Kỷ yếu Hội nghị KHCN Quốc gia lần thứ XII về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR); Huế, ngày 07-08/6/2019
DOI: 10.15625/vap.2019.00017

ĐÁNH GIÁ CÁC PHƯƠNG PHÁP DỰA TRÊN DEEP LEARNING CHO BÀI TOÁN PHÁT HIỆN LOGO

Nguyễn Nhật Duy, Đỗ Văn Tiến, Ngô Đức Thành, Huỳnh Ngọc Tín, Lê Đình Duy

Phòng Thí nghiệm Truyền thông Đa phương tiện, Trường Đại học Công nghệ Thông tin
ĐHQG TP. Hồ Chí Minh

{duynn, tiendv, thanhnd, tinhn, duyld}@uit.edu.vn

***TÓM TẮT**: Phát hiện sự xuất hiện của logo để quản lý thương hiệu là một ứng dụng điển hình của việc áp dụng kết quả của các bài toán thị giác vào ứng dụng thực tiễn. Trước đây, các ứng dụng dạng này thường dựa trên dữ liệu dạng văn bản để xử lý. Tuy nhiên, với sự phổ biến của ảnh và video thì hướng tiếp cận dựa trên phát hiện logo đang là hướng đi mới với nhiều tiềm năng. Hiện nay, đối với bài toán phát hiện logo có khá nhiều hướng giải quyết, đặc biệt các hướng tiếp cận tiên tiến hiện nay là sử dụng học sâu (Deep learning) đang mang lại hiệu quả cao. Tuy nhiên, khi triển khai vào một ứng dụng thì việc lựa chọn phương pháp để đảm bảo cân bằng giữa các yếu tố như độ chính xác trên dữ liệu thực tế, tốc độ cũng như tài nguyên cần để xử lý là thách thức cần được giải quyết. Theo đó, trong bài báo này chúng tôi đã (1) xây dựng tập dữ liệu thực tế được thu thập về bao gồm 15035 ảnh của 15 thương hiệu từ các diễn đàn, mạng xã hội, cũng như các công cụ tìm kiếm hình ảnh; (2) thực hiện đánh giá các phương pháp Deep learning tốt nhất hiện nay bao gồm YOLO, RetinaNet, Faster RCNN, Mask RCNN, trên tập dữ liệu thu thập được về các yếu tố độ chính xác, tốc độ xử lý và tài nguyên tính toán. Cùng với đó, các phân tích trên kết quả đánh giá là một tài liệu tham khảo hữu ích cho các nhà phát triển ứng dụng.*

***Từ khóa**: Phát hiện đối tượng, phát hiện logo, mô hình mạng học sâu, Deep learning.*

I. GIỚI THIỆU

Quản lý thương hiệu thông qua những bài viết, hình ảnh, video có sự xuất hiện của thương hiệu được chia sẻ trên Internet là một trong những vấn đề được các công ty đặc biệt quan tâm. Thông qua việc phân tích dữ liệu công ty có thể nhận được các đánh giá về sản phẩm dịch vụ của mình, qua đó có thể đưa ra các đánh giá tổng thể và giải pháp phát triển mô hình kinh doanh một cách hiệu quả. Các hệ thống quản lý thương hiệu thông qua phân tích nội dung đa phương tiện hiện nay thường xử lý trên dữ liệu dạng văn bản (text) như nội dung trong các bài viết, nhận xét từ người dùng, v.v. Đây cũng là dạng dữ liệu dễ xử lý hơn so với dữ liệu dạng ảnh, video mặc dù với sự phát triển của các phần cứng thì dữ liệu ảnh và video có nội dung liên quan đến thương hiệu khá phổ biến. Hiện nay với sự phát triển vượt bậc của các kỹ thuật trong lĩnh vực thị giác máy đặc biệt là các kỹ thuật theo hướng tiếp cận Deep learning thì bài toán phát hiện logo đã đạt được nhiều kết quả khả quan trên các tập dữ liệu chuẩn [1], [2]. Tuy nhiên, khi áp dụng các phương pháp này vào dữ liệu thực tế - dữ liệu là các ảnh chụp hoặc video từ người dùng được đưa lên thành các bài viết trên internet thì gặp khá nhiều thách thức như bị mờ, che khuất, kích thước nhỏ, v.v dẫn đến độ chính xác không còn đảm bảo như trên các tập dữ liệu chuẩn (Hình 1). Ngoài ra, muốn lựa chọn phương pháp phù hợp để xây dựng ứng dụng thực tiễn đạt được hiệu quả cao thì nhà phát triển phải cân bằng giữa các yếu tố như độ chính xác, tốc độ xử lý cũng như tài nguyên tính toán. Với những lý do trên, trong bài báo này chúng tôi có những đóng góp chính sau đây:

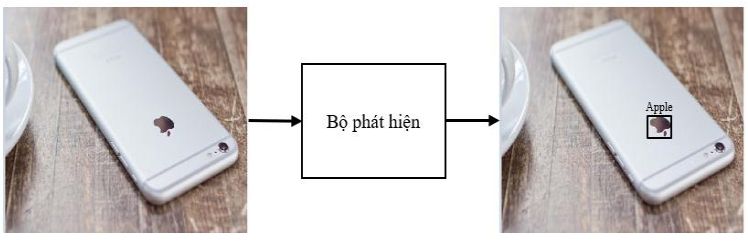
1. Xây dựng tập dữ liệu thực tế của 15 thương hiệu phổ biến ở Việt Nam với nhiều lĩnh vực khác nhau. Tập dữ liệu bao gồm 15035 ảnh được thu thập về từ nhiều nguồn khác nhau bao gồm từ các diễn đàn, mạng xã hội và các công cụ tìm kiếm ảnh phổ biến. Với các tiêu chí lựa chọn như: các hình ảnh phải là hình ảnh được người dùng chụp hoặc cắt từ video mà không phải là hình ảnh quảng cáo, cũng như có sự đa dạng về góc nhìn, kích thước. Tập dữ liệu mà chúng tôi xây dựng được không những là một tập dữ liệu có nhiều thách thức cần giải quyết cho các nhóm nghiên cứu trong và ngoài nước, mà còn là nguồn tham khảo hữu ích cho các nhà phát triển ứng dụng.
2. Chúng tôi tiến hành thực nghiệm bốn phương pháp tiên tiến nhất hiện nay cho bài toán phát hiện đối tượng trên tập dữ liệu xây dựng được bao gồm YOLO [3], Faster RCNN [4], Mask RCNN [5], RetinaNet [6]. Việc đánh giá thực hiện trên các yếu tố như độ chính xác, tốc độ xử lý cũng như tài nguyên tính toán cần thiết để chạy. Bên cạnh đó, ứng với mỗi phương pháp cụ thể chúng tôi còn tiến hành trên nhiều thiết bị đặt khác nhau nhằm đưa ra được thông số tốt nhất. Phân tích so sánh kết quả thực nghiệm cho thấy khi phát triển ứng dụng thực tế với yêu cầu cân bằng giữa các yếu tố thì phương pháp YOLO sẽ là lựa chọn hợp lý vì phương pháp này cho kết quả tốt nhất với độ chính xác trung bình là 51.5% và tốc độ xử lý 0.03 giây.

Bố cục của bài báo được trình bày như sau: phần II sẽ khảo sát một số công trình liên quan; phần III và IV trình bày về tập dữ liệu thu thập được cũng như các phân tích và nhận định dựa trên kết quả đánh giá; kết luận được trình trong phần V.

Hình 1. Một số hình ảnh chứa các logo (a) từ tập dữ liệu thực chuẩn và (b) từ tập dữ liệu thực tế

II. MỘT SỐ NGHIÊN CỨU LIÊN QUAN

Trong phần này, chúng tôi giới thiệu một số phương pháp liên quan đến bài toán phát hiện đối tượng nói chung cũng như áp dụng cho bài toán phát hiện logo. Đầu vào của bài toán là hình ảnh cần được phát hiện, đầu ra là vị trí của đối tượng nếu có (vị trí được thể hiện thông qua một hình chữ nhật bao quanh đối tượng - bounding box (Hình 2). Trong lĩnh vực thị giác máy thì bài toán phát hiện đối tượng nói chung hay logo nói riêng đã đạt được nhiều kết quả khi áp dụng hướng tiếp cận Deep learning. Có thể kể đến một số hướng tiếp cận tiên tiến nhất hiện nay bao gồm RCNN [7], Fast RCNN [8], Faster RCNN [4], Mask RCNN [5], RetinaNet [6], YOLO [3], v.v. Trong đó, các phương pháp này được chia thành 2 loại cơ bản [9]: loại có hướng tiếp cận two-stage - tức là quá trình phát hiện dựa trên các vùng đề xuất (region proposals) là kết quả từ một phương pháp khác như Selective Search [10] hoặc Region Proposal Network [4] và loại có hướng tiếp cận one-stage - tức là ngay trong bản thân cấu trúc mạng của phương pháp đã bao gồm thao tác đưa ra vùng đề xuất. Trong nghiên cứu này, chúng tôi sẽ thực hiện các đánh giá trên cả hai loại, cụ thể các phương pháp được chọn bao gồm Faster RCNN, Mask RCNN (two-stage), RetinaNet, YOLO (one-stage).



Hình 2. Ví dụ minh họa về ảnh đầu vào và ảnh đầu ra của bài toán phát hiện logo

A. Faster RCNN

Phương pháp Faster RCNN là một trong các phương pháp phát hiện đối tượng sử dụng mạng Deep learning đạt độ chính xác cao trên các tập dữ liệu chuẩn như COCO [11]. Faster RCNN được cải tiến dựa trên 2 phương pháp trước đó là RCNN và Fast RCNN. Trong Faster RCNN, tác giả đề xuất sử dụng mạng các vùng đề xuất RPN (Region Proposal Network) để tạo ra các vùng đề xuất. Sau khi có được các đặc trưng học sâu (deep feature) từ các lớp tích chập (convolutional) đầu tiên, mạng RPN sử dụng cửa sổ trượt trên bản đồ đặc trưng (feature map) để rút trích đặc trưng cho mỗi vùng đề xuất. RPN được xem như là một mạng liên kết đầy đủ cùng lúc thực hiện 2 nhiệm vụ đó là dự đoán tọa độ cho các đối tượng và độ tin cậy cho đối tượng đó (objectness score). So với các phương pháp trước đó Faster RCNN đạt kết quả cao hơn và có thời gian xử lý nhanh hơn, tuy nhiên tốc độ vẫn vẫn chưa thể đáp ứng xử lý theo thời gian thực.

