PHÁT HIỆN PHƯƠNG TIỆN GIAO THÔNG VÀ NGƯỜI ĐI BỘ DỰA VÀO THUẬT TOÁN HỌC SÂU Ở CÁC HỆ THỐNG HỖ TRỢ LÁI THÔNG MINH

DEEP LEARNING-BASED VEHICLES AND PEDESTRIAN DETECTION IN INTELLIGENT DRIVER ASSISTANCE SYSTEMS

Vũ Hồng Sơn[1,*]

DOI: https://doi.org/10.57001/huih5804.2023.106

TÓM TẮT

Để phát triển một hệ thống giao thông thông minh và an toàn, việc đầu tiên chúng ta cần quan tâm là xây dựng hệ thống trợ lái thông minh ADAS. Hệ thống trợ lái ADAS thường yêu cầu các ràng buộc cực đại về tốc độ xử lý nhanh và hiệu suất phát hiện chính xác. Tuy thế mà có nhiều các ràng buộc đang đặt ra cho hệ thống, cụ thể là: do những biến đổi về ánh sáng nền, cấu trúc, tình trạng bị tắc nghẽn từng phần (có nhiều phần xuất hiện trong cùng ngữ cảnh), đối tượng và camera cùng di chuyển, và ở các ngữ cảnh phức tạp... Ngoài ra, một thử thách cực đại cho hệ thống là yêu cầu đáp ứng thời gian thực. Để cải thiện các ràng buộc cực đại này, chúng tôi đề xuất một mô hình sử dụng thuật toán học sâu. Trước tiên, bài báo sử dụng mô hình YOLO (You Only Look One), ngoài ra để bổ sung cho tập dữ liệu đào tạo, chúng tôi đã phân loại và thu thập tập dữ liệu mẫu phù hợp với giao thông Việt Nam. Sau đó, máy tính nhúng NVIDIA Jetson TX2 đã được sử dụng để thực hiện các thí nghiệm. Các kết quả đạt được đã chứng minh rằng, công việc đề xuất có khả năng tăng tốc độ xử lý ít nhất 1,6 lần với tỷ lệ phát hiện đạt 90% cho hệ thống camera tĩnh; và tăng tốc độ ít nhất 1,36 lần với tỷ lệ phát hiện đạt 90% cho hệ thống camera động với các ảnh có độ phân giải cao 1280x720 pixel.

Từ khóa: *Hệ thống trợ lái thông minh ADAS, mô hình YOLO, thuật toán học sâu và trí tuệ nhân tạo.*

ABSTRACT

In order to build a traffic safety system, we first need to develop ADASs. These systems normally require real-time and reliable detection performance. However, moving vehicles and pedestrian detection is critical requirement due to their challenges in the real-world environments such as complicated background, shadow, partial occlusion, articulation and illumination variations. Besides, one of the most important challenges in ADASs is real-time requirement. This paper proposes a model using deep-learning algorithm in order to increase accuracy and processing time for ADASs. Accordingly, we first propose the YOLO (You Only Look One) model, moreover in order to improve detection performance we add sample datasets for traing model. Experimental results are then conducted in a NVIDIA Jetson TX2 embedded computer. Achievable results prove that the proposed work can speed up processing time of at least 1.6x with detection rate of 90% for static cameras; and speed up processing time of at least 1.36x with detection rate of 90% in high resolution images (1280x720 pixel) for moving cameras.

Keywords**:** *Advanced driver assistance systems, YOLO recognition model, deep-learning algorithm and artificial intelligence.*

[1]Khoa Điện - Điện tử, Trường Đại học Sư phạm Kỹ thuật Hưng Yên

[*]Email: hongson.ute@gmail.com

Ngày nhận bài: 15/3/2023

Ngày nhận bài sửa sau phản biện: 29/5/2023

Ngày chấp nhận đăng: 15/6/2023

1. GIỚI THIỆU

Tai nạn đường bộ đang là một vấn nạn toàn cầu và là nhân tố trực tiếp ảnh hưởng tới sự gia tăng các phương tiện giao thông trên thế giới. Mỗi năm có hàng chục triệu các vụ tai nạn giao thông trên khắp thế giới với khoảng 10 triệu người trở thành nạn nhân, trong số đó có từ hai đến ba triệu người bị thương tật vĩnh viễn [1].

Cả cộng đồng khoa học và ngành công nghiệp ô tô đang tập trung phát triển và xây dựng các hệ thống cảnh báo/bảo vệ nhằm cải thiện hệ thống giao thông thông minh và an toàn. Ở đó, các hệ thống trợ lái ADAS đang trở thành một phạm vi nghiên cứu tích cực nhằm cải tiến các tính năng thông minh và an toàn cho các phương tiện giao thông.

