PHÁT HIỆN ĐỐI TƯỢNG DỰA VÀO HỌC SÂU TRÊN RASPBERRY PI

Lê Quang Chiến

Khoa Công nghệ Thông tin, Trường Đại học Khoa học, Đại học Huế

Email: lqchien@hueuni.edu.vn

Ngày nhận bài: 13/3/2021; ngày hoàn thành phản biện: 6/7/2021; ngày duyệt đăng: 4/4/2022

TÓM TẮT

Với sự phát triển gần đây của lĩnh vực học sâu, các phương pháp phát hiện đối tượng đã đạt được hiệu suất cao trên cả tốc độ và độ chính xác trên các hệ thống máy tính để bàn hiện đại. Bên cạnh đó, việc phát triển các mô hình học sâu nhỏ hơn và nhanh hơn để phù hợp với các thiết bị IoT đang thu hút được nhiều sự quan tâm. Bài báo này tìm hiểu sự phù hợp của các mô hình phát hiện đối tượng trên Raspberry Pi, một bo mạch máy tính nhúng phổ biến có thể được tích hợp vào các hệ thống IoT để giúp công việc trở nên dễ dàng. Chúng tôi tiến hành khảo sát ảnh hưởng của hai mô hình phát hiện đối tượng hiện đại là Single Shot Detector (SSD) và You Only Look Once (YOLO). Hai mô hình này sẽ được đánh giá dựa trên tốc độ xử lý khung hình và độ chính xác trung bình khi thực hiện suy luận. Các kết quả thí nghiệm cho thấy tính khả thi của các mô hình này khi được sử dụng trên các thiết bị máy tính cấu hình thấp.

Từ khóa: Phát hiện đối tượng, YOLO, SSD, Raspberry Pi.

1. MỞ ĐẦU

Phát hiện đối tượng có lẽ là nhiệm vụ quan trọng nhất trong các hệ thống giám sát. Mục tiêu của nhiệm vụ này là phát hiện sự hiện diện của đối tượng từ một tập các lớp nhất định và xác định vị trí chính xác trong một hình ảnh. Trước khi AlexNet [1], một mạng nơron tích chập (CNN), được giới thiệu, đây được xem là một vấn đề khó giải quyết đối với các nhà nghiên cứu trong việc tìm giải pháp phân loại hình ảnh với tỷ lệ lỗi rất thấp. Từ cột mốc này, nhiều phương pháp phát hiện đối tượng áp dụng CNN đã được trình bày cho thấy hiệu suất và hiệu quả tuyệt vời. Tuy nhiên, để thực thi các tác vụ sử dụng CNN một cách hiệu quả, chúng ta vẫn cần nhiều sức mạnh tính toán. Do vậy, việc chạy một hệ thống phát hiện đối tượng trên một thiết bị có tài nguyên phần cứng hạn chế có thể là một thách thức.

Raspberry Pi [2] là một loại máy tính cỡ nhỏ với kích thước không lớn hơn một thẻ tín dụng. Loại máy tính này thiếu sức mạnh tính toán như của các hệ thống máy tính

để bàn truyền thống. Tuy nhiên, do kích thước và chi phí thấp, chúng có thể được sử dụng cho các tác vụ nhất định.

Hiện tại, có nhiều phương pháp khác nhau có thể được sử dụng để phát hiện các đối tượng. Tuy nhiên, không có phương pháp nào là lựa chọn tối ưu để sử dụng trong các hệ thống phát hiện đối tượng. Các phương pháp được đề xuất đều tập trung vào việc đạt được độ chính xác cao mà không cần xem xét những giới hạn về thời gian chạy hoặc phần cứng. Trong các ứng dụng thực tế, việc cân nhắc đến các yếu tố trên sẽ khiến cho việc tìm ra một phương pháp phù hợp để sử dụng là rất khó.

Gần đây, hai phương pháp SSD [3], YOLO [4] được giới thiệu như những phương pháp đạt được hiệu quả tốt nhất trên độ chính xác phát hiện và tốc độ xử lý. Trong nghiên cứu này, chúng tôi triển khai hai phương pháp nêu trên với Raspberry Pi 3 để đánh giá xem sự phù hợp khi chạy trên phần cứng hiệu suất thấp. Một bộ phát hiện đối tượng được thực hiện được coi là phù hợp nếu nó đạt được độ chính xác và tốc độ khung hình đủ cao để có thể triển khai trong các ứng dụng thực tế.

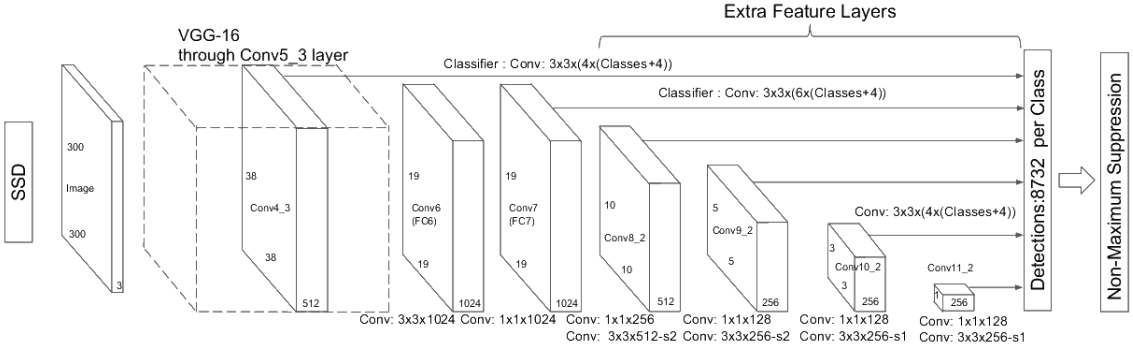
2. PHƯƠNG PHÁP NGHIÊN CỨU

Phát hiện đối tượng là một trong những vấn đề kinh điển trong thị giác máy tính với mục tiêu là để nhận ra cái gì và ở đâu. Tức là, chúng ta phải xác định những vật thể nào bên trong một hình ảnh và vị trí của chúng ở trong hình ảnh đó. Do đó, một hệ thống phát hiện đối tượng phải bao gồm quá trình phân loại và định vị không chỉ một đối tượng trong ảnh mà còn mọi đối tượng được tham chiếu. Đây là một nhiệm vụ khó khăn hơn nhiều so với phân loại hình ảnh truyền thống. Trong phần này, chúng tôi trình bày một cách tổng quan hai mô hình phát hiện đối tượng hiện đại: SSD và YOLO. Đây là hai mô hình được đề xuất dựa trên các CNN để tạo thành một hệ thống phát hiện đối tượng end-to-end.

2.1. Mô hình SSD

SSD [3] là mô hình được thiết kế để phát hiện đối tượng trong thời gian thực. Bài toán phát hiện đối tượng đánh dấu sự đột phá với mô hình R-CNN và các cải tiến của nó như: Fast R-CNN và Faster R-CNN. Các mô hình này thực thi qua hai giai đoạn: (i) giai đoạn đầu tiên là sử dụng mạng đề xuất vùng (Region Proposal Network) để tạo ra các vùng ứng cử viên (proposals); (ii) giai đoạn thứ hai là phân loại các proposals này bởi một bộ phân lớp mạnh. Mặc dù, mô hình Faster R-CNN đạt được độ chính xác cao, nhưng tốc độ thực thị của toàn bộ quá trình thấp hơn nhiều so với yêu cầu về xử lý theo thời gian thực. Trong khi đó, mô hình SSD chỉ gồm một giai đoạn duy nhất. Kiến trúc của SSD chỉ bao gồm một mạng noron cho toàn bộ quá trình phát hiện đối tượng (xem Hình 1). Trong kiến trúc này, mạng RPN được loại bỏ hoàn toàn. Để gia tăng độ chính xác, SSD áp dụng một vài cải tiến bao gồm các đặc trưng đa tỷ lệ (multi-scale features)

và các hộp mặc định (default boxes). Những cải tiến này cho phép SSD đạt được độ chính xác tương đương với Faster R-CNN (thậm chí là cao hơn) khi sử dụng các hình ảnh có độ phân giải thấp hơn, giúp nâng cao tốc độ hơn.



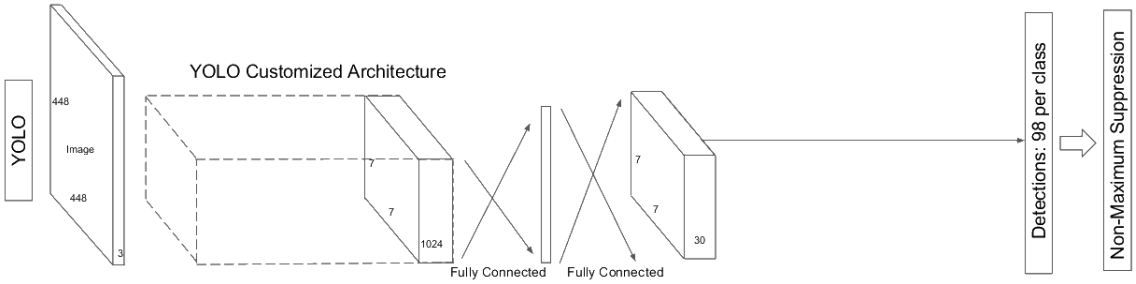
Hình 1. Kiến trúc của mô hình SSD.

Kiến trúc tổng quát của SSD bao gồm hai phần: (i) trích xuất feature maps, và (ii) áp dụng các bộ lọc tích chập (convolutional filters) để phát hiện đối tượng. Giống như các mô hình R-CNN, SSD cũng sử dụng các mạng CNN để trích xuất đặc trưng. Trong kiến trúc được mô tả ở Hình 1, mạng VGG-16 [5] được sử dụng để thực hiện nhiệm vụ này. Các đặc trưng này, sau đó, sẽ được dùng để tính toán vị trí và độ tin cậy của đối tượng được phân lớp. Đặc biệt, trong kiến trúc của SSD, đầu ra của các lớp Convolutional sẽ được sử dụng để đóng góp vào quá trình tính toán cuối cùng.

2.2. Mô hình YOLO

Tương tự như SSD [3], YOLO [4] cũng là một kiểu kiến trúc end-to-end, bao gồm cả hai quá trình trích chọn đặc trưng và suy luận (phân loại và định vị). YOLO cũng là một trong những mô hình đạt được độ chính xác cao đồng thời cũng có thể thực thi suy luận trong thời gian thực. Mô hình này chỉ cần một lần truyền thông tin chuyển tiếp qua mạng nơron để đưa ra dự đoán. Sau đó, một thuật toán Non-Maximum Suppression sẽ được áp dụng để giữ lại một dự đoán tốt nhất trên mỗi đối tượng được phát hiện.

Khác với kiến trúc của SSD, YOLO chỉ sử dụng thông tin từ lớp Convolutional cuối cùng cho nhiệm vụ phát hiện và phân lớp đối tượng (xem Hình 2) để tăng tốc độ xử lý. YOLO cũng đề xuất nhiều cải tiến như: (i) sử dụng bộ trích xuất đặc trưng tốt hơn; (ii) bổ sung Feature Pyramid để phát hiện các đối tượng nhỏ tốt hơn.



Hình 2. Kiến trúc của mô hình YOLO.

3. KẾT QUẢ VÀ THẢO LUẬN

Trong phần này, chúng tôi trình bày các thiết lập thí nghiệm để thực hiện đánh giá hai mô hình SSD và YOLO. Bên cạnh đó, chúng tôi cũng báo cáo các kết quả thí nghiệm đã tiến hành. Cuối cùng, chúng tôi phân tích, thảo luận dựa trên các kết quả đã báo cáo để đánh giá vai trò của hai mô hình trong việc triển khai trên Raspberry Pi 3.

3.1. Các thiết lập thí nghiệm

Chúng tôi thực hiện đánh giá SSD và YOLO thông qua hai mô hình đã được huấn luyện sẵn: MobileNetV2-SSDLite [7] và YOLOv3-tiny [8]. Hai mô hình này đều được huấn luyện cùng bộ dữ liệu COCO [6].

Bên cạnh đó, do SSD và YOLO đều là các CNN đầy đủ nên cả hai mô hình đều có thể thực hiện suy luận trên các hình ảnh ở nhiều kích thước khác nhau. Do vậy, chúng tôi sử dụng kích thước ảnh đầu vào như một tham số để đánh giá tốc độ và độ chính xác phát hiện. Trong các thí nghiệm, chúng tôi đánh giá trên ba kích thước khác nhau của ảnh đầu vào, lần lượt là: 96x96, 160x160, 224x224.

Chúng tôi cung cấp các đánh giá liên quan dựa trên độ chính xác và tốc độ xử lý của SSD và YOLO. Các kết quả được báo cáo trong bài báo này đều được đánh giá trên Raspberry Pi 3. Đánh giá về độ chính xác được khảo sát dựa trên các công trình đã được công bố. Riêng đánh giá về tốc độ, chúng tôi thực hiện thí nghiệm trên 1000 khung hình với các kích cỡ đầu vào khác nhau.

3.2. Các kết quả thí nghiệm

3.2.1. Tốc độ xử lý

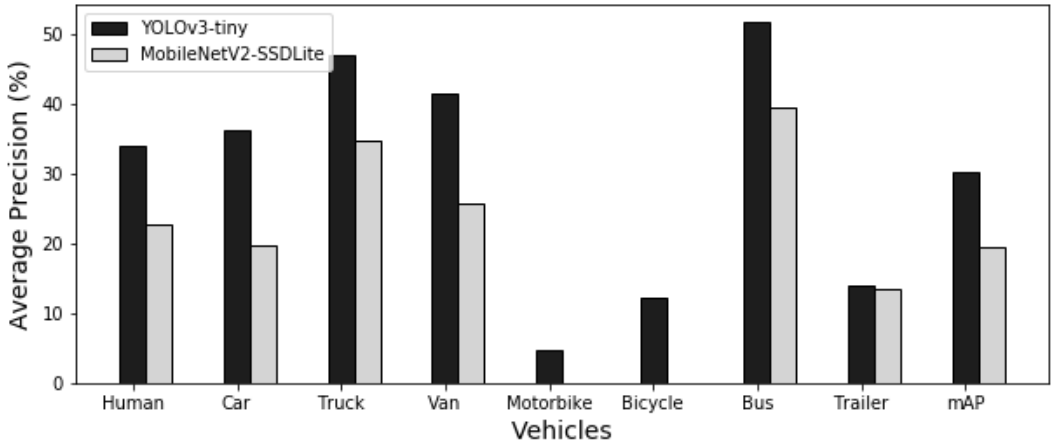
Bảng 1 thể hiện các kết quả các kết quả thí nghiệm khi đánh giá tốc độ xử lý của SSD và YOLO. Tốc độ xử lý được tính bằng số lượng khung hình được xử lý trong một giây. Các kết quả cũng được báo cáo với các kích cỡ đầu vào khác nhau từ 1000 khung hình được quay sẵn.