B. Mask RCNN

Phương pháp Mask RCNN là phương pháp thực hiện song song 2 bài toán là phân vùng đối tượng (Instance Segmentation) và phát hiện đối tượng. Mask RCNN là phương pháp được cải tiến từ Faster RCNN, trong đó Mask RCNN đề xuất sử dụng lớp RoI Align thay thế cho RoI pooling trong Faster RCNN việc sử dụng RoI Align giúp Mask RCNN cải thiện đáng kể trong việc chọn vùng rút trích đặc trưng. Điều này giúp cải thiện độ chính xác trên phát hiện đối tượng và cơ sở tốt cho bài toán phân vùng đối tượng. Mask RCNN thực hiện 2 bài toán cùng một lúc nên thiết kế mạng Mask RCNN có 2 nhánh song song nhau. Nhiệm vụ của nhánh phát hiện đối tượng thì tương tự như Faster RCNN và nhánh phân vùng là tính toán các đặc trưng từ lớp RoI Align để đưa ra các mặt nạ (mask) phân vùng cho các đối tượng. Mặc dù Mask RCNN cải thiện được độ chính hơn so với Faster RCNN nhưng vẫn chưa cải thiện được phần tính toán cho nên về mặt thời gian Mask RCNN vẫn chưa được cải thiện nhiều.

C. RetinaNet

RetinaNet là một phương pháp tiếp cận one-stage, trong đó RetinaNet thực hiện tính toán dựa trên các anchor boxes mặc định tại mỗi vị trí cần tính toán thay vì sử dụng các vùng đề xuất được tạo ra từ một nghiên cứu khác. Dữ liệu đầu vào của RetinaNet được đưa qua mô hình mạng có tên là FPN [12] nhằm rút trích các ma trận đặc trưng với cùng một tỉ lệ nhưng theo nhiều kích thước khác nhau. Sau đó từ ma trận đặc trưng mới bắt đầu tính toán các vùng đề xuất. Cuối cùng các vùng đề xuất được đưa qua hai mạng phụ để tính ra vị trí của các bounding box và lớp của đối tượng mà bounding box đó bao quanh.

D. You Only Look Once (YOLO)

YOLO được xem là phương pháp đầu tiên xử lý dữ liệu theo thời gian thực và vẫn đạt được độ chính xác cao. Ý tưởng cốt lõi của YOLO là thay vì sử dụng các vùng đề xuất để rút trích đặc trưng thì YOLO sử dụng các thông tin cục bộ từ dữ liệu huấn luyện để học các đặc trưng cần quan tâm bằng cách chia ảnh dữ liệu đầu vào thành lưới (grid view) để khai thác đặc trưng trên lưới. Nếu trọng tâm của đối tượng rơi vào ô nào trong lưới thì ô đó chịu trách nhiệm phát hiện đối tượng. Kích thước lưới như thế nào phụ thuộc vào phiên bản của YOLO, vì hiện tại YOLO có đến 3 phiên bản gồm YOLOv1 [3], YOLOv2 [13], YOLOv3 [14] và mỗi phiên bản có cách chia lưới và thực hiện khác nhau. Nổi bật trong 3 phiên bản này là YOLOv3 mặc dù về tốc độ thì chậm hơn YOLOv2 nhưng độ chính xác được cải thiện đáng kể so với YOLOv2.

Các kết quả thực nghiệm trên các tập dữ liệu chuẩn của bài toán phát hiện đối tượng cho thấy các phương pháp two-stage thường cho kết quả với độ chính xác cao hơn hướng tiếp cận one-stage. Nhưng về thời gian thực hiện thì phương pháp one-stage thường nhanh hơn, cụ thể là có thể xử lý gần thời gian thực. Tuy nhiên, hiệu suất của các phương pháp có kết quả khác nhau tùy thuộc và kích thước, thuộc tính, độ phức tạp, tính đa dạng của các tập dữ liệu sử dụng cho việc huấn luyện cũng như cách thiết kế của mạng Deep learning. Do đó, việc đánh giá được sự ảnh hưởng của các yếu tố này khi áp dụng các phương pháp được liệt kê ra trên đây trên tập dữ liệu thực sẽ giúp các nhà phát triển lựa chọn phù hợp nhất với yêu cầu của mình.

III. THỰC NGHIỆM VÀ KẾT QUẢ

Để đánh giá các phương pháp theo hướng tiếp cận Deep learning trên dữ liệu thực tế. Chúng tôi đã tiến hành thu thập và gán nhãn dữ liệu của 15 thương hiệu với nhiều lĩnh vực khác nhau từ nhiều nguồn. Sau đó, chúng tôi tiến hành đánh giá các phương pháp trên nhiều yếu tố bao gồm độ chính xác, thời gian và tài nguyên cần xử lý. Ngoài ra ứng với mỗi phương pháp khác nhau chúng tôi còn đánh giá với nhiều thiết lập, tham số (model hyperparameters) khác nhau để chọn được tham số phù hợp nhất ứng với mỗi phương pháp. Theo đó, trong nội dung này chúng tôi sẽ nêu các tiêu chí cũng như độ đo sử dụng trong việc đánh giá, thông tin chi tiết về tập dữ liệu thu thập được và các thiết đặt khác nhau tương ứng với mỗi phương pháp.