Để phát triển một hệ thống giao thông thông minh và an toàn, việc đầu tiên chúng ta cần quan tâm là xây dựng hệ thống trợ lái thông minh ADAS. Ngoài việc phát triển cơ sở hạ tầng hiện đại và đồng bộ thì việc phát triển các công nghệ mới cho các hệ thống trợ lái thông minh cũng là một trong những nhân tố quan trọng cần được quan tâm. Thực tế chứng minh rằng, với các phương tiện được tích hợp các công nghệ trợ lái như phát hiện, nhận dạng các phương tiện xung quanh, từ đó hệ thống có thể đưa ra các cảnh báo tới người điều khiển hoặc tác động vào hệ thống điều khiển trung tâm giúp cho việc lái xe an toàn hơn. Những năm gần đây, với sự phát triển của kĩ thuật học sâu (Deep

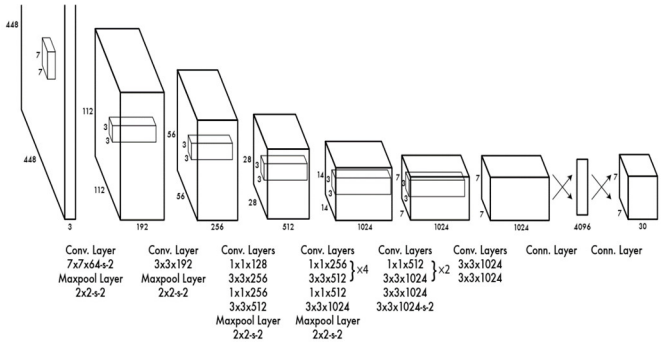
Learning) được các tác giả đề xuất [2-8], các kĩ thuật mới này không những giúp hệ thống cải thiện hiệu suất, giảm các cảnh báo lỗi mà còn có khả năng nhận dạng và phân loại nhiều lớp đối tượng. Ngoài ra với những công nghệ mới trong cấu trúc phần cứng như tốc độ xử lý nhanh, đa nhiệm và việc tích hợp nhiều lõi GPU trên một đơn vị chip, điều này giúp các hệ thống nhúng có thể thực hiện các thuật toán phức tạp on-board.

Thử thách chính của hệ thống là làm sao có thể phát hiện được các đối tượng quan tâm chính xác on-board. Điều này là do sự xuất hiện hay biến đổi của các đối tượng (ví dụ như quần áo, kích thước, tỉ lệ, hình dạng động và tình trạng tắc nghẽn từng phần) cùng môi trường phi cấu trúc và yêu cầu chi phí tính toán lớn để tìm phân vùng tọa độ các đối tượng đang di chuyển, từ đó phân loại và nhận dạng các đối tượng quan tâm.

Tác giả đề xuất một mô hình sử dụng trí tuệ nhân tạo để tăng hiệu suất phát hiện và tăng tốc độ tính toán cho hệ thống ADAS. Trước tiên, bài báo sử dụng mô hình YOLO (You Only Look One) [9], ngoài ra để bổ sung cho tập dữ liệu đào tạo, tác giả đã phân loại và thu thập tập dữ liệu mẫu phù hợp với giao thông Việt Nam. Sau đó, máy tính nhúng NVIDIA Jetson TX2 đã được sử dụng để thực hiện các thí nghiệm. Các kết quả đạt được đã chứng minh rằng, công việc đề xuất có khả năng tăng tốc độ xử lý ít nhất 1,6 lần với tỷ lệ phát hiện đạt 90% cho hệ thống camera tĩnh; và tăng tốc độ ít nhất 1,36 lần với tỷ lệ phát hiện đạt 90% cho hệ thống camera động với các ảnh có độ phân giải cao 1280x720 pixel.

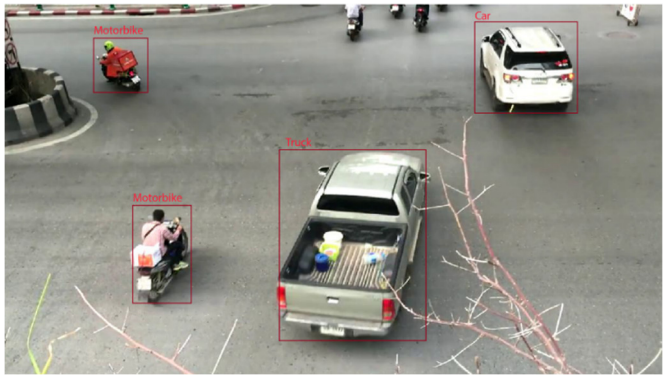
2. CÁC NGHIÊN CỨU LIÊN QUAN

2.1. Mô hình YOLO



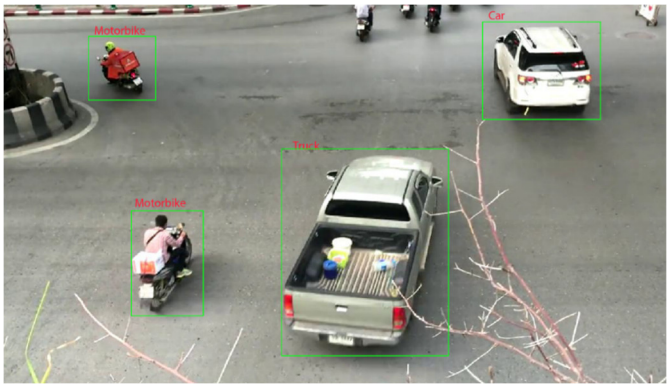
Hình 1. Mô hình thuật toán YOLO [9]

Các tác giả trong [9] đề xuất một thuật toán dựa vào cấu trúc mạng nơ ron tích chập với tên gọi YOLO, từ việc kết hợp giữa các lớp kết và các lớp tích chập. Ở đó các lớp tích chập sẽ rút trích ra các đặc trưng của ảnh đầu vào, còn các lớp kết nối sẽ dự báo xác xuất và tọa độ của đối tượng quan tâm. Cùng với mô hình YOLO, các tác giả trong [9] cũng đưa ra một cách thức để đánh giá hiệu suất thông qua tham số ground truth. Ground truth là tọa độ của đối tượng quan tâm, nó được cung cấp trong tập dữ liệu đào tạo của hệ thống, được kiểm tra và đánh giá. Với các ứng dụng thị giác máy tính, ground truth sẽ được biểu diễn bởi hình ảnh, các lớp của đối tượng và các khung hình bao quanh nó.



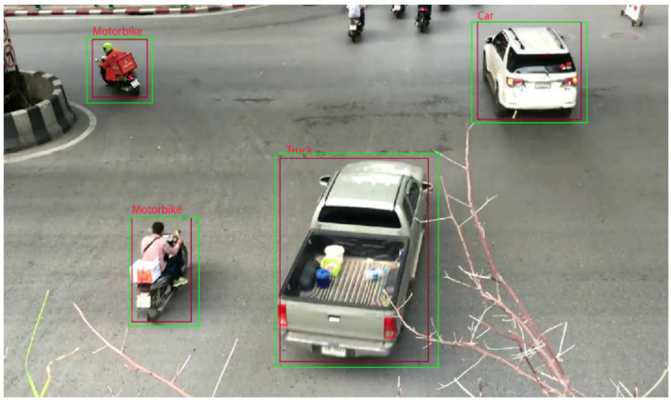
Hình 2. Minh họa tọa độ, khung hình ground truth của các đối tượng

Hình 2 minh họa các khung hình chữ nhật bao quanh ground truth, chúng được xác định trên ảnh thông qua tập dữ liệu kiểm tra và huấn luyện. Giả định rằng, tọa độ của các đối tượng quan tâm như hình 2, sau đó mô hình sẽ được đào tạo trên tập dữ liệu huấn luyện, ngay sau khi mô hình được đào tạo xong giai đoạn kiểm tra sẽ được thực hiện. Các hình ảnh gốc chứa đối tượng quan tâm sẽ được mô hình phát hiện và trả về thông tin tọa độ tương ứng của các đối tượng trên mỗi khung hình, như được minh họa ở hình 3.

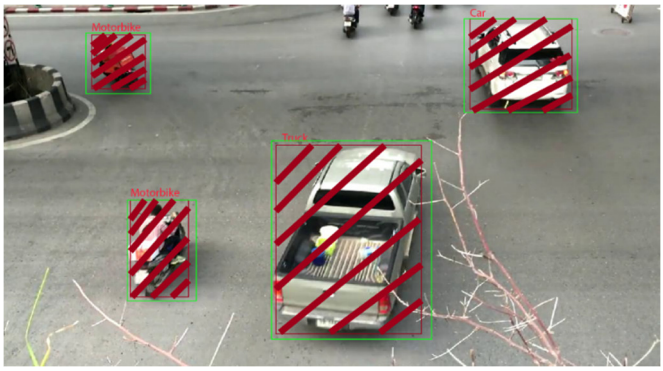


Hình 3. Minh họa kết quả dự đoán

Để tính được độ chính xác của một hộp giới hạn ground truth, chúng ta sẽ sử dụng chỉ số IoU (Intersection over Union). IoU là tỷ lệ mức độ giao nhau giữa khung hình ground truth và khung hình dự báo. Hình 4 và 5 minh họa lần lượt dự đoán vị trí của đối tượng và ground truth, vùng giao nhau dự đoán vị trí của đối tượng và ground truth.

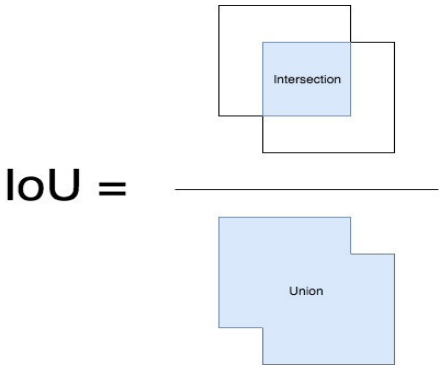


Hình 4. Minh họa dự đoán vị trí của đối tượng và ground truth



Hình 5. Vùng giao nhau giữa dự đoán vị trí của đối tượng và ground truth

IoU sẽ được tính như thể hiện trong hình 6.



Hình 6. Mô tả cách tính IoU

IoU được định nghĩa là thương số của phần diện tích giao nhau giữa hai khung hình ground truth và khung hình dự báo so với phần tổng diện tích giao và không giao của hai khung hình.

2.2. Phương pháp huấn luyện

Trong phần này, chúng tôi trình bày cách thức xây dựng và tối ưu hàm mất mát dựa trên mô hình YOLO đã được đề xuất trong [9], ngoài ra để bổ sung cho tập dữ liệu đào tạo, chúng tôi đã phân loại và thu thập tập dữ liệu mẫu phù hợp với giao thông Việt Nam.

Tác giả [9] sử dụng hàm tổng bình phương lỗi giữa hai tham số mong muốn và tham số dự đoán. Tổng quát, chúng ta có thể biểu diễn hàm mất mát như sau:

$$\sum_{i=0}^{S^2} 1_{ij}^{obj} \sum_{c \in Classes} (p_i(c) - \hat{p}_i(c))^2 \quad (1)$$

Trong đó:

- $1_i^{obj} = 0$ nếu không có đối tượng và bằng 1 nếu có bất kỳ đối tượng nào
- $\hat{p}_i(c)$: Hàm xác suất

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + [(y_i - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \quad (2)$$

Trong đó:

- $1^{obj}_{ij} = 0$ nếu không có đối tượng và bằng 1 nếu có bất kỳ đối tượng nào
- λ_{coord}: Trọng số của hàm mất mát
- $\hat{x}, \hat{y}, \hat{w}, \hat{h}$: Vị trí của đối tượng

Để giảm các cảnh báo lỗi: trong trường hợp khung hình bao quanh đối tượng lớn hơn hoặc nhỏ hơn khung hình thực chứa đối tượng. Mô hình YOLO đề xuất hàm lấy căn bậc 2 chiều rộng và chiều cao của tọa độ khung hình bao quanh đối tượng. Ngoài ra, để giảm sai số của hộp giới hạn chứa đối tượng, mô hình YOLO còn đề xuất phép nhân hàm mất mát cục bộ với λ_{coord} (mặc định bằng 5).

$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (C_i - \hat{C}_i)^2 \quad (3)$$

Trong đó:

- $\hat{C}_i$: Độ tin cậy của khung hình bao quanh đối tượng j trong ô i
- $1_{ij}^{obj} = 0$ nếu không có đối tượng và bằng 1 nếu có bất kỳ đối tượng nào
- Khi trong hộp giới hạn không chứa đối tượng thì Confidence loss là:

$$\lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (C_i - \hat{C}_i)^2 \quad (4)$$

Trong đó:

- 1_{ij}^{noobi} là phần bổ sung cho 1_{ij}^{obj}
- $\hat{C}_i$: Độ tin cậy của khung hình bao quanh đối tượng j trong ô i
- λ_{noobi} : Trọng số

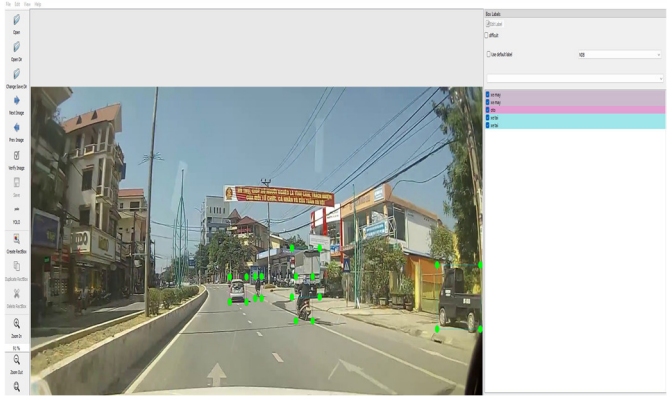
Số lượng đối tượng trong khung hình thường nhỏ hơn số lượng trong môi trường thực tế, vì vậy mà nhiều hộp có giá trị trống. Để giảm sai số cho hàm dự báo này, công việc trong [9] đề xuất khởi tạo thêm một tham số λ_{noobi} (mặc định bằng 0,5).

Từ phương trình (1)-(4), chúng ta có:

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2] + \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(C_i - \hat{C}_i)^2 + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(C_i - \hat{C}_i)^2 + \sum_{i=0}^{S^2} 1_{ij}^{obj} \sum_{c \in Classes} (p_i(c) - \hat{p}_i(c))^2 \quad (5)$$

2.3. Chuẩn bị dữ liệu phục vụ đào tạo

Để đào tạo mô hình được đề xuất, hai nguồn dữ liệu chính bao gồm tập các ảnh được trích xuất từ camera hành trình, camera cố định tại các ngã ba, ngã tư… đã được chúng tôi tập hợp. Tập dữ liệu này bao gồm 1199 ảnh có độ phân giải 1280x720 pixel, trong đó 999 ảnh được sử dụng để huấn luyện mô hình và 200 ảnh được sử dụng để kiểm tra và đánh giá.

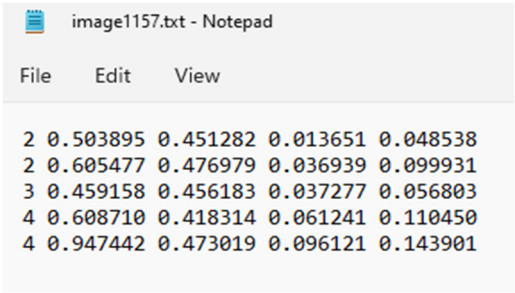


Hình 7. Tạo nhãn cho các layer

Đầu tiên chúng tôi sử dụng Makesense.ai để tiền xử lý các tập ảnh mẫu, theo cấu trúc:

[thứ tự của lớp đối tượng] [tọa độ x] [tọa độ y] [chiều rộng của đối tượng] [chiều cao của đối tượng]

Với mỗi ảnh đầu ra của tập dữ liệu đào tạo được định nghĩa dưới dạng file .txt.



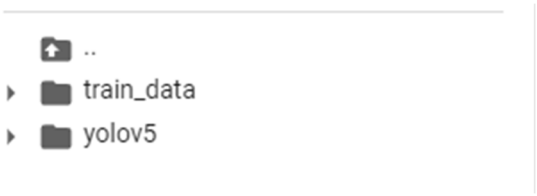
Hình 8. Dữ liệu file nhãn layer

Trong đó ID tương ứng: lớp 0: NĐB, lớp 1: xe đạp, lớp 2: xe máy, lớp 3: ô tô, và lớp 4: xe tải.

Ngoài ra, để giảm thời gian huấn luyện, một phương pháp sử dụng kỹ thuật học chuyển tiếp của mô hình đã được đào tạo từ trước là được sử dụng. Sau đó, tập dữ liệu đào tạo riêng được bổ sung với bối cảnh chuyên biệt phù hợp với bài toán ở Việt Nam là được đào tạo.

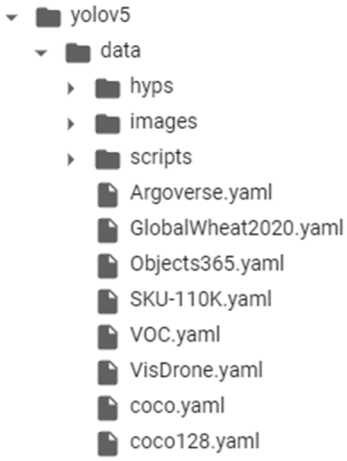
2.4. Huấn luyện mô hình được đề xuất

Chuẩn bị hình ảnh và các tệp .txt tương ứng với từng hình ảnh đã tạo bằng Makesense.ai. Dữ liệu gồm hai phần riêng biệt là train và val, trong đó thư mục train chứa các hình ảnh đào tạo, thư mục val chứa các hình ảnh kiểm tra sau khi quá trình đào tạo thành công. Truy cập Google Colab và Git Clone dự án YOLOv5 về máy chủ Google Colab. Hệ thống sau đó sẽ tự động cài các thư viện cần thiết cho quá trình huấn luyện.

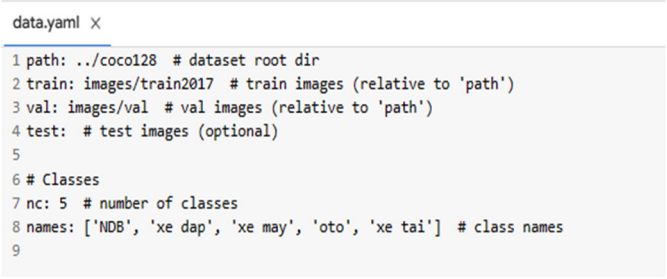


Hình 9. Thư mục YOLOv5 và train_data

Tải thư mục train_data đã tạo lên máy chủ Colab.

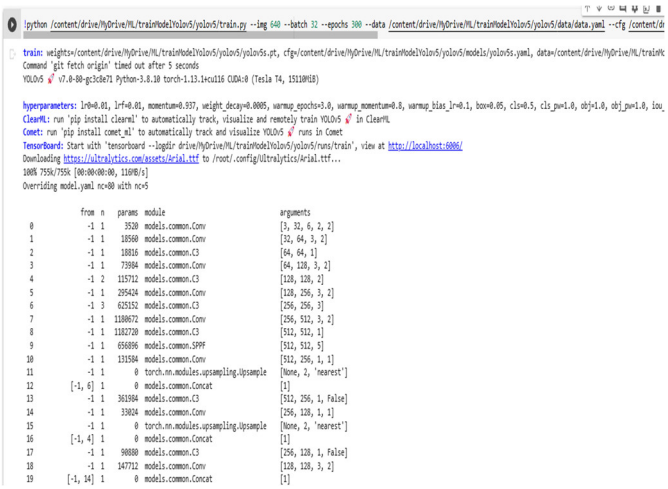


Hình 10. Đường dẫn coco128.yaml



Hình 11. Dữ liệu file coco128.yaml

Tìm và chỉnh sửa các đường dẫn cho file coco128.yaml nhằm nhận diện được vị trí dữ liệu đã tải lên cũng như xác định số đối tượng và tên tương ứng cần đào tạo.



Hình 12. Quá trình đào tạo mô hình

Số lượng mẫu và lớp đối tượng cần được phát hiện sẽ quyết định thời gian đào tạo.

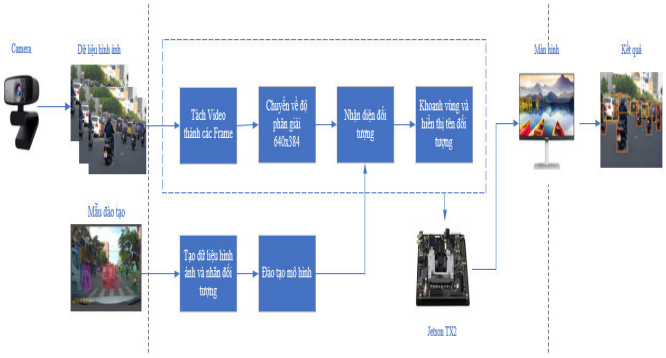
```
Validating drive/MyDrive/ML/trainModelYolov5/yolov5/runs/train/Chi3_results/weights/best.pt...
Fusing layers...
YOLOv5s summary: 157 layers, 7023610 parameters, 0 gradients, 15.8 GFLOPs
               Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 4/4 [00:03<00:00,  1.22it/s]
                 all        199       1517      0.727      0.397      0.453      0.202
                 NDB        199        341      0.659      0.592      0.571       0.19
              xe dap        199         21          1          0      0.147     0.0842
              xe may        199        421      0.557      0.618      0.593      0.283
                 oto        199        649      0.651       0.54       0.58      0.268
              xe tai        199         85       0.77      0.237      0.376      0.184
```

Hình 13. Minh họa kết quả đào tạo

Hình 13 minh họa các tham số của mô hình được đề xuất sau khi mô hình được đào tạo thành công. Sau đó file best.pt sẽ được sử dụng để làm file mô hình đào tạo trong quá trình nhận diện.

3. MÔ HÌNH ĐƯỢC ĐỀ XUẤT

Mô hình hệ thống được đề xuất là được mô tả chi tiết ở hình 14. Đầu tiên dữ liệu đầu vào từ camera được tách thành các khung hình và được chuyển về độ phân giải 640x384 pixel, đây là độ phân giải tối ưu để tăng tốc độ cho hệ thống cũng như đảm bảo chất lượng ảnh cho giai đoạn nhận dạng và phân loại đối tượng quan tâm ở các khung hình. Sau đó dữ liệu sẽ được trích xuất và được xử lý bởi mô hình đã được đào tạo và huấn luyện, từ đó mô hình sẽ đưa ra các dự báo về các đối tượng quan tâm ở các khung hình. Đầu ra bao gồm tọa độ đối tượng, id đối tượng sẽ được khoanh vùng và gắn tên đối tượng tương ứng.



Hình 14. Đề xuất mô hình hệ thống trong bài báo

4. ĐÁNH GIÁ VÀ SO SÁNH

Để đánh giá hiệu suất của hệ thống, một tập hợp bộ dữ liệu với các kịch bản: ban ngày, ban đêm với các ràng buộc bao gồm sự biến đổi của người đi bộ (ví dụ như quần áo, kích thước, tỉ lệ, hình dạng động và tình trạng tắc nghẽn từng phần) cùng môi trường phi cấu trúc, nhiều phương tiện giao thông: xe máy, ô tô…, với những biến đổi về ánh sáng, bóng nền, ở các môi trường phức tạp...

Đầu tiên, tác giả đánh giá các kĩ thuật được đề xuất trên mô hình truyền thống với các kịch bản: chỉ có một đối tượng di chuyển ở các điều kiện cực đại như sự thay đổi ánh sáng, bóng nền, cùng bối cảnh phức tạp bao gồm các tòa nhà, xe, bầu trời… sau đó mô hình được đánh ở giá ở kịch bản nhiều phương tiện tham gia giao thông ở cả ban ngày và ban đêm. Đầu vào là các ảnh có độ phân giải cao 1280x720 pixel với tổng số khung hình là 2032 để đánh giá hiệu suất. Máy tính nhúng NVIDIA Jetson TX2 đã được sử dụng để thực hiện các thí nghiệm.

Ngoài ra, tác giả định nghĩa 3 tham số bao gồm tỷ lệ phát hiện - DR, tỷ lệ lỗi - MR và tỷ lệ phát hiện lỗi - FDR để đánh giá hiệu suất của hệ thống.

$$DR = \frac{TP}{TOC} * 100\% \quad (6)$$

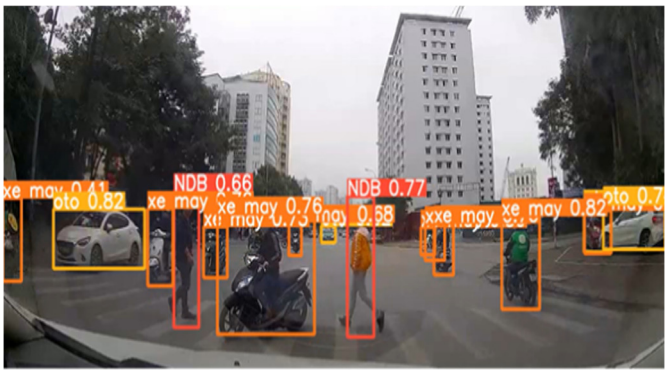
$$MR = 100\% - DP \quad (7)$$

$$FDR = \frac{FP}{TP + FP} * 100\% \quad (8)$$

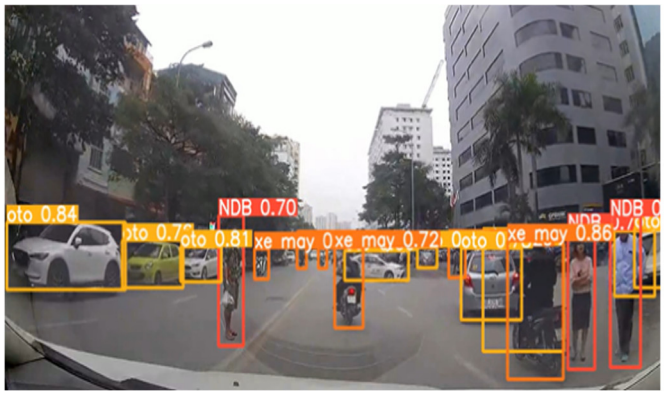
Trong phương trình (6)-(8), tổng số các dự đoán đối tượng: TOC (Total Object Collections), các dự đoán đúng biểu diễn cho đối tượng được phát hiện: TP (True Positives), và các dự đoán lỗi biểu diễn cho số lượng các mẫu không phải đối tượng quan tâm: FP (False Positives).

Bảng 1. So sánh hiệu suất của công việc đề xuất với các công việc hiện tại

Dữ liệu đầu vào	Tổng số (khung hình)	Mô hình đào tạo	DR (%)	MR (%)	FDR (%)	Thời gian (FPS)
Camera cố định	500	YOLOv5s	50	50	0	10
		YOLOv5m	80	20	0	4
		Proposed	90	10	0	16
Camera di động (camera hành trình)	1532	YOLOv5m	80	20	0	11
		Proposed	90	10	0	15



(a)



(b)

Hình 15. Minh họa dự đoán các đối tượng được nhận dạng và phân loại

Bảng 1 mô tả chi tiết các kết quả đạt được, từ các kết quả ở bảng 1 nhận thấy rằng, công việc đề xuất có khả năng tăng tốc độ xử lý ít nhất 1,6 lần với tỷ lệ phát hiện đạt

90% cho hệ thống camera tĩnh, với tỷ lệ cảnh báo lỗi giảm ít nhất 10% và tỷ lệ phát hiện tăng 10%; và tăng tốc độ ít nhất 1,36 lần với tỷ lệ phát hiện đạt 90% cho hệ thống camera động, với tỷ lệ phát hiện tăng 10% khi so sánh với mô hình YOLOv5m.

5. KẾT LUẬN

Bài báo đã phát triển một mô hình kết hợp giữa học máy và các kĩ thuật được đề xuất nhằm nâng cao hiệu suất và giảm tỷ lệ cảnh báo lỗi cho các hệ thống hỗ trợ lái ADAS. Thông qua phương pháp được đề xuất, các kĩ thuật mới đã được thực nghiệm, đánh giá trên nền tảng máy tính nhúng Jestson TX2 với nhiều điều kiện môi trường, bối cảnh khác nhau, số lượng tập dữ liệu đủ lớn để có được độ tin cậy cao. Mô hình được đề xuất rất có tiềm năng trong việc thực hiện ở các ứng dụng thực tế chỉ có camera bao gồm những đối tượng chuyển động như các hệ thống giám sát thông minh, hệ thống hỗ trợ lái thông minh và robotics,...

LỜI CẢM ƠN

Công trình này được hỗ trợ bởi Bộ Giáo dục và Đào tạo thông qua đề tài mã số: B2020-SKH-02

TÀI LIỆU THAM KHẢO

[1]. P. Dollar, C. Wojek, B. Schiele, P. Perona, 2012. *Pedestrian detection: An evaluation of the state of the art.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 4, pp. 743–761.

[2]. R. Girshick, J. Donahue, T. Darrell, J. Malik. Rich, 2014. *Feature Hierarchies for Accurate Object Detection and Semantic Segmentation*. in Proc. Comput. Vis. Patt. Recognit. (CVPR).

[3]. R. Girshick, 2015. *Fast R-CNN*. in Proc. Comput. Vis. Patt. Recognit. (CVPR), pp. 1–9.

[4]. S. Ren, K. He, R. Girshick, J. Sun, 2016. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.* in Proc. Comput. Vis. Patt. Recognit. (CVPR), pp. 1–14.

[5]. H. Mao, S. Yao, T. Tang, B. Li, J. Yao, Y. Wang, 2018. *Towards Real-Time Object Detection on Embedded Systems*. IEEE Transactions on Emerging Topics in Computing, vol. 6, no. 3, pp. 417 – 431.

[6]. A. F. Agarap, 2018. *Deep Learning using Rectified Linear Units (ReLU)*. arXiv:1803.08375.

[7]. G. S. W. Luger, 2020. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving.* Benjamin/Cummings. ISBN 978-0-8053-4780-7.

[8]. M. Galvani, 2019. *History and future of driver assistance*. IEEE Instrumentation Measurement Magazine, ISSN 1941-0123.

[9]. S. D. R. G. A. F. Joseph Redmon, 2015. *You Only Look Once: Unified, Real-Time Object Detection*. arXiv:1506.02640 [cs.CV].

[10]. D. Thuan, 2021. *Evolution of YOLO Algorithm and YOLOv5: The State-of-the-art Object Detection*. Bachelor thesis (3.092Mt).

[11]. M. Schumann, 2015. A Book about Colab and related activities. Printed Matter Inc, ISBN 978-0-89439-085-2.

[11]. GeeksforGeeks, 2020. *Python Virtual Environment | Introduction*. Available: https://www.geeksforgeeks.

AUTHOR INFORMATION

Vu Hong Son

Faculty of Electrical and Eletronics Engineering, Hung Yen University of Technology and Education, Vietnam