CẢI TIẾN KIẾN TRÚC MẠNG YOLO CHO BÀI TOÁN NHẬN DẠNG LOGO

Lê Đình Nghiệp[1], Phạm Việt Bình[2], Đỗ Năng Toàn[3], Phạm Thu Hà[4],Trần Văn Huy[1]
[1]Trường Đại học Hồng Đức,
[2]Trường Đại học Công nghệ thông tin & Truyền thông – ĐH Thái Nguyên,
[3]Viện Công nghệ thông tin – ĐH Quốc gia Hà Nội, [4]ĐH Điện Lực

TÓM TẮT

Ngày nay, logo không những được xem là thương hiệu gắn liền với doanh nghiệp mà còn có nhiều ý nghĩa khác. Vì vậy, nhận dạng logo là bài toán rất được quan tâm. Bài báo này trình bày một phương pháp nhận dạng logo dựa trên kiến trúc mạng học sâu. Thay vì sử dụng tiếp cận kiểu RCNN hoặc biến thể FRCNN, chúng tôi đã cải tiến mạng học sâu Yolo để dò tìm vùng logo đồng thời với nhận dạng logo trong ảnh mầu đầu vào. Kết quả thực nghiệm với mẫu tập flickrlogo47 cho thấy phương pháp đề xuất đạt được độ chính xác cao. Hơn nữa, phương pháp đề xuất đơn giản, hiệu quả và có thời gian thực hiện nhanh, phù hợp với các hệ thống nhận dạng logo yêu cầu tính thời gian thực.

Từ khóa: *dò tìm đối tượng, dò tìm logo, mạng YOLO, FRCNN, nhận dạng thời gian thực*

Ngày nhận bài: 22/4/2019;Ngày hoàn thiện: 07/5/2019;Ngày duyệt đăng: 16/5/2019

CUSTUMIZED YOLO ARCHITECTURE FOR LOGO RECOGNITION

Nghiep Le Dinh[1], Binh Pham Viet[2], Toan Do Nang[3], Ha Pham Thu[4], Huy Tran Van[1]
[1]Hong Duc University,
[2]University of information and communication technology – Thai Nguyen University,
[3]The Information Technology Institute - Vietnam National University, Hanoi,
[4]Electric power University

ABSTRACT

Today, logos not only are considered trademarks associated with businesses, but also have others meaningfull. Therefore, logo identification is a very important problem in image processing. This article presents a method of identifying logos in real time with a deep learning network architecture. Instead of using an RCNN type approach or FRCNN variant, we custumized Yolo algorithm to detect the logo area simultaneously with the logo identification in the input color image. Experimental results with popular logo dataset flickrlogos-47, show that the proposed method achieves high accuracy. Furthermore, the proposed method is simple, effective and has a fast execution time, in accordance with the logo recognition system that requires real-time computing.

Keywords. *Object detection , Logo detection, YOLO, FRCNN, Realtime recognition.*

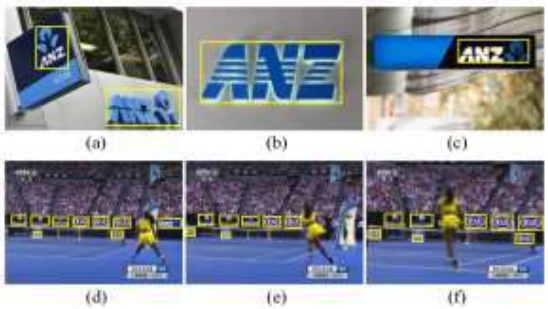
Received: 22/4/2019; Revised: 07/5/2019; Approved: 16/5/2019

* Corresponding author. *Email: ledinhnghiep@hdu.edu.vn*

1. Giới thiệu

Trong thời gian gần đây nhận dạng logo trong ảnh và video nhận được nhiều sự quan tâm, nghiên cứu vì vai trò quan trọng của nó trong rất nhiều ứng dụng thực tế như trong các hệ thống điều khiển giao thông tự động, thống kê tự động các nhãn hiệu xuất hiện trên phương tiện thông tin, kiểm soát sự bùng nổ của các thương hiệu, định vị trí đặt quảng cáo phù hợp với ngữ cảnh trong video. Ngoài ra dò tìm, phân loại logo trong các video còn có giá trị to lớn trong việc phân tích thị trường đặc biệt là trong các quảng cáo trên tivi hay internet do tần xuất xuất hiện và thời lượng xuất hiện của logo cho phép các nhà tài trợ ước lượng được hiệu quả của việc quảng cáo mà họ đặt ra.

Các logo xuất hiện trong các ảnh thường rõ nét vì chủ yếu được chụp chính diện, có độ phân giải cao. Do đó việc dò tìm và nhận dạng trong ảnh thường cho kết quả tốt [1,2]. Tuy nhiên ngoài các yếu tố tác động giống như trong ảnh như sự đa dạng về kích thước và hình thức, logo có thể được biểu diễn bởi các ký tự hay biểu tượng đồ họa hoặc bằng sự kết hợp các đặc trưng này. Trong các video logo còn bị tác động bởi nhiều yếu tố khác như: sự phóng to thu nhỏ, sự đang dạng trong điểm nhìn, trong điều kiện khác nhau của cường độ sáng, bị che khuất một phần bởi nền hoặc đối tượng khác, sự lộn xộn phức tạp của nền ảnh hay đa dạng về chủng loại vật thể (hình 1)… ảnh hưởng đến độ chính xác và thời gian nhận dạng.



Hình 1. *Các yếu tố ảnh hưởng đến nhận dạng logo. Các logo cùng một nhà tài trợ (a), (b) và (c) có các thể hiện, biến dạng khác nhau trong video tennis. Các logos trong các frame lân cận (d), (e) và (f) bị tác động bởi nhiễu và che khuất*

Hiện nay, có ba hướng tiếp cận chính trong nhận dạng logo. Hướng tiếp cận cổ điển dựa trên đặc trưng toàn cục như đặc trưng hình dáng, lược đồ màu, kết cấu có ưu điểm giúp cho việc phát hiện logo nhanh nhưng việc nhận dạng thì không thích hợp trong thế giới thực bởi vì nó không đầy đủ thông tin để phân biệt các logo thuộc các lớp khác nhau [3]. Ngoài ra, đặc trưng toàn cục thường không bất biến với những phép biến đổi hình học và việc sử dụng đặc trưng toàn cục đòi hỏi tập ảnh dùng để huấn luyện rất lớn, chi phí tính toán cao. Các hệ thống này thường sử dụng phương pháp đối sánh mẫu, đơn giản là so sánh các điểm ảnh của các đối tượng cần nhận dạng với nhau. Tuy nhiên, việc so sánh này tốn nhiều thời gian và không thu được độ chính xác cao.

Hướng tiếp cận thứ hai dựa trên đặc trưng cục bộ [4,5,6,7] đã được nhiều tác giả nghiên cứu và vận dụng vào bài toán phát hiện và nhận dạng logo. Hướng tiếp cận này được đánh giá cao bởi vì các đặc trưng cục bộ này có thể bất biến với những phép biến đổi hình học và mạnh đối với sự thay đổi về điều kiện chiếu sáng, nhiễu, sự che khuất một phần. Mặc dù vậy cách tiếp cận này vẫn chưa cho độ chính xác cao và tốc độ tìm kiếm tương đối chậm.

Hướng tiếp cận thứ ba được phát triển trong thời gian gần đây tập trung nghiên cứu sử dụng mạng học sâu để nhận dạng logo[8,9,10]. Oliveria cùng cộng sự [11] đã đưa ra các mô hình mạng nhân chập huấn luyện trước và sử dụng chúng như là một phần của mạng nhân chập nhanh dựa trên vùng đề xuất. Với số lượng dữ liệu huấn luyện có giới hạn cho việc nhận dạng logo, tất cả những phương pháp này làm việc trên các mạng huấn luyện trước với các mục đích khác nhau. Bianco và các cộng sự [13] thay vì sử dụng các mạng huấn luyện trước đã sử dụng mạng norron nhân chập tự định nghĩa kết hợp với vùng lựa chọn đề xuất [12] trên tập huấn luyện FlickrLogos-32*[14]* để nhận dạng logo. Hướng tiếp cận thứ ba này cho độ chính xác

cao hơn hẳn hai hướng một và hai trên. Tuy nhiên, việc sử dụng mô hình học sâu thường phức tạp, tốc độ xử lý có thể chậm do phải tính toán rất nhiều trên các tầng mạng học sâu.

