TẠP CHÍ KHOA HỌC CÔNG NGHỆ HÀNG HẢI
JOURNAL OF MARINE SCIENCE AND TECHNOLOGY
ISSN: 1859-316X

XÂY DỰNG HỆ THỐNG PHÁT HIỆN PHƯƠNG TIỆN GIAO THÔNG SỬ DỤNG MÔ HÌNH HỌC SÂU YOLO3

BUILDING A VEHICLE DETECTION SYSTEM BY USING DEEP LEARNING MODEL YOLO3

NGUYỄN HỮU TUÂN*, NGUYỄN VĂN THUỶ

Khoa Công nghệ Thông tin, Trường Đại học Hàng hải Việt Nam
**Email liên hệ: huu-tuan.nguyen@vimaru.edu.vn*

Tóm tắt

Bài toán phát hiện phương tiện giao thông là một bài toán thuộc lĩnh vực thị giác máy tính có nhiều ứng dụng hữu ích trong các hệ thống xe tự hành, quản lý phương tiện giao thông và xác định lưu lượng giao thông tại các điểm, đường giao thông quan trọng. Có nhiều cách tiếp cận cho bài toán này, từ phương pháp trừ nền cho tới các phương pháp học sâu hiện đại. Trong bài báo này, nhóm tác giả tập trung vào việc ứng dụng mô hình học sâu YOLO3 (You Only Look Once version 3) để giải quyết bài toán. Một hệ thống demo cũng được xây dựng bằng cách sử dụng nền tảng Darknet-53 và thử nghiệm với các dữ liệu do nhóm tác giả tự thu thập. Kết quả cho thấy hệ thống xây dựng có độ chính xác cao và khả thi khi cần áp dụng cho các ứng dụng thực tế.

Từ khóa: *Phát hiện phương tiện giao thông, phát hiện xe ô tô, học sâu, mô hình YOLO3.*

Abstract

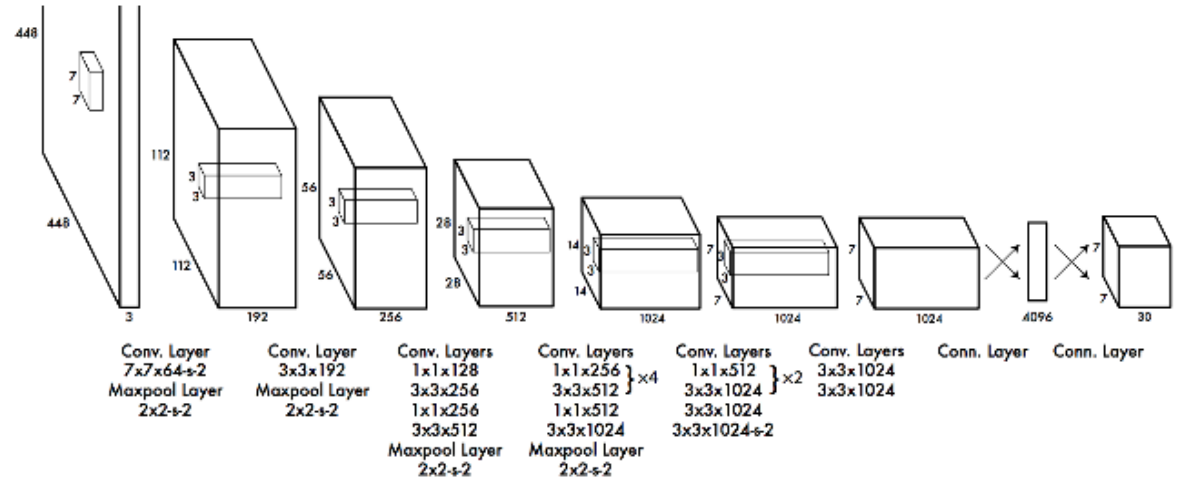
Vehicle detection is a computer vision problem that has many useful applications in automatic driving systems, transportation vehicles management and traffic flow at important intersections and roads. There are a lot of approaches for this problem, from background subtraction ones to modern deep learning methods. In this paper, authors focus on applying deep learning model YOLO3 (You Only Look Once version 3) to deal with the problem. A demo system is built based on Darknet-53 and tested with self-collected data. The obtained results show that our system gains high accuracy and is viable in real life situations.

Keywords: *Vehicles detection, car detection, deep learning, model YOLO3.*

1. Mở đầu

Phát hiện phương tiện giao thông là một bài toán thuộc nhóm các bài toán phát hiện đối tượng (một lĩnh vực con của ngành thị giác máy tính) và có nhiều ứng dụng trên thực tế. Bài toán này nhận được sự quan tâm của các nhà khoa học và các hãng sản xuất công nghiệp lớn nhằm phát triển các hệ thống lái tự động cũng như kiểm soát giao thông (xác định lưu lượng giao thông tại các điểm nút để điều chỉnh, phân luồng và quy hoạch hạ tầng giao thông). Có thể chia các phương pháp giải quyết bài toán phát hiện phương tiện giao thông thành 4 nhóm: các phương pháp dựa vào kỹ thuật trừ nền [1], các phương pháp dựa vào hiệu giữa các frame liên tiếp trong dữ liệu video [2], các phương pháp dựa vào luồng quang học (optical flow) [3] và các phương pháp dựa vào các mô hình mạng học sâu phát hiện đối tượng như YOLO [4], Retinanet [5], SSD [6] hay Fast R-CNN [7]. Các phương pháp thuộc các cách tiếp cận [1], [2] và [3] có ưu điểm là không cần nhiều dữ liệu huấn luyện hay năng lực xử lý mạnh mẽ của các hệ thống phần cứng, tốc độ nhanh nhưng hạn chế về độ chính xác. Các mô hình học sâu ([4], [5], [6] và [7]) có điểm chung là cần dữ liệu huấn luyện lớn, các phần cứng triển khai phải có năng lực xử lý mạnh mẽ (các card đồ hoạ GPU chuyên dụng) và tốc độ thực thi chậm hơn, nhưng lại có độ chính xác cao hơn.

Trong phạm vi của bài báo này, chúng tôi tập trung vào việc sử dụng mô hình mạng học sâu YOLO3 [8] để xây dựng một hệ thống phát hiện các phương tiện giao thông, chủ yếu là các loại phương tiện giao thông phổ biến (xe đạp, xe máy, xe con, xe tải và xe buýt), dựa trên nền tảng Darknet-53 và ngôn ngữ lập trình Python 3. Hệ thống được kiểm thử với các dữ liệu được download từ Internet (github.com) và dữ liệu thực tế do nhóm tác giả thu thập. Các phần tiếp theo của bài báo được cấu trúc như sau: trong phần 2 mô hình mạng YOLO3 sẽ được trình bày chi tiết, tiếp đến là phần cài đặt và kiểm thử hệ thống, cuối cùng là kết luận và một số đề xuất.



Hình 1. Mô hình mạng YOLO (còn gọi là YOLO1) *[4]*

2. Mô hình mạng học sâu YOLO3

2.1. Mô hình mạng học sâu YOLO

YOLO [4] (xem Hình 1) là một thuật toán phát hiện đối tượng được đề xuất năm 2015 nhằm mục đích có thể triển khai trong các ứng dụng thời gian thực (nhanh-Yolo3 [8] có tốc độ xử lý 28,2ms/1 ảnh, tức khoảng 35,5 FPS với ảnh 320x320) và là một trong các thuật toán hiệu quả nhất. So với bài toán phân lớp đối tượng (classification), phát hiện đối tượng phức tạp hơn vì phải trả lời hai câu hỏi: có loại đối tượng nào trong ảnh (bản chất ngang với bài toán phân lớp) và nếu có thì vị trí của các đối tượng đó ở đâu trong ảnh input. Để có tốc độ nhanh, YOLO được thiết kế với số lớp khá nhỏ so với các mô hình mạng CNNs (YOLO3 chỉ có 53 lớp nhân chập) khác.

