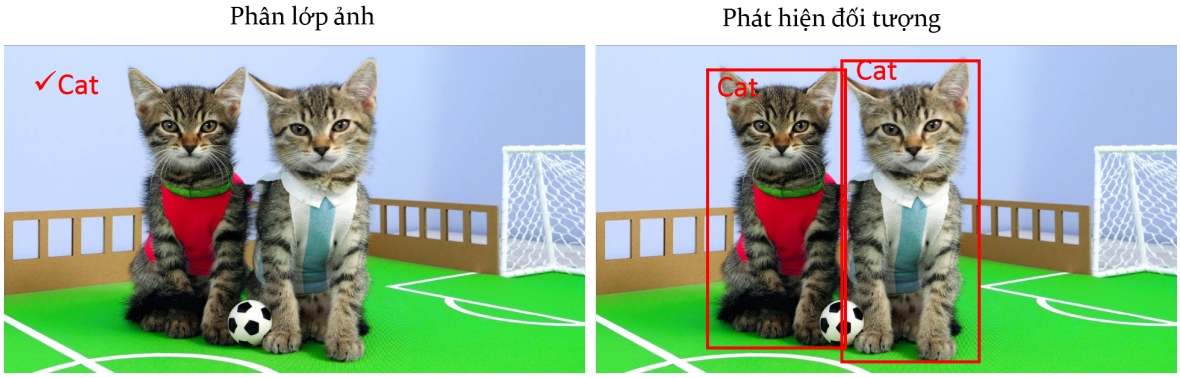
Trong thị giác máy tính, phân lớp đối tượng và phát hiện đối tượng là hai bài toán thu hút được khá nhiều quan tâm. Phân lớp đối tượng tập trung vào việc dự đoán sự có mặt của các đối tượng trong ảnh, trong khi đó phát hiện đối tượng nhằm vào việc xác định vị trí của các đối tượng (xem hình 2.1). So sánh với bài toán phân lớp, phát hiện đối tượng mang tính thách thức cao hơn và cần có các phương pháp phức tạp để giải quyết.

Bài toán phát hiện đối tượng rất hữu ích trong nhiều ứng dụng thực tế, ví dụ như xe tự vận hành, công nghệ robot, và thực tế tăng cường. Trong giai đoạn đầu, nhiều nghiên cứu đã được thực hiện điển hình là phát hiện khuôn mặt [10]. Phương pháp này sử dụng đặc trưng có chi phí thấp Haar và bộ phân lớp đa tầng (*cascade*) để phát hiện khuôn mặt rất hiệu quả. Tiếp đó Dalal và cộng sự đã đề xuất đặc trưng biểu đồ tần suất hướng - HOG (*Histogram of Gradients*) [11] đem lại hiệu quả cao cho bài toán phát hiện người bộ hành. HOG sau đó được sử dụng rộng rãi cho nhiều loại đối tượng khác. Felzenszwalb và cộng sự đã đề xuất mô hình huấn luyện đối tượng dựa trên các bộ phận DPM (*Discriminatively trained Part based Models*) [12] để phát hiện các đối tượng có khả năng biến dạng cao, ví dụ như người bộ hành có nhiều hình dáng khác nhau. Gần đây, các bước tiến trong lĩnh vực học sâu, ví dụ như R-CNN [13], Fast R-CNN [14], Faster R-CNN [6], SSD [17], và YOLO [16], đã cải tiến đáng kể kết quả đạt được trên bài toán phát hiện đối tượng.

2.2 Mô hình bài toán

Bài toán phát hiện đối tượng cơ sở được mô hình hóa dưới dạng ánh xạ sau [70]:

$$f : X \times Y \to \mathbb{R} \tag{2.1}$$



Hình 2.1: Phân biệt phân lớp đối tượng và phát hiện đối tượng.

trong đó X là không gian toàn bộ các ảnh và Y là không gian của các vùng ảnh con (được xác định bằng hình bao chữ nhật). $f(x, y)$ là hàm mục tiêu của việc dự đoán một đối tượng thuộc một lớp cho trước xuất hiện tại vị trí y trong ảnh x.

Trong trường hợp x là một ảnh đơn, ta có thể sử dụng $f(y)$ thay cho $f(x, y)$ mà không bị nhập nhằng. Để dự đoán vị trí của đối tượng, ta cần giải quyết tối ưu sau:

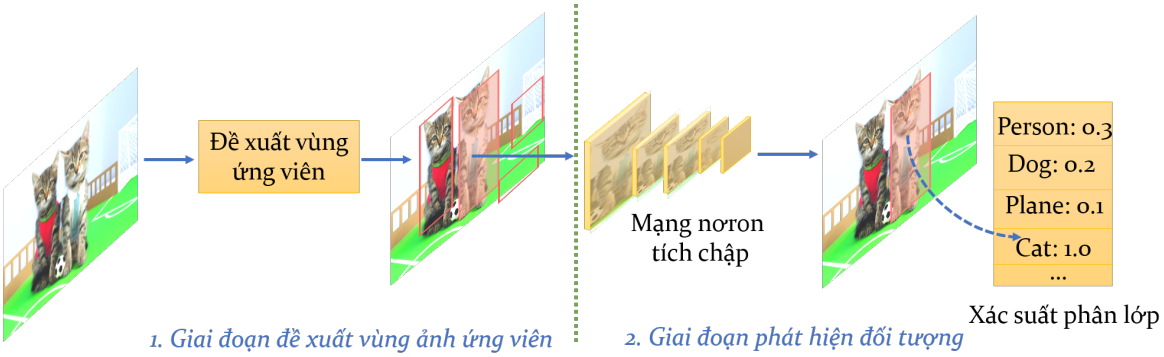
$$y_{opt} = \underset{y \in Y}{\operatorname{argmax}} f(y) \tag{2.2}$$

Do Y là không gian có số lượng phần tử $O(n^2m^2)$ tương ứng với một ảnh kích thước $n \times m$, việc sử dụng chiến lược vét cạn để dò tìm vị trí của đối tượng là không khả thi, trừ trường hợp ảnh có kích thước rất nhỏ.

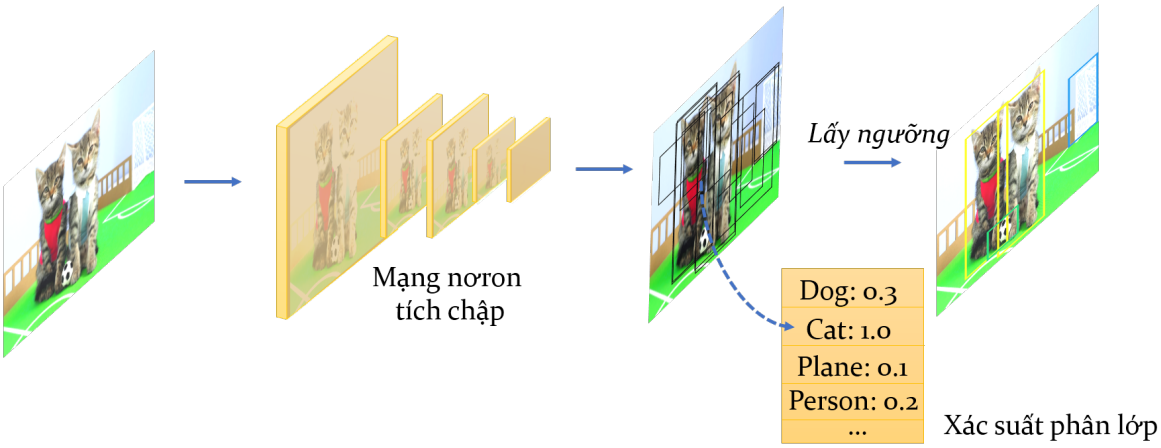
Trong thực tế, việc đánh giá các phát hiện đối tượng không chỉ trên một ảnh duy nhất và có nhiều lớp đối tượng được quan tâm đồng thời. Trong trường hợp này, hàm mục tiêu được biểu diễn tổng quát như sau:

$$(y_{opt}, x_{opt}, \omega_{opt}) = \underset{\substack{y \in Y, \omega \in \Omega \\ x \in \{x_1, \ldots, x_n\}}}{\operatorname{argmax}} f_\omega(x, y) \tag{2.3}$$

trong đó f_ω là hàm mục tiêu tương ứng cho một lớp đối tượng ω trong tập hợp các lớp được quan tâm Ω, và x là một ảnh trong tập huấn luyện $\{x_1, ..., x_N\}$.



(a) Các bộ phát hiện đối tượng hai giai đoạn.



(b) Các bộ phát hiện đối tượng một giai đoạn.

Hình 2.2: Kiến trúc tổng quát của các mô hình phát hiện đối tượng dựa trên mạng học sâu.

2.3 Các hướng tiến cận dựa trên mạng học sâu

Các phương pháp phát hiện đối tượng sử dụng mạng học sâu được chia thành hai nhóm chính:

- Các bộ phát hiện hai giai đoạn (*Two-stage Detectors*): Công trình tiên phong trong nhóm các bộ phát hiện này là R-CNN [13]. R-CNN đề xuất phương pháp phát hiện gồm hai bước, bước đầu tiên sử dụng một thuật toán lựa chọn vùng (*Selective Search* [38]) để phát sinh ra một tập các vùng ảnh tiềm năng có thể chứa đối tượng, bước thứ hai sử dụng các bộ phân lớp (*linear SVM*) dựa trên đặc trưng rút trích từ mạng học sâu để phân loại vùng ảnh là đối tượng/ảnh nền. R-CNN có độ chính xác vượt trội một khoảng đáng kể so với các bộ phát hiện đối tượng trước đó và mở ra bước

phát triển mới cho bài toán. R-CNN được tiếp tục cải tiến qua nhiều năm, về cả tốc độ [39, 14] và phương pháp phát sinh vùng tiềm năng [71, 6]. RPN (*Region Proposal Network*) được đề xuất nhằm tích hợp giai đoạn phát sinh vùng tiềm năng và phân lớp đối tượng vào một kiến trúc mạng duy nhất, được đặt tên là Faster R-CNN [6]. Nhiều phương pháp mở rộng tiếp tục được đề xuất trên kiến trúc mạng này ví dụ như [15, 32, 51, 72].

- Các bộ phát hiện một giai đoạn (*One-stage Detectors*): OverFeat [73] là một trong những bộ phát hiện đối tượng hiện đại dựa trên mạng học sâu theo cơ chế một giai đoạn. Gần đây, SSD [17], YOLO [16, 9, 37], và RetinaNet [5] được đề xuất và thu hút được rất nhiều quan tâm. Đặc điểm của các bộ phát hiện này là chúng được thiết kế để có tốc độ phát hiện đối tượng rất nhanh (ví dụ như 45 fps đối với YOLO), tuy nhiên độ chính xác thường không theo kịp với các kiến trúc mạng hai giai đoạn. Gần đây, RetinaNet [5] là bộ phát hiện đối tượng một giai đoạn được đề xuất với tốc độ tương đương với các bộ phát hiện một giai đoạn cùng loại nhưng độ chính xác đạt được được cải tiến đáng kể.

2.3.1 Kiến trúc tổng quát của các bộ phát hiện một giai đoạn và hai giai đoạn

Kiến trúc tổng quát của các bộ phát hiện hai giai đoạn được trình bày như Hình 2.2(a). Về bản chất, cả hai giai đoạn (xác định vùng ứng viên và phân lớp) đều phải khai thác các đặc trưng của đối tượng, trong đó giai đoạn xác định vùng ứng viên có thể xem là một bước dò tìm sơ lược, và kết quả được tinh chỉnh ở bước phân lớp. Các đặc trưng được sử dụng trong bước dò tìm vùng ứng viên thường có chi phí tính toán thấp để dò tìm tại nhiều vị trí khác nhau trong ảnh. Sau đó các vùng ứng viên sẽ được đưa vào các mạng học sâu có kiến trúc phức tạp để xác định chính xác nhãn của đối tượng (và loại bỏ các phát hiện sai).

Kiến trúc tổng quát của các bộ phát hiện một giai đoạn được minh họa như Hình 2.2(b). Đặc điểm khác biệt chính của các bộ phát hiện một giai đoạn chính là kiến trúc mạng của chúng có khả năng dự đoán đồng thời tất cả các vị trí đối tượng trong ảnh kèm theo xác suất lớp đối tượng đồng thời. Điều này khác với cơ chế phát hiện tuần tự của các bộ phát hiện hai giai đoạn, trong đó lần lượt từng vùng ảnh tiềm năng sẽ được đưa vào mạng nơron sau đó để rút trích đặc trưng và phân lớp. Sự khác biệt này cho phép các bộ phát hiện dựa trên một giai đoạn có tốc độ vượt trội so với các bộ phát hiện đối tượng một giai đoạn.

2.3.2 Các bộ phát hiện đối tượng hai giai đoạn

R-CNN là một bộ phát hiện sơ khai với cách giải quyết khá đơn giản là chỉ sử dụng mạng học sâu để rút trích đặc trưng của đối tượng. Việc dò tìm các vùng đối tượng ban đầu cũng như phân lớp đối tượng đều sử dụng các phương pháp trước đó, cụ thể là Selective Search và SVM. SPPnet đã được đề xuất để khắc phục yếu điểm này. Sự khác biệt cơ bản giữa SPPnet và R-CNN chính là SPPnet tính toán một bản đồ đặc trưng dùng chung cho tất cả các vùng đối tượng. Tuy nhiên, các vùng đối tượng có kích thước khác nhau, nên việc truyền trực tiếp các vùng tương ứng trên bản đồ đặc trưng vào các lớp mạng sau đó (các lớp kết nối đầy đủ) là không thể, vì các lớp kết nối đầy đủ có số lượng nơron cố định. Để giải quyết điều này SPP đề xuất sử dụng *"Spatial Pyramid Pooling"*. Đây là một cơ chế chia vùng đặc trưng thành lưới có kích thước cố định và lấy giá trị cao nhất trong mỗi ô (*max-pooling*), sau đó kết hợp các giá trị này lại để tạo thành vectơ biểu diễu có số chiều cố định. SPPnet kết hợp nhiều lưới có kích thước khác nhau để biểu diễn đối tượng bền vững hơn. Các vectơ biểu diễn đối tượng được tính toán dựa trên tháp không gian *"Spatial Pyramid"* sẽ được đưa vào SVM để phân lớp tương tự như R-CNN. Fast R-CNN tiếp tục cải tiến yếu điểm của SPPnet, trong đó đề xuất phương pháp huấn luyện có thể cập nhật trọng số cho các lớp mạng phía trước bản đồ đặc trưng, đồng thời sử dụng các lớp kết nối đầy đủ để thay thế cho SVM và thuật toán hồi qui vùng phát hiện. Hàm mất mát đa tác được đề xuất để thống nhất các giai đoạn huấn luyện cho bộ phát hiện. Tuy nhiên, tương tự SPPnet và R-CNN, Fast R-CNN cũng cần sử dụng bộ dò tìm vùng ứng viên độc lập và có chi phí tính toán cao (Selective Search). Từ đó Faster R-CNN đề xuất một kiến trúc mạng con (RPN) có thể dò tìm đối tượng trực tiếp trên bản đồ đặc trưng. RPN sử dụng các khung neo mặc định (*anchor box*) trên bản đồ đặc trưng, từ đó dự đoán khả năng xuất hiện đối tượng trong các khung neo này cũng như hồi qui vị trí chính xác của đối tượng (tương đối với khung neo). RPN được tích hợp vào một kiến trúc thống nhất với kiến trúc mạng chính làm nhiệm vụ phân lớp sau đó (giống Fast R-CNN). Việc đề xuất RPN thay cho Selective Search giúp bộ phát hiện đối tượng hoạt động nhanh hơn gấp 10 lần, đồng thời độ chính xác cũng được tăng cường đáng kể.