A. Tiêu chí và độ đo đánh giá

Chúng tôi đánh giá và so sánh các phương pháp trên các yếu tố sau:

1. Độ chính xác của các phương pháp trên dữ liệu thực tế. Một trong những lý do cần xét tới tiêu chí này vì có rất nhiều trường hợp phương pháp chạy tốt trên tập dữ liệu chuẩn, nhưng bị hạn chế trên tập dữ liệu thực tế do tính phức tạp của dữ liệu.
2. Tốc độ cũng như thời gian xử lý đóng vai trò quan trọng trong các ứng dụng, việc cân bằng giữa độ chính xác và tốc độ xử lý cũng là một thách thức.
3. Hầu hết các phương pháp tiếp cận theo hướng Deep learning đều yêu cầu tài nguyên tính toán rất lớn. Do đó thống kê các tài nguyên cần thiết giữa các phương pháp khác nhau là một yêu cầu cần thiết khi muốn triển khai kết quả vào ứng dụng thực tế.

Để đánh giá các yếu tố này, chúng tôi sử dụng các độ đo chuẩn của bài toán phát hiện đối tượng. Trong đó, với tiêu chí chính xác chúng tôi sử dụng độ đo mAP (mean average precision) theo chuẩn đánh giá của PASCAL VOC [15] như công thức bên dưới. Đo thời gian huấn luyện mô hình bằng giờ và xác định tài nguyên sử dụng cho mỗi mô hình bằng dung lượng RAM của GPU dùng trong quá trình huấn luyện và chạy dự đoán.

Công thức tính độ chính xác trung bình (mAP):

$$\text{mAP} = \sum_{q=1}^{Q} \frac{AP(q)}{Q}$$

Trong đó Q là số lượng lớp đối tượng có trong tập dữ liệu, AP là độ chính xác trung bình của từng lớp được tính bằng công thức như sau:

$$\text{AP} = \frac{1}{11} \sum_{r\epsilon\{0,0.1,\ldots,11)} P_{inter}p(r)$$

Với $P_{inter}p(r)$ được tính bằng công thức:

$$P_{inter}p(r) = \max_{\tilde{r}:\tilde{r}\geq r} p(\tilde{r})$$

Trong đó $p(\tilde{r})$ là độ chính xác (precision) được đo tại độ phủ (recall) $\tilde{r}$.

B. Tập dữ liệu đề xuất

Với tiêu chí cần xây dựng tập dữ liệu thực tế cho nhu cầu xây dựng các ứng dụng phát hiện sự xuất hiện của logo như bài toán quản lý thương hiệu. Chúng tôi đã tiến hành thu thập dữ liệu bằng cách xây dựng công cụ để thu thập dữ liệu từ các diễn đàn, mạng xã hội và các công cụ tìm kiếm. Trong đó, dữ liệu thu thập được có tính đa dạng về hình dáng, kích thước, góc nhìn (Hình 3 là một số ví dụ trong tập dữ liệu thu thập được). Hầu hết các ảnh mà chúng tôi thu thập được đều là ảnh được người dùng chụp trực tiếp bằng các loại máy ảnh khác nhau. Do đó tập dữ liệu thu thập được khá đa dạng về chủng loại, kích csm góc nhìn, chất lượng ảnh. Kết quả của quá trình thu thập là tập dữ liệu với tổng cộng khoảng 15035 ảnh cho 15 loại thương hiệu-lớp với nhiều lĩnh vực khác nhau (thông tin chi tiết có thể xem trong Bảng 1). Trong đó, lớp có nhiều ảnh nhất là "Cocacola" 1648 ảnh và lớp có số lượng ảnh ít nhất là "Tiki" 354 ảnh. Sau quá trình gán nhãn bằng tay, chúng tôi chia tập dữ liệu xây dựng được theo tỉ lệ như sau: tập huấn luyện ảnh chiếm 60%, số lượng ảnh của tập validate là 20% và tập test chiếm 20% phục vụ cho quá trình đánh giá.

Hình 3. Một số hình ảnh từ dữ liệu thu thập được

Bảng 1. Bảng chi tiết về số lượng ảnh trong tập dữ liệu của chúng tôi

Tên logo	Xe ô tô				Nước uống				Thương hiệu khác						
	Hyundai	Honda	Toyota	Mercedes	Pepsi	Cocacola	Lavie	Aquafina	FPT	Grab	Thegioididong	Apple	Tiki	VNPT	Nike
Số lượng ảnh	1550	1049	942	1871	940	1648	976	783	619	575	652	967	354	769	1340

C. Một số thiết đặt tương ứng với các phương pháp đánh giá

Đối với các phương pháp phát hiện đối tượng thì bản thân mỗi phương pháp đều có những thiết đặt ứng với các tham số trong mô hình như tốc độ học (Learning rate), số lần lặp (Iteration), kích thước ảnh đầu vào (Image size), v.v. Bên cạnh đó việc lựa chọn mô hình cho quá trình trích xuất đặc trưng (backbone) cũng đóng vai trò quan trọng trong quá trình chạy phương pháp. Tất cả các những thiết đặt này đều ảnh hưởng tới các yếu tố như độ chính xác, tốc độ, thời gian xử lý và tài nguyên của hệ thống. Theo đó, trong phần thực nghiệm, ngoài việc so sánh các phương pháp khác nhau chúng tôi còn thiết đặt các tham số khác nhau trong từng phương pháp, từ đó có thể đưa ra các tham số tốt nhất. Các mô hình mà chúng tôi đánh giá gồm có YOLOv3, RetinaNet, Mask RCNN, Faster RCNN với các backbone gồm FPN [12], ResNet [16].