Trong bài báo này chúng tôi đề xuất một phương pháp nhận dạng logo dựa trên mạng học sâu YOLO cải tiến với hai mục tiêu là định vị vùng chứa logo với các đặc trưng của nó, phân lớp, nhận dạng nhãn hiệu của logo có trong vùng này trong khi hình ảnh chỉ được truyền qua mạng một lần duy nhất.

Phần còn lại của bài báo được tổ chức như sau: phần 2 trình bày các nghiên cứu liên quan; phần 3 đưa ra mô hình đề xuất; phần 4 trình bày kết quả thực nghiệm; cuối cùng kết luận sẽ được trình bày trong phần 5.

2. Các nghiên cứu liên quan

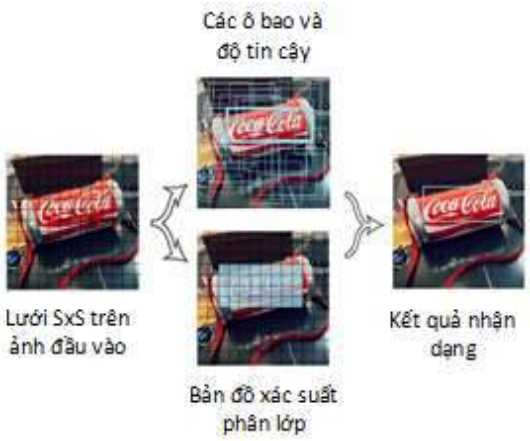
Hiện tại có nhiều kiến trúc CNN dùng cho việc dò tìm đối tượng, tuy nhiên các kiến trúc Faster R-CNN[15], SDD[16] và YOLOv3[17] cho độ chính xác cao với kết quả đầu ra gồm hai thành phần là vùng chứa đối tượng và lớp của đối tượng trong các vùng tìm thấy.

Faster R-CNN là bộ dò tìm đối tượng 2 bước có độ chính xác cao trên tập dữ liệu chuẩn nhưng lại chậm hơn 2 bộ dò một bước SDD và YOLO. FRCNN sử dụng kỹ thuật RPN (region Proposal Network)[15] thay thế cho thuật toán tìm vùng lựa chọn (selective search)[18] vốn khá chậm. Thay vì phải rút trích đặc trưng của mỗi vùng đề xuất, FRCNN sử dụng CNN rút trích đặc trưng của toàn bộ bức ảnh trước, đồng thời rút trích các vùng đề xuất, lấy các vùng đề xuất tương ứng trên đặc trưng đối hợp, chuẩn hóa tỷ lệ và cuối cùng là phân lớp và tìm vị trí của đối tượng. Với việc không phải lặp lại trên quá nhiều vùng đề xuất việc rút trích đặc trưng, FRCNN giảm thời gian xử lý một cách đáng kể. Để nhận dạng, FRCNN sử dụng kiến trúc mạng Fast R-CNN[20] trên tập vùng đề xuất vừa thu được. Để tăng tốc độ xử lý RPN và Fast R-CNN được hợp nhất với các tầng nhân chập dùng chung.

Kiến trúc của YOLO [19] giống với FRCNN, sử dụng kiến trúc mạng CNN, trong đó các tầng trích xuất đặc trưng từ ảnh đầu vào và các tầng liên kết đầy đủ sẽ dự báo xác suất lớp của đối tượng và toạ độ đầu ra vùng bao chứa đối tượng. Hình ảnh chỉ được truyền qua một lần duy nhất, sử dụng các đặc trưng từ toàn bức ảnh để dự báo mỗi vùng bao với tất cả các đối tượng. Ảnh đầu vào được chia thành SxS ô lưới (hình 2), nếu tâm của đối tượng nằm trong một ô thì ô đó sẽ chịu trách nhiệm phát hiện ra đối tượng này.

Hình 2. *Vùng bao và các ô lưới*

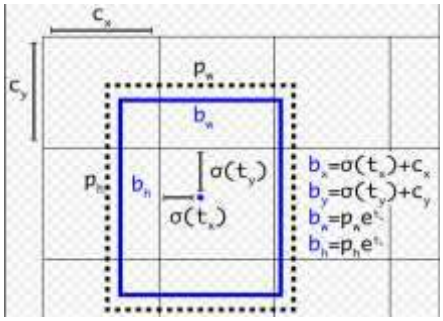
Mỗi ô lưới dự đoán vùng bao và độ tin cậy tương ứng mỗi vùng bao trong B vùng bao đối tượng được dự báo. Độ tin cậy phản ánh khả năng của đối tượng nằm trong vùng bao, đồng thời là độ chính xác, độ khớp của vùng bao so với đối tượng. Độ tin cậy kết nhập giá trị đo khả năng vùng bao có chứa đối tượng và giá trị đo độ khớp của vùng bao dự báo so với vùng bao đúng. Nếu không tồn tại đối tượng nào trong ô lưới, giá trị độ tin cậy sẽ gán là 0.