2.3.3 Các bộ phát hiện đối tượng một giai đoạn

YOLO là bộ phát hiện đối tượng một giai đoạn được đề xuất với tốc độ vượt bậc so với các bộ phát hiện đối tượng hai giai đoạn (như họ R-CNN). Cách tiếp

cận của YOLO là chia lưới bản đồ đặc trưng với kích thước cố định và sử dụng các lớp kết nối đầy đủ để dự đoán đồng thời khung bao đối tượng kèm theo giá trị xác suất xuất hiện đối tượng tại tất cả các vị trí trong ảnh (được tính toán tương ứng với các vị trí của ô trong lưới đã chia). YOLO đạt được hai ưu điểm chính để cải tiến về tốc độ, đó là chia sẻ chi phí tính toán bản đồ đặc trưng, và dự đoán đồng thời vị trí của đối tượng kèm nhãn lớp đối tượng (tránh được việc phân lớp vùng đối tượng theo kiểu tuần tự đưa vào mạng CNN như đối với các bộ phát hiện hai giai đoạn). Tuy nhiên về mặt độ chính xác, YOLO có hiệu quả kém hơn so với Fast/Faster R-CNN, đặc biệt đối với các đối tượng có kích thước nhỏ. SSD khắc phục khuyết điểm của YOLO bằng cách sử dụng các khung bao đối tượng mặc định với nhiều tỉ lệ khác nhau, và dự đoán đồng thời trên nhiều bản đồ đặc trưng với kích thước khác nhau. Các bộ dự đoán cũng được xây dựng dưa trên các lớp tích chập (*convolutional layer*) thay cho các lớp kết nối đầy đủ (*fully connected layer*) để có thể hoạt động trên các kích thước khung bao và bản đồ đặc trưng khác nhau. Tuy nhiên, SSD vẫn tồn tại các yếu điểm chính, thứ nhất là các tầng bản đồ đặc trưng đầu tiên có kích thước lớn tuy nhiên lại thiếu khả năng biểu diễn các đặc trưng cấp cao (ngữ nghĩa) của đối tượng, từ đó làm giả khả năng phát hiện các đối tượng nhỏ trên các tầng bản đồ đặc trưng này. Thứ hai là sự mất cân bằng dữ liệu trong quá trình huấn luyện (thường xảy ra ở các bộ phát hiện đối tượng một giai đoạn), tỉ lệ đối tượng/ảnh nền có thể lên đến 1/1,000. RetinaNet được đề xuất nhằm khắc phục hai yếu điểm trên, đối với việc tính toán bản đồ đặc trưng, RetinaNet sử dụng FPN (*Feature Pyramid Network*) để tái xây dựng bản đồ đặc trưng bằng việc bổ sung đặc trưng ngữ nghĩa cao hơn ở các tầng phía sau cho tầng phía trước. Đối với vấn đề mất cân bằng dữ liệu, RetinaNet đề xuất một hàm mất mát nhằm cân bằng sự ảnh hưởng của số lượng mẫu các lớp và quá trình huấn luyện, được gọi là *Focal loss*. RetinaNet có thể hoạt động với tốc độ 5 fps và đạt độ chính xác cao hơn so với tất cả các bộ phát hiện đối tượng một giai đoạn và hai giai đoạn trước đó.

Chương 3
CÁC ĐỀ XUẤT CHO VIỆC PHÁT HIỆN ĐỐI TƯỢNG KHÓ

3.1 Phương pháp YALA

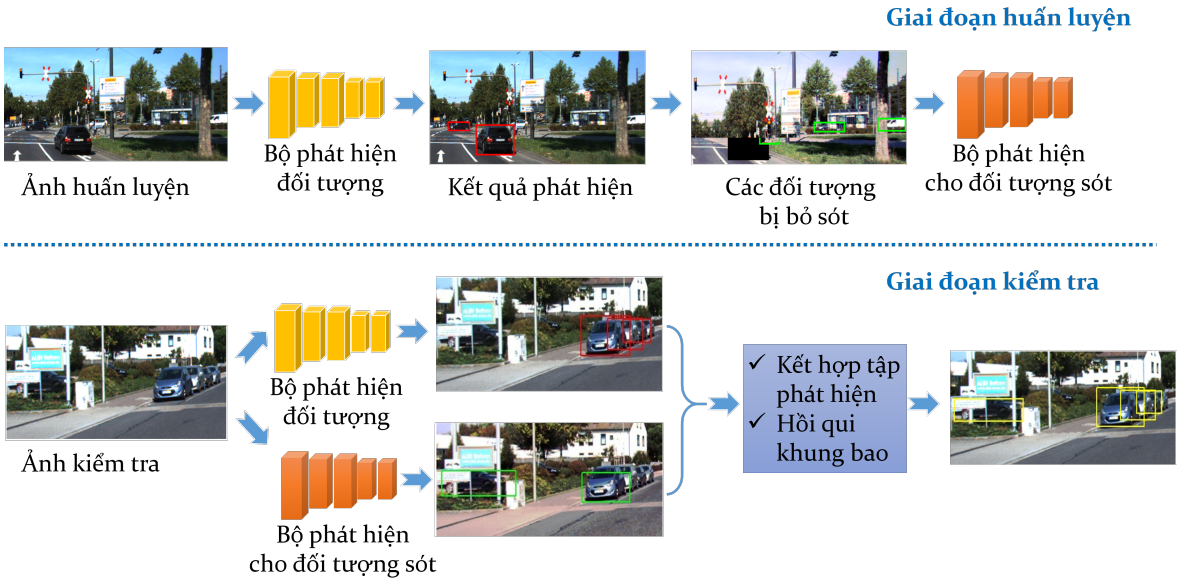
Trong mục này, chúng tôi sẽ trình bày chi tiết phương pháp được đề xuất - YALA. Hình 3.1 minh họa kiến trúc tổng quan của phương pháp này.

3.1.1 Động lực

Như đã trình bày ở chương đầu, chúng tôi nhận thấy có một số lượng đáng kể các đối tượng khó xuất hiện trong ảnh. Các thuật toán hiện đại vẫn thất bại trong việc phát hiện các đối tượng khó này. Những đối tượng này khó phát hiện do nhiều nguyên nhân, như mức độ che khuất lớn, có hình dáng biến đổi lớn, hoặc bị nhập nhằng với ảnh nền. Các đối tượng này thường chiếm tỉ lệ nhỏ hơn so với các đối tượng được phân lớp dễ dàng trong tập huấn luyện. Vì vậy, như đã được chỉ ra trong RetinaNet [5], trong quá trình huấn luyện, các bộ phát hiện đối tượng có xu hướng học lệch về các mẫu dễ (chiếm đa số, và gây ảnh hưởng lớn đến hàm mất mát do tính chất cộng dồn các giá trị mất mát với trọng số như nhau). Để khắc phục vấn đề này, giải pháp được đề ra trong luận án là khai thác các đối tượng khó và huấn luyện một kiến trúc mạng để phát hiện độc lập cho các đối tượng này, loại bỏ ảnh hưởng của yếu tố mất cân bằng dữ liệu.

3.1.2 Phát hiện trên tập đối tượng bị bỏ sót

Như đã trình bày ở phần trước, quan sát chính của chúng tôi đó là có một số lượng đối tượng khó bị bỏ sót bởi các phương pháp hiện đại. Do đó, chúng tôi trước hết huấn luyện một bộ phát hiện Faster R-CNN [6] cho tầng thứ nhất, sau đó chúng tôi áp dụng bộ phát hiện này trên dữ liệu ảnh huấn luyện để khai thác các đối tượng bị bỏ sót dựa vào dữ liệu gán nhãn đã được cung cấp. Tiếp đó, chúng tôi huấn luyện một bộ phát hiện Faster R-CNN khác trên tập các



Hình 3.1: Phương pháp YALA được minh họa bằng ví dụ thực tế. Trong giai đoạn huấn luyện, bộ phát hiện đối tượng khó sẽ được huấn luyện trên tập các đối tượng bị bỏ sót bởi bộ phát hiện đối tượng thông thường. Trong giai đoạn phát hiện, ảnh đầu vào sẽ được đưa vào cả hai bộ phát hiện đối tượng, các kết quả phát hiện sẽ được kết hợp và tinh chỉnh với thuật toán hồi qui khung bao đối tượng.

đối tượng bị sót này để tập trung vào việc phát hiện các đối tượng thách thức. Thuật toán cân bằng biểu đồ tần suất màu (*histogram equalization*) sẽ được áp dụng trên từng kênh của ảnh màu RGB nhằm hỗ trợ cho bộ phát hiện đối tượng thứ hai.

Cụ thể hơn, quá trình phát hiện trên tập đối tượng bị bỏ sót như sau. Trước tiên, một bộ phát hiện Faster R-CNN được huấn luyện dựa trên tập dữ liệu huấn luyện sử dụng kiến trúc mạng VGG [77] với trọng số khởi tạo từ mô hình đã được huấn luyện trên tập ImageNet. Sau đó, bộ phát hiện này được áp dụng để phát hiện các đối tượng trong tập huấn luyện. Để tính toán các đối tượng bị bỏ sót, chúng tôi sử dụng dữ liệu gán nhãn được cung cấp kèm theo để xác định các phát hiện chính xác, và loại bỏ chúng ra khỏi tập huấn luyện bằng thao tác che mặt nạ (*masking*) tất cả các điểm ảnh trong vùng phát hiện được. Cụ thể, chúng tôi thiết lập giá trị của mỗi điểm ảnh (x, y) trong mặt nạ M như sau đây:

$$M(x,y) = \begin{cases} 0, & \text{nếu } (x,y) \in \mathrm{B}_1. \\ 1, & \text{ngược lại.} \end{cases} \tag{3.1}$$

, trong đó B_1 là tập khung bao đối tượng được phát hiện từ bộ phát hiện thứ nhất. Sau đó chúng tôi thực hiện thao tác nhân từng phần tử (*element-wise multiplication*) giữa mỗi kênh màu của ảnh huấn luyện và mặt nạ M tương ứng của nó nhằm tạo ra tập dữ liệu đối tượng còn lại. Cùng với việc loại bỏ các kết quả phát hiện ở giai đoạn thứ nhất, chúng tôi đồng thời làm nổi bật các vùng ảnh cho tập đối tượng còn lại. Cụ thể, chúng tôi áp dụng kỹ thuật cân bằng biểu đồ tần suất màu cho các ảnh nhằm làm cho các đối tượng nổi bật hơn. Chi tiết của thao tác này được trình bày như trong công thức bên dưới:

$$C_{eq_i} = (L-1) * \tau(C_i), \tag{3.2}$$

trong đó $\tau(C_i) = \sum_{k=0}^{N}(C_k \leqslant C_i)/N$, C biểu diễn một kênh màu (R/G/B), C_{eq} là kênh màu sau khi áp dụng cân bằng biểu đồ tần suất màu, C_i biểu diễn điểm ảnh i trong ảnh, L là số giá trị màu và được cố định là 256, và N là tổng số điểm ảnh có trong ảnh.

Tiếp đó, bộ phát hiện Faster R-CNN thứ hai được huấn luyện để phát hiện ra các đối tượng bị bỏ sót từ bộ phát hiện Faster R-CNN đầu tiên. Nói cách khác, bộ phát hiện này nhắm tới việc phát hiện các đối tượng có trong các vùng ảnh khác thay vì các vùng ảnh đã được phát hiện. Đối với quá trình kiểm tra, ảnh đầu vào sẽ được đưa vào bộ phát hiện Faster R-CNN thứ nhất để phát hiện ra các đối tượng thông thường. Sau đó, bộ phát hiện Faster R-CNN được huấn luyện tập trung cho các đối tượng bị bỏ sót sẽ được sử dụng để phát hiện ra các đối tượng chưa được khám phá trước đó. Thực tế, các đối tượng đã được phát hiện từ bộ phát hiện Faster R-CNN đầu tiên có thể được xóa khỏi ảnh bằng kỹ thuật che mặt nạ như đã trình bày phía trên. Tuy nhiên thao tác này có thể làm giảm độ phủ bởi vì nó có khả năng loại bỏ luôn một số lượng đối tượng bị che khuất bởi các phát hiện trước đó. Kết quả đầu ra của bộ phát hiện Faster R-CNN thứ hai là một tập khung bao đối tượng khác được biểu diễn bởi B_2. Tiếp theo, kết quả phát hiện cuối cùng được tạo ra bằng việc kết hợp các kết quả phát hiện của cả hai bộ phát hiện. Chi tiết hơn, chúng tôi kết hợp các tập khung bao B_1 và B_2 như sau đây:

$$B = \{b_i \in B_1, b_j \in B_2 : \max_{i=1:n} \frac{area(b_j \cap b_i)}{area(b_j \cup b_i)} \leq \zeta\} \tag{3.3}$$

trong đó b_i là một khung bao trong B_1, b_j là một khung bao trong B_2, n là tổng số khung bao của B_1, và ζ là ngưỡng sử dụng của thuật toán *non-maximal-*

suppression, được cố định là 0.7. $\cap$ biểu diễn phép lấy giao của hai tập khung bao, và $\cup$ biểu diễn phép lấy hợp của hai tập khung bao.

Tất cả các vùng phát hiện sau đó sẽ được sắp xếp theo giá trị xác suất phân lớp (*classification probabilities*). Giá trị điều chỉnh (*penalty*) sẽ được thêm vào cho các vùng được phát hiện bởi bộ Faster R-CNN thứ hai để ước lượng khả năng không chắc chắn của các phát hiện này so với các phát hiện của bộ phát hiện Faster R-CNN thứ nhất. Chi tiết trình bày như công thức dưới đây:

$$\theta(b_k) = \begin{cases} \theta(b_k), & \text{nếu } b_k \in \mathrm{B}_1. \\ \theta(b_k) - \delta, & \text{nếu } b_k \in \mathrm{B}_2. \end{cases} \tag{3.4}$$

trong đó b_k là một khung bao trong tập kết quả phát hiện cuối cùng B, $\theta(b_k)$ là giá trị xác suất phân lớp của b_k, và δ là giá trị điều chỉnh, được cố định bằng 0.7. Chú ý là tham số này có giá trị mặc định và chưa được lựa chọn cho từng tập dữ liệu cụ thể.

3.1.3 Phát hiện dựa trên các kích thước mặc định và tinh chỉnh khung bao đối tượng bằng thuật toán hồi qui

Với YALA chúng tôi kế thừa cơ chế phát hiện dựa trên các "neo" mặc định của bộ phát hiện Faster R-CNN, và đề xuất 2 vấn đề:

- Tăng cường đối với giai đoạn kiểm tra. Trong đó, chúng tôi sử dụng nhiều kích thước ảnh khác nhau. Kết quả phát hiện tương ứng với các kích thước ảnh khác nhau sẽ được kết hợp và các phát hiện trùng sẽ được loại bỏ sử dụng thuật toán *non-maximum suppression*.

- Tăng cường thuật toán hồi qui của Faster R-CNN để hiệu quả hơn cho các đối tượng khó. Theo đó, vùng ảnh đối tượng sẽ được cắt và thay đổi vào kích thước đầu vào của mạng, sau đó đặc trưng ở tầng fc_7 sẽ được rút trích để huấn luyện thuật toán hồi qui.

Thay vì sử dụng thuật toán hồi qui được cung cấp bởi RPN, chúng tôi đề xuất sử dụng một thuật toán khác để tinh chỉnh lại kết quả. Cụ thể, chúng tôi rút trích đặc trưng học sâu từ tầng kế cuối của mô hình CNN cho các vùng đối tượng phát hiện được, và sau đó sử dụng các đặc trưng này cho mô hình hồi qui.

Chúng tôi xây dựng mô hình hồi qui vùng đối tượng tương ứng với phép biến đổi của bốn tọa độ (x, y, w, h) tương tự như [13]. Lưu ý rằng chúng tôi sử dụng đặc trưng được rút trích từ tầng fc_7 thay vì tầng $pool_5$ của mô hình CNN. Chi tiết hơn, chúng tôi ứng dụng mô hình hồi qui Ridge như dưới đây:

$$\beta_{ridge} = \arg\min_{\beta \in R^p} \sum_{i=1}^{n} (t_i - \beta^T \phi_{fc_7}(x_i))^2 + \lambda \sum_{j=1}^{p} \beta_j^2, \tag{3.5}$$

trong đó $\phi_{fc_7}(x_i)$ biểu diễn đặc trưng rút trích từ tầng fc_7 của vùng đối tượng x_i, n là số lượng vùng đối tượng huấn luyện, p là số chiều của đặc trưng được sử dụng (p = 4096 đối với đặc trưng học sâu rút trích từ tầng fc_7), $\lambda \geqslant 0$ là tham số kiểm soát mức độ ảnh hưởng của các hệ số học có giá trị lớn (*penalty term*), β là các hệ số học, và t_i là mục tiêu hồi qui tương ứng với mỗi bộ huấn luyện (x_i, y_i), được định nghĩa theo như [13].

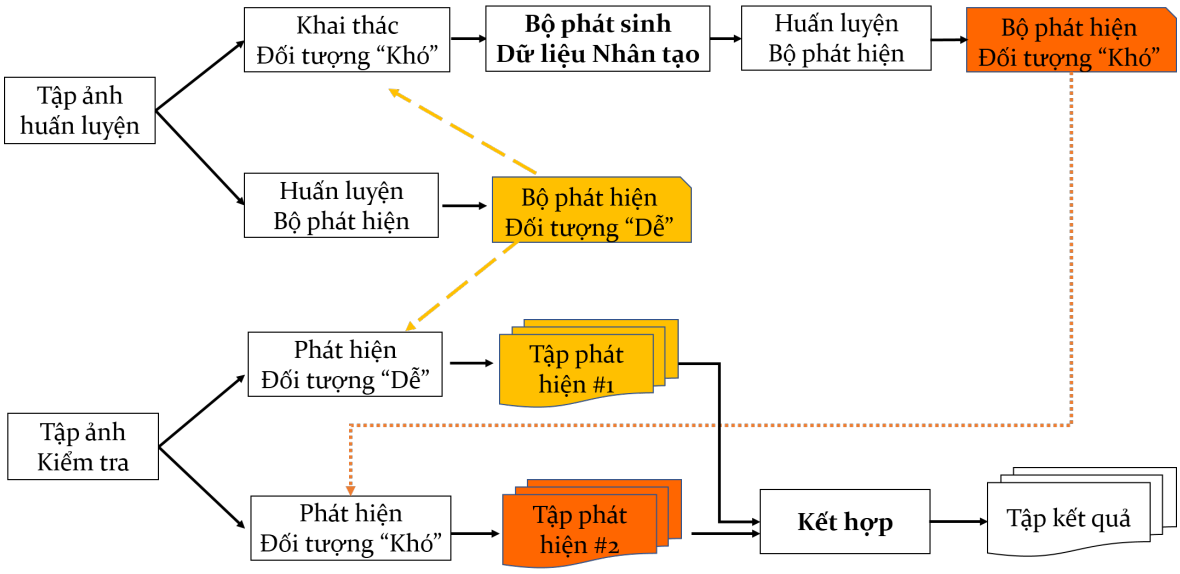
Lưu ý thêm rằng, điểm khác biệt của phương pháp đề xuất so với Faster R-CNN là không sử bản đồ đặc trưng (dùng chung) đã được tính toán trước đó, thay vào đó sử dụng phương pháp cắt ảnh và đưa vào mô hình mạng để rút trích đặc trưng. Các công trình công bố gần đây (điển hình như SSD [17] hoặc Feature Pyramid Network [72]) đã chứng minh nếu sử dụng các tầng phía trước của mạng thì độ phân giải sẽ cao nhưng mức độ biểu đạt ngữ nghĩa thấp, và **việc phát hiện đối tượng khó trên các tầng này sẽ không đạt hiệu quả tốt nhất**. Do đó giải pháp được đề xuất trong các phương pháp nêu trên là kết hợp thông tin ở nhiều tầng bản đồ đặc trưng khác nhau. Trong luận án, chúng tôi đề xuất giải pháp cắt vùng ảnh đối tượng và thay đổi về kích thước đầu vào của mạng nhằm nâng cao độ phân giải đầu vào của đối tượng. Sau đó sẽ rút trích ở tầng fc_7 nhằm tăng mức độ biểu đạt ngữ nghĩa.

3.2 Phương pháp YADA

Trong phần này, chúng tôi sẽ trình bày chi tiết về phương pháp đề xuất - YADA. Hình 3.2 minh họa tổng quan các giai đoạn thực hiện của YADA.

3.2.1 Động lực

Phương pháp đề xuất YADA được phát triển trên nền tảng phương pháp đã đề xuất trước đó (YALA). Trong phương pháp trước, vấn đề mất cân bằng dữ



Hình 3.2: Sơ đồ minh họa các bước của phương pháp đề xuất YADA.

liệu trong quá trình huấn luyện các đối tượng khó (chiếm thiểu số) được đặt ra, từ đó được giải quyết bằng phương pháp khai thác các đối tượng khó và huấn luyện một mạng học sâu độc lập cho tập các đối tượng này. Trong phương pháp này, chúng tôi đề xuất một hướng giải quyết khác dựa trên kỹ thuật tăng cường dữ liệu nhằm giúp dữ liệu được cân bằng hơn. Cụ thể, các đối tượng khó sẽ được khai thác từ tập huấn luyện thông qua một bộ phát hiện cơ sở (tương tự như YALA). Sau đó chúng tôi áp dụng một qui trình phát sinh tăng cường dữ liệu cho các đối tượng này dựa trên việc tổng hợp dữ liệu nhân tạo. Qui trình này chúng tôi gọi là *"lucid dreaming"*, nghĩa là việc phát sinh dữ liệu được định hướng, theo đó các dữ liệu mong muốn sẽ được phát sinh thêm để tăng cường cho bộ phát hiện đối tượng. Kết quả phát hiện đối tượng từ bộ phát hiện được huấn luyện trên tập dữ liệu được tăng cường bằng dữ liệu nhân tạo có thể bổ sung rất hiệu quả cho bộ phát hiện được huấn luyện trên dữ liệu thông thường.

3.2.2 Phát sinh dữ liệu nhân tạo

Tìm kiếm cảnh tương tự

Trong ngữ cảnh này, chúng tôi định nghĩa các đối tượng *"dễ"* và *"khó"* tương ứng là các đối tượng có thể phát hiện được và các đối tượng bị bỏ sót bởi một bộ phát hiện đối tượng tân tiến hiện nay, như Faster R-CNN. Bước đầu tiên để phát sinh dữ liệu nhân tạo về các đối tượng khó là tạo ra một tập các ảnh

tương tự. Mục đích là đảm bảo dữ liệu phát sinh nhân tạo có ngữ cảnh trung thực như dữ liệu thực tế. Cho trước tập dữ liệu ảnh huấn luyện T, với mỗi ảnh I_q trong T chúng tôi sử dụng đặc trưng được rút trích từ tầng fc_7 của kiến trúc mạng VGG (đã được huấn luyện trên tập dữ liệu ImageNet) để tìm kiếm k ảnh tương tự nhất từ tập ảnh huấn luyện, tạo thành tập các ảnh tương tự.

Giả sử V_q và V_i^P là các vectơ đặc trưng tương ứng của ảnh truy vấn I_q và một ảnh I_i từ tập ảnh tương tự. Khi đó, chúng tôi định nghĩa mức độ tương đồng giữa ảnh I_q và I_i dựa trên khoảng cách Euclid giữa hai vectơ đặc trưng tương ứng: $s_i = \|V_q - V_i^p\|$. Khoảng cách này càng nhỏ thì mức độ tương đồng giữa hai ảnh càng cao. Mỗi ảnh I_i trong tập ảnh tương tự sẽ được xếp hạng theo độ tương tự giảm dần.

Bộ phát sinh dữ liệu nhân tạo cho các đối tượng *"khó"*

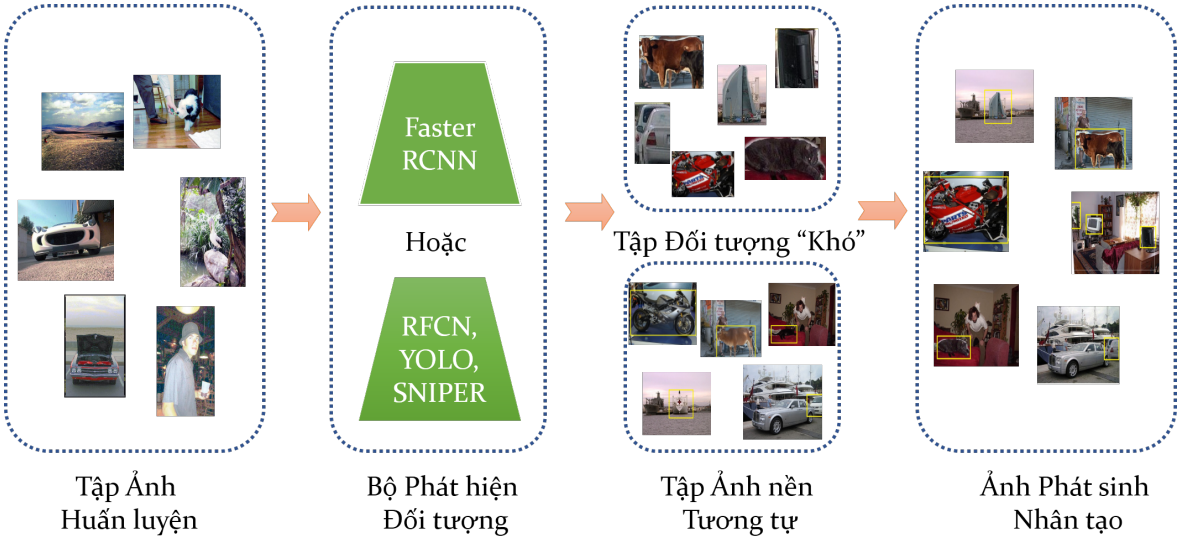
Để phát sinh dữ liệu nhân tạo, chúng tôi sử dụng kỹ thuật thay thế các đối tượng *"dễ"* bằng các đối tượng *"khó"* trong ảnh để đảm bảo các ảnh phát sinh nhân tạo sẽ có ngữ cảnh tương tự như ảnh thực tế. Minh họa trực quan cho quá trình phát sinh dữ liệu được trình bày trong Hình 3.3. Cụ thể với mỗi ảnh I trong tập dữ liệu huấn luyện, chúng tôi tìm kiếm tất cả các vị trí có thể sử dụng - đó là các khung bao của đối tượng đã được phát hiện dựa trên bộ phát hiện đối tượng (Faster R-CNN) được huấn luyện trước đó. Tiếp theo, chúng tôi tạo một tập các đối tượng *"khó"* được thu thập từ k ảnh tương tự nhất với ảnh I. Các đối tượng khó được xác định dựa trên việc sử dụng bộ phát hiện đối tượng đã được huấn luyện trước đó để dò tìm các đối tượng chưa phát hiện được. Độ tương tự của hai ảnh được xác định bằng khoảng cách Euclid giữa hai vectơ đặc trưng tương ứng, như trình bày ở mục 3.2.2. Khi đó, mỗi vị trí xuất hiện của đối tượng trong ảnh I sẽ được gán với một đối tượng phù hợp trong tập các đối tượng khó này. Đối tượng được xác định có phù hợp để gán vào một vị trí trong ảnh dựa vào các đặc điểm tương tự như kích thước chiều rộng (*width*), chiều cao (*height*), và tỉ lệ giữa các chiều (*aspect ratio*). Thuật toán 1 thể hiện chi tiết các bước của phương pháp phát sinh dữ liệu nhân tạo cho các đối tượng *"khó"*.

Sau khi phát sinh các ảnh chứa tập đối tượng *"khó"*, thao tác cân bằng biểu đồ màu (*histogram equalization*) sẽ được áp dụng để làm nổi bật hơn các vùng đối tượng trong ảnh. Chi tiết thao tác này được thể hiện theo như công thức 3.2.

Để hạn chế các dấu vết nhân tạo ở đường biên của đối tượng có thể làm sai lệch các đặc trưng thật của đối tượng và ảnh hưởng trực tiếp tới quá trình

Algorithm 1 "Hard" Object Lucid Data Synthesizer

procedure SYNTHESIZER(T) ▷ T: Training image set
  $T_{syn} \leftarrow \emptyset$ ▷ T_{syn}: Synthesized image set
  for an image I in T **do**
    $L \leftarrow$ All available locations in I ▷ locations of well detected objects
    $S \leftarrow$ Set of k similar images ▷ using $L2$ distance of CNN features
    $O \leftarrow$ Hard objects in S
    for a location L_i in L **do**
      $D \leftarrow$ Similarity distances between L_i and objects in O ▷ using object class, size, and ratio
      $O_{closest} \leftarrow$ Object in O which has smallest distance
      if $D_{O_{closest}} > threshold$ **then**
        $I_{syn} \leftarrow$ Paste $O_{closest}$ into I at L_i location
      end if
    end for
    $T_{syn} \leftarrow T_{syn} \cup I_{syn}$
  end for
end procedure



Hình 3.3: Minh họa quá trình phát sinh các ảnh nhân tạo cho các đối tượng "khó".

huấn luyện, chúng tôi sử dụng kỹ thuật trộn màu Gaussian (*Gaussian blending*) để làm mờ các cạnh của đường biên đối tượng. Đồng thời, phương pháp Mask R-CNN [15] cũng được sử dụng để tách chính xác đối tượng ra khỏi vùng ảnh nền của ảnh gốc trước khi dán các đối tượng này vào các ảnh tương tự

3.2.3 Kết hợp vùng phát hiện đối tượng

Giả sử B_1 và B_2 lần lượt là tập các vùng đối tượng đã được phát hiện bởi bộ phát hiện đối tượng thứ nhất (các đối tượng thông thường) và thứ hai (các đối tượng khó), kỹ thuật kết hợp hai tập kết quả này được trình bày chi tiết như dưới đây.

Loại bỏ các vùng phát hiện trùng lắp

Trước tiên, các vùng phát hiện trùng lắp cần được xử lý. Các vùng trùng lắp được xác định theo công thức sau:

$$D = \{(b_i, b_j) : \frac{area(b_j \cap b_i)}{area(b_j \cup b_i)} > \zeta\}, \tag{3.6}$$

trong đó b_i là một khung bao đối tượng trong B_1, b_j là một khung bao đối tượng trong B_2, và ζ là ngưỡng trùng lắp tối đa được thiết lập cố định 0.7. Đồng thời, $\cap$ biểu diễn phép lấy giao giữa hai khung bao, và $\cup$ biểu diễn phép lấy hợp hai khung bao.

Chúng tôi loại bỏ các khung bao trùng lắp khỏi tập B_2 thay vì tập B_1 bởi vì các kết quả phát hiện từ bộ phát hiện đối tượng khó thông thường có độ tin cậy thấp hơn so với kết quả từ bộ phát hiện đối tượng thông thường.

Tái điều chỉnh độ tin cậy

Sau khi loại bỏ các vùng phát hiện trùng lắp, thao tác tái điều chỉnh độ tin cậy sẽ được áp dụng cho các vùng phát hiện đối tượng của bộ phát hiện đối tượng khó. Thao tác này nhằm giải quyết hai vấn đề. Thứ nhất, chúng tôi muốn kiểm soát mức độ không chắc chắn của các vùng phát hiện này. Các vùng phát hiện này tập trung vào các đối tượng khó (không thể phát hiện được với bộ phát hiện đối tượng thông thường), do vậy giá trị độ tin cậy của các vùng phát hiện cần được điều chỉnh để không cao hơn so với các vùng phát hiện của các đối tượng dễ hơn (được phát hiện hiệu quả bởi bộ phát hiện đối tượng thông thường). Thứ hai, việc tái điều chỉnh giá trị độ tin cậy của các vùng phát hiện này vào trong một khoảng hợp lý sẽ bổ trợ hiệu quả cho các phát hiện từ bộ phát hiện đối tượng thông thường.

Để thực hiện, với mỗi lớp đối tượng C chúng tôi tính toán giá trị độ tin cậy trung bình μ_C của các vùng phát hiện đối tượng của bộ phát hiện đối tượng thứ

nhất (đối tượng dễ). Sau đó chúng tôi tái điều chỉnh giá trị độ tin cậy của các vùng phát hiện đối tượng của bộ phát hiện đối tượng thứ hai (đối tượng khó) theo công thức dưới đây:

$$S^{Normalized} = S^{Original} \times \frac{1}{\mu'_C} \times (1 - \mu_C - \gamma \times \sigma_C), \tag{3.7}$$

trong đó μ'_C là trung bình giá trị độ tin cậy của các vùng phát hiện bởi bộ phát hiện đối tượng thứ hai (đối tượng khó), σ_C là độ lệch chuẩn (*standard deviation*) của các giá trị độ tin cậy, and γ là hệ số điều chỉnh việc kết hợp giữa hai phân bố giá trị độ tin cậy của hai bộ phát hiện đối tượng. Trong các thử nghiệm, chúng tôi thiết lập $\gamma = 0.2$.

Để công thức 3.7 có thể hoạt động được trong mọi trường hợp, chúng tôi bổ sung ràng buộc sau:

$$S^{Normalized} = S^{Original} \times \frac{1}{\mu'_C} \times max(0, (1 - \mu_C - \gamma \times \sigma_C)), \tag{3.8}$$

Lưu ý rằng, trường hợp $(1 - \mu_C - \gamma \times \sigma_C) < 0$ chưa xảy ra với các phương pháp phát hiện đối tượng và tập dữ liệu chúng tôi đã thử nghiệm.

Ngoài ra, chúng tôi cũng bổ sung thêm ràng buộc $\mu_C > 0.5$. Ràng buộc này để đảm bảo giá trị độ tin cậy của các đối tượng được phát hiện bởi bộ phát hiện đối tượng khó phải có phân bố với giá trị trung bình thấp hơn so với các đối tượng được phát hiện bởi bộ phát hiện đối tượng dễ. Trong trường hợp ràng buộc này bị vi phạm, các đối tượng được phát hiện sẽ được lọc lại thông qua thao tác lấy ngưỡng giá trị độ tin cậy cao hơn (thông thường được thiết lập là 0.3 như Fast/Faster R-CNN).

Chương 4
THỬ NGHIỆM VÀ KẾT QUẢ

4.1 Giới thiệu các Datasets

Trong nghiên cứu này, chúng tôi đánh giá phương pháp đề xuất (YALA) trên ba tập dữ liệu chuẩn là PASCAL VOC, KITTI, và MS-COCO.

PASCAL VOC (Visual Object Classes Challenge) [78] là một tập dữ liệu thông dụng trong lĩnh vực phát hiện và nhận dạng đối tượng trong cảnh thực tế. Tập dữ liệu này bao gồm 20 lớp đối tượng bao gồm nhiều loại: người, động vật, phương tiện giao thông, và các loại đối tượng trong nhà (như chai nước, ghế, ti vi, ghế dài - sofa, ...). Các thử nghiệm của chúng tôi được tiến hành trên tập PASCAL VOC 2007, bao gồm 9,936 ảnh, chứa tổng cộng 24,640 đối tượng được gán nhãn.

Trong khi đó, tập dữ liệu KITTI được thu thập bởi Geiger và các cộng sự [62]. Mục đích chính của tập dữ liệu này là xây dựng một tập tiêu chuẩn (*benchmark*) cho bài toán xe vận hành tự động với các thách thức thực tế. Tập dữ liệu này bao gồm 7,481 ảnh huấn luyện và 7,518 ảnh kiểm tra, bao gồm tổng cộng 80,256 đối tượng được gán nhãn thuộc về 8 lớp đối tượng tham gia giao thông.

Một tập dữ liệu khác rất phổ biến gần đây là MS-COCCO (Common Objects in Context [65]). Đây là tập dữ liệu qui mô lớn (*large-scale*) được Microsoft xây dựng cho bài các bài toán phát hiện đối tượng, phân vùng ảnh, và tạo tiêu đề tự động cho ảnh (*captioning*). Tập dữ liệu này chứa tổng cộng 80 lớp đối tượng. Các ảnh được gán nhãn thủ công và được chia thành các tập huấn luyện (*training*) chứa 80,000 ảnh, và tập kiểm chứng (*validation*) chứa 40,000 ảnh.

4.2 Giới thiệu các độ đo được sử dụng

Để đánh giá hiệu quả của thuật toán phát hiện đối tượng, chúng tôi sử dụng độ đo Độ chính xác bình quân (*Average precision - AP*). Độ đo AP được tính dựa trên tất cả các giá trị độ phủ (*recall*) tương ứng theo danh sách các vùng đối tượng đã được sắp xếp giảm dần theo độ tin cậy. Chúng tôi sử dụng phương

Bảng 4.1: Kết quả phát hiện trên tập kiểm tra (*testing set*) của PASCAL VOC 2007. Độ đo được sử dụng là AP (%). Tất cả các phương pháp sử dụng kiến trúc mạng VGG16. Tập dữ liệu huấn luyện là tập huấn luyện/kiểm chứng (*trainval set*) của PASCAL VOC 2007.

Phương pháp	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow
Zhu *et al.*	29.4	55.8	9.4	14.3	28.6	44.0	51.3	21.3	20.0	19.3
Felzenszwalb *et al.*	31.2	61.5	11.9	17.4	27.0	49.1	59.6	23.1	23.0	26.3
Harzallah *et al.*	35.1	45.6	10.9	12.0	23.2	42.1	50.9	19.0	18.0	31.5
Chen *et al.*	38.6	58.7	18.0	18.7	31.8	53.6	56.0	30.6	23.5	31.1
Fast R-CNN	74.6	79.0	68.6	57.0	39.3	79.5	78.6	81.9	48.0	74.0
Faster R-CNN	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3
OHEM	71.2	78.3	69.2	57.9	46.5	81.8	79.1	83.2	47.9	76.2
SSD300	73.4	77.5	64.1	59.0	38.9	75.2	80.8	78.5	46.0	67.8
SSD512	75.1	81.4	69.8	60.8	46.3	**82.6**	84.7	84.1	48.5	75.0
CC-Net	**78.3**	79.4	69.1	**63.5**	**53.2**	82.1	79.7	**86.3**	**56.0**	75.6
YALA	75.2	**84.1**	**72.4**	60.4	**53.2**	82.3	**85.8**	85.5	53.1	**81.0**

Phương pháp	mAP	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
Zhu *et al.*[80]	29.6	25.2	12.5	50.4	38.4	36.6	15.1	19.7	25.1	36.8	39.3
Felzenszwalb *et al.*[12]	34.1	24.9	12.9	60.1	51.0	43.2	13.4	18.8	36.2	49.1	43.0
Harzallah *et al.*[81]	28.9	17.2	17.6	49.6	43.1	21.0	18.9	27.3	24.7	29.9	39.7
Chen *et al.*[82]	37.7	36.6	20.9	62.6	47.9	41.2	18.8	23.5	41.8	53.6	45.3
Fast R-CNN[14]	68.1	67.4	80.5	80.7	74.1	69.6	31.8	67.1	68.4	75.3	65.5
Faster R-CNN[6]	69.9	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	**81.1**	67.6
OHEM[51]	69.9	68.9	83.2	80.8	75.8	72.7	39.9	67.5	66.2	75.6	**75.9**
SSD300[17]	68.0	69.2	76.6	82.1	77.0	72.5	41.2	64.2	69.1	78.0	68.5
SSD512[17]	71.6	67.4	82.3	83.9	79.4	76.6	**44.9**	69.9	69.1	78.1	71.8
CC-Net [83]	72.4	**72.3**	**83.4**	79.0	76.3	76.4	43.1	67.6	**71.8**	77.3	**76.6**
YALA	**72.8**	67.7	82.9	**84.1**	**79.6**	**80.3**	41.1	**71.7**	64.3	76.1	75.4

pháp tính AP sử dụng tất cả các điểm dữ liệu [76] thay vì chỉ sử dụng 11 điểm dữ liệu dãn đều như được trình bày ở [78].

4.3 Kết quả phương pháp YALA

4.3.1 Kết quả trên tập dữ liệu PASCAL VOC

Chúng tôi so sánh phương pháp của chúng tôi với các thuật toán hiện đại khác, bao gồm: Fast R-CNN, Faster R-CNN, OHEM, SSD300, và SSD512. **Lưu ý rằng trong số các phương pháp này, OHEM được đánh giá là một trong những phương pháp phát hiện đối tượng khó đạt hiệu quả cao**. Kết quả được thể hiện trong Bảng 4.1 cho thấy hiệu quả của phương pháp đề xuất. Phương pháp của chúng tôi có độ chính xác cao hơn tất cả các phương pháp được so sánh. Cụ thể, phương pháp đề xuất đạt độ chính xác cao hơn Fast R-CNN **4.7%** mAP, và Faster R-CNN **2.9%** mAP. So sánh với phương pháp tương tự theo cách tiếp cận khai thác các đối tượng khó OHEM (*Online Hard*

Bảng 4.2: Kết quả phát hiện trên tập dữ liệu kiểm chứng (*validation set*) của KITTI sử dụng độ đo AP (%) với nhiều kiến trúc mạng khác nhau của thuật toán Faster R-CNN và phương pháp đề xuất - YALA. Dấu (*) ghi chú phương pháp Faster R-CNN được cài đặt lại.

Phương pháp	Mô hình	mAP	car	pedes.	cyclist	truck	van	tram	misc
Faster R-CNN*-SS	VGGM	74.1	80.4	56.9	66.4	86.2	81.4	81.6	65.9
Faster R-CNN*-MS	VGGM	75.9	83.4	58.9	65.9	89.3	83.1	81.1	69.8
Faster R-CNN*-SS	VGG16	79.8	82.5	62.6	72.1	92.2	87.2	88.4	73.5
Faster R-CNN*-MS	VGG16	**83.0**	**87.0**	**67.1**	**75.0**	**94.1**	**89.2**	**92.1**	**76.3**
YALA-SS	VGGM	79.0	86.1	61.6	70.8	91.4	85.1	87.1	70.9
YALA-MS	VGGM	79.4	87.3	64.2	70.3	91.2	84.8	84.8	73.3
YALA-SS	VGG16	83.8	87.6	69.8	76.8	94.1	90.1	89.5	78.4
YALA-MS	VGG16	**85.9**	**90.1**	**72.5**	**80.2**	**95.0**	**91.2**	**92.7**	**79.4**

Example Mining) [51], phương pháp của chúng tôi có độ chính xác cao hơn **2.9%** mAP.

4.3.2 Kết quả trên tập dữ liệu KITTI

Các kết quả dựa trên các cấu hình thử nghiệm khác nhau trên tập dữ liệu KITTI được trình bày trong Bảng 4.2. Độ chính xác tổng thể của phương pháp đề xuất với chế độ đơn tỉ lệ (YALA-SS) là **83.8%**, cải tiến **4.0%** so với Faster R-CNN đơn tỉ lệ (Faster R-CNN-SS). Khi phát hiện trên đa tỉ lệ phương pháp của chúng tôi (YALA-MS) cải tiến độ chính xác thêm 2.1% so với đơn tỉ lệ.

4.3.3 Kết quả trên tập dữ liệu MS-COCO

Hiệu quả của YALA cũng được đánh giá trên tập dữ liệu lớn hơn - MS-COCO. Bảng 4.3 trình bày kết quả của phương pháp đề xuất trên tập dữ liệu *test-dev2015*. Tương tự như kết quả quan sát được trên PASCAL VOC và KITTI, phương pháp đề xuất YALA tăng cường đáng kể độ chính xác của bộ phát hiện đối tượng cơ sở Faster RCNN - **2.1**% trên độ đo mAP@0.5. So sánh với các bộ phát hiện đối tượng hiện đại khác, phương pháp YALA có độ chính xác cao nhất (mAP@0.5). Hiệu quả của YALA tốt hơn SSD512 và SSD300 tương ứng 0.9% và 6.2% trên độ đo mAP. Đáng chú ý, YALA đạt được độ phủ (*recall*) cao hơn đáng kể so với SSD512 đó là 2.9%, 2.4%, và 1.5% mAR đối với các kích thước đối tượng lần lượt là nhỏ, trung bình, và lớn (dựa theo phân loại của COCO).

Bảng 4.3: Độ chính xác của các phương pháp phát hiện đối tượng trên tập dữ liệu MS-COCO *test-dev2015*. Kiến trúc mạng VGG16 được sử dụng cho tất các phương pháp.

Phương pháp	Tập huấn luyện	AP[0.5]	AP[0.5:0.95], Area:			AR[0.5:0.95], #Dets:			AR[0.5:0.95], Area:		
			S	M	L	1	10	100	S	M	L
Fast R-CNN [14]	train	39.9	4.1	20	35.8	21.3	29.5	30.1	7.3	32.1	52.0
ION [47]	train	43.2	6.4	24.1	38.3	23.2	32.7	33.5	10.1	37.7	53.6
OHEM [51]	trainval	45.9	7.4	27.7	40.3	-	-	-	-	-	-
Cascade R-CNN [84]	train	44.3	8.3	28.2	41.1	-	-	-	-	-	-
Faster R-CNN [6]	trainval35k	45.3	7.7	26.4	37.1	23.8	34.0	34.6	12.0	38.5	54.4
SSD300 [17]	trainval35k	41.2	5.3	23.2	39.6	22.5	33.2	35.3	9.6	37.6	56.5
SSD512 [17]	trainval35k	46.5	9.0	**28.9**	**41.9**	**24.8**	37.5	39.8	14.0	43.5	59.0
YALA	trainval35k	**47.4**	**9.6**	28.4	37.7	24.6	**38.9**	**40.8**	**16.9**	**45.9**	**60.5**

Bảng 4.4: So sánh tốc độ thực thi và độ chính xác của các phương pháp khác nhau trên tập dữ liệu kiểm tra (*testing set*) của PASCAL VOC 2007. Tập dữ liệu huấn luyện được sử dụng là tập huấn luyện/kiểm chứng (*trainval set*) của PASCAL VOC 2007+PASCAL VOC 2012. Kiến trúc mạng VGG16 được sử dụng cho tất cả các phương pháp.

Phương pháp	Fast R-CNN	Faster R-CNN	YOLOv1	SSD300	SSD512	YALA-SS	YALA-MS
mAP (%)	70.0	73.2	66.4	74.3	76.8	77.5	79.0
Tốc độ (fps)	0.5	7	21	46	19	4	1

4.3.4 So sánh về thời gian thực thi

Bảng 4.4 trình bày thời gian thực thi của các phương pháp phát hiện đối tượng khác nhau bao gồm Fast R-CNN, Faster R-CNN, YOLO (*phiên bản đầu tiên được công bố năm 2016* - YOLOv1), SSD, và phương pháp đề xuất - YALA. Phương pháp YALA được xây dựng dựa trên hai tầng Faster R-CNN, và do vậy có tốc độ thực thi chậm hơn khoảng 2 lần. Tuy nhiên xét về mặt độ chính xác, YALA cao hơn Faster R-CNN 4.3% mAP (cấu hình đơn tỉ lệ). So sánh với các bộ phát hiện có tốc độ nhanh như YOLOv1 và SS, YALA có thời gian thực thi chậm hơn nhưng cho độ chính xác cao hơn.

4.4 Kết quả phương pháp YADA

Trong phần này chúng tôi tiến hành đánh giá kết quả của phương pháp YADA đề xuất, đồng thời so sánh kết quả của YADA với các phương pháp phát hiện đối tượng hiện đại bao gồm: Fast R-CNN, Faster R-CNN, SSD, OHEM, SNIPER, RFCN, YOLOv2 (phiên bản thứ hai hay còn gọi là YOLO9000 [9], được công bố vào năm 2017).

Bảng 4.5: Độ chính xác của các thuật toán trên tập dữ liệu kiểm tra (*testing set*) của PASCAL VOC 2007 (% mAP). Kiến trúc mạng VGG16 được sử dụng cho tất các các phương pháp. Tập dữ liệu huấn luyện được sử dụng là tập huấn luyện/kiểm chứng (*trainval set*) của PASCAL VOC 2007.

Phương pháp	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow
Fast R-CNN	74.6	79.0	68.6	57.0	39.3	79.5	78.6	81.9	48.0	74.0
Faster R-CNN	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3
OHEM	71.2	78.3	69.2	57.9	46.5	81.8	79.1	83.2	47.9	76.2
SSD300	73.4	77.5	64.1	59.0	38.9	75.2	80.8	78.5	46.0	67.8
SSD512	75.1	81.4	69.8	60.8	46.3	82.6	84.7	84.1	48.5	75.0
YADA-SS	73.0	83.3	72.4	58.1	51.5	83.1	85.5	**87.5**	50.2	78.5
YADA-MS	**75.5**	**84.7**	**74.3**	**61.2**	**54.0**	**83.2**	**87.1**	85.6	**53.1**	**80.3**

Phương pháp	mAP	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast R-CNN[14]	68.1	67.4	80.5	80.7	74.1	69.6	31.8	67.1	68.4	75.3	65.5
Faster R-CNN[6]	69.9	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	**81.1**	67.6
OHEM[51]	69.9	68.9	83.2	80.8	75.8	72.7	39.9	67.5	66.2	75.6	75.9
SSD300[17]	68.0	**69.2**	76.6	82.1	77.0	72.5	41.2	64.2	**69.1**	78.0	68.5
SSD512[17]	71.6	67.4	82.3	83.9	79.4	76.6	**44.9**	69.9	**69.1**	78.1	71.8
YADA-SS	72.5	68.7	**84.1**	**86.1**	77.9	80.3	38.7	**72.7**	65.7	77.9	74.3
YADA-MS	**73.5**	68.7	83.3	85.1	**80.4**	**80.6**	42.2	72.0	65.2	77.9	**76.4**

Bảng 4.6: Độ chính xác của các thuật toán trên tập dữ liệu kiểm chứng (*validation set*) của KITTI (% mAP). Faster R-CNN được cài đặt với các kiến trúc mạng khác nhau - VGGM và VGG16. Ký hiệu '*' ghi chú thuật toán Faster R-CNN cơ sở được cài đặt. Các kết quả cao nhất trên mỗi nhóm đối tượng được tô đậm.

Phương pháp	Mô hình	mAP	car	pedes.	cyclist	truck	van	tram	misc
Faster R-CNN*-SS	VGGM	74.1	80.4	56.9	66.4	86.2	81.4	81.6	65.9
Faster R-CNN*-MS	VGGM	75.9	83.4	58.9	65.9	89.3	83.1	81.1	69.8
Faster R-CNN*-SS	VGG16	79.8	82.5	62.6	72.1	92.2	87.2	88.4	73.5
Faster R-CNN*-MS	VGG16	**83.0**	**87.0**	**67.1**	**75.0**	**94.1**	**89.2**	**92.1**	**76.3**
YADA-SS	VGG16	83.2	87.1	68.2	75.9	93.4	89.8	90.7	77.2
YADA-MS	VGG16	**85.0**	**89.8**	**70.8**	**77.5**	**94.7**	**90.6**	**92.6**	**79.1**

4.4.1 Kết quả thử nghiệm trên tập dữ liệu PASCAL VOC

Hiệu quả của phương pháp đề xuất YADA được so sánh với các phương pháp tân tiến hiện nay. Kết quả trình bày trong Bảng 4.5 cho thấy tính hiệu quả của YADA. Phương pháp đề xuất của chúng tôi đem lại độ chính xác cao hơn so với các phương pháp được liệt kê. Cụ thể, YADA có độ chính xác cao hơn Fast R-CNN **5.4%** mAP, và Faster R-CNN **3.6%** mAP. So sánh với phương pháp khai thác đối tượng khó tân tiến - OHEM [51], phương pháp đề xuất cũng đạt được độ chính xác cao hơn **3.6%** mAP.

Bảng 4.7: Độ chính xác (mAP %) của phương pháp đề xuất YADA trên tập dữ liệu kiểm tra (*testing set*) của PASCAL VOC 2007 với các cấu hình sử dụng/không sử dụng tái điều chỉnh độ tin cậy. Tập dữ liệu huấn luyện là tập huấn luyện/kiểm chứng (*trainval set*) của PASCAL VOC 2007+PASCAL VOC 2012. Kiến trúc mạng VGG-16 được sử dụng.

Method		aero	bike	bird	boat	bottle	bus	car
Faster-RCNN*-SS		79.9	83.4	77.2	64.9	57.4	88.0	88.1
Faster-RCNN*-MS		80.4	84.3	80.9	69.5	62.3	89.0	89.2
YADA-SS	không	76.8	81.3	72.1	59.9	56.1	85.9	86.7
YADA-MS	điều chỉnh	79.2	81.5	74.2	61.8	59.1	87.2	88.1
YADA-SS	có	81.6	85.3	78.7	65.8	59.0	88.5	89.2
YADA-MS	điều chỉnh	**82.0**	**85.4**	**82.0**	**70.1**	**63.0**	**89.2**	**90.1**

Method		cat	chair	cow	table	dog	horse	mbike
Faster-RCNN*-SS		**92.9**	55.8	83.6	70.8	87.5	89.1	78.2
Faster-RCNN*-MS		91.8	57.9	87.3	71.9	87.8	89.1	79.2
YADA-SS	không	89.0	51.8	81.2	68.5	84.0	87.3	73.7
YADA-MS	điều chỉnh	89.2	53.0	84.5	69.1	84.1	86.8	73.8
YADA-SS	có	93.3	56.6	85.5	71.0	**88.9**	**89.9**	**80.6**
YADA-MS	điều chỉnh	92.4	**58.6**	**88.3**	**72.0**	88.4	89.5	80.5

Method		mAP	person	plant	sheep	sofa	train	tv
Faster-RCNN*-SS		76.7	81.2	42.4	79.3	72.1	83.2	78.4
Faster-RCNN*-MS		78.5	83.1	45.8	81.5	73.8	85.1	79.3
YADA-SS	không	73.9	79.1	42.2	75.7	69.8	80.9	75.8
YADA-MS	điều chỉnh	75.3	80.6	43.9	77.9	70.8	82.7	77.5
YADA-SS	có	77.9	83.1	44.1	80.9	73.4	84.2	79.0
YADA-MS	điều chỉnh	**79.3**	**84.4**	**46.8**	**82.1**	**74.8**	**85.9**	**80.2**

4.4.2 Kết quả thử nghiệm trên tập dữ liệu KITTI

Kết quả thử nghiệm với các cấu hình khác nhau trên tập dữ liệu KITTI được trình bày trong Bảng 4.6. Phương pháp đề xuất-YADA cho độ chính xác cao hơn so với Faster R-CNN với tất cả các kiến trúc mạng khác nhau. Độ chính xác trên toàn bộ các lớp của YADA-SS (đơn tỉ lệ) là **83.2%**, tăng cường **3.4%** so với Faster R-CNN-SS (đơn tỉ lệ). YADA-MS (đa tỉ lệ) nâng cao độ chính xác với 1.8% cao hơn so với YADA-SS.

4.4.3 Hiệu quả của phương pháp tái điều chỉnh độ tin cậy trong YADA

Để làm rõ hiệu quả của phương pháp của phương pháp tái điều chỉnh độ tin cậy (*score-scaling*), chúng tôi đánh giá độ chính xác của YALA với hai thiết lập: không sử dụng/có sử dụng tái điều chỉnh độ tin cậy. Các kết quả được trình bày trong Bảng 4.7. Kết quả cho thấy YADA trong thiết lập không sử dụng tái điều

Bảng 4.8: Độ chính xác (mAP %) của YADA khi sử dụng các phương pháp cơ sơ khác nhau trên tập dữ liệu kiểm tra (*testing set*) của PASCAL VOC 2007. RFCN và SNIPER sử dung kiến trúc mạng Resnet-101. YOLOv2 sử dụng kiến trúc mạng Darknet-19. Tập dữ liệu dùng huấn luyện là tập huấn luyện/kiểm chứng (*trainval set*) của PASCAL VOC 2007+PASCAL VOC 2012.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow
YOLOv2 352x352	72.1	80.3	66.6	56.0	41.1	80.5	79.6	85.5	46.3	69.1
RFCN	82.8	88.4	83.2	71.9	70.3	88.6	90.7	91.7	67.8	88.6
SNIPER	91.3	93.8	88.3	81.2	78.3	91.3	95.1	89.1	75.7	**89.6**
YADA-YOLOv2	72.5	81.0	67.7	56.6	42.0	80.6	80.4	86.0	47.6	69.6
YADA-RFCN	84.5	89.3	83.5	72.3	71.2	89.2	91.3	**91.8**	68.1	89.3
YADA-SNIPER	**91.4**	**93.9**	**88.5**	**81.4**	**78.9**	**91.4**	**95.2**	89.1	**76.3**	89.5

Method	mAP	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
YOLOv2 352x352[9]	70.2	72.2	80.5	84.7	83.5	71.8	39.8	66.4	73.9	81.3	72.4
RFCN[85]	82.0	75.6	92.4	89.8	85.4	85.2	52.5	82.9	81.8	88.6	80.9
SNIPER[86]	86.7	84.8	86.8	91.6	93.1	91.5	67.4	86.9	86.7	90.1	80.9
YADA-YOLOv2	70.9	73.1	80.7	85.0	83.9	73.0	41.0	67.3	75.1	82.1	73.1
YADA-RFCN	82.6	75.7	**92.7**	90.0	86.4	86.3	53.3	83.0	82.1	89.2	**81.8**
YADA-SNIPER	**87.0**	**85.1**	87.2	**92.0**	**93.2**	**91.7**	**68.1**	**87.0**	**86.9**	**90.7**	81.5

chỉnh độ tin cậy có độ chính xác (mAP) giảm 2.8% và 3.2% tương ứng với các cấu hình đơn tỉ lệ và đa tỉ lệ so với Faster R-CNN. Phương pháp đề xuất YADA khi sử dụng tái điều chỉnh độ tin cậy cho độ chính xác (mAP) cao hơn Faster R-CNN tương ứng với các cấu hình đơn tỉ lệ và đa tỉ lệ là 1.2% and 0.8%. Giải pháp được đề xuất giúp hạn chế ảnh hưởng của các phát hiện nhầm lẫn (do mức độ không chắc chắn của các phát hiện từ mô hình phát hiện các đối tượng khó).

4.4.4 Hiệu quả của phương pháp đề xuất YADA trên các kiến trúc mạng và phương pháp phát hiện cơ sở khác

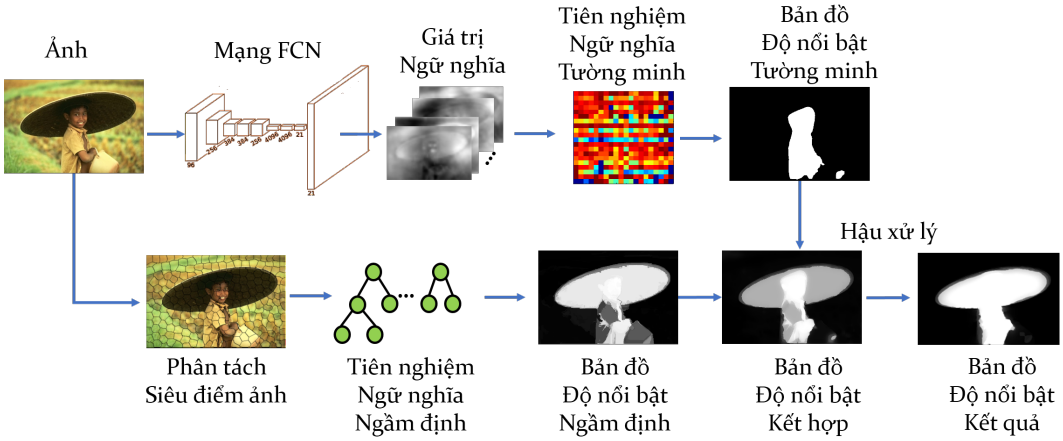
Trong phần này, chúng tôi tập trung vào việc đánh giá hiệu quả của YADA trên các phương pháp phát hiện đối tượng và kiến trúc mạng khác nhau. Thay vì sử dụng Faster RCNN trong các thử nghiệm trước, YOLOv2 [9], RFCN [85], và SNIPER [86] được tích hợp vào phương pháp đề xuất YADA. Chú ý là RFCN và SNIPER sử dụng kiến trúc mạng Resnet-101, và YOLOv2 sử dụng kiến trúc mạng Darknet-19. Kết quả trình bày trong Bảng 4.8 cho thấy hiệu quả của YADA với các phương pháp này. Trên tập dữ liệu PASCAL VOC 2007, YADA cho kết quả tốt hơn so với YOLOv2, RFCN, và SNIPER lần lượt là 0.7%, 0.6%, và 0.3% mAP. Kết quả này cho thấy phương pháp đề xuất của chúng tôi đem lại hiệu quả tốt cho cho những bộ phát hiện đối tượng và kiến trúc mạng hiện đại.

Chương 5
ÁP DỤNG CHO BÀI TOÁN PHÁT HIỆN ĐỐI TƯỢNG CHÍNH TRONG ẢNH

5.1 Động lực

Các kết quả phát hiện đối tượng trong ảnh có thể được áp dụng cho các bài toán lớn hơn như quản lý giao thông hay tìm kiếm đối tượng,... Trong luận án này, chúng tôi thử nghiệm kết quả phát hiện đối tượng trên bài toán phát hiện đối tượng chính (*salient object detection*) trong ảnh. Cụ thể hơn, đối tượng chính ở đây là đối tượng nổi bật thu hút ngay sự chú ý của người nhìn vào ảnh. Bài toán nghiên cứu này ngày càng được quan tâm bởi cộng đồng thị giác máy tính. Bài toán đã được áp dụng cho nhiều ứng dụng như phân loại hình ảnh [66], phân loại video [67], điều hướng chú ý [68], và quảng cáo có chủ đích [69].

Nhiều nghiên cứu tập trung vào phát triển và cải thiện tính chính xác của các mô hình phát hiện đối tượng chính, như sử dụng các đặc trưng thủ công như tương phản toàn cục [88], tương phản cục bộ [89, 90], hoặc chiến lược phân chia ảnh đầu vào thành các siêu điểm ảnh [91, 92]. Trước những tiến bộ trong lĩnh vực, động lực của chúng tôi bắt nguồn từ một câu hỏi nghiên cứu đơn giản "Tại sao đối tượng này được coi là nổi bật hơn các đối tượng khác trong cùng một hình ảnh?". Câu hỏi này trở nên cần thiết khi các bộ dữ liệu phức tạp như ECSSD [94, 95] và HKUIS [96] được công bố với một hoặc nhiều đối tượng trên ảnh nền phức tạp. Trong trường hợp có nhiều đối tượng, người gán nhãn sẽ xác định nhãn ngữ nghĩa cho từng đối tượng và sau đó mới xác định xem đối tượng nào là đối tượng chính. Điều này truyền cảm hứng cho chúng tôi để kết nối vấn đề phát hiện đối tượng (như trong YALA/YADA đề xuất phía trên) với phát hiện đối tượng chính. Cụ thể hơn, chúng tôi áp dụng việc trích xuất mặt nạ trong những khung bao (*bounding box*) chứa đối tượng để tính toán thông tin ngữ nghĩa. Từ đó, chúng tôi nghiên cứu tường minh ảnh hưởng của thông tin ngữ nghĩa đối với bài toán phát hiện đối tượng chính. Cụ thể, chúng tôi đề



Hình 5.1: Quá trình hoạt động của phương pháp đề xuất "Tiên nghiệm ngữ nghĩa" (tên tiếng Anh là Semantic Priors, viết tắt là SP): đầu tiên điểm số ngữ nghĩa được rút trích từ bộ phân tích ngữ nghĩa (mục 5.2.1), sau đó sẽ tính toán các bản đồ độ nổi bật tường minh (mục 5.2.2) và bản đồ độ nổi bật ngầm định (mục 5.2.3), tích hợp các bản đồ (mục 5.2.4).

xuất các xác suất tiên nghiệm ngữ nghĩa để xây dựng các ánh xạ tường minh và không tường minh nhằm cho ra mô hình phát hiện đối tượng chính chất lượng.

5.2 Phương pháp đề xuất

Trong phần này, chúng tôi trình bày chi tiết phương pháp đề xuất "**Tiên nghiệm ngữ nghĩa**" (tên tiếng Anh là Semantic Priors, viết tắt là **SP**). Hình 5.1 minh họa quá trình hoạt động của phương pháp đề xuất.

5.2.1 Rút trích ngữ nghĩa

Trong luận án này, để tiết kiệm thời gian cài đặt các trình ngữ nghĩa hiện có vào YALA/YADA, chúng tôi tận dụng sẵn các trình rút trích ngữ nghĩa hiện có dựa trên CNN [26] hoặc RPN [6]. Cụ thể hơn, chúng tôi tích hợp các phương pháp phân tích ngữ nghĩa theo cơ chế liền mạch (*end-to-end*) khác nhau, cụ thể là FCN [101], và Mask RCNN [15] vào hệ thống đề xuất của chúng tôi. Trong ngữ cảnh này, "liền mạch" có nghĩa là một bản đồ ngữ nghĩa hoàn chỉnh $\mathbb{C}$ có thể trích xuất trực tiếp ở tầng cuối của mạng khi đưa ảnh vào các mạng học sâu. Cụ thể, chúng tôi sử dụng các giá trị $\mathbb{C}_{x,y}$ cho mỗi điểm ảnh (x, y) như dưới đây:

$$\mathbb{C}_{x,y} = \{\mathbb{C}^1_{x,y}, \mathbb{C}^2_{x,y}, \cdots, \mathbb{C}^{n_c}_{x,y}\}, \tag{5.1}$$

trong đó $\mathbb{C}^1_{x,y}, \mathbb{C}^2_{x,y}, \cdots, \mathbb{C}^{n_c}_{x,y}$ chỉ ra khả năng điểm ảnh (x, y) thuộc về các lớp ngữ nghĩa n_c được liệt kê. Với một hình ảnh đầu vào có kích thước $h \times w$, kích thước của $\mathbb{C}$ là $h \times w \times n_c$.

5.2.2 Bản đồ độ nổi bật tường minh

Bản đồ độ nổi bật tường minh (*Explicit Saliency Map*) nhằm mục đích nắm bắt thứ tự chú ý ưu tiên của con người đối với các lớp ngữ nghĩa khác nhau như "người", "xe hơi" hoặc "con ngựa". Cụ thể, chúng tôi hướng đến việc cho máy học xem lớp nào sẽ gây chú ý hơn nếu tồn tại nhiều hơn hai lớp ngữ nghĩa trong hình ảnh đầu vào. Từ bản đồ phản hồi $\mathbb{C}$ thu được từ bước trước, chúng tôi tính toán nhãn lớp $L_{x,y}$ của mỗi điểm ảnh đơn lẻ (x, y) dưới dạng:

$$\mathbb{L}_{x,y} = \arg\max \mathbb{C}_{x,y}. \tag{5.2}$$

$\mathbb{L}_{x,y}$ sẽ là chỉ mục của lớp ngữ nghĩa được gán cho điểm ảnh (x, y).

Trong giai đoạn huấn luyện, với một bản đồ độ nổi bật đã được gán nhãn $\mathbb{G}$ trong dữ liệu huấn luyện, mật độ của mỗi lớp ngữ nghĩa k trong hình ảnh đầu vào được tính bằng:

$$p_k = \frac{\sum_{x,y}(\mathbb{L}_{x,y} = k) \times \mathbb{G}_{x,y}}{\sum_{x,y}(\mathbb{L}_{x,y} = k)}, \tag{5.3}$$

trong đó $(\mathbb{L}_{x,y} = k)$ là phép so sánh nhị phân xác nhận xem liệu chỉ số lớp được gán $\mathbb{L}_{x,y}$ bằng k hay không.

Chúng tôi định nghĩa **Tiên nghiệm ngữ nghĩa tường minh** là sự tích lũy của độ nổi bật theo cặp của tất cả các lớp. Các tiên nghiệm ngữ nghĩa tường minh của hai lớp k và t được định nghĩa như sau:

$$sp^{Explicit}_{k,t} = \frac{\sum_{i=1}^{n_t} p^i_k \theta^i_{k,t}}{\sum_{i=1}^{n_t} \theta^i_{k,t} + \epsilon}, \tag{5.4}$$

trong đó n_t là số lượng hình ảnh trong tập huấn luyện, ϵ được thêm vào để tránh chia cho 0 và giá trị cặp $\theta_{g,t}$ của bất kỳ cặp lớp ngữ nghĩa nào k và t được tính như dưới đây.

$$\theta_{k,t} = \begin{cases} 1 & \text{, nếu } \exists \mathbb{L}_{x',y'} = k \wedge \mathbb{L}_{x'',y''} = t \\ 0 & \text{, ngược lại} \end{cases}. \tag{5.5}$$

Lưu ý rằng chúng tôi trích xuất độ nổi bật đồng thời xuất hiện theo cặp của một lớp ngữ nghĩa với $n_c - 1$ lớp khác từ dữ liệu huấn luyện.

Trong giai đoạn kiểm tra, với một hình ảnh cho trước, giá trị độ nổi bật tường minh của từng điểm ảnh (x, y) được tính như sau:

$$S_{x,y}^{Explicit} = \sum_{k=1}^{n_c} \sum_{t=1}^{n_c} (\mathbb{L}_{x,y} = k) \times \theta_{k,t} \times sp_{k,t}^{Explicit}. \tag{5.6}$$

5.2.3 Bản đồ độ nổi bật ngầm định

Chúng tôi đề xuất một bản đồ bổ sung, cụ thể là bản đồ độ nổi bật ngầm định (*Implicit Saliency Map*), nhằm mục đích khám phá các đối tượng nổi bật không thuộc các lớp ngữ nghĩa được liệt kê. Để xây dựng bản đồ này, chúng tôi tách nhỏ ảnh đầu vào thành các siêu điểm ảnh (*super-pixel*) không chồng lấp và trích xuất các đặc trưng của siêu điểm ảnh. Ngoài các đặc trưng siêu điểm ảnh hiện có, chúng tôi tích hợp hai đặc trưng mới cho mỗi siêu điểm ảnh, được đặt tên là đặc trưng ngữ nghĩa cục bộ và đặc trưng ngữ nghĩa toàn cục. Đặc trưng ngữ nghĩa cục bộ của mỗi siêu điểm ảnh q được định nghĩa là: $sp_1 = \frac{\sum_{x,y} \mathbb{C}_{x,y} \times (idx(x,y)=q)}{\sum_{x,y} (idx(x,y)=q)}$, trong đó $idx(x, y)$ là một hàm trả về chỉ số siêu điểm ảnh của điểm ảnh (x, y). Lưu ý rằng $\mathbb{C}$ có thể thu được thông qua trình phân tích ngữ nghĩa như được đề cập trong Công thức 5.1. Trong khi đó, đặc trưng ngữ nghĩa toàn cục được định nghĩa là: $sp_2 = \frac{\sum_{x,y} \mathbb{C}_{x,y}}{h \times w}$.

Các đặc trưng ngữ nghĩa $sp^{Implicit} = \{sp_1, sp_2\}$ cuối cùng được kết hợp với các đặc trưng siêu điểm ảnh khác. Chúng tôi coi các đặc trưng ngữ nghĩa ở đây là **tiên nghiệm ngữ nghĩa ngầm định** vì chúng có các ảnh hưởng không thể chỉ rõ một cách tường minh đến việc ánh xạ các đặc trưng siêu điểm ảnh vào giá trị độ nổi bật.

Trong giai đoạn huấn luyện, chúng tôi trích xuất một tập hợp n_r siêu điểm ảnh $\{\{r_1, sp_1^{Implicit}\}, \{r_2, sp_2^{Implicit}\}, \cdots, \{r_{n_r}, sp_{n_r}^{Implicit}\}\}$ từ bộ ảnh huấn luyện. Tiếp đó, chúng tôi huấn luyện một bộ hồi quy rf để ước tính các giá trị nổi bật của các đặc trưng siêu điểm ảnh được trích xuất nói trên.

Trong giai đoạn kiểm thử, trước tiên chúng tôi tách ảnh đầu vào thành các siêu điểm ảnh và trích xuất các đặc trưng tương ứng của các siêu điểm ảnh đó. Giá trị độ nổi bật không tường minh của mỗi siêu điểm ảnh kiểm thử q sau đó được tính bằng cách cung cấp các đặc trưng được trích xuất vào bộ hồi quy rf:

$$S_q^{Implicit} = rf(\{r_q, sp_q^{Implicit}\}). \tag{5.7}$$

Quá trình này được thực hiện trên tất cả các siêu điểm ảnh để tạo thành bản đồ độ nổi bật không tường minh $S^{Implicit}$.

5.2.4 Kết hợp các bản đồ độ nổi bật

Trước tiên, chúng tôi điều chỉnh (theo phép tỉ lệ) giá trị độ nổi bật của hai bản đồ $S^{Implicit}$ và $S^{Explicit}$ về phạm vi [0..1]. Tiếp đó, chúng tôi sử dụng phương pháp hợp nhất dựa trên trọng số thích nghi (*adaptive weight*) hai bản đồ này để tính giá trị độ nổi bật S^{Fusion} cho mỗi điểm ảnh:

$$S^{Fusion} = \alpha S^{Explicit} + (1 - \alpha) S^{Implicit}, \tag{5.8}$$

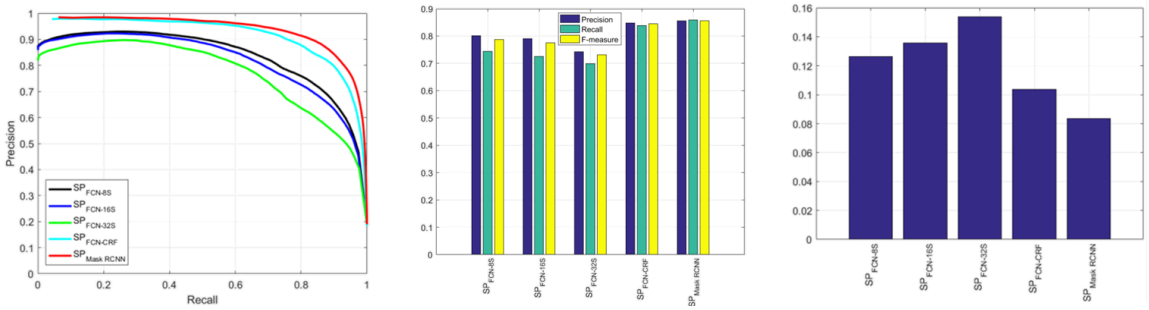
trong đó trọng số α dùng để điều chỉnh mức độ ảnh hưởng của các điểm ảnh mang ngữ nghĩa (bản đồ độ nổi bật tường minh) đối với giá trị độ nổi bật hợp nhất, được xác định tự động là $\frac{\sum_{x,y} S_{x,y}^{Implicit}}{h \times w}$.

5.3 Kết quả thử nghiệm

5.3.1 Thiết lập thí nghiệm

Các bộ dữ liệu chuẩn

Để đánh giá, chúng tôi tiến hành so sánh hiệu quả của phương pháp đề xuất với các phương pháp khác đã được công bố trên ba tập dữ liệu chuẩn: ECSSD [95], iCoSeg [109], HKUIS [96] (trên tập kiểm thử). **Tập dữ liệu ECSSD:** tập dữ liệu này chứa $1,000$ ảnh với cảnh nền phức tạp (chứa nhiều vật thể). **Tập dữ liệu HKUIS:** tập dữ liệu này bao gồm $5,447$ ảnh, được chia thành hai tập con: tập huấn luyện chứa $4,000$ ảnh và tập kiểm tra chứa $1,447$ ảnh. **Tập dữ liệu iCoSeg:** tập dữ liệu này chứa 643 ảnh. Chú ý rằng mỗi ảnh trong cả 3 tập dữ liệu có thể chứa một hoặc nhiều đối tượng chính.



(a) Ngưỡng cố định (b) Ngưỡng tương thích (c) Độ lỗi trung bình

Hình 5.2: Kết quả so sánh trên tập dữ liệu HKUIS cho những phương pháp rút trích ngữ nghĩa khác nhau.

Độ đo dùng để đánh giá:

Chúng tôi đánh giá các kết quả đạt được với các công cụ bao gồm biểu đồ Độ chính xác-Độ phủ (*Precision-Recall Curve - PRC*), độ đo F (*F-measure*), và sai số tuyệt đối trung bình (*Mean Absolute Error - MAE*).

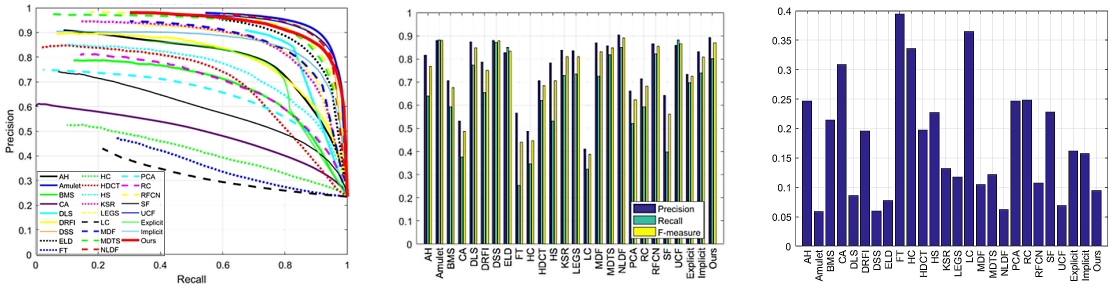
5.3.2 Hiệu quả của phương pháp phân tích ngữ nghĩa

Trước tiên, chúng tôi đánh giá mô hình SP được đề xuất với nhiều bộ phân tích ngữ nghĩa khác nhau. Như đã đề cập, chúng tôi tiến hành khảo sát nhiều phương pháp khác nhau bao gồm mạng tính chập đầy đủ (FCN) [101] với 3 thiết đặt 'FCN-8S', 'FCN-16S', 'FCN-32S', 'FCN-CRF' [107] và Mask R-CNN [15].

Hiệu quả của từng bộ phân tích được thể hiện trong Hình 5.2. Các độ đo bao gồm biểu đồ PRC, giá trị $F-measure$, và độ lỗi MAE, đều cho thấy phương pháp đề xuất SP kết hợp với bộ phân tích ngữ nghĩa Mask R-CNN đạt được kết quả tốt nhất. Hay nói cách khác, bản đồ độ nổi bật được tính toán bởi $SP_{MaskRCNN}$ gần giống với bản đồ độ nổi bật mong muốn (được gán nhãn) nhất. Do đó, chúng tôi sẽ sử dụng Mask R-CNN cho các bước thí nghiệm tiếp theo.

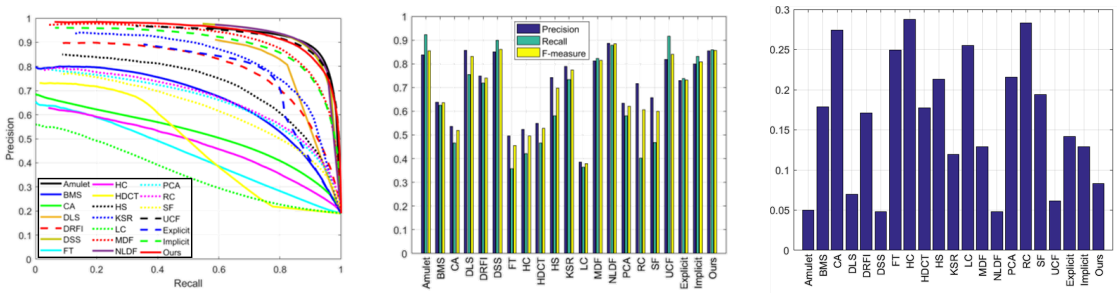
5.3.3 So sánh với các phương pháp hiện đại

Trong phần này chúng tôi đánh giá phương pháp đề xuất SP đồng thời với các phương pháp hiện đại trên 3 tập dữ liệu có độ khó cao bao gồm: ECSSD, HKUIS, và iCoSeg.



(a) Ngưỡng cố định (b) Ngưỡng tương thích (c) Độ lỗi trung bình

Hình 5.3: Kết quả so sánh trên tập dữ liệu ECSSD.



(a) Ngưỡng cố định (b) Ngưỡng tương thích (c) Độ lỗi trung bình

Hình 5.4: Kết quả so sánh trên tập dữ liệu HKUIS.

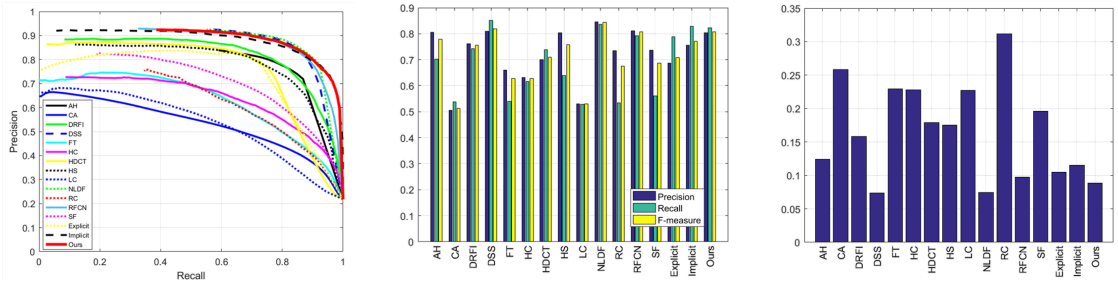
Đánh giá hiệu quả trên tập dữ liệu ECSSD

Chúng tôi so sánh hiệu quả của hệ thống đề xuất với các phương pháp hiện đại (23 phương pháp) bằng cách tiến hành thử nghiệm với các mã nguồn được cung cấp, hoặc so sánh với các kết quả đã được công bố.

Theo hình 5.3, phương pháp đề xuất của chúng tôi có kết quả vượt trội nhóm các phương pháp sử dụng đặc trưng tự thiết kế, và đạt độ chính xác cao trong nhóm các phương pháp dựa trên mạng học sâu (Amulet, DLS, DSS, ELD, KSR, LEGS, MDF, NLDF, MTDS, RFCN, và UCF).

Đánh giá hiệu quả trên tập dữ liệu HKUIS

Kết quả quan sát được tương tự như kết quả trên tập dữ liệu ECSSD ở mục trước. Phương pháp của chúng tôi đề xuất đạt được kết quả có độ chính xác khi so với các phương pháp được thử nghiệm. Như được thể hiện trong Hình 5.4, phương pháp chúng tôi đạt được độ chính xác rất cao trên độ đo đánh giá MAE (< 0.1).



(a) Ngưỡng cố định (b) Ngưỡng tương thích (c) Độ lỗi trung bình

Hình 5.5: Kết quả so sánh trên tập dữ liệu iCoSeg.

Đánh giá hiệu quả trên trên tập iCoSeg

Chúng tôi so sánh phương pháp được đề xuất SP với 13 phương pháp khác. Như quan sát được ở Hình 5.5a và 5.5b, phương pháp chúng tôi đạt được đường PRC tương đương với NLDF [119] và DSS [123], và cao hơn trên độ đo $F-measure$ so với các phương pháp khác. Chú ý rằng, kết quả của các phương pháp không hoàn toàn nhất quán khi sử dụng 3 độ đo đánh giá khác nhau. Cụ thể, HDCT [116] và DRFI [92] cho kết quả tốt hơn AH [110] trên đường cong PRC, tuy nhiên AH lại đạt được giá trị MAE tốt hơn. Ngoài ra, những phương pháp dựa trên mạng học sâu như NLDF và DSS không đạt được độ chính xác cao như kỳ vọng.

5.3.4 Thảo luận

Chúng tôi muốn nhấn mạnh vào mục tiêu chính của chương này là áp dụng kết quả của việc phát hiện đối tượng vào bài toán phát hiện đối tượng chính. Việc kết hợp khung bao và mặt nạ phân vùng ảnh đã góp phần tính được ngữ nghĩa cho mỗi điểm ảnh. Từ đó chúng tôi có thể áp dụng để phát hiện đối tượng chính. Phương pháp chúng tôi đề xuất đơn giản hơn nhiều so với các mô hình học sâu tân tiên, đồng thời đạt được độ chính xác cao. Mask R-CNN được sử dụng để rút trích ngữ nghĩa. Bản chất của Mask R-CNN là sử dụng phương pháp phát hiện khung bao từ phương pháp Faster R-CNN. Do đó nếu áp dụng phương pháp YALA/YADA như đã trình bày ở các chương trước, chúng tôi kỳ vọng sẽ đạt được kết quả cao hơn nữa trong việc phát hiện đối tượng chính trong ảnh.

Chương 6
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Đóng góp của luận án

Trong luận án này, chúng tôi tập trung vào việc phát hiện các đối tượng khó trong ảnh với mục tiêu cải tiến những phương pháp phát hiện đối tượng hiện có. Cụ thể, chúng tôi trình bày hai đề xuất hiệu quả cho bài toán phát hiện đối tượng khó và áp dụng kết quả vào bài toán phát hiện đối tượng chính:

- Phương pháp YALA: tập trung vào các đối tượng thách thức (khó phát hiện được dựa vào các phương pháp tân tiến hiện nay, cụ thể là Faster R-CNN), phương pháp của chúng tôi sử dụng hai giai đoạn phát hiện dựa trên mạng học sâu kết hợp với kỹ thuật phát hiện đa tỉ lệ (*multi-scale*) và thuật toán hồi qui nhằm tinh chỉnh vị trí phát hiện đối tượng (*bounding box regression*). Các thử nghiệm cho thấy phương pháp được đề xuất tăng cường đáng kể khả năng phát hiện đối tượng của các mạng học sâu. Phương pháp đề xuất có khả năng phát hiện các đối tượng bị bỏ sót ở giai đoạn đầu và cho độ chính xác tốt hơn trên tất cả các tập dữ liệu thử nghiệm. Nguyên lý của phương pháp đề xuất cũng có thể áp dụng được cho các nghiên cứu khác về phát hiện đối tượng.

- Phương pháp YADA: đóng góp của chúng tôi bao gồm hai điểm chính: đầu tiên, chúng tôi trình bày một phương pháp phát sinh dữ liệu nhân tạo nhằm định hướng cho bộ phát hiện (có thể gọi là *"lucid data synthesizing"*). Phương pháp này được thực hiện thông qua khám phá các đối tượng khó trong tập huấn luyện, sau đó sao chép các đối tượng này vào các vị trí khác nhau trong các ảnh tương tự. Khác với các phương pháp đề xuất trong việc làm giàu dữ liệu trước đó, phương pháp của chúng tôi tạo ra dữ liệu trung thực và được định hướng với các tiêu chí rõ ràng. Từ đó tăng cường độ chính xác cho bộ phát hiện đối tượng với các đối tượng khó. Thứ hai, chúng tôi đề xuất sử dụng hai giai đoạn phát hiện dựa trên mạng học

sâu, trong đó giai đoạn hai được huấn luyện dựa trên dữ liệu nhân tạo. Nhờ đó khai thác được tốt hơn các ưu điểm của mô hình huấn luyện trên dữ liệu nhân tạo. Tập đối tượng phát hiện ở hai giai đoạn được thông qua một kỹ thuật kết hợp hiệu quả. Các thử nghiệm trên các tập dữ liệu thông dụng hiện nay bao gồm PASCAL VOC, KITTI, và COCO cho thấy tính hiệu quả của các phương pháp đề xuất so với các phương pháp tân tiến trong bài toán phát hiện đối tượng hiện nay.

- Ngoài ra, chúng tôi cũng đã áp dụng kết quả của việc phát hiện đối tượng vào trong bài toán phát hiện đối tượng chính. Kết quả thực nghiệm tốt trên các tập dữ liệu khác nhau như đã trình bày trong Chương 5 đã cho thấy đây là một hướng áp dụng đúng.

6.2 Ưu điểm và khuyết điểm của các phương pháp đề xuất

- Ưu điểm: Đối với phương pháp YALA, cách tiếp cận của chúng tôi cho phép phát hiện ra nhiều đối tượng khó mà phương pháp cơ sở không phát hiện được. Đối với YADA, chúng tôi tiếp tục phát triển YALA với tầng phát hiện thứ hai được huấn luyện dựa trên dữ liệu phát sinh nhân tạo, từ đó tiếp tục tăng cường độ chính xác của bộ phát hiện đối với các đối tượng khó. Các thử nghiệm trên các tập dữ liệu chuẩn cho thấy độ chính xác cao của YALA/YADA so với các cách tiếp cận tân tiến.
- Khuyết điểm: Khuyết điểm của cả hai phương pháp đề xuất là về mặt tốc độ xử lý. Chúng tôi sử dụng hai giai đoạn phát hiện đối tượng, do đó chi phí của thuật toán cơ sở sẽ bị nhân đôi.

6.3 Hướng phát triển

Điểm yếu của các phương pháp đề xuất bao gồm YALA và YADA là chi phí tính toán lớn và quá trình huấn luyện trải qua nhiều giai đoạn. Do đó, các phương pháp này có thể được cải tiến theo hướng tích hợp các bước này vào một kiến trúc mạng thống nhất, hỗ trợ quá trình huấn luyện theo cơ chế liền mạch. Đồng thời với đó là việc rút trích đặc trưng (bản đồ đặc trưng) có thể được chia sẻ ở các bộ phát hiện để giảm thiểu chi phí tính toán.

CÁC CÔNG TRÌNH ĐÃ CÔNG BỐ CỦA TÁC GIẢ

1. Các công bố chính

Tạp chí chuyên ngành

[CT.1] **Khanh-Duy Nguyen**, Khang Nguyen, Duy-Dinh Le, Duc Anh Duong, and Tam V. Nguyen. You always look again: Learning to detect the unseen objects. Journal of Visual Communication and Image Representation, vol. 60, pp. 206-216, 2019. **(ISI, Q1, Impact Factor: 2.259)**.

[CT.2] **Khanh-Duy Nguyen**, Khang Nguyen, Duy-Dinh Le, Duc Anh Duong, and Tam V. Nguyen. YADA: You Always Dream Again for Better Object Detection. Multimedia Tools and Applications, vol. 78, no. 19, pp. 28189–28208, 2019. **(ISI, Q1, Impact Factor: 2.101)**.

[CT.3] Tam V. Nguyen, **Khanh Nguyen**, and Toan Do. Semantic Prior Analysis for Salient Object Detection. IEEE Transactions on Image Processing, vol. 28, no. 6, pp. 3130-3141, 2019. **(ISI, Q1, Impact Factor: 6.79)**.

2. Các công bố khác có liên quan đến luận án

Tạp chí chuyên ngành

[CT.4] Khang Nguyen, Nhut T. Huynh, Phat C. Nguyen, **Khanh-Duy Nguyen**, Nguyen D. Vo, and Tam V. Nguyen. Detecting Objects from Space: An Evaluation of Deep-Learning Modern Approaches. Electronics, vol. 9, no. 4, pp. 583:1-18, 2020. **(ISI, Q2, Impact Factor: 1.764)**.

Hội thảo quốc tế

[CT.5] **Khanh-Duy Nguyen**, Duy-Dinh Le, and Duc Anh Duong. Efficient traffic sign detection using bag of visual words and multi-scales SIFT. International Conference on Neural Information Processing (ICONIP), pp. 433-441. Springer, Berlin, Heidelberg, 2013. **(ERA conference ranking: A)**.

[CT.6] **Khanh Nguyen**, and Ngo Duc Thanh. Scene text detection based on structural features. In Computer, Control, Informatics and its Applications (IC3INA), 2016 International Conference on, pp. 48-53. IEEE, 2016.

[CT.7] Nguyen D. Vo, **Khanh Nguyen**, Tam V. Nguyen, and Khang Nguyen. Ensemble of Deep Object Detectors for Page Object Detection. In Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication, p. 11. ACM, 2018.

Hội thảo trong nước

[CT.8] Nguyen D. Vo, **Khanh Nguyen**, Tam V. Nguyen, Khang Nguyen. Evaluation of State-of-the-art Object Detection Methods for Document Image Understanding. Kỷ yếu Hội nghị khoa học quốc gia lần thứ 10: Nghiên cứu cơ bản và ứng dụng công nghệ thông tin (FAIR'10), 2017.

Tài liệu tham khảo

[1] Umar Asif, Mohammed Bennamoun, and Ferdous A Sohel. Rgb-d object recognition and grasp detection using hierarchical cascaded forests. *IEEE Transactions on Robotics*, 33(3):547–564, 2017.

[2] Zhuoling Li, Minghui Dong, Shiping Wen, Xiang Hu, Pan Zhou, and Zhigang Zeng. Clu-cnns: Object detection for medical images. *Neurocomputing*, 350:53–59, 2019.

[3] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 2019.

[4] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *arXiv preprint arXiv:1809.02165*, 2018.

[5] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017.

[6] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Proceedings of Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[7] Manolis Loukadakis, José Cano, and Michael O'Boyle. Accelerating deep neural networks on low power heterogeneous architectures. 2018.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint*, 2017.

[10] Paul A. Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[11] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pages 886–893, 2005.

[12] Pedro F. Felzenszwalb, David A. McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[13] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[14] Ross B. Girshick. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[17] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *14th European Conference on Computer Vision, ECCV 2016*. Springer Verlag, 2016.

[18] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. A survey of deep learning-based object detection. *arXiv preprint arXiv:1907.09408*, 2019.

[19] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019.

[20] Shivang Agarwal, Jean Ogier Du Terrail, and Frédéric Jurie. Recent advances in object detection in the age of deep convolutional neural networks. *arXiv preprint arXiv:1809.03193*, 2018.

[21] Junwei Han, Dingwen Zhang, Gong Cheng, Nian Liu, and Dong Xu. Advanced deep-learning techniques for salient and category-specific object detection: a survey. *IEEE Signal Processing Magazine*, 35(1):84–100, 2018.

[22] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[23] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[24] Hedi Harzallah, Frédéric Jurie, and Cordelia Schmid. Combining efficient object localization and image classification. In *IEEE 12th International Conference on Computer Vision*, pages 237–244, 2009.

[25] Zheng Song, Qiang Chen, ZhongYang Huang, Yang Hua, and Shuicheng Yan. Contextualizing object detection and classification. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition*, pages 1585–1592, 2011.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1106–1114, 2012.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[29] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[30] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[31] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[33] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3, 2017.

[34] Zhi-Hua Zhou and Ji Feng. Deep forest: Towards an alternative to deep neural networks. *arXiv preprint arXiv:1702.08835*, 2017.

[35] David Rolnick and Max Tegmark. The power of deeper networks for expressing natural functions. *arXiv preprint arXiv:1705.05502*, 2017.

[36] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[37] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[38] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pages 346–361. Springer, 2014.

[40] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. Subcategory-aware convolutional neural networks for object proposals and detection. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 924–933. IEEE, 2017.

[41] Yao Ding, Yanzhao Zhou, Yi Zhu, Qixiang Ye, and Jianbin Jiao. Selective sparse sampling for fine-grained image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6599–6608, 2019.

[42] Sanja Fidler, Roozbeh Mottaghi, Alan Yuille, and Raquel Urtasun. Bottom-up segmentation for top-down detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3294–3301, 2013.

[43] Yukun Zhu, Raquel Urtasun, Ruslan Salakhutdinov, and Sanja Fidler. segdeepm: Exploiting segmentation and context in deep neural networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4703–4711, 2015.

[44] Spyros Gidaris and Nikos Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1134–1142, 2015.

[45] S. Zagoruyko, A. Lerer, T.-Y. Lin, P. O. Pinheiro, S. Gross, S. Chintala, and P. Dollár. A multipath network for object detection. In *BMVC*, 2016.

[46] Tianfu Wu, Bo Li, and Song-Chun Zhu. Learning and-or model to represent context and occlusion for car detection and viewpoint estimation. *IEEE transactions on pattern analysis and machine intelligence*, 38(9): 1829–1843, 2016.

[47] Sean Bell, C Lawrence Zitnick, Kavita Bala, and Ross Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2874–2883, 2016.

[48] Liliang Zhang, Liang Lin, Xiaodan Liang, and Kaiming He. Is faster r-cnn doing well for pedestrian detection? In *European Conference on Computer Vision*, pages 443–457. Springer, 2016.

[49] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, and Francesc Moreno-Noguer. Fracking deep convolutional image descriptors. *arXiv preprint arXiv:1412.6537*, 2014.

[50] Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. *arXiv preprint arXiv:1511.06343*, 2015.

[51] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 761–769, 2016.

[52] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. *arXiv preprint arXiv:1805.00932*, 2018.

[53] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016.

[54] Ankur Handa, Viorica Pătrăucean, Simon Stent, and Roberto Cipolla. Scenenet: An annotated model generator for indoor scene understanding. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 5737–5743. IEEE, 2016.

[55] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016.

[56] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, pages 4627–4635. IEEE, 2017.

[57] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation.

In *European Conference on Computer Vision*, pages 345–360. Springer, 2014.

[58] Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. Learning deep object detectors from 3d models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1278–1286, 2015.

[59] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. *arXiv preprint arXiv:1804.06516*, 2018.

[60] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 746–753. IEEE, 2017.

[61] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[62] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361, 2012.

[63] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1301–1310, 2017.

[64] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[65] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, pages 740–755, 2014.

[66] Qiang Chen, Zheng Song, Yang Hua, ZhongYang Huang, and Shuicheng Yan. Hierarchical matching with side information for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3426–3433, 2012.

[67] Tam V. Nguyen, Zheng Song, and Shuicheng Yan. STAP: Spatial-Temporal Attention-Aware Pooling for Action Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(1):77–86, 2015.

[68] Tam V. Nguyen, Bingbing Ni, Hairong Liu, Wei Xia, Jiebo Luo, Mohan S. Kankanhalli, and Shuicheng Yan. Image re-attentionizing. *IEEE Transactions on Multimedia*, 15(8):1910–1919, 2013.

[69] Tao Mei, Lusong Li, Xinmei Tian, Dacheng Tao, and Chong-Wah Ngo. Pagesense: Toward stylewise contextual advertising via visual analysis of web pages. *IEEE Trans. Circuits Syst. Video Techn.*, 28(1):254–266, 2018.

[70] Christoph H Lampert, Matthew B Blaschko, and Thomas Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):2129–2142, 2009.

[71] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2147–2154, 2014.

[72] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[73] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and

detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[74] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[75] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[76] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.

[77] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[78] Mark Everingham, Luc J. Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[79] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678, 2014.

[80] Long Zhu, Yuanhao Chen, Alan Yuille, and William Freeman. Latent hierarchical structural learning for object detection. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1062–1069. IEEE, 2010.

[81] Hedi Harzallah, Frédéric Jurie, and Cordelia Schmid. Combining efficient object localization and image classification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 237–244. IEEE, 2009.

[82] Qiang Chen, Zheng Song, Jian Dong, Zhongyang Huang, Yang Hua, and Shuicheng Yan. Contextualizing object detection and classification. *IEEE transactions on pattern analysis and machine intelligence*, 37(1):13–27, 2015.

[83] Wanli Ouyang, Kun Wang, Xin Zhu, and Xiaogang Wang. Chained cascade network for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1938–1946, 2017.

[84] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: delving into high quality object detection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018.

[85] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[86] Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: Efficient multi-scale training. In *Advances in Neural Information Processing Systems*, pages 9310–9320, 2018.

[87] Tam V. Nguyen, Qi Zhao, and Shuicheng Yan. Attentive systems: A survey. *International Journal of Computer Vision*, 126(1):86–110, 2018.

[88] Radhakrishna Achanta, Sheila S. Hemami, Francisco J. Estrada, and Sabine Süsstrunk. Frequency-tuned salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1597–1604, 2009.

[89] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):569–582, 2015.

[90] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 733–740, 2012.

[91] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2376–2383, 2010.

[92] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2083–2090, 2013.

[93] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(2):353–367, 2011.

[94] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1155–1162, 2013.

[95] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended CSSD. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(4):717–729, 2016.

[96] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5455–5463, 2015.

[97] Ce Liu, Jenny Yuen, and Antonio Torralba. Nonparametric scene parsing via label transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2368–2382, 2011.

[98] Joseph Tighe and Svetlana Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3001–3008, 2013.

[99] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Deep networks for saliency detection via local estimation and global search. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3183–3192, 2015.

[100] Xi Li, Liming Zhao, Lina Wei, Ming-Hsuan Yang, Fei Wu, Yueting Zhuang, Haibin Ling, and Jingdong Wang. Deepsaliency: Multi-task deep neural network model for salient object detection. *IEEE Transactions on Image Processing*, 25(8):3919–3930, 2016.

[101] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[102] Thomas K. Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.

[103] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *International Conference on Computer Vision*, pages 2106–2113, 2009.

[104] Congyan Lang, Tam V. Nguyen, Harish Katti, Karthik Yadati, Mohan S. Kankanhalli, and Shuicheng Yan. Depth matters: Influence of depth cues on visual saliency. In *European Conference on Computer Vision*, pages 101–115, 2012.

[105] Eduardo Simoes Lopes Gastal and Manuel M. Oliveira. Domain transform for edge-aware image and video processing. *ACM Trans. Graph.*, 30(4): 69:1–69:12, 2011.

[106] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002.

[107] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. In *International Conference on Computer Vision*, pages 1529–1537, 2015.

[108] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

[109] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3169–3176, 2010.

[110] Tam V. Nguyen and Jose Sepulveda. Salient object detection via augmented hypotheses. In *International Joint Conference on Artificial Intelligence*, pages 2176–2182, 2015.

[111] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *IEEE International Conference on Computer Vision*, pages 202–211, 2017.

[112] Jianming Zhang and Stan Sclaroff. Exploiting surroundedness for saliency detection: A boolean map approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(5):889–902, 2016.

[113] Ping Hu, Bing Shuai, Jun Liu, and Gang Wang. Deep level sets for salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 540–549, 2017.

[114] Gayoung Lee, Yu-Wing Tai, and Junmo Kim. Deep saliency with encoded low level distance map and high level features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 660–668, 2016.

[115] Ming-Ming Cheng, Guo-Xin Zhang, Niloy J. Mitra, Xiaolei Huang, and Shi-Min Hu. Global contrast based salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 409–416, 2011.

[116] Jiwhan Kim, Dongyoon Han, Yu-Wing Tai, and Junmo Kim. Salient region detection via high-dimensional color transform. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 883–890, 2014.

[117] Tiantian Wang, Lihe Zhang, Huchuan Lu, Chong Sun, and Jinqing Qi. Kernelized subspace ranking for saliency detection. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, pages 450–466, 2016.

[118] Yun Zhai and Mubarak Shah. Visual attention detection in video sequences using spatiotemporal cues. In *ACM Multimedia*, pages 815–824, 2006.

[119] Zhiming Luo, Akshaya Kumar Mishra, Andrew Achkar, Justin A. Eichel, Shaozi Li, and Pierre-Marc Jodoin. Non-local deep features for salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6593–6601, 2017.

[120] Ran Margolin, Ayellet Tal, and Lihi Zelnik-Manor. What makes a patch distinct? In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1139–1146, 2013.

[121] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Saliency detection with recurrent fully convolutional networks. In *European Conference on Computer Vision*, pages 825–841, 2016.

[122] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Baocai Yin. Learning uncertain convolutional features for accurate saliency detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 212–221, 2017.

[123] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip H. S. Torr. Deeply supervised salient object detection with short connections. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5300–5309, 2017.

[124] Olivier Le Meur and Zhi Liu. Saliency aggregation: Does unity make strength? In *Asian Conference on Computer Vision*, pages 18–32, 2014.

[125] Xuanyi Dong, Liang Zheng, Fan Ma, Yi Yang, and Deyu Meng. Few-example object detection with model communication. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1641–1654, 2018.