Chúng tôi huấn luyện các mô hình mạng Deep learning này bằng phương pháp transfer learning- tức là sử dụng bộ trọng số đã được huấn luyện trước đó trên các tập dữ liệu lớn như ImageNet [17], sau đó bằng cách sử dụng trọng số

đã được học và tiếp tục huấn luyện trên tập dữ liệu thực của bài toán. Việc huấn luyện theo phương pháp này giúp chúng tôi giải quyết được vấn đề thiếu dữ liệu trong việc huấn luyện các mạng Deep learning.

Đối với YOLO chúng tôi sử dụng K-means như tác giả đề cập tính lại kích thước và tỉ lệ của các anchors, theo thực nghiệm thì sau khi tính lại các anchors kết quả trên YOLOv3 cho kết quả kém hơn anchors mặc định cho nên chúng tôi quyết định sử dụng 9 anchors mặc định gồm: [10, 13], [16, 30], [33, 23], [30, 61], [62, 45], [59, 119], [116, 90], [156, 198], [373, 326]. Phần còn lại liên quan đến các siêu thông số được trình bày trong Bảng 2.

Chúng tôi huấn luyện các mô hình tổng cộng 30 epochs. Từ kết quả thu được chúng tôi sẽ tiến hành so sánh các phương pháp lẫn nhau dựa trên 3 tiêu chí đã đặt ra từ đó tìm ra những ưu điểm và hạn chế của từng phương pháp.

Bảng 2. Thông tin thống kê chi tiết về các siêu tham số khi huấn luyện mô hình

Phương pháp	Learning rate	Batchsize	Weight decay	Max iteration	Gamma	Scales	Stepsize	Image size
Mask RCNN	0.001	2	0.0001	135000	0.1		60000, 120000	500, 833
Faster RCNN	0.001	2	0.0001	135000	0.1		60000, 120000	500, 833
RetinaNet	0.001	2	0.0001	135000	0.1		60000, 120000	500, 833
YOLOv3	0.001	32	0.0005	422202		0.1	40000	416, 416

IV. PHÂN TÍCH VÀ ĐÁNH GIÁ

Trong phần này chúng tôi trình bày các kết quả đạt được thông qua phần đánh giá thực nghiệm. Tất cả các mô hình được huấn luyện được chạy trên môi trường Ubuntu 14.04 64 bits với cấu hình Intel(R) Xeon(R) CPU E5-2620b3 @ 2.40GHz, 65 GB RAM DDR3, GPU Tesla P100 12Gb RAM. Như đã đề cập ở các phần trước, chúng tôi chủ yếu so sánh 3 tiêu chí đánh giá là độ chính xác, tốc độ và tài nguyên sử dụng cho nên trong phần này chúng tôi chia ra thành 3 mục chính như sau:

A. Độ chính xác

Dựa trên kết quả về độ chính xác của các phương pháp (Bảng 3), chúng tôi thấy rằng đối với các phương pháp two-stage mô hình Mask RCNN là mô hình đạt kết quả mAP cao nhất đặc biệt là Mask RCNN với mạng cơ bản là ResNet50-FPN đạt 80.9% cao hơn khoảng 2% so với mạng cơ bản ResNet101-FPN đạt 78.5%. Điều này cho thấy rằng việc sử dụng mạng có nhiều lớp chưa chắc đã cho kết quả tốt hơn. Mạng càng có nhiều lớp thì số thông số cần học càng lớn điều này có nghĩa rằng cần rất nhiều dữ liệu để huấn luyện mô hình mà có nhiều lớp. Vì trong quá trình huấn luyện giá trị của các thông số sẽ được cập nhật bằng các thuật toán tối ưu như Gradient Descent thông qua dữ liệu truyền vào. Nếu phương pháp chúng ta sử dụng không xử lý tốt khi rút trích đặc trưng từ dữ liệu thì phần kết quả khi phân lớp và dự đoán tọa độ cho các đối tượng sẽ bị ảnh hưởng. Điều này tương tự xảy ra với phương pháp Faster RCNN, mạng cơ bản ResNet50-FPN đạt 78.4% và ResNet101-FPN đạt 78.1%, trong trường hợp này chênh lệch giữa 2 mạng cơ bản không đáng kể chỉ 0.3%. Mặc dù kết quả của Faster RCNN thấp hơn Mask RCNN khoảng 2% nhưng việc chênh lệch giữa 2 mạng cơ bản cho thấy việc kết hợp các mạng cơ bản với Faster RCNN cho kết quả tốt hơn so với Mask RCNN.