Bảng 1. Tốc độ xử lý của SSD và YOLO với kích cỡ đầu vào khác nhau

Kích cỡ đầu vào	Mô hình	FPS
96 x 96	SSD	4.41
	YOLO	2.71
160 x 160	SSD	2.66
	YOLO	1.92
224 x 224	SSD	1.73
	YOLO	1.03

3.2.2. Độ chính xác

Hình 3 thể hiện độ chính xác trung bình của hai mô hình YOLO và SSD khi được thực nghiệm trên tập dữ liệu AU-AIR [9]. Đây là tập dữ liệu được ghi hình bởi các cảm biến gắn trên một thiết bị bay không người lái. Tập dữ liệu này gồm nhiều loại dữ liệu khác nhau, như: hình ảnh, thời gian, GPS, IMU, vận tốc. Mục đích của tập dữ liệu này là để phục vụ cho các hệ thống giám sát giao thông từ trên không.



Hình 3. Độ chính xác trung bình của hai mô hình trên tập dữ liệu AU-AIR [9].

3.3. Phân tích và thảo luận

Các kết quả được báo cáo trong thí nghiệm thể hiện tốc độ xử lý của hai mô hình SSD và YOLO với các kích cỡ đầu vào khác nhau. Kết quả thí nghiệm cho thấy rằng bằng cách sử dụng kích thước đầu vào nhỏ hơn, chúng ta có thể đạt được tốc độ xử lý cao hơn. Tất nhiên, chúng ta phải đánh đổi với sự suy giảm của độ chính xác trong dự đoán đối tượng. Trong Bảng 1, chúng ta thấy rằng SSD có tốc độ xử lý nhanh hơn so với YOLO. Khi kích thước đầu vào là 96x96, SSD đạt tới 4,41 fps trong khi YOLO chỉ đạt 2,71 fps, nhanh hơn một chút so với SSD tại kích thước đầu vào 160x160. Cả hai mô hình đều có tốc độ xử lý khá chậm khi kích thước đầu vào là 224x224, YOLO chỉ xử lý ở tốc độ 1,03 fps và SSD xử lý ở tốc độ 1,73 fps.

Trong báo cáo thí nghiệm về độ chính xác dự đoán, YOLO cho kết quả dự đoán tốt hơn đáng kể so với SSD. Với nhiệm vụ phát hiện người, YOLO có thể dự đoán với độ chính xác 34,05%, trong khi SSD chỉ đạt 22,86%. Đặc biệt với nhiệm vụ phát hiện xe đạp hay xe gắn máy, SSD gần như là thất bại trong hầu hết các mẫu kiểm thử, chỉ đạt độ chính xác 0,01% ở cả hai nhiệm vụ này, trong khi YOLO đạt được độ chính xác lần lượt là 12,34% và 4,80%. Kết quả dự đoán trung bình cũng thể hiện rõ ràng sự khác biệt đáng kể giữa YOLO (đạt 30,22%) và SSD (đạt 19,50%).

Các báo cáo thí nghiệm cho thấy việc thực thi nhiệm vụ phát hiện đối tượng trên các thiết bị nhúng như Raspberry Pi 3 là khá chậm khi kích thước đầu vào tăng lên. Bên

cạnh đó, để đảm bảo được độ chính xác phù hợp với các bài toán thực tế, việc sử dụng kích thước đầu vào nhỏ là không hợp lý nhất là các nhiệm vụ liên quan đến hoạt động giám sát. Tuy nhiên, chúng ta cần nhận thức rằng các ứng dụng khác nhau có các yêu cầu khác nhau về tốc độ và độ chính xác. Do vậy, khi triển khai nhiệm vụ giám sát trên thiết bị cấp thấp, chúng ta cần phải cân bằng tốc độ và độ chính xác.