Mô hình YOLO là một mạng nơ ron nhân chập thông minh với lớp input là ảnh đầu vào của hệ thống. Thuật toán sẽ chia ảnh input thành các vùng con và dự đoán các hình chữ nhật bao gói các đối tượng và các xác suất tương ứng cho mỗi vùng. Thuật toán YOLO được sử dụng trong nhiều bài toán có sử dụng bước phát hiện đối tượng vì hai nguyên nhân: độ chính xác cao và tốc độ theo thời gian thực. Cách làm việc của mô hình mạng YOLO chỉ sử dụng một phương pháp lan truyền thẳng trong mạng nơ ron mà nó sử dụng nên các tác giả đã đặt tên hệ thống là "*You Only Look Once*" với hàm ý là thuật toán này cũng giống như hệ thống thị giác của con người, chỉ cần nhìn một lần đã có thể đưa ra các dự đoán chính xác về các đối tượng trong khung hình được nhìn thấy.

2.2. Phiên bản YOLO3

Từ phiên bản đầu tiên năm 2015 [4], các kỹ thuật mới đã được áp dụng để cải thiện độ chính xác và thời gian thực hiện và mô hình YOLO3 (version 3) đã được đề xuất năm 2018 [8] (xem Hình 2).

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Hình 2. Kiến trúc mạng YOLO3 *[8]*

YOLO3 có dữ liệu input là một lô các ảnh có kích thước (m, 416, 416, 3) với kết quả đầu ra là một danh sách các hộp bao đối tượng (bounding box - bb) cùng với lớp nhận dạng. Mỗi bb được biểu diễn bởi 6 giá trị (pc, bx, by, bh, bw, c) trong đó pc là xác suất dự đoán, (bx, by) là toạ độ điểm phía trên bên trái của bb dự đoán, (bh, bw) là kích thước chiều cao và chiều rộng của bb, còn c là nhãn của đối tượng phát hiện được. Trong YOLO3, việc dự đoán được thực hiện bằng cách sử dụng một lớp nhân chập sử dụng các phép nhân chập 1x1. Do đó, điều đầu tiên cần lưu ý là kết quả đầu ra sẽ là các bản đồ đặc trưng (feature map). Vì chỉ có các phép nhân chập 1x1, kích thước của bản đồ dự đoán (prediction map) sẽ bằng với kích thước

TẠP CHÍ ISSN: 1859-316X
KHOA HỌC CÔNG NGHỆ HÀNG HẢI
JOURNAL OF MARINE SCIENCE AND TECHNOLOGY

của feature map ngay trước nó. Trong YOLO3, ý nghĩa của prediction map này là mỗi ô (cell) của nó sẽ có thể dự đoán một số lượng cố định các bb. Chẳng hạn nếu chúng ta có B*(5+C) giá trị trong feature map. B là số bb mà mỗi cell có thể dự đoán. Mỗi một trong số B các bb này có thể chuyên biệt cho việc dự đoán một loại đối tượng cụ thể. Mỗi bb phải có 5+C thuộc tính, mô tả các giá trị về toạ độ trung tâm, kích thước, giá trị điểm số và C độ tin cậy cho mỗi bb. YOLO3 dự đoán 3 bb cho mỗi cell.

YOLO3 huấn luyện mô hình mạng theo cách thức mà trong đó chỉ một bb sẽ chịu trách nhiệm cho việc phát hiện 1 đối tượng. Đầu tiên, chúng ta cần phải xác minh các ô mà bb này thuộc về. Để làm điều đó, YOLO3 sẽ chia bức ảnh input thành các lưới kích thước bằng với feature map cuối cùng. Ở ví dụ minh hoạ bên dưới (Hình 3), trong đó ảnh input là 416x416, và bước của mạng là 32. Như đã đề cập trước đó, kích thước của feature map cuối cùng sẽ là 13x13. Do đó chúng ta sẽ chia bức ảnh thành các ô 13x13.