Đối với các phương pháp one-stage mô hình YOLOv3 đạt kết quả thấp nhất 51.5% nhưng đánh đổi lại YOLOv3 có khả năng xử lý dữ liệu theo thời gian thực. Đối với RetinaNet thì ngược lại khi kết hợp với mạng cơ bản ResNet-101 đạt 78.1% cho kết quả tốt hơn ResNet-50 1.1%. Điều này có thể giải thích rằng RetinaNet sử dụng Focal Loss để tính độ lỗi mà Focal Loss được đề xuất để sử dụng trong các trường hợp dữ liệu giữa các lớp không cân bằng nhau giữa foreground và background cho nên trong trường hợp như tập dữ liệu của chúng tôi RetinaNet cho kết quả ổn định khi kết hợp với các mạng cơ bản. Ngoài ra, các lớp có AP tương đương nhau, riêng chỉ có lớp thương hiệu Toyota là có AP thấp hơn. Để giải thích cho điều này, các dữ liệu được huấn luyện cho Toyota thường là các hình ảnh có chứa logo nhỏ, đồng thời các đặc trưng của Toyota lại khá tương đồng với logo Hyundai. Và dựa vào kiểm chứng thực nghiệm thì với các hình ảnh có chứa logo Toyota thì mô hình thường nhận dạng sai sang Hyundai.

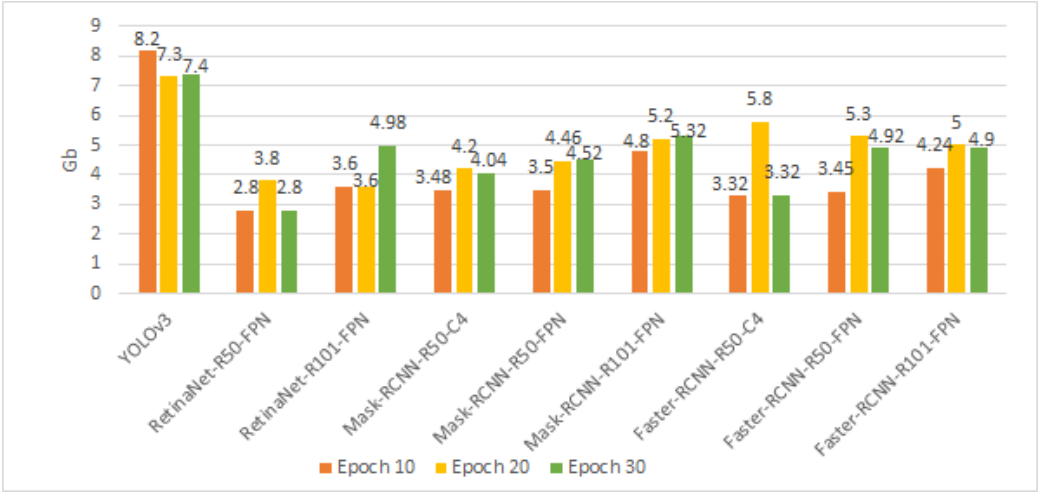
Thông qua kết quả này chúng tôi thấy rằng độ chính xác của các mô hình ảnh hưởng nhiều vào mạng cơ sở khi rút trích đặc trưng và phần mạng phía sau dùng để xử lý đặc trưng cũng như hàm tính độ lỗi khi huấn luyện của các phương pháp.

Bảng 3. Thông tin thống kê chi tiết về kết quả thực nghiệm trên tập dữ liệu thu thập

Tên logo	Tên phương pháp								
	YOLO v3	RetinaNet ResNet50	ReitnaNet ResNet101	Mask ResNet50 C4	Mask ResNet50 FPN	Mask ResNet101 FPN	Faster ResNet50 C4	Faster ResNet50 FPN	Faster ResNet101 FPN
Hyundai	45.5	68	69	67.1	**76.5**	73.2	74	72.9	70.2
Honda	32.9	65.7	68.3	68	66.9	**69.1**	66.4	69	65.5
Toyota	24.8	45.1	47.7	51	57	55.1	50.8	51.1	55.5
Mercedes	71.5	88.3	89.4	85.4	**89.5**	86.5	88.2	85.9	86.6
FPT	60.2	91.9	92.5	91.9	**93.1**	94.5	91.2	92.2	91.9
Pepsi	64.9	73.4	69.2	71.8	75.2	68.1	**78.6**	71.8	70.9
Coca Cola	68.6	86.9	**87.9**	82.4	83.7	76.1	84.8	76.7	75.9
Grab	45.1	79.1	77.5	78.6	**86**	79.7	75.6	81.1	84.6
Lavie	69.6	92.4	**93.5**	77.5	88.2	84.9	87.2	85.4	85.7
Aquafina	62.7	88.2	**88.6**	83.2	85.7	83	85.4	86.8	83.1
Thế giới di động	53.3	65.1	73.2	92	93.2	**94.1**	91.2	91.5	93.6
Apple	63.1	**97.7**	85.6	78.2	79.3	79.9	88.4	81.3	78.1
Tiki	36.8	74.7	80.1	65.8	**80.8**	75.3	78.8	76.1	73.7
VNPT	26.4	79	78.4	85.6	**89.7**	88.5	84.4	86.6	87.4
Nike	46.8	69.4	70.2	66.5	69.1	**70.3**	68.4	67.9	69.2
Trung bình (mAP %)	51.5	77	78.1	76.3	**80.9**	78.5	79.6	78.4	78.1

B. Tài nguyên tính toán

Khi xét về tài nguyên tính toán cần sử dụng khi huấn luyện và sử dụng mô hình **(**Hình 4**)**, chúng tôi nhận thấy rằng bình quân dung lượng GPU RAM cần khi sử dụng huấn luyện mô hình bằng phương pháp RetinaNet với backbone ResNet50-FPN là thấp nhất 3.13GB, phương pháp YOLOv3 tốn nhiều tài nguyên sử dụng nhất khi sử dụng đến gấp đôi tài nguyên mà các phương pháp khác cần. YOLOv3 sử dụng nhiều tài nguyên hơn do thay vì sử dụng mạng cơ bản darknet19 như 2 phiên bản trước đó, YOLOv3 sử dụng mạng cơ bản là darknet53 với 3 vị trí dự đoán kết quả với 3 tỉ lệ đối tượng khác nhau thay vì chỉ 1 vị trí dự đoán cho tất cả các đối tượng như 2 phiên bản trước. Đứng sau Retinanet về dung lượng sử dụng tài nguyên là Mask RCNN 3.9GB với mạng cơ bản là ResNet50-C4 và kết hợp với độ chính xác là 76.3% thì RetinaNet có phần vượt trội hơn. Nhưng so sánh RetinaNet ResNet50-FPN với Mask RCNN ResNet50-FPN thì Mask RCNN sử dụng khoảng 4.14 GB lớn hơn 1GB để tăng độ chính xác khoảng 2%. Nếu so sánh Faster RCNN với RetinaNet và Mask RCNN trong trường hợp tài nguyên sử dụng kết hợp với độ chính xác thì RetinaNet với Mask RCNN thì hiệu suất tốt hơn.

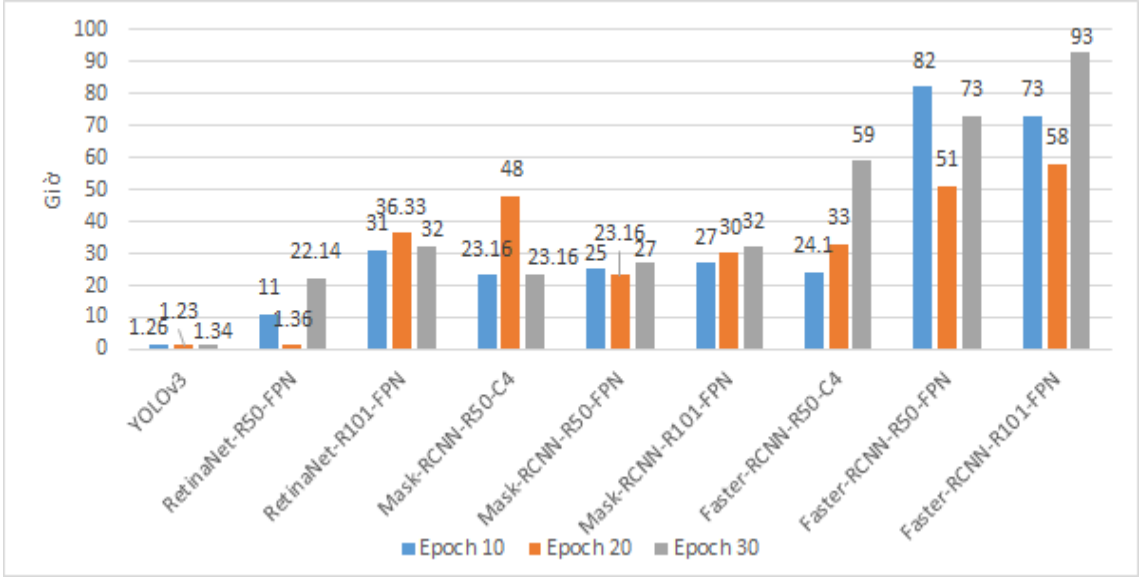


Hình 4. Thông số đánh giá dựa vào GPU RAM sử dụng khi huấn luyện trên tập dữ liệu chuẩn

C. Tốc độ

Khi xét về thông số thời gian huấn luyện của các phương pháp (Hình 5), phương pháp YOLOv3 là mô hình có thời gian huấn luyện trung bình nhanh chỉ khoảng 1.5 giờ, so với mặt bằng chung của các mô hình khác là tốn thời gian khá lâu. Đặc biệt, phương pháp Faster RCNN với các mạng cơ bản ResNet101-FPN là có thời gian tối đa lên đến 3

ngày cho mỗi 10 epoch khác nhau. Đồng thời các mạng cơ bản khác khi kết hợp với Faster RCNN cũng tốn rất nhiều thời gian để huấn luyện. Mask RCNN là phương pháp có tốc độ huấn luyện nhanh thứ 2 nếu kết hợp với độ chính xác và tài nguyên sử dụng thì Mask RCNN là phương pháp hoạt động khi chúng tôi tập trung vào độ chính xác. So về tốc độ huấn luyện cũng như tốc độ khi xử lý dữ liệu thì YOLOv3 vẫn là phương pháp dẫn đầu mặc dù lượng tài nguyên sử dụng khi huấn luyện khá cao.