Khi đối tượng được yêu cầu giám sát là con người, thì các mô hình YOLO hay SSD sẽ dễ dàng phát hiện ngay cả ở khoảng cách xa do bởi kích thước lớn và hình dạng đặc thù. Do vậy, việc sử dụng các mô hình này ở tốc độ cao là không cần thiết. Chúng ta có thể triển khai với kích thước đầu vào là 224x224 hoặc lớn hơn để phục vụ cho mục đích giám sát. Trong trường hợp này, mô hình YOLO sẽ được ưu tiên lựa chọn do có độ chính xác cao hơn đáng kể. Bên cạnh đó, chúng ta có thể tăng tốc độ xử lý thông qua việc sử dụng các kỹ thuật tiền xử lý như trừ nền, hoặc các kỹ thuật theo dấu cũng như nâng cấp phần cứng và phụ thuộc yêu cầu của bài toán mà chúng ta có thể có những tùy chọn thích hợp.

4. KẾT LUẬN

Mục đích của nghiên cứu này là đánh giá sự phù hợp của việc áp dụng hệ thống phát hiện đối tượng thời gian thực trên máy tính có cấu hình thấp như Raspberry Pi. Hai mô hình SSD và YOLO đã được lựa chọn để đánh giá trên độ chính xác và tốc độ xử lý. Kết quả cho thấy cả hai model đều không đáp ứng được tốc độ xử lý theo thời gian thực. Tuy nhiên, trong các ứng dụng không yêu cầu tốc độ cao thì mô hình YOLO có thể được sử dụng trong các máy tính nhúng như Raspberry Pi.

Việc đạt được sự cân bằng giữa tốc độ xử lý và độ chính xác phụ thuộc rất lớn vào yêu cầu của từng ứng dụng thực tế. Điều này dẫn đến việc lựa chọn kích thước đầu vào phù hợp là rất quan trọng để có được sự cân bằng cần thiết cho một ứng dụng cụ thể. Nghiên cứu này có thể giúp ích cho việc triển khai các hệ thống giám sát trên phần cứng tương tự để đạt được hiệu quả cao nhất.

TÀI LIỆU THAM KHẢO

[1]. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[2]. Pi, R. What is a Raspberry Pi? [Online]. Available from: https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/

[3]. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.

[4]. Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).

[5]. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[6]. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.

[7]. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).

[8]. Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[9]. Bozcan, I., & Kayacan, E. (2020). AU-AIR: A Multi-modal Unmanned Aerial Vehicle Dataset for Low Altitude Traffic Surveillance. IEEE International Conference on Robotics and Automation (ICRA).

DEEP LEARNING - BASED OBJECT DETECTION ON RASPBERRY PI

Le Quang Chien

Faculty of Information Technology, University of Sciences, Hue University

Email: lqchien@hueuni.edu.vn

ABSTRACT

With the recent development of deep learning, object detection methods have achieved high performance on both speed and accuracy on modern desktop systems. In addition, the development of smaller and faster deep learning models to fit IoT devices has been attracting much attention. This article explores the relevance of object detection models on the Raspberry Pi, a popular embedded computer that can be integrated into IoT systems to make work easy. We consider the effects of two modern object detection models, Single Shot Detector (SSD) and You Only Look Once (YOLO). The models will be evaluated based on the frame rate and the average precision in the phase of inference. The experimental results show the feasibility of the models on low-profile computers.

Keywords: object detection, YOLO, SSD, Raspberry Pi.

Lê Quang Chiến sinh ngày 15/09/1983 tại Thừa Thiên Huế. Năm 2005, ông tốt nghiệp cử nhân chuyên ngành Tin học tại trường Đại học Khoa học, Đại học Huế. Năm 2007, ông nhận bằng thạc sĩ chuyên ngành khoa học máy tính tại trường Đại học Khoa học, Đại học Huế. Năm 2016, ông nhận học vị tiến sĩ chuyên ngành Tin học tại trường SOKENDAI (The Graduate University for Advanced Studies), Nhật Bản. Hiện nay, ông đang công tác tại khoa Công nghệ Thông tin, trường Đại học Khoa học, Đại học Huế.

Lĩnh vực nghiên cứu: Xử lý và nhận dạng ảnh, xử lý video, học máy, thị giác máy tính.