Hình 3. *Quy trình thuật toán YOLO*

Mỗi vùng bao chứa 5 giá trị *x, y, w, h* và *độ tin cậy*. Toạ độ *(x, y)* thể hiện *vị trí tương đối*

của tâm đối tượng so với viền *ô lưới*. *w*, *h* là *kích thước tương đối* của chiều rộng và chiều cao của vùng bao so với kích thước ảnh đầu vào. Mỗi ô lưới đồng thời cũng dự đoán C xác suất có điều kiện của C lớp cho trước. Xác suất này phụ thuộc việc tồn tại đối tượng ở ô lưới, và chỉ dự báo một tập C xác suất tương ứng mỗi ô lưới, không cần quan tâm đến vùng bao. Tại pha nhận dạng, 2 giá trị trên được nhân với nhau để tạo thành độ tin cậy - từng lớp cho mỗi vùng bao.



Hình 4. *Dự báo vùng bao tại một ô lưới*

Hàm loss của thuật toán YOLO được cho như sau:

$$
\begin{aligned}
Loss = \quad & \{ \\
& \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{A} \mathbb{1}_{ij}^{obj} \left[\left(b_{x_i} - b_{\hat{x}_i}\right)^2 + \left(b_{y_i} - b_{\hat{y}_i}\right)^2 \right] \\
& + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{A} \mathbb{1}_{ij}^{obj} \left[\left(\sqrt{b_{w_i}} - \sqrt{b_{\hat{w}_i}}\right)^2 + \left(\sqrt{b_{h_i}} - \sqrt{b_{\hat{h}_i}}\right)^2 \right] \\
& + \sum_{i=0}^{S^2} \sum_{j=0}^{A} \mathbb{1}_{ij}^{obj} \left(C_i - \hat{C}_i\right)^2 \\
& + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{A} \mathbb{1}_{ij}^{noobj} \left(C_i - \hat{C}_i\right)^2 \\
& + \sum_{i=0}^{S^2} \mathbb{1}_{i}^{obj} \sum_{c \in classes} \left(p_i(c) - \hat{p}_i(c)\right)^2 \ \}
\end{aligned} \tag{1}
$$

Trong đó

$\mathbb{1}_i^{obj}$: Nếu tâm của vùng bao đối tượng xuất hiện trong ô lưới thứ i

$1 \le i \le S^2$

$\mathbb{1}_{ij}^{obj}$: Nếu đối tượng xuất hiện trong cell i và đối tượng được bao bởi box j

$1 \le i \le S^2; 1 \le j \le B$

$\mathbb{1}_{ij}^{noobj}$: Nếu một phần đối tượng xuất hiện trong cell i nhưng box j không thật sự chứa đối tượng

$1 \le i \le S^2; 1 \le j \le B$

$1 - \mathbb{1}_{ij}^{obj}$

và S, B, λ_{coord}, λ_{noobj} là các tham số thực nghiệm.

Mô hình này có nhiều điểm vượt trội so với FRCNN. Tại pha nhận dạng, YOLO sẽ "nhìn" toàn bộ bức ảnh (thay vì từng phần bức ảnh), tức là những kết quả dự báo của nó được cung cấp thông tin bởi nội dung toàn cục của bức ảnh. YOLO thống nhất toàn bộ các thành phần riêng biệt trong bộ dò đối tượng vào một mạng duy nhất. Vì vậy tốc độ của YOLO là nhanh hơn rất nhiều so với FRCNN nhưng độ chính xác vẫn không suy giảm. Tuy nhiên, hạn chế của YOLO là chỉ nhận dạng ra vùng bao quanh đối tượng là hình chữ nhật.