Hình 5. Thông số đánh giá dựa vào thời gian khi huấn luyện trên tập dữ liệu chuẩn

Khi so sánh về mặt tốc độ xử lý khi sử dụng mô hình (Bảng 4) chúng tối thấy rằng YOLO là phương pháp xử lý nhanh nhất trong khi đó Mask RCNN chậm nhất. Nếu xét về tổng thể, bao gồm cả mặt thời gian xử lý và độ chính xác thì YOLO là phương pháp sở hữu khá nhiều ưu thế mặc dù tốn nhiều tài nguyên GPU RAM.

Bảng 4. Thời gian khi sử dụng các mô hình để phát hiện logo

Tên các phương pháp	Thời gian (s/ảnh)
YOLO	0.03
RetinaNet-Resnet50-FPN	0.1
RetinaNet-Resnet101-FPN	0.13
Mask RCNN-Resnet50-C4	0.41
Mask RCNN-Resnet50-FPN	0.1
Mask RCNN-Resnet101-FPN	0.12
Faster RCNN-Resnet50-C4	0.41
Faster RCNN-Resnet50-FPN	0.1
Faster RCNN-Resnet50-FPN	0.12

V. KẾT LUẬN

Nội dung của nghiên cứu tập trung vào việc đánh giá 4 phương pháp phát hiện đối tượng bao gồm YOLO, RetinaNet, Faster RCNN, Mask RCNN trên tập dữ liệu thực tế tự thu thập và gán nhãn. Chúng tôi đánh giá, so sánh và phân tích trên ba yếu tố chính bao gồm độ chính xác, tốc độ xử lý và tài nguyên cần tính toán - đây là các yếu tố quan trọng cần xem xét khi áp dụng kết quả tính toán vào các ứng dụng thực tiễn. Bên cạnh đó với tập dữ liệu xây dựng được của 15 loại logo với tổng 15035 là nguồn tham khảo hữu ích cho các nhóm nghiên cứu quan tâm đến bài toán này. Kết quả đánh giá cho thấy, YOLO là phương pháp cho tốc độ xử lý dữ liệu nhanh chóng nhưng độ chính xác không cao (mAP = 51.5%), trong khi đó Mask RCNN là phương pháp cho độ chính xác tốt nhất (mAP = 80.9) nhưng thời gian xử lý lâu. Để cân bằng giữa các yếu tố về tốc độ, thời gian và tài nguyên sử dụng thì YOLO là lựa chọn tốt nhất khi muốn phát triển ứng dụng. Bên cạnh đó, với tập dữ liệu thực tế của 15 loại logo của khoảng 15035 bức ảnh được thu thập và gán nhãn chúng tôi hy vọng rằng là một tập dữ liệu tham khảo hữu ích cho cộng đồng quan tâm nghiên cứu đến bài toán này.

VI. LỜI CẢM ƠN

Nghiên cứu này được tài trợ bởi Trường Đại học Công nghệ thông tin - ĐHQG-HCM trong khuôn khổ Đề tài mã số D2-2019-017.

TÀI LIỆU THAM KHẢO

[1] C. Eggert, A. Winschel, D. Zecha, and R. Lienhart, "Saliency-guided selective magnification for company logo detection," *Proc. - Int. Conf. Pattern Recognit.*, pp. 651-656, 2017.

[2] H. Su, X. Zhu, and S. Gong, "Deep learning logo detection with data expansion by synthesising context," *Proc. - 2017 IEEE Winter Conf. Appl. Comput. Vision, WACV 2017*, pp. 530-539, 2017.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2015.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137-1149, 2017.

[5] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-October, pp. 2980-2988, 2017.

[6] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-October, pp. 2999-3007, 2017.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 580-587, 2014.

[8] R. Girshick, "Fast R-CNN," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 International Conference on Computer Vision, ICCV 2015, pp. 1440-1448, 2015.

[9] L. Liu *et al.*, "Deep Learning for Generic Object Detection: A Survey," 2018.

[10] T. G. A. W. M. Smeulders, "Selective Search for Object Recognition," pp. 154-171, 2013.

[11] T. Lin, C. L. Zitnick, and P. Doll, "Microsoft COCO : Common Objects in Context," pp. 1-15.

[12] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 936-944, 2017.

[13] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 6517-6525, 2017.

[14] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018.

[15] M. Everingham, L. Van Gool, C. K. I. Williams, and J. Winn, "The P ASCAL Visual Object Classes (VOC) Challenge," pp. 303-338, 2010.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2015.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097-1105.

EVALUATION OF DEEP LEARNING BASED APPROACHES FOR LOGO DETECTION

Duy Nguyen, Tien Do, Thanh Duc Ngo, Tin Huynh, Duy Dinh Le

***ABSTRACT**: Detecting the appearance of logos to manage trademark is a typical application of computer vision to practical applications. In the past, applications of this type were often based on textual data for processing. With the popularity of images and videos, approaches based on logo detection is a new one with great potential. There are many solutions to deal with logo detection, especially the state-of-the-art approaches based on Deep learning which achieve high performance. However, the choice of approaches to ensure a balance between factors such as accuracy, speed of processing and resource usage is a challenge to handle when deploying them into an application. In this paper, we have (1) built the actual dataset consisting of 15035 images of 15 type of logos collected from social networks and search engines for images; (2) evaluated the state of the art models based on Deep learning including YOLO, Faster RCNN, Mask RCNN, RetinaNet on our proposed dataset with factors of accuracy, processing speed and resource usage. The analysis on the experimental evaluation is a useful reference for application developers.*