Sau đó, ô (trên bức ảnh input) chứa trung tâm của bb đúng của một đối tượng sẽ được chọn là đối tượng chịu trách nhiệm cho việc dự đoán đối tượng. Trong bức ảnh, chính là ô được đánh dấu đỏ, chứa trung tâm của bb đúng (đánh dấu vàng).

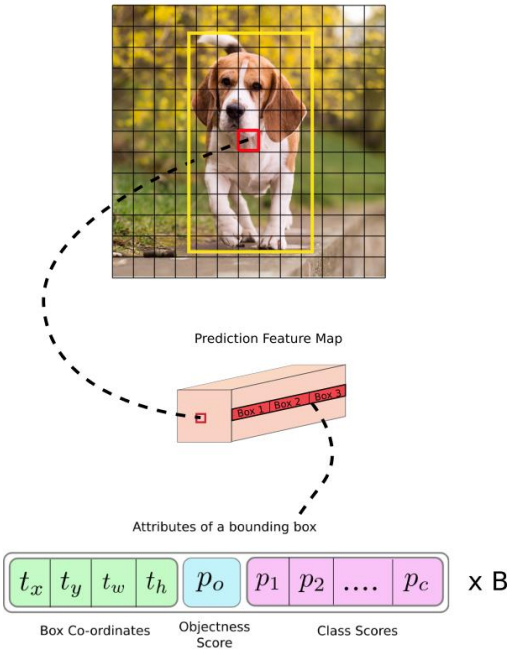
YOLO chỉ sử dụng các lớp nhân chập (minh hoạ trong Hình 2), do đó nó là một mạng nơ ron nhân chập đầy đủ (FCN). YOLO3 [8] là một kiến trúc mới (so với YOLO1 [4]), sâu hơn để trích chọn đặc trưng gọi là Darknet-53. Như cái tên của nó, Darknet-53 sử dụng 53 lớp nhân chập, mỗi lớp được theo sau bởi một lớp chuẩn hoá theo lô và một lớp kích hoạt sử dụng hàm Leaky ReLU. Không có lớp tổng hợp nào được sử dụng, và một lớp nhân chập với giá trị bước giảm bằng 2 được sử dụng để giảm kích thước của các feature map. Điều này giúp cho việc ngăn chặn các mất mát về các đặc trưng ở mức thấp thường gây ra do các lớp tổng hợp.

Một điểm đáng lưu ý nữa của YOLO3 là nó có khả năng phát hiện các đối tượng có kích thước tương đối bé (tốt hơn so với các phiên bản trước của nó).

3. Xây dựng hệ thống

3.1. Ngôn ngữ và nền tảng lập trình

Để xây dựng hệ thống demo, chúng tôi sử dụng ngôn ngữ lập trình Python 3 và thư viện Darknet [9] trên hệ điều hành Windows 10 Enterprise với một hệ thống phần cứng có 1 card GPU Nvidia Geforce 1050 Ti 4 GB bộ nhớ, CPU core i7 9750H 6 lõi, 12 luồng, 32 GB RAM, 1 TB ổ cứng SSD. Về nền tảng phần mềm, chúng tôi sử dụng phiên bản Tensorflow 1.16, Spyder 4 và OpenCV 4.3. Hệ thống demo có khả năng phát hiện các phương tiện giao thông phổ biến trên đường, cụ thể là: xe máy (motorbike), xe con (car), xe tải (truck), xe đạp (bicycle) và xe buýt (bus).



Hình 3. Các ô dự đoán/phát hiện đối tượng

Hình 4. Hình ảnh kết quả phát hiện phương tiện giao thông từ camera giám sát hành trình

Hình 5. Hình ảnh kết quả phát hiện phương tiện giao thông từ đường Nguyễn Bỉnh Khiêm

3.2. Dữ liệu và kết quả thử nghiệm

Để thử nghiệm hiệu năng (độ chính xác và tốc độ) của hệ thống demo, chúng tôi đã thu thập dữ liệu (video) từ Internet [10] và tự quay một số hình ảnh từ khung cảnh ngoài trời (tại nút giao thông Cầu vượt Đông Hải và từ camera giám sát hành trình đặt trên ô tô). Mô hình YOLO3 được training trên cơ sở dữ liệu COCO [8]. Kết quả thực nghiệm cho thấy: 1) hệ thống demo có thể phát hiện tốt 5 loại phương tiện giao thông cơ bản (xe đạp, xe máy, xe ô tô con, xe tải, xe buýt) với tỉ lệ chính xác trên 180 ảnh test [10] là 94%, 2) hệ thống có khả năng chạy trong thời gian thực (vẫn chưa dùng hết tài nguyên phần cứng) khi đạt tốc độ xử lý 20 FPS, 3) vẫn còn một số trường hợp nhầm lẫn giữa xe tải và xe con (xem thêm Hình 4, 5) khi số lượng đối tượng trong 1 ảnh lớn.

4. Kết luận

Phát hiện phương tiện giao thông là một vấn đề thực tế và có nhiều ứng dụng trong thực tế. Trong bài báo này, nhóm tác giả đã đề xuất một hệ thống phát hiện các phương tiện giao thông phổ biến (xe đạp, xe máy, xe ô tô con, xe ô tô tải, xe buýt) bằng cách sử dụng mô hình mạng học sâu YOLO3. Kết quả thực nghiệm cho thấy đây là một hướng tiếp cận có độ chính xác cao và thời gian thực hiện nhanh, có khả năng áp dụng vào các tình huống thực tế.

Trong tương lai, nhóm tác giả mong muốn ứng dụng mô hình YOLO3 vào các bài toán có yêu cầu phát hiện đối tượng khác (như phát hiện mặt người, cảnh báo các tình huống giao thông nguy hiểm) và kết hợp với các phương pháp học sâu khác như retinanet hay SSD để nâng cao độ chính xác.

Lời cảm ơn

Bài báo này là sản phẩm của đề tài nghiên cứu khoa học cấp Trường năm học 2019-2020, tên đề tài: "*Ứng dụng mạng nơ ron học sâu xây dựng hệ thống phát hiện và thống kê phương tiện giao thông*", được hỗ trợ kinh phí bởi Trường Đại học Hàng hải Việt Nam.

TÀI LIỆU THAM KHẢO

[1] Radhakrishnan, M, *Video object extraction by using background subtraction techniques for sports applications*, Digital Image Processing, Vol.5(9), pp.91-97, 2013.

[2] Qiu-Lin, L.I., & Jia-Feng, H.E., *Vehicles detection based on three-frame-difference method and cross-entropy threshold method*, Computer Engineering, Vol.37(4), pp.172-174, 2011.

[3] Liu, Y., Yao, L., Shi, Q., Ding, J., *Optical flow based urban road vehicle tracking*, Ninth International Conference on Computational Intelligence and Security, 2014.

[4] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, *You Only Look Once: Unified, Real-Time Object Detection*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[5] T. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, *Focal Loss for Dense Object Detection*, IEEE International Conference on Computer Vision (ICCV), 2017.

[6] Liu W. et al, *SSD: Single Shot MultiBox Detector*, Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision - ECCV, 2016.

[7] R. Girshick, *Fast R-CNN*, IEEE International Conference on Computer Vision (ICCV), Santiago, 2015.

[8] Redmon, Joseph & Farhadi, Ali., *YOLOv3: An Incremental Improvement*, 2018.

[9] *Darknet*, [Online]. Available: https://pjreddie.com/darknet/.

[10] *Test dataset*, [Online]. Available: https://github.com/ahmetozlu/vehicle_counting_tensorflow/tree/master/custom_vehicle_training/images/train.

Ngày nhận bài:	11/05/2020
Ngày nhận bản sửa:	28/05/2020
Ngày duyệt đăng:	03/06/2020