

# NHẬN DẠNG LOGO SỬ DỤNG NAÏVE BAYES NEAREST NEIGHBOR VÀ ĐẶC TRƯNG DSIFT

Võ Thành C<sup>1</sup>

**Tóm tắt** – Ngày nay, logo được xem là thương hiệu gắn liền với doanh nghiệp. Vì vậy, nhận dạng logo là bài toán rất được quan tâm trong xử lý ảnh. Bài báo này trình bày một phương pháp nhận dạng logo trong ảnh sử dụng giải thuật Naïve Bayes Nearest Neighbor (NBNN) và đặc trưng Dense Scale Invariant Feature Transform (DSIFT). Trước tiên, chúng tôi sử dụng Cascades of Boosted với tập đặc trưng Haar-like để dò tìm logo trong ảnh đầu vào. Tiếp theo, chúng tôi áp dụng giải thuật NBNN để nhận dạng logo với đặc trưng DSIFT. Tập dữ liệu thực nghiệm được thu thập từ Internet. Kết quả thực nghiệm 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:** Nhận dạng logo, đặc trưng SIFT, đặc trưng DSIFT, NBNN.

**Abstract** – Nowadays, logo is a trademark associated with company. Therefore, logo recognition is one of the great interest issues in image processing. This paper presents a method for logo recognition in the input image by using Naïve Bayes Nearest Neighbor (NBNN) algorithm and Dense Scale Invariant Feature Transform (DSIFT) feature. Firstly, we use Cascades of Boosted with a set of Haar-like features to detect the logo over the input image. Then, NBNN algorithm is employed to recognize the logo with DSIFT feature. Dataset is downloaded from Internet. The experimental results showed that this method achieved higher accuracy. Furthermore, it is extremely simple, efficient with less execution time and suitable to real-time logo recognition systems.

**Keywords:** Logo recognition, SIFT feature, DSIFT feature, NBNN.

## I. GIỚI THIỆU

Trong nền kinh tế thị trường ngày nay, logo được xem là biểu tượng quan trọng trong việc xây dựng, phát triển hình ảnh, thương hiệu của doanh nghiệp và trở thành một trong những sản phẩm sở hữu trí tuệ quan trọng của các doanh nghiệp trong kinh doanh. Hiện nay logo xuất hiện ngày càng nhiều trong các nhãn hiệu hàng hóa, các bảng tiếp thị, quảng cáo hoặc trên truyền hình, Internet. Bên cạnh đó, việc các doanh nghiệp mới ra đời sao chép hay “nhái” logo của những thương hiệu nổi tiếng đang là vấn đề làm đau đầu các nhà quản lý, nó trực tiếp gây hại đến thương hiệu của các doanh nghiệp và cả người tiêu dùng.

Hiện nay, có hai hướng tiếp cận chính trong nhận dạng logo [1]. Hướng thứ nhất 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 [2]. 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 so khớp mẫu, đơn giản là so sánh các điểm ảnh của các đối tượng cần nhận dạng với nhau. Tuy nhiên, việc so sánh này thường tốn nhiều thời gian và không thu được độ chính xác cao.

Những năm gần đây, hướng tiếp cận thứ hai dựa trên đặc trưng cục bộ đã được nhiều tác giả nghiên cứu và vận dụng thành công vào bài toán phát hiện và nhận dạng logo như đặc trưng Harris, SIFT [3], PCