PHÁT HIỆN ĐỐI TƯỢNG MỘT GIAI ĐOẠN CHO GIÁM SÁT VIDEO TRONG THÀNH PHỐ THÔNG MINH

Lê Quang Chiến

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

Email: lqchien@hueuni.edu.vn

Ngày nhận bài: 14/6/2022; ngày hoàn thành phản biện: 16/6/2022; ngày duyệt đăng: 22/6/2022

TÓM TẮT

Gần đây các phương pháp phát hiện đối tượng đã đạt được hiệu suất cao trên cả tốc độ và độ chính xác. Bên cạnh đó, việc ứng dụng các phương pháp này trong các hệ thống giám sát thông minh đang thu hút được nhiều sự quan tâm. Bài báo này tìm hiểu sự phát triển của mô hình phát hiện đối tượng một giai đoạn You Only Look Once (YOLO) gồm: YOLOv3, YOLOv4 và YOLOv5. Các phiên bản này sẽ được đánh giá dựa trên tốc độ xử lý khung hình, F1-score 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 cho giám sát video trong thành phố thông minh.

Từ khóa: Phát hiện đối tượng, YOLO, giám sát video, thành phố thông minh.

1. MỞ ĐẦU

Trong các hệ thống thông minh, phát hiện đối tượng đóng một vai trò quan trọng để giám sát hiệu quả các sự vật, sự kiện cụ thể. Mục tiêu của nhiệm vụ này là phát hiện sự hiện diện của đối tượng trong một lớp cụ thể và xác định vị trí chính xác của đối tượng đó trong hình ảnh. Các phương pháp phát hiện đối tượng hiện tại có thể được chia thành hai kiểu: một giai đoạn và hai giai đoạn. Phương pháp phát hiện một giai đoạn thực hiện định vị đối tượng và phân loại cùng một lúc trong khi phương pháp phát hiện hai giai đoạn phân tách nhiệm vụ này cho mỗi hộp giới hạn (bounding box). Trong bài báo này, chúng tôi nghiên cứu và đánh giá phương pháp phát hiện một giai đoạn do bởi những ưu thế về tốc độ xử lý so với phương pháp phát hiện hai giai đoạn.

Dựa trên những báo cáo gần đây, YOLO là một trong những kiến trúc một giai đoạn đạt được hiệu quả cao trên cả hai khía cạnh độ chính xác và tốc độ xử lý. Cụ thể, các phiên bản YOLOv3 [1], YOLOv4 [2] và YOLOv5 [3] sẽ được tìm hiểu và so sánh trên cả hai khía cạnh này. Các đánh giá của các phiên bản này được thực hiện trên tập dữ liệu HumanSurveillance thu được từ camera giám sát thực tế kết hợp với dữ liệu được

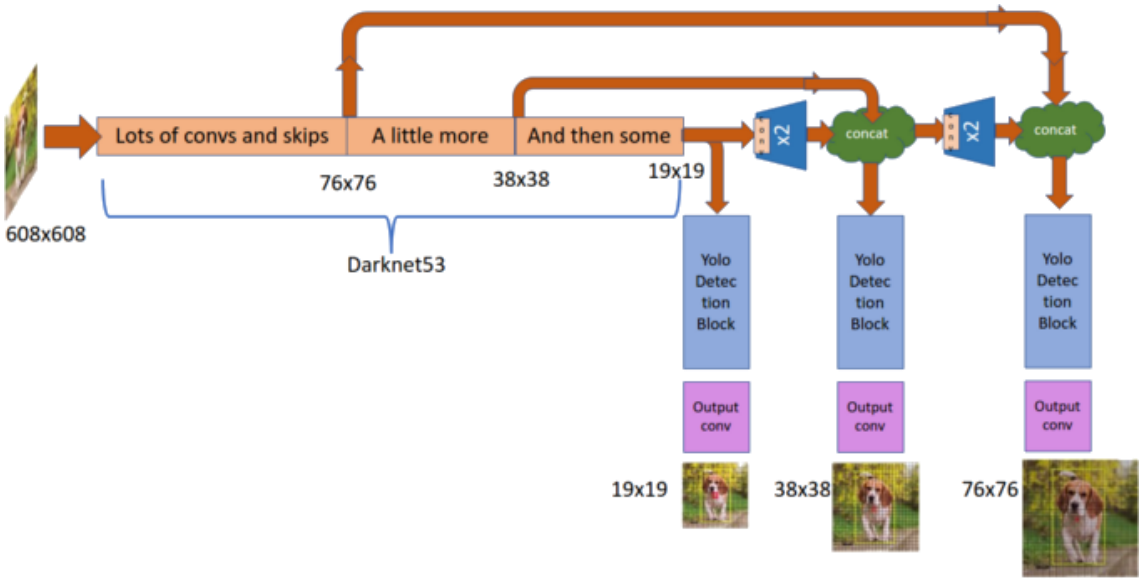
thu thập từ trang web gettyimages. Các kết quả thí nghiệm chỉ ra được tính khả thi của phương pháp YOLO cho các bài toán phát hiện đối tượng trong các hệ thống giám sát thông minh.

2. PHƯƠNG PHÁP NGHIÊN CỨU

Phát hiện đối tượng là một trong những bài toán quan trọng trong thị giác máy tính với mục tiêu là để nhận biết một đối tượng là gì và ở đâu. Hiện tại, có nhiều thuật toán học sâu dựa trên kiến trúc hai giai đoạn đã được đề xuất cho bài toán này. Tuy nhiên, các thuật toán này thường không thể đáp ứng yêu cầu về tốc độ xử lý do bởi hai giai đoạn xử lý tách biệt nhau. Điều đó đã giới hạn khả năng ứng dụng của kiểu kiến trúc này cho các bài toán thực tế. Khác với kiểu kiến trúc hai giai đoạn, YOLO là một kiến trúc học sâu một giai đoạn sử dụng mạng nơ-ron tích chập để phát hiện đối tượng. Kiểu kiến trúc này giúp cho nhiệm vụ phát hiện đối tượng đạt hiệu suất tốt hơn cả về độ chính xác và tốc độ xử lý. Trong phần này, chúng tôi trình bày một cách tổng quan ba phiên bản của YOLO cho bài toán phát hiện đối tượng hiện đại: YOLOv3, YOLOv4 và YOLOv5.

2.1. Kiến trúc YOLOv3

Trong YOLOv3, Darknet53 được sử dụng làm backbone để trích xuất các đặc trưng từ hình ảnh đầu vào. Backbone này gồm 52 phép tích chập kết hợp với kỹ thuật bỏ qua một số kết nối như trong ResNet. Bên cạnh đó, kiểu kiến trúc này cũng sử dụng mạng kim tự tháp đặc trưng (Feature Pyramid Network - FPN) [4] để xử lý hình ảnh tại các mức nén không gian khác nhau.



Hình 1. Kiến trúc của YOLOv3.

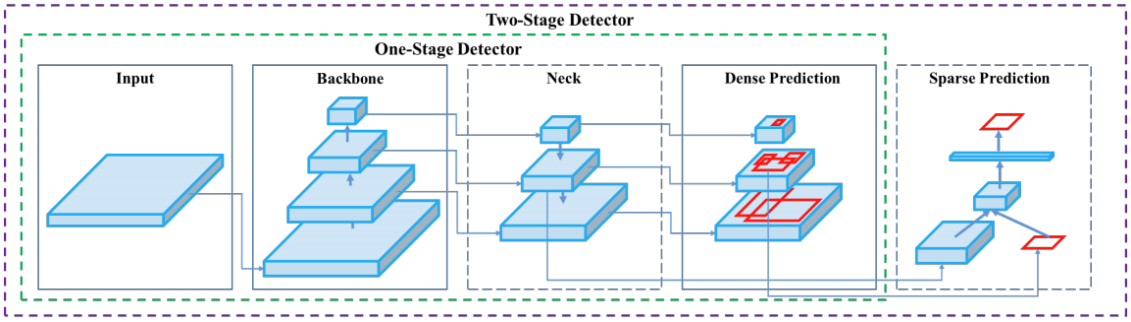
So sánh với các phiên bản trước đó, YOLOv3 cung cấp hiệu suất tốt trên các đầu vào với độ phân giải khác nhau. Thêm vào đó, YOLOv3 cũng cho tốc độ xử lý nhanh hơn đáng kể so với các mô hình dựa trên hai giai đoạn (ví dụ, RCNN và Faster-RCNN với ResNet50 làm backbone) và đạt được độ chính xác tương đương. Ngoài ra, khi so sánh với kiểu kiến trúc một giai đoạn khác như Mobilenet-SSD, YOLOv3 cũng đạt được tốc độ xử lý tương tự nhưng lại cung cấp độ chính xác tốt hơn.

2.2. Kiến trúc YOLOv4

Kiến trúc của YOLOv4 là một phiên bản sửa đổi cao cấp hơn của YOLOv3. Những sửa đổi này thể hiện ở ba thành phần chính trong YOLOv4: Backbone, Neck và Head. YOLOv4 sử dụng Cross Stage Partial Network (CSPNet) trong Darknet, tạo ra một backbone trích xuất đặc trưng mới được gọi là CSPDarknet53. Kiến trúc tích chập này dựa trên kiến trúc DenseNet đã sửa đổi [5]. Nó chuyển một bản sao của bản đồ đặc trưng từ layer cơ sở sang layer tiếp theo thông qua dense block. Ý tưởng ở đây là giúp cho vừa lưu giữ được một phần thông tin từ các layer trước, vừa giảm độ phức tạp của mô hình.

Phần Neck có nhiệm vụ trộn và kết hợp các bản đồ đặc trưng (feature maps) đã học được thông qua quá trình trích xuất đặc trưng (ở Backbone) và quá trình nhận dạng (YOLOv4 gọi là Dense prediction). Mục tiêu của việc tổng hợp này dùng để làm giàu thông tin đẩy về Head. Trước khi chuyển tiếp đến kiến trúc tổng hợp đặc trưng trong Neck, các feature maps đầu ra của CSPDarknet53 sẽ được gửi đến một khối bổ sung (khối SCP - Spatial Pyramid Pooling) để gia tăng trường tiếp nhận và tách ra các đặc trưng quan trọng nhất. Trong YOLOv3, các tác giả đã giới thiệu một cách thức áp dụng khối SPP (YOLOv3-SPP). Khối SPP áp dụng max-pooling với các kernel kích thước khác nhau. Các feature maps thu được từ việc áp dụng max-pooling (với kernel size khác nhau) sẽ được kết nối lại với nhau. YOLOv4 cũng áp dụng lại kỹ thuật này. Thêm vào đó, trong YOLOv4, các tác giả sử dụng mạng tổng hợp đường dẫn (Path Aggregation Network - PANet) [6] làm phương pháp tổng hợp tham số từ các mức backbone khác nhau cho các mức phát hiện khác nhau, thay vì FPN được sử dụng trong YOLOv3. PANet hay FPN là các phương pháp được sử dụng nhằm làm giảm sự mất mát thông tin khi mạng càng sâu. Trong đó, thông tin ở các mức khác nhau sẽ được kết hợp với nhau.

Phần Head có chức năng là thực hiện các dự đoán dày đặc. Dự đoán dày đặc là dự đoán cuối cùng bao gồm một vectơ có chứa tọa độ bounding box (tâm, chiều cao, chiều rộng), độ tin cậy của dự đoán và các lớp xác suất. YOLOv4 triển khai Head giống như YOLOv3 với các bước phát hiện dựa trên anchor.



Hình 2. Kiến trúc YOLOv4.

2.3. Kiến trúc YOLOv5

YOLOv5 khác với các phiên bản trước. Phiên bản này sử dụng framework PyTorch thay vì Darknet. Vì vậy, YOLOv5 thân thiện với người dùng và có thể dễ dàng huấn luyện trên các tập dữ liệu tùy chỉnh. So với framework Darknet được YOLOv4 áp dụng, framework Pytorch dễ đưa vào sản phẩm hơn.

So sánh về mặt kiến trúc, YOLOv5 đã tích hợp những cải tiến mới nhất tương tự như kiến trúc YOLOv4, do đó không có nhiều khác biệt về lý thuyết. Nhìn chung, kiến trúc YOLOv5 có thể được tóm tắt như sau:

- Backbone: Cấu trúc Focus, mạng CSP
- Neck: Khối SPP, PANet
- Head: YOLOv3, sử dụng GIoU-loss

3. KẾT QUẢ VÀ THẢO LUẬN

Trong phần này, chúng tôi trình bày các thiết lập thí nghiệm để thực hiện đánh giá ba phiên bản của mô hình YOLO: YOLOv3, YOLOv4 và YOLOv5 (chúng tôi lựa chọn sử dụng YOLOv5-large với kích thước mô hình tương đương với YOLOv4.). 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 hiệu quả của từng phiên bản khi áp dụng cho hệ thống giám sát thực tế.

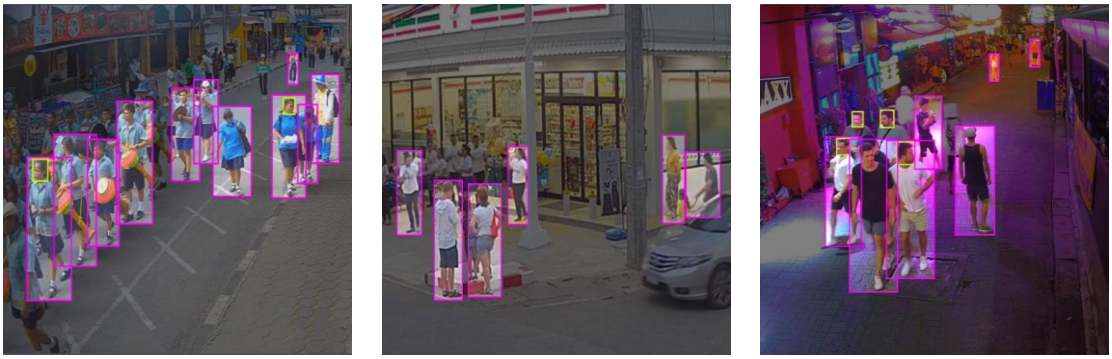
3.1. Các thiết lập thí nghiệm

Chúng tôi thực hiện đánh giá ba phiên bản YOLO trên tập dữ liệu HumanSurveillance. Đây là tập dữ liệu được thu thập từ hai nguồn chính: (1) Từ trang web gettyimages; (2) Từ các camera giám sát được chia sẻ trên youtube. Tập dữ liệu HumanSurveillance gồm hai tập train và test. Trong đó, tập dữ liệu train chứa 13.564 ảnh và tập dữ liệu test chứa 3.393 ảnh. HumanSurveillance gồm hai lớp đối tượng: head

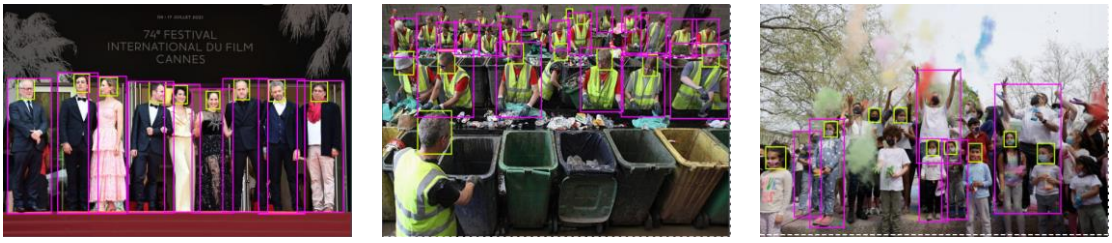
và body. Tập dữ liệu này cũng chứa tổng cộng 125.959 bounding box với trung bình 4 người mỗi ảnh (Hình 3).

Trong phần huấn luyện, chúng tôi sử dụng cùng chung một thiết lập cho cả ba phiên bản. Các tham số đầu vào cụ thể gồm: (1) kích thước đầu vào: 416; (2) kích thước batch: 32; (3) số vòng lặp cần thực thi: 1000; (4) optimizer sử dụng stochastic gradient descent với momentum 0.9; (5) learning rate: 0.001 và (6) weight decay: 0.0005.

Chúng tôi cung cấp các đánh giá liên quan dựa trên các độ đo như: Precision, Recall, F1-score, mAP và tốc độ xử lý của ba phiên bản. Các kết quả được báo cáo trong bài báo này đều được đánh giá trên PC với các tham số cơ bản (CPU: 10th Gen. Intel CoreTM i7-10750H Processor; GPU: NVIDIA GeForce RTX 2080Ti with 11 GB GDDR6).



a) Dữ liệu từ các camera giám sát



b) Dữ liệu từ trang web gettyimages

Hình 3. Một số hình ảnh minh họa tập dữ liệu HumanSurveillance.

3.2. Các kết quả thí nghiệm

3.2.1. Tốc độ xử lý

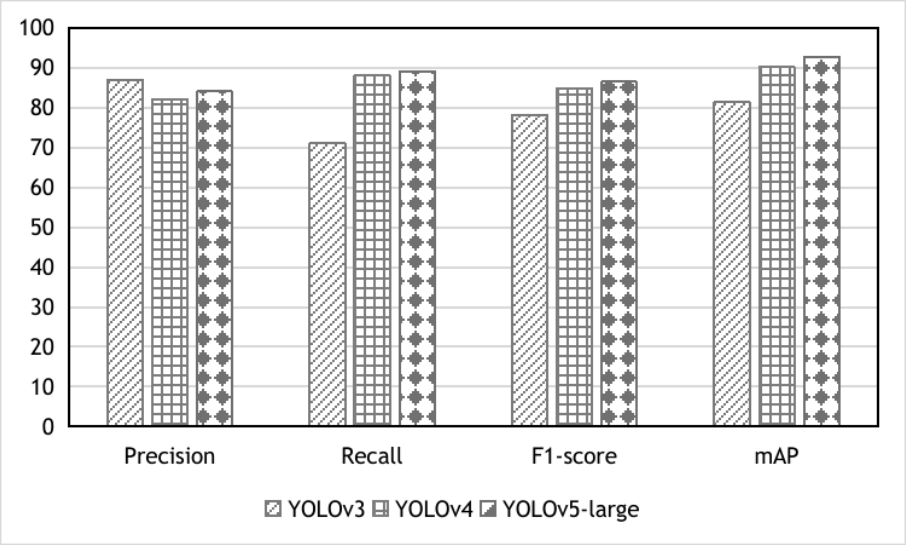
Bảng 1 thể hiện các kết quả các kết quả thí nghiệm khi đánh giá tốc độ xử lý của ba mô hình YOLO. Tốc độ xử lý được tính bằng số lượng khung hình được xử lý trong một giây.

Bảng 1. Tốc độ xử lý của các mô hình YOLO

Độ đo	YOLOv3	YOLOv4	YOLOv5
FPS	64.8	60	59.8

3.2.2. Độ chính xác

Bảng 2 thể hiện độ chính xác trung bình của hai mô hình YOLO và SSD khi được thực nghiệm trên tập dữ liệu HumanSurveillance. Đây là tập dữ liệu được ghi hình bởi các camera giám sát đường phố và từ trang gettyimages. Tập dữ liệu này thể hiện sự đa dạng trong góc nhìn với nhiều độ phân giải khác nhau. Mục đích của tập dữ liệu này là để phục vụ cho các hệ thống giám sát giao thông trong các điều kiện khác nhau.



Hình 3. Precision, Recall, F1-score và mAP của ba mô hình trên tập dữ liệu HumanSurveillance.

3.3. Phân tích và thảo luận

Các kết quả được báo cáo trong Bảng 1 thể hiện tốc độ xử lý của ba mô hình YOLOv3, YOLOv4 và YOLOv5-large. Kết quả thí nghiệm cho thấy YOLOv3 có tốc độ xử lý nhanh nhất (64.8 fps), tiếp theo đến YOLOv4 (60 fps) và YOLOv5-large (59.8 fps). Điều này có thể giải thích do bởi YOLOv4 và YOLOv5-large có kiến trúc phức tạp hơn YOLOv3 khi bổ sung thêm thành phần Neck để xử lý feature maps được trích xuất từ thành phần Backbone. Kết quả này cũng khẳng định cả ba mô hình đều có thể đáp ứng các ứng dụng đòi hỏi tốc độ xử lý theo thời gian thực.

Hình 3 cho thấy kết quả so sánh của ba mô hình YOLO khác nhau được huấn luyện và kiểm thử trên tập dữ liệu HumanSurveillance để phát hiện người từ các camera giám sát. YOLOv3 có precision cao nhưng recall thấp và điều đó cho thấy mô hình cần được cải thiện. Để một thuật toán được coi là hiệu quả, phải có sự cân bằng giữa precision và recall. Sự cân bằng đó được phản ánh bởi độ đo F1-score. Như được chỉ ra trong Hình 3, precision và recall của YOLOv4 và YOLOv5-large cân bằng tốt hơn. Do

đó, F1-score của YOLOv4 và YOLOv5-large cao hơn so với YOLOv3, mặc dù YOLOv3 có precision cao hơn. Như một hệ quả, mAP của YOLOv4 (90.1%) và YOLOv5-large (92.6%) cũng cao hơn đáng kể so với YOLOv3 (81.3%).

Các báo cáo thí nghiệm cho thấy việc thực thi nhiệm vụ phát hiện đối tượng trong ngữ cảnh của camera giám sát có thể đáp ứng tốc độ xử lý thời gian thực. Bên cạnh đó, để đảm bảo được độ chính xác phù hợp với các bài toán thực tế, việc sử dụng các phiên bản YOLOv4 và YOLOv5 là thích hợp hơn so với YOLOv3. Tuy nhiên, chúng ta cần xác định 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 cho từng ngữ cảnh, chúng ta cần phải cân bằng hai yếu tố này.

Với đối tượng được giám sát là con người, thì các mô hình YOLO đều dễ dàng phát hiện ngay cả ở khoảng cách xa do bởi kích thước và hình dạng đặc trưng. Vì vậy, việc sử dụng các mô hình có kiến trúc lớn như YOLOv4 và YOLOv5-large đôi khi không cần thiết. Chúng ta có thể triển khai với các mô hình YOLO với kiến trúc nhỏ gọn hơn để phục vụ cho mục đích giám sát theo thời gian thực khi mà phần cứng của hệ thống không đáp ứng đủ yêu cầu. Trong trường hợp này, các mô hình như YOLOv5-small hay YOLOv5-medium sẽ được ưu tiên lựa chọn do có tốc độ xử lý nhanh hơn đáng kể khi thực hiện suy luận trên CPU. 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, cũng như các kỹ thuật theo dấu để 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 các mô hình YOLO cho nhiệm vụ phát hiện đối tượng thời gian thực. Ba mô hình YOLOv3, YOLOv4 và YOLOv5-large đã được lựa chọn để đánh giá trên độ chính xác và tốc độ xử lý. Kết quả cho thấy ba mô hình YOLOv3, YOLOv4 và YOLOv5-large đều đáp ứng được tốc độ xử lý theo thời gian thực với sự hỗ trợ của GPU. Tuy nhiên, trong các hệ thống giám sát với cấu hình phần cứng không đảm bảo thì YOLOv5 có thể linh hoạt được sử dụng với các kiến trúc small hay medium.

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 mô hình ban đầu 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 đạt được hiệu quả cao nhất.

TÀI LIỆU THAM KHẢO

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

[2]. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

[3]. https://github.com/ultralytics/yolov5

[4]. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).

[5]. Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T., & Keutzer, K. (2014). Densenet: Implementing efficient convnet descriptor pyramids. arXiv preprint arXiv:1404.1869.

[6]. Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8759-8768).

ONE-STAGE OBJECT DETECTION FOR VIDEO SURVEILLANCE IN SMART CITIES

Le Quang Chien

Faculty of Information Technology, University of Sciences, Hue University

Email: lqchien@hueuni.edu.vn

ABSTRACT

Object detection methods have recently achieved high performance in both speed and accuracy. Besides, the application of these methods in intelligent surveillance systems has been attracting much attention. This article explores the evolution of a single-stage object detection model, You Only Look Once (YOLO) including: YOLOv3, YOLOv4 and YOLOv5. The versions will be evaluated based on the frame rate, F1-score and mean average precision in the phase of the inference. The experimental results show the feasibility of the models when being applied for video surveillance in smart cities.

Keywords: Object detection, YOLO, Video surveillance, smart cities.

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.