3. Mô hình đề xuất

3.1 Kiến trúc mạng

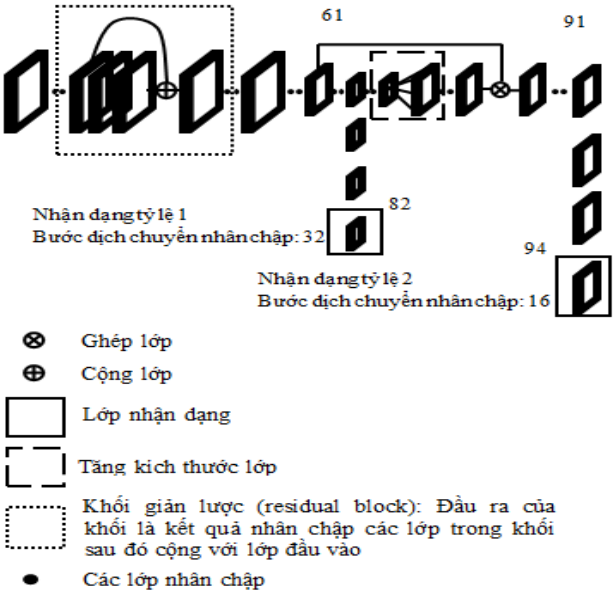
Thay vì sử dụng kiến trúc mạng CNN trong YOLO chúng tôi đề xuất sử dụng biến thể darknet53 với 94 tầng (hình 5) dự đoán ở 2 tỷ lệ như trong hình 4 phù hợp cho việc nhận dạng logo. Tỷ lệ logo to được nhận dạng ở tầng 82 và tại tầng 94 được dùng để nhận dạng logo có kích thước vừa.

3.2 Lựa chọn các hộp neo

Khác với YOLOv3 sử dụng tổng cộng 9 hộp neo để nhận dạng ở 3 tỷ lệ. Chúng tôi sử dụng 6 hộp neo cho 2 tỷ lệ. Để tìm ra các hộp neo này chúng tôi đã sử dụng giải thuật phân cụm K-Means tạo ra 6 hộp neo. Sắp xếp chúng theo chiều giảm kích thước, gán 3 hộp neo đầu cho tỷ lệ đầu và 3 hộp neo còn lại cho tỷ lệ sau.

3.3 Thay đổi hàm loss

Các phiên bản khác nhau của YOLOv1-3 chỉ xác định vùng box đứng là hình chữ nhật ngang vùng logo cần định vị, nhưng trong thực tế logo xuất hiện ở nhiều vị trí chéo. Để định vị được logo xuất hiện chéo trong video, chúng tôi đề xuất cách dự báo vùng bao là tứ giác lồi chứa ảnh logo (hình 6).



Hình 5. *Kiến trúc mạng*

Hình 6. *Tứ giác lồi bao một logo xuất hiện chéo trong ảnh.*

Quá trình học tham số dự báo sẽ thay công thức hàm loss (1) bởi công thức mới sau:

$$
\begin{aligned}
Loss = \\
&\lambda_{coord}\sum_{i=0}^{S^*}\sum_{j=0}^{A}\mathbb{1}_{ij}^{obj}\left[\left(b_{x_A}-b_{\hat{x}_A}\right)^2+\left(b_{y_A}-b_{\hat{y}_A}\right)^2+\left(b_{x_B}-b_{\hat{x}_B}\right)^2+\left(b_{y_B}-b_{\hat{y}_B}\right)^2\right. \\
&\left.+\left(b_{x_C}-b_{\hat{x}_C}\right)^2+\left(b_{y_C}-b_{\hat{y}_C}\right)^2+\left(b_{x_D}-b_{\hat{x}_D}\right)^2+\left(b_{y_A}-b_{\hat{y}_A}\right)^2\right] \\
&+\sum_{i=0}^{S^*}\sum_{j=0}^{A}\mathbb{1}_{ij}^{obj}\left(C_i-\hat{C}_i\right)^2 \\
&+\lambda_{noobj}\sum_{i=0}^{S^*}\sum_{j=0}^{A}\mathbb{1}_{ij}^{noobj}\left(C_i-\hat{C}_i\right)^2 \\
&+\sum_{i=0}^{S^*}\mathbb{1}_{i}^{obj}\sum_{c\in classes}\left(p_i(c)-\hat{p}_i(c)\right)^2
\end{aligned}
\tag{2}
$$

Với ABCD là tứ giác lồi được đánh vị trí như sau: A là đỉnh có tổng tọa độ x+y nhỏ nhất, C là đỉnh sao cho đoạn AC cắt đoạn nối hai đỉnh còn lại tại điểm trong của đoạn, B và D là 2 đỉnh còn lại sao cho ABCD thuận chiều kim đồng hồ.

Các cải tiến này cho phép nhận dạng chính xác hơn do nhận dạng ở 2 tỷ lệ, đồng thời cũng xác định vị trí của logo đa dạng hơn và tốc độ không suy giảm so với YOLO.

4. Thực nghiệm

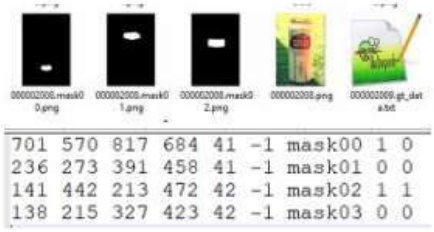
4.1 FlickrLogos-47

Để huấn luyện và kiểm thử mô hình, chúng tôi đã sử dụng bộ dữ liệu flickrlogos-47[21] được mở rộng, điều chỉnh từ bộ dữ liệu flickrlogos-32 rất phổ biến cho bài toán nhận dạng logo. Bộ dữ liệu flickrlogos-32 như tên gọi gồm 32 loại nhãn hiệu khác nhau với tổng 8240 ảnh trong đó có 4230 ảnh huấn luyện và 3960 ảnh kiểm thử. Trong mỗi tập huấn luyện và kiểm thử mỗi tập có 3000 ảnh nhiễu không chứa logo. Flickrlogos-47 bổ sung thêm dữ liệu mẫu và cải tiến từ flickrlogos-32 nhằm khắc phục một số hạn chế của tập dữ liệu này.

Số lớp trong flickrlogos-47 được nâng lên 47 lớp bằng cách bổ sung thêm ảnh, tách số lớp trong flickrlogos-32. Các nhãn hiệu trong flickrlogos-32 gồm cả biểu tượng và ký tự thì được tách thành 2 lớp trong flickrlogos-47. Mỗi ảnh trong flickrlogos-32 chỉ chứa một logo thuộc về một lớp duy nhất thì trong flickrlogos-47 trong một ảnh có thể có nhiều biến thể của một logo hoặc nhiều loại logo thuộc về một lớp hoặc nhiều lớp khác nhau. Ảnh nhiễu trong flickrlogos-32 bị loại bỏ trong flickrlogos-47. Một khác nữa của bộ dữ liệu flickrlogos-47 so với flickrlogos-32 là sự đa dạng về kích thước, đặc biệt là xuất hiện nhiều ảnh chứa các logo nhỏ nhằm tạo thêm độ khó cho việc nhận dạng.

4.2 Kết quả thực nghiệm

Chúng tôi thiết lập kiến trúc mạng Darknet53 biến thể 94 lớp như trong hình 3 trên GPU để huấn luyện và thử nghiệm các mô hình đề xuất với tập dữ liệu ảnh huấn luyện RGB flickrlogos-47 như đã giới thiệu phần trên sử dụng công cụ python 3.6.5 với dark flow.



Hình 7. *Ảnh huấn luyện minh họa*

Tập dữ liệu thử ảnh chứa logo lấy từ tập dữ liệu ảnh *Flickrlogos-47, t*ham số thực nghiệm S chọn S=7, B = 2 như minh họa tập ảnh trong hình 7.

Kết quả nhận dạng mAP với các kiến trúc học sâu tương ứng là:

Bảng 1. *Kết quả định vị và nhận dạng với chỉ số mAP của các kiến trúc học sâu.*

Dựa trên kiến trúc R-CNN	Dựa trên kiến trúc YOLO cải tiến
46%	**95%**

Thời gian chạy của thuật toán dò tìm và nhận dạng logo với các kiến trúc R-CNN và YOLOv3 tương ứng với chế độ với cấu hình CPU và GPU Nvidia 3.5 7GB RAM được cho ở bảng 2.

Bảng 2. *So sánh thời gian trung bình dò tìm và nhận dạng logo được cho ở bảng 2 sau*

Cấu hình	Dựa trên kiến trúc R-CNN	Dựa trên kiến trúc YOLO cải tiến
CPU	4,2 giây	**0,61 giây**
GPU	0.46 giây	**0,067 giây**

Dưới đây là một vài minh họa kết quả dò tìm:

Hình 8. *Kết quả dò tìm chính xác bởi YOLOv3 cho những logo biến dạng*

5. Kết luận

Nhận dạng logo là nền tảng cho nhiều lĩnh vực ứng dụng. Tuy nhiên có nhiều yếu tố ảnh hưởng đến việc nhận dạng này như: logo có thể xuất hiện ở nhiều vị trí, tỷ lệ và khung nhìn khác nhau trong ảnh. Hơn nữa các ảnh có thể bị tác động bởi nhiễu hay các yếu tố khác.

Cách tiếp cận truyền thống là sử dụng các bộ dò tìm và mô tả dựa trên điểm đặc trưng hoặc sử dụng các mạng CNN huấn luyện trước cho các ứng dụng cụ thể. Giải pháp của chúng tôi là cải tiến YOLO cho việc phân loại và nhận dạng logo. Kết quả thực nghiệm trên tập dữ liệu flickrlogo-47 cho độ chính xác cao so với FRCNN và YOLO nguyên bản.

TÀI LIỆU THAM KHẢO

[1]. S. Bianco, M. Buzzelli, D. Mazzini, and R. Schettini, "Logo recognition using cnn features", In *International Conference on Image Analysis and Processing*, pp. 438–448. Springer, 2015.

[2]. R. Boia and C. Florea, "Homographic class template for logo localization and recognition", In *Iberian Conference on Pattern Recognition and Image Analysis*, pp. 487-495, Springer, 2015.

[3]. Souvik Ghosh, Ranjan Parekh, "Automated Color Logo Recognition System based on shape and Color Features", *International Journal of Computer Applications*, 118(12), pp. 14-20, 2015.

[4]. A. D. Bagdanov, L. Ballan, M. Bertini, A. Del Bimbo, "Trademark matching and retrieval in sports video databases", in: *Proceedings of the international workshop on Workshop on multimedia information retrieval, ACM*, pp. 79-86, 2007.

[5]. J. Kleban, X. Xie, W.-Y. Ma, "Spatial pyramid mining for logo detection in natural scenes", in: *Multimedia and Expo, 2008 IEEE International Conference on, IEEE,* pp. 1077-1080, 2008.

[6]. A. Joly, O. Buisson, "Logo retrieval with a contrario visual query expansion", in: *Proceedings of the 17th ACM international conference on Multimedia, ACM,* pp. 581-584, 2009.

[7]. J. Meng, J. Yuan, Y. Jiang, N. Narasimhan, V. Vasudevan, Y. Wu, "Interactive visual object search through mutual information maximization", in: *Proceedings of the 18th ACM international conference on Multimedia, ACM*, pp. 1147-1150, 2010.

[8]. S. Bianco, M. Buzzelli, D. Mazzini, R. Schettini, "Logo recognition using cnn features", in: *Image Analysis and Processing ICIAP 2015, Springer,* pp. 438-448, 2015.

[9]. C. Eggert, A. Winschel, R. Lienhart, "On the benefit of synthetic data for company logo detection", in: *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, ACM,* pp. 1283-1286, 2015.

[10]. F. N. Iandola, A. Shen, P. Gao, K. Keutzer, "Deeplogo: Hitting logo recognition with the deep neural network hammer", arXiv preprint arXiv:1510.02131.

[11]. G. Oliveira, X. Fraz~ao, A. Pimentel, B. Ribeiro, "Automatic graphic logo detection via fast region-based convolutional networks", in: *Neural Networks (IJCNN), 2016 International JointConference on, IEEE,* pp. 985-991, 2016.

[12]. R. Girshick, J. Donahue, T. Darrell, J. Malik, "Region-based convolutional networks for accurate object detection and segmentation", *Pattern Analysis and Machine Intelligence, IEEE Transactions* on 38 (1), pp. 142-158, 2016.

[13]. S. Bianco, M. Buzzelli, D. Mazzini, R. Schettini, "Deep learning for logo recognition", in: *Neurocomputing*, 245, pp. 23-30, 2017.

[14]. S. Romberg, L. G. Pueyo, R. Lienhart, and R. van Zwol, "Scalable logo recognition in real-world images", In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ICMR '11, pp. 25:1–25:8, New York, NY, USA, 2011. ACM.

[15]. 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.*, 39(6), pp.1137–1149, June 2017.

[16]. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector", In *European conference on computer vision*, pp. 21–37.Springer, 2016.

[17]. J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018

[18]. J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision (IJCV)*, 2013.

[19]. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779– 788, 2016.

[20]. R. B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[21]. FlickrLogo-47 Dataset http://www.multimediacomputing.de/flickrlogos/