

NHẬN DIỆN BIẾN BÁO VÀ TÍN HIỆU ĐÈN GIAO THÔNG SỬ DỤNG YOLOv4 TRÊN PHẦN CỨNG JETSON TX2

Bùi Quốc Tú^{1*}, Nguyễn Huy Hoàng¹, Trương Quang Phúc¹
Lê Quang Bình¹, Hồ Nhật Minh²

¹Trường Đại học Sư phạm Kỹ thuật TP.HCM

²Học viện Công nghệ Bưu chính Viễn thông

*Email: tubui2263@gmail.com

Ngày nhận bài: 22/6/2022; Ngày chấp nhận đăng: 27/7/2022

TÓM TẮT

Phát hiện, nhận diện biến báo và tín hiệu đèn giao thông là một phần quan trọng của xe tự hành nhằm tăng cường sự an toàn khi tham gia giao thông. Trong bài báo này, nhóm nghiên cứu sử dụng mô hình YOLOv4, một trong những kiến trúc phổ biến của học sâu để phát hiện, nhận diện biến báo và các tín hiệu đèn giao thông. Một cách tổng quát, mô hình YOLOv4 có nhiều phiên bản với kiến trúc và tham số khác nhau, nhóm nghiên cứu tập trung vào việc so sánh và đánh giá để lựa chọn phiên bản phù hợp với bộ dữ liệu. Ngoài ra, card đồ họa tích hợp (GPU) trong Jetson TX2 của NVIDIA được sử dụng làm nền tảng phần cứng với khả năng tối ưu thời gian huấn luyện. Trong đó, dữ liệu sử dụng được tạo độc lập bao gồm 32 lớp để huấn luyện và kiểm tra. Phương pháp sử dụng đã đạt được chỉ số 91% mAP và 31,2 FPS trên tập dữ liệu kiểm tra.

Từ khóa: Nhận diện, biến báo, tín hiệu đèn, YOLOv4, Jetson TX2.

1. MỞ ĐẦU

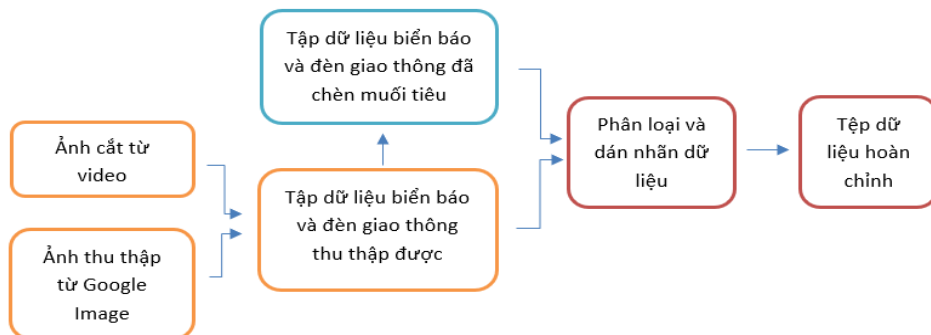
Trong những năm gần đây, với sự phát triển của các phương tiện tự hành và nhu cầu sử dụng xe ô tô tự hành ngày càng tăng, nhu cầu phát hiện các biến báo và tín hiệu đèn giao thông đã trở nên cần thiết và ngày càng phổ biến. Việc phát hiện biến báo và tín hiệu đèn giao thông sẽ giúp người lái xe nhận thức rõ hơn về môi trường xung quanh và giúp họ nhận ra các biến báo, đèn tín hiệu ở xa, khi thời tiết xấu hoặc vào ban đêm khi tầm nhìn bị hạn chế. Và một hệ thống phát hiện tốt là cần thiết để làm cho tất cả những điều đó trở nên khả thi. Hệ thống này có giá trị trong việc duy trì an toàn giao thông, ngăn ngừa tai nạn và các vấn đề giao thông. Nhưng việc nhận diện và phát hiện vẫn là một vấn đề đầy thách thức trong thế giới thực vì những vấn đề này sinh về chất lượng hình ảnh, ánh sáng, môi trường xung quanh. Do đó, để đối phó với tất cả những thách thức này, nhiều phương pháp và thuật toán học máy mới đã được giới thiệu. Trước đây, hệ thống chủ yếu dựa vào các thuật toán phát hiện đối tượng truyền thống. Đường ống của hệ thống thường sử dụng các tính năng thủ công để trích xuất các đặc trưng khu vực và sau đó kết hợp các bộ phân loại để loại bỏ các vùng không chứa đối tượng. Ngày nay, các phương pháp học sâu đang gia tăng và được áp dụng để nhận dạng hình ảnh và phát hiện đối tượng, ngày càng trở nên nhanh chóng và chính xác. Hầu hết các nghiên cứu liên quan đến nhận dạng hình ảnh và phát hiện đối tượng xoay quanh mạng nơ-ron tích chập (CNN) để tăng tốc độ và độ chính xác. CNN sở hữu khả năng học các đặc điểm từ một lượng lớn dữ liệu mà không cần xử lý trước, điều này tránh được khó khăn trong thiết kế của các tính năng thủ công và tìm hiểu các tính năng tổng quát hơn [1-3]. CNN đã đóng vai trò là một công cụ phân loại trong học máy và cũng đã được triển khai trong phân loại biến báo và tín hiệu đèn giao thông.

Gần đây, các thuật toán như SSD, Fast R - CNN, Faster R - CNN, R - FCN và YOLO là những tiến bộ trong việc phát hiện đối tượng đều sử dụng CNN. Trong bài báo này, nhóm nghiên cứu đã sử dụng kiến trúc "You Only Look Once" (YOLO) cụ thể là YOLOv4, là mạng phát hiện một giai đoạn và có độ trễ lan truyền thấp và hiệu suất phát hiện tuyệt vời. Nhiều mạng nơ-ron hiện đại chính xác không hoạt động trong thời gian thực và yêu cầu một số lượng lớn GPU để đào tạo. YOLO giải quyết những vấn đề như vậy bằng cách tạo CNN hoạt động trong thời gian thực trên GPU thông thường và đào tạo chỉ yêu cầu một GPU thông thường [4, 5].

Ngoài sự phát triển của các thuật toán, một yếu tố quan trọng khác là yếu tố phần cứng. Trong bài báo này, nhóm nghiên cứu lựa chọn phần cứng Jetson TX2 để thực hiện mô hình nhận diện các biển báo và tín hiệu giao thông. Jetson TX2 tích hợp nhân CUDA và hệ thống GPU để đạt tốc độ xử lý cao [6]. Hơn nữa, Jetson TX2 đáp ứng các yêu cầu khắt khe nhất về thời gian thực nhưng vẫn nhỏ gọn và dễ cài đặt [7]. Nhóm nghiên cứu thực hiện một mô hình phát hiện và nhận diện các biển báo và tín hiệu giao thông sử dụng YOLOv4 trên phần cứng nhúng Jetson TX2. Mục tiêu là phát triển một mô hình có khả năng nhận diện các biển báo: hạn chế tốc độ, biển cấm, hiệu lệnh và 3 tín hiệu đèn giao thông cơ bản: Xanh, vàng, đỏ, có thể hoạt động được trong thời gian thực. Mô hình được nghiên cứu và phát triển trên một hệ thống nhúng riêng biệt (Jetson TX2), mô hình có thể đạt được độ chính xác cao (hơn 90% bao gồm cả phát hiện lẫn nhận diện đèn giao thông và biển báo) với tốc độ khung hình nhanh nhất lên đến 43 FPS.

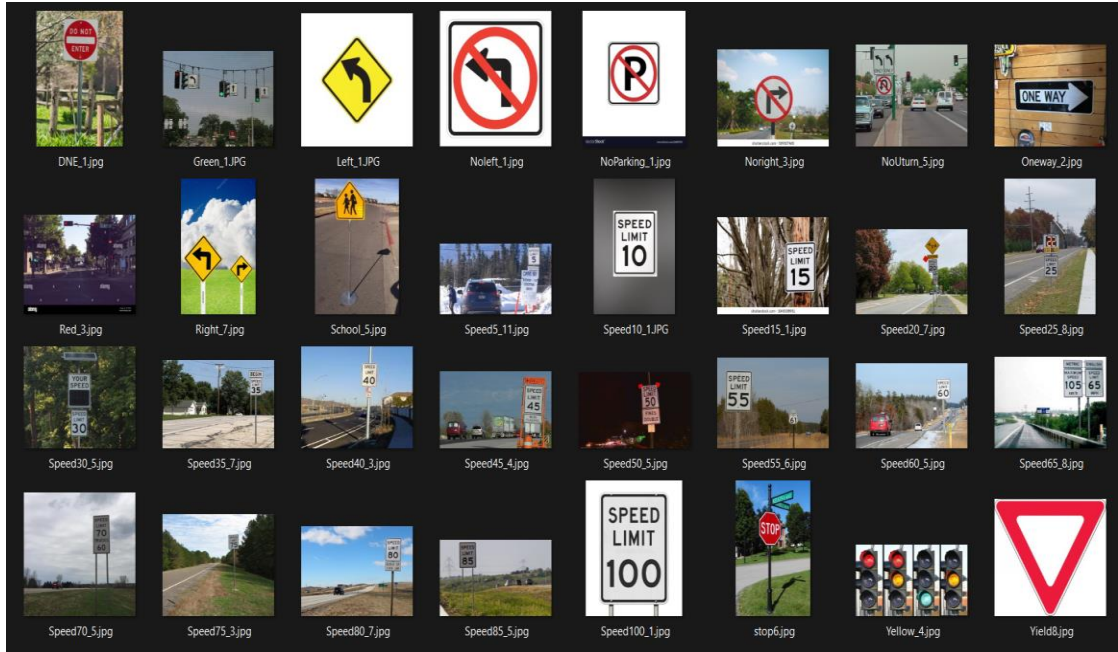
2. TẬP DỮ LIỆU

Có rất nhiều tập dữ liệu khác nhau có thể dùng để phát triển mô hình phát hiện, nhận diện biển báo và các tín hiệu đèn giao thông như bộ dữ liệu phát hiện biển báo giao thông của Đức (GTSDDB), bộ dữ liệu biển báo giao thông LISA (LISATSD), bộ dữ liệu biển báo giao thông Thụy Điển (STSD) và bộ dữ liệu biển báo giao thông Trung Quốc (CTSD), nhưng các tập dữ liệu kể trên rất lớn và phức tạp, cần tiêu tốn một lượng lớn thời gian để xử lý cho phù hợp với mô hình. Trong bài báo này, nhóm nghiên cứu tập trung thử nghiệm với tập dữ liệu được tạo độc lập với mục tiêu đơn giản nhưng vẫn mang lại độ chính xác cao cho mô hình. Nhóm nghiên cứu sử dụng tập dữ liệu được thu thập từ các biển báo và tín hiệu đèn giao thông ở Los Angeles gồm có 32 lớp và hơn 1500 ảnh dữ liệu đường phố. Trong đó có khoảng 800 ảnh dữ liệu được thu thập và 700 ảnh được tăng cường từ 800 ảnh dữ liệu đó bằng cách chèn nhiễu muối tiêu vào. Hình 1 là tổng quan về tập dữ liệu và cách xử lý dữ liệu để đưa vào mô hình.



Hình 1. Tập dữ liệu huấn luyện

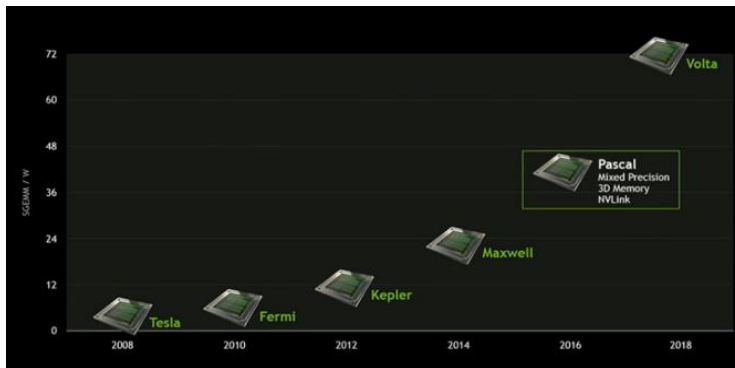
Hình 2 là ảnh của từng lớp trong tập dữ liệu, trong đó có 3 lớp thuộc tín hiệu đèn và 29 lớp là các biển báo.



Hình 2. Ảnh các lớp trong tập dữ liệu

3. PHẦN CỨNG JETSON TX2

Jetson TX2 với bản chất là một máy tính nhưng có hiệu quả năng lượng cao của NVIDIA, tuy nhỏ nhưng nó khá mạnh mẽ, TX2 được sử dụng rộng rãi ở các lĩnh vực máy tính, trí tuệ nhân tạo, máy học. Chiếc máy tính nhúng này được trang bị 2 CPU gồm: ARM Cortex - A57 (4 lõi) 2GHz và NVIDIA Denver2 (lõi kép) @ 2GHz [7], song song đó, GPU mà TX2 sở hữu là Pascal 256 lõi 1300 MHz - một kiến trúc được ra mắt vào 2016. Tất cả những tác vụ tính toán có thể chỉ tiêu thụ 7,5W năng lượng. Do đó, TX2 là một phần cứng tốt để thực hiện các giải thuật học sâu trên nền tảng hệ thống nhúng [8].



Hình 3. Quá trình phát triển kiến trúc GPU của NVIDIA [9]

Hình trên cho thấy quá trình phát triển lõi GPU của NVIDIA đặc biệt là sự đột phá của NVIDIA Pascal [9]. Kiến trúc này có thể đạt gấp 10 lần hiệu năng so với kiến trúc trước đó là Maxwell:

Pascal có hiệu suất điểm nổi gấp 2 lần Maxwell.

Pascal có dung lượng bộ nhớ gấp 2,7 lần Maxwell.

Pascal có độ chính xác gấp 4 lần Maxwell.

4. THUẬT TOÁN YOLOv4

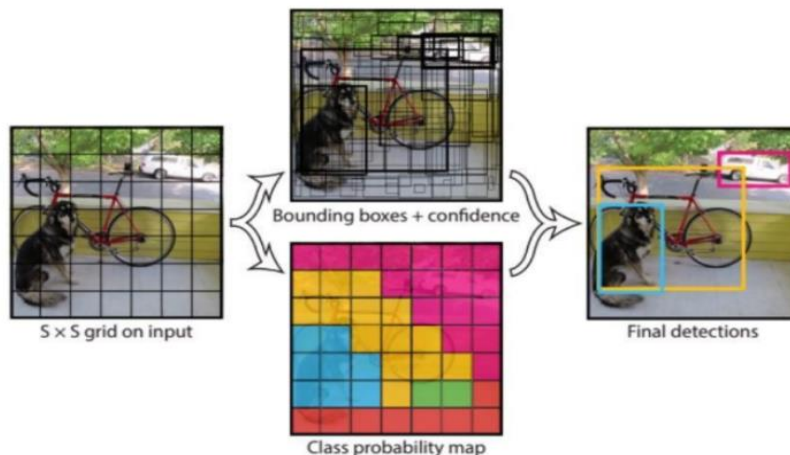
4.1. Giới thiệu về YOLO

YOLO là viết tắt của "You Only Look Once", nghĩa là "Bạn chỉ nhìn một lần", tên gọi bắt nguồn từ cách hoạt động của nó. Khi mới ra mắt, YOLO đã cho thấy một tốc độ vượt trội và độ chính xác cao hơn so với những đối thủ của mình. Mạng YOLO có cấu trúc tương tự như cấu trúc của một mạng nơ-ron tích chập bình thường, bao gồm một số lớp tích chập và Max - pooling, kết thúc bằng hai lớp Fully - connected. YOLO tuân theo một cách tiếp cận rất độc đáo, trong đó nó sử dụng một mạng nơ-ron duy nhất để xem toàn bộ hình ảnh chỉ một lần, vì điều này, nó có nhiều ưu điểm khác nhau so với các hệ thống dựa trên bộ phân loại. Để dự đoán cho từng hộp giới hạn, nó sử dụng các đặc trưng từ toàn bộ hình ảnh.

Mô hình YOLO chia hình ảnh đầu vào thành $S \times S$ ô lưới như trong Hình 4. Mỗi ô lưới dự đoán k hộp giới hạn và giá trị tin cậy của các hộp giới hạn, cũng như xác suất lớp có điều kiện C [10]. Điểm tin cậy có nghĩa là mức độ tin cậy của mô hình về đối tượng trong hộp và cũng là độ chính xác của hộp mà nó dự đoán. Mỗi hộp giới hạn đưa ra 5 dự đoán: x, y, w, h và độ tin cậy. Các tọa độ (x, y) là tọa độ tâm của hộp giới hạn. w và h là chiều rộng và chiều cao, được dự đoán so với toàn bộ hình ảnh. Dự đoán độ tin cậy (Confidence prediction - cfd) được tính toán bằng:

$$cfd = P_r(object) * IOU_{pred}^{truth} \quad (1)$$

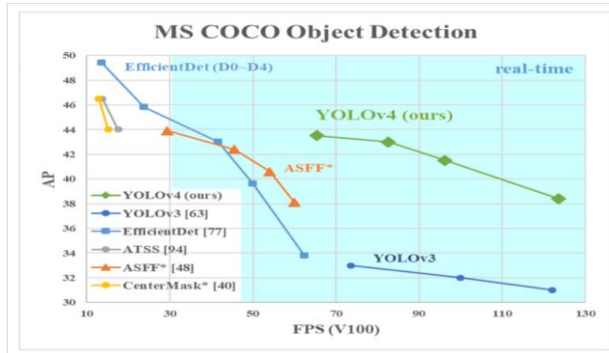
Khi một ô lưới chứa một phần của hộp ground truth thì giá trị của $Pr(object)$ bằng 1 ngược lại bằng 0 [11]. IOU là viết tắt của "Intersection Over the Union", là phần giao nhau giữa hộp giới hạn dự đoán và hộp ground truth. Với sự trợ giúp của các dự đoán này, cuối cùng lấy được điểm tin cậy theo lớp cụ thể của các hộp giới hạn riêng lẻ và chọn các hộp giới hạn có điểm tin cậy cao trong mỗi ô lưới để đưa ra dự đoán chung về một biển báo hoặc đèn tín hiệu giao thông trong hình ảnh. Như vậy việc chỉ sử dụng một bài toán hồi quy duy nhất cho toàn bộ ảnh, thuật toán YOLO giúp giảm số lượng phép toán, tăng tốc độ xử lý khi đó có thể đáp ứng bài toán thời gian thực tốt hơn so với các thuật toán R - CNN.



Hình 4. Cách YOLO phát hiện đối tượng [12]

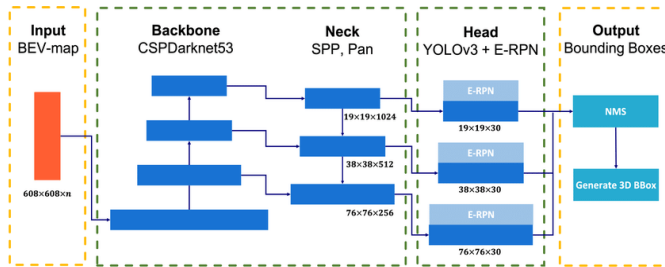
4.2. YOLOv4

Alexey Bochkovski và cộng sự đã phát triển YOLOv4 và phát hành vào tháng 4 năm 2020. Đây được coi là một trong những mô hình phát hiện vật thể thời gian thực hiện đại nhất vào thời điểm đó. So với YOLOv3, YOLOv4 nhanh hơn 12% và chính xác hơn 10% [4].



Hình 5. So sánh tốc độ xử lý và độ chính xác của YOLOv4 với các thuật toán khác trong cùng tập dữ liệu MS COCO. Có thể thấy so với YOLOv3 với cùng một FPS (Frame per Second), YOLOv4 cho độ chính xác (AP) cao hơn hẳn [4]

Kiến trúc mới của YOLOv4 được xây dựng với CSPDarknet53 làm xương sống, giúp tăng khả năng học tập của mạng nơ-ron tích chập. Mô hình YOLOv4 có thể được đào tạo trên một GPU duy nhất.



Hình 6. Kiến trúc mô hình YOLOv4 cơ bản [11]

Kiến trúc của YOLOv4 bao gồm [4]:

Bảng 1. Kiến trúc YOLOv4

Backbone	Neck	Head
CSPDarknet53	SPP, PAN, FPN, SSP...	YOLOv3

Các tác giả đã phát triển các kỹ thuật như bag of freebies và bag of specials và ứng dụng trong YOLOv4 để cải thiện độ chính xác của mô hình trong khi đào tạo và trong quá trình tiền xử lý, góp phần làm cho YOLOv4 trở thành một mô hình hiện đại và đứng đầu trong học sâu.

Bag of Freebies (BoF): Bag of Freebies là tập hợp các kỹ thuật được sử dụng trong quá trình đào tạo để giúp tăng độ chính xác của mô hình mà không tăng thời gian suy luận [13].

Bag of Specials (BoS): Bag of Specials là tập hợp các kỹ thuật được sử dụng trong quá trình đào tạo nhằm thay đổi kiến trúc của mạng và tăng thời gian suy luận một chút nhưng giúp cải thiện độ chính xác của mô hình [13].

Bảng 2. Các kỹ thuật BoF, BoS của Backbone và Detector

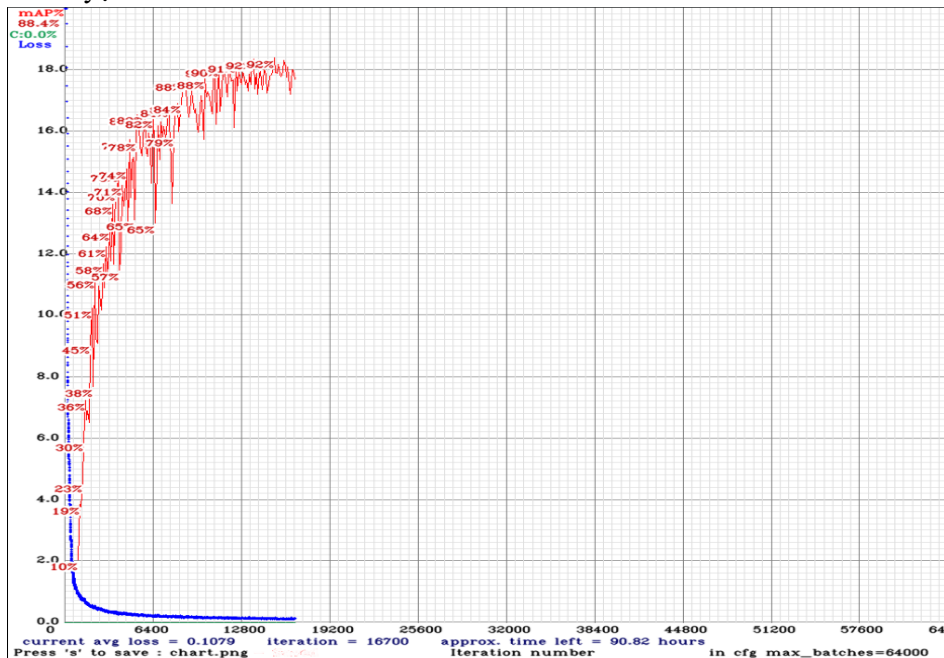
	Backbone	Detector
Bag of Freebies	DropBlock regularization, CutMix & Mosaic data augmentation, Class label smoothing.	Cosine annealing scheduler, Mosaic data augmentation, Optimal hyperparameters, Eliminate grid sensitivity, Multiple anchors for a single ground truth, Random training shapes, ...
Bag of Specials	CSP - Cross - stage partial connections, MiWRC - Multi - input weighted residual connections, Mish activation.	Spatial Pyramid Pooling SPP - block, PAN path - aggregation block, DfIoU - NMS, Mish activation, Self Attention Module SAM - block.

5. KẾT QUẢ ĐÁNH GIÁ

Để cung cấp tập dữ liệu cho YOLOv4, chúng phải ở định dạng mà YOLOv4 có thể sử dụng. Tập dữ liệu của nhóm nghiên cứu đã ở định dạng jpg. Tiếp theo nhóm nghiên cứu tiến hành gắn nhãn cho các hình ảnh và tạo tệp .txt cho mỗi hình ảnh. Sau đó nhóm nghiên cứu tạo 4 tệp: yolo.names, yolo.data, train.txt, valid.txt. Tệp yolo.names chứa tất cả các lớp của tập dữ liệu, tệp train.txt và valid.txt chứa các đường dẫn đến hình ảnh được dùng cho đào tạo và thử nghiệm. Nhóm nghiên cứu chia tập dữ liệu thành các tập huấn luyện và thử nghiệm, đồng thời huấn luyện mô hình trên tập huấn luyện và xác thực nó trên tập thử nghiệm.

Đối với mỗi mô hình phát hiện vật thể, sau khi đào tạo cần có những thang điểm để đánh giá sự chính xác của nó. Trong bài báo này nhóm nghiên cứu đánh giá mô hình dựa trên: loss function, precision, recall, IoU average, mAP, đánh giá trực quan, FPS với mAP là thước đo chính. Nếu dữ liệu đào tạo có chỉ số loss cao, điều đó có nghĩa là mô hình đang hoạt động kém hiệu quả và nó cần được đào tạo lâu hơn. Nếu chỉ số loss trên dữ liệu đào tạo thấp nhưng chỉ số mAP trên dữ liệu thử nghiệm không cao, điều đó có nghĩa là mô hình đang bị overfit và cần phải thêm nhiều dữ liệu hơn.

Hình 7 biểu diễn các thông số loss và mAP trong quá trình huấn luyện mô hình và kiểm thử với tập valid.txt. Trong quá trình huấn luyện nhóm nghiên cứu sẽ theo dõi biểu đồ loss và chỉ số mAP (sẽ được cập nhật sau mỗi 1000 vòng). Khi hai chỉ số ngừng có sự thay đổi lớn thì đó là lúc dừng huấn luyện mô hình. Có thể thấy ở Hình 7, từ vòng thứ 12800 trở đi thì chỉ số loss đã ngừng giảm, luôn giữ ở mức gần bằng 0,1, còn chỉ số mAP luôn dao động trong khoảng 91% đến 92%, dựa theo những thông tin có được từ biểu đồ, nhóm nghiên cứu đã quyết định dừng huấn luyện mô hình.



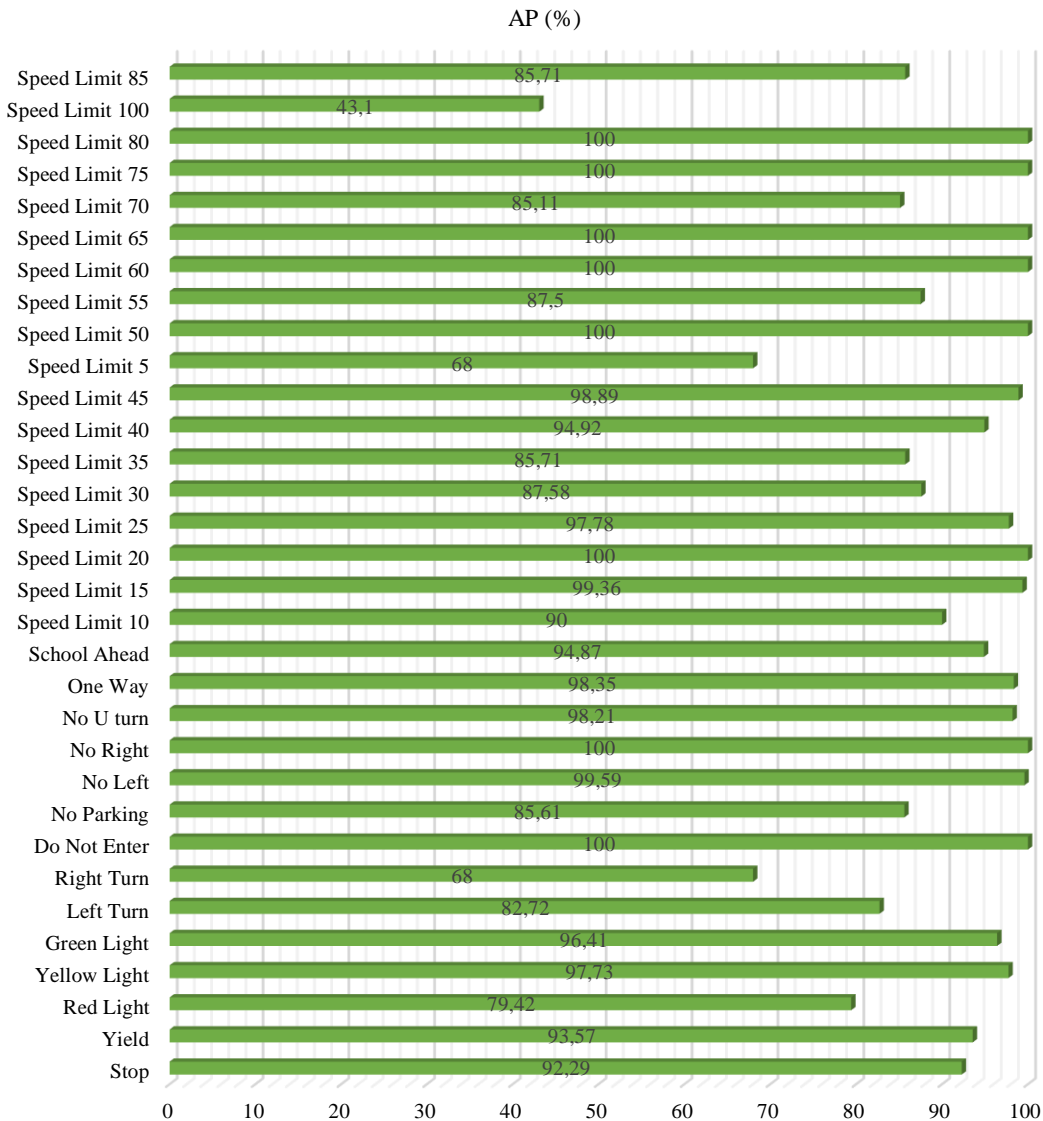
Hình 7. Biểu đồ thông số Loss và mAP khi huấn luyện

Trong bài toán phát hiện và nhận diện vật thể, các chỉ số AP (Average Precision) và mAP (Mean Average Precision) là các thước đo đánh giá được sử dụng phổ biến. Các chỉ số giúp xem xét cả về độ chính xác và độ bao quát của các dự đoán cho từng lớp. Trong phát hiện nhiều loại biển báo giao thông, mỗi loại biển báo giao thông có thể vẽ một đường cong theo độ nhớ (Recall) và độ chính xác (Precision). Diện tích dưới đường cong đó là AP và giá trị trung bình AP của nhiều lớp là mAP. Cách tính như sau:

$$recall = \frac{TP}{TP+FN} \quad (2)$$

$$precision = \frac{TP}{TP+FP} \quad (3)$$

Trong đó chỉ số TP (True Positive) thể hiện số lượng biển báo giao thông được phát hiện chính xác, chỉ số FN (False Negative) thể hiện số lượng biển báo giao thông bị bỏ sót và chỉ số FP (False Positive) thể hiện số lượng biển báo giao thông được phát hiện là sai, ngoài ra còn có thêm chỉ số TN (True Negative) thể hiện số lượng ảnh được nhận diện không có vật thể nào là đúng. Hình 8, bảng 3 và bảng 4 là chỉ số AP từng lớp và các thông số đạt được giữa các phiên bản của mô hình YOLOv4.



Hình 8. Chỉ số AP của các lớp

Dựa trên kết quả thu được ở Hình 8, Bảng 3 và Bảng 4 thì các lớp có chỉ số AP tương đối tốt, dao động từ 80 - 100%, với 8 lớp đạt 100%. Riêng 3 lớp Right Turn, Speed Limit 5 và Speed Limit 100 có chỉ số AP thấp từ 40 - 70%. Các chỉ số của các phiên bản YOLOv4 cơ bản có phần trăm precision đạt từ 91% đến 99%, phần trăm recall từ 97% đến 100%, mAP từ 95%

Nhận diện biển báo và tín hiệu đèn giao thông sử dụng YOLOv4 trên phần cứng Jetson TX2

đến 99% và average IoU từ 72% đến 81%, riêng phiên bản YOLOv4 - tiny có các chỉ số thấp hơn các phiên bản còn lại với các chỉ số precision, recall, mAP và average IoU lần lượt đạt 73%, 93%, 91% và 61%. Tuy nhiên ở Bảng 5 có thể thấy chỉ số FPS của YOLOv4 - tiny cao hơn hẳn các phiên bản khác với 31,2 FPS, cho thấy phiên bản này đã trade - off giữa độ chính xác và tốc độ nhận diện để có thể chạy trên các nền tảng CPU và GPU có cấu hình thấp.

Bảng 3. Chỉ số TP, FP, FN của các phiên bản

Phiên bản	TP	FP	FN	% precision	% recall
YOLOv4	440	6	0	99%	100%
YOLOv4 – Leaky	432	28	8	94%	98%
YOLOv4 – Tiny	385	145	31	73%	93%
YOLOv4 – Mish	420	40	12	91%	97%

Bảng 4. So sánh chỉ số mAP, IoU và FPS giữa các phiên bản YOLOv4

Phần cứng	Phiên bản	Chỉ số mAP	Chỉ số Average Iou
Jetson TX2	YOLOv4	0,99	0,8
Jetson TX2	YOLOv4 – Leaky	0,97	0,74
Jetson TX2	YOLOv4 – Tiny	0,91	0,61
Jetson TX2	YOLOv4 – Mish	0,95	0,72
Google Colab	YOLOv4	0,96	0,81

Nhóm nghiên cứu cũng thử nghiệm các mô hình trên video và nhận được phản hồi tốt khi mô hình có thể phát hiện các biển báo, tín hiệu đèn giao thông và phân loại chúng. Bảng 5 là chỉ số FPS mà nhóm nghiên cứu đã ghi nhận được khi thử nghiệm trên GPU phần cứng Jetson TX2 và GPU 12GB do Colab cung cấp. Trên Colab, mô hình có chỉ số FPS trong khoảng 30 - 43 FPS, trên phần cứng Jetson TX2 đạt khoảng 3 - 4 FPS đối với các phiên bản YOLOv4 cơ bản và 30 - 32 FPS đối với phiên bản YOLOv4 - tiny.

Bảng 5. So sánh chỉ số FPS giữa các phiên bản YOLOv4 trên Jetson TX2 và Colab

Phần cứng	Phiên bản	Chỉ số Average FPS
Jetson TX2	YOLOv4	4
Google Colab	YOLOv4	36,2
Jetson TX2	YOLOv4 – Leaky	4
Google Colab	YOLOv4 – Leaky	33,5
Jetson TX2	YOLOv4 – Tiny	31,2
Google Colab	YOLOv4 – Tiny	42,5
Jetson TX2	YOLOv4 – Mish	3,8
Google Colab	YOLOv4 – Mish	31,7

Đối với các phiên bản YOLOv4 cơ bản, mô hình đạt được thông số mAP tốt từ 96% đến 98%, tuy nhiên các mô hình này có lượng tham số cao dẫn tới dung lượng lớn và tốc độ phát hiện vật thể không cao, không phù hợp với các bài toán nhúng trên các phương tiện tự hành vốn ưu tiên dung lượng thấp. Đối với phiên bản YOLOv4 - tiny tuy có lượng tham số thấp nên

độ chính xác sẽ không bằng nhưng cũng đạt yêu cầu của đề tài, tầm 91%, bù lại YOLOv4 - tiny có dung lượng thấp và tốc độ phát hiện vật thể cao, phù hợp áp dụng trên các phương tiện tự hành.

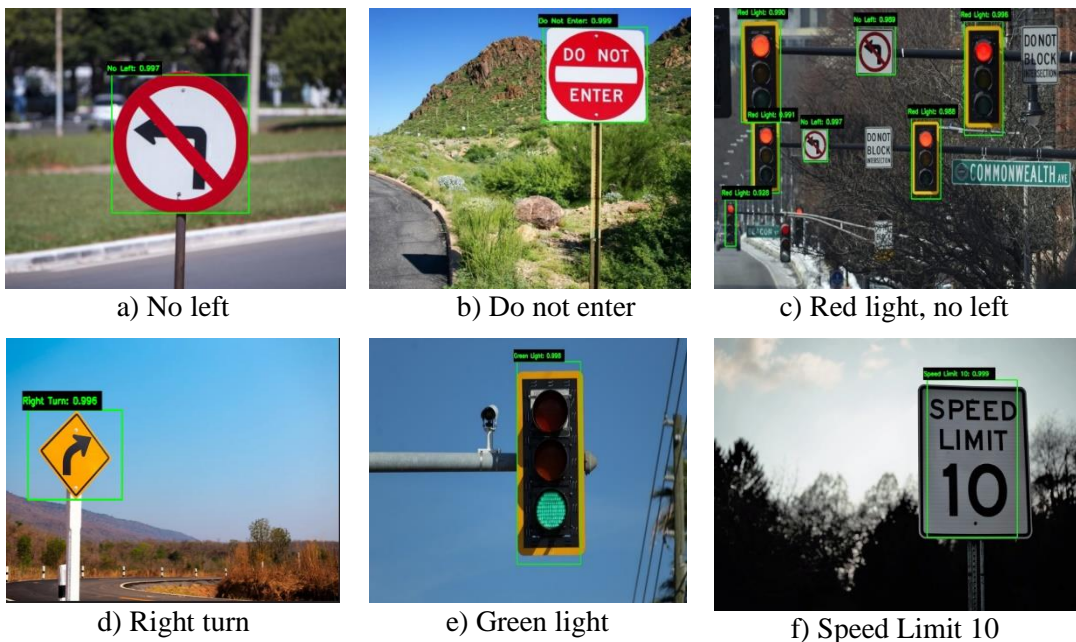
Ngoài ra, nhóm nghiên cứu còn tính toán thêm chỉ số F1, là chỉ số để đánh giá một mô hình là tốt hay không. Chỉ số được tính toán bằng công thức:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

Precision và Recall là các thông số được sử dụng để tính độ chính xác cho quá trình đánh giá kết quả của hệ thống. Precision đại diện cho độ tin cậy của mô hình, thông số cho biết rằng bao nhiêu dự đoán đúng được đưa ra trong tổng số các dự đoán là thật sự đúng. Recall đại diện cho độ nhạy của mô hình, thông số cho biết mô hình có thể dự đoán đúng được bao nhiêu trong tổng số đúng trong dữ liệu. Và chỉ số F1 được sử dụng để cân bằng cả hai, mô hình có chỉ số F1 càng cao thì càng hiệu quả.

Bảng 6. So sánh chỉ số F1 khi thay đổi các thông số

Phần cứng	Jetson TX2	Jetson TX2	Jetson TX2	Jetson TX2	Google Colab
Phiên bản	YOLOv4	YOLOv4	YOLOv4 – Tiny	YOLOv4 – Tiny	YOLOv4
Learning rate	0,001	0,001	0,001	0,0001	0,001
Batch size	32	64	64	64	32
Subdivisons	16	64	64	32	16
Thời gian train	102 h	110 h	32 h	37 h	21 h
F1 Score	0,98	0,98	0,85	0,81	0,99
Loss	0,6	0,58	0,1	0,24	0,57



Hình 9. Các hình ảnh nhận diện thực tế

6. KẾT LUẬN

Trong bài báo này, nhóm đã nghiên cứu và phát triển mô hình nhận diện biển báo và các tín hiệu đèn giao thông sử dụng thuật toán YOLOv4. Mô hình được triển khai trên phần cứng Jetson TX2 dựa trên tập dữ liệu được tạo độc lập. Mô hình của nhóm nghiên cứu hoạt động tốt trong việc nhận diện thời gian thực với chỉ số mAP và average FPS tốt nhất lần lượt là 91% và 31,2 FPS thuộc về phiên bản YOLOv4 – tiny. Mô hình có thể nhận diện các biển báo và tín hiệu đèn với tỷ lệ nhận dạng chính xác khá cao, riêng với ba lớp Right Turn, Speed Limit 5 và Speed Limit 100 có tỷ lệ nhận dạng chính xác thấp. Nguyên nhân có thể do số lượng hình ảnh trong tập dữ liệu có sự chênh lệch lớn giữa các lớp, lớp Right Turn, Speed Limit 5 và Speed Limit 100 chỉ có số lượng lần lượt là 34 ảnh, 24 ảnh và 8 ảnh trong tổng số 1500 ảnh dữ liệu. Điều này dẫn đến sự mất cân bằng dữ liệu giữa các lớp, từ đó phần nào gây sự ảnh hưởng đến độ chính xác của mô hình. Trong tương lai, nhóm nghiên cứu sẽ xây dựng một tập dữ liệu biển báo giao thông gắn liền với thực tế ở Việt Nam với số lượng hình ảnh lớn, đầy đủ các điều kiện thời tiết, ánh sáng và đồng đều cho mỗi lớp để có thể cải thiện thêm mô hình của nhóm.

TÀI LIỆU THAM KHẢO

1. Jianming Zhang, Manting Huang, Xiaokang Jin, Xudong Li - A Real-Time Chinese Traffic Sign Detection Algorithm Based on Modified YOLOv2, *Algorithms* **10** (2017) 1-5.
2. Shuai Liu, Hongxin Li - Application of Chinese Traffic Sign Detection Based on Yolov4, in: *International Conference on Computer and Communications (ICCC)*, Chengdu (2021).
3. Si Chen, Fang Lv, Ping Huo - Improved detection of yolov4 sunflower leaf diseases, in: *International Symposium on Computer Engineering and Intelligent Communications (ISCEIC)*, Nanjing (2021).
4. Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao - YOLOv4: Optimal Speed and Accuracy of Object Detection, *arXiv* **2004** (2020) 1-16.
5. Jie Liu, Lizhi Liu - Helmet Wearing Detection Based on YOLOv4-MT, in: *International Conference on Robotics, Control and Automation Engineering (RCAE)*, Wuhan (2021).
6. Xun Yin, Li Chen, Xiaoyun Zhang, Zhiyong Gao - Object Detection Implementation and Optimization on Embedded GPU System, in: *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Valencia (2018).
7. Artiom Basulto-Lantsova, Jose A. Padilla-Medina, Francisco J. Perez-Pinal, Alejandro I. Barranco-Gutierrez - Performance comparative of OpenCV Template Matching method on Jetson TX2 and Jetson Nano developer kits, in: *Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas (2020).
8. Sützen, Ahmet Ali and Duman, Burhan and Şen, Betül - Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN, in: *International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Turkey (2020).
9. NVIDIA - Unveils roadmap for 2015 - 2018, in *NVIDIA GTC*, San Jose (2014).
10. Chung Yu Wang, Royce Cheng-Yue - Traffic Sign Detection using You Only Look Once Framework, *Stanford University* 263 (2016) 1-6.

11. Rishabh Singh, Momin Danish, Vipul Purohit, Ashraf Siddiqui - Traffic Sign Detection using YOLOv4, International Journal of Creative Research Thoughts **9** (2021) 1-6.
12. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi - You Only Look Once: Unified, Real-Time Object Detection, arXiv **1506** (2016) 1-7.
13. Zhi Zhang, Tong He, Hang Zhang, Zhongyue Zhang, Junyuan Xie, Mu Li - Bag of Freebies for Training Object Detection Neural Networks, arXiv **1902** (2019) 1-8.

ABSTRACT

IMPLEMENTATION OF TRAFFIC SIGNS DETECTION USING YOLOv4 ON JETSON TX2

Bui Quoc Tu^{1*}, Nguyen Huy Hoang¹, Truong Quang Phuc¹
Le Quang Binh¹, Ho Nhut Minh²

¹*HCMC University of Technology and Education*

²*Posts and Telecommunications Institute of Technology, HCMC*

*Email: *tubui2263@gmail.com*

Traffic signs and traffic lights detection is an important aspects of autonomous vehicles to prevent and reduce accidents. In this paper, the YOLOv4 model, one of the popular architectures of deep learning used to detect and recognize traffic signs and traffic lights has been implemented on NVIDIA Jetson TX2 hardware. In general, the YOLOv4 model has many variants with different structures and parameters, the author focuses on comparison and evaluation to select the right variant for the data set. Specifically, the author chooses NVIDIA's Jetson TX2 as a hardware platform to take advantage of the GPU's computing power to optimize data training time. In particular, the usage data is generated independently including 32 classes for training and testing. The method used achieved 91% mAP and 31.2 FPS on the test dataset.

Keywords: Detection, traffic signs, traffic lights, YOLOv4, Jetson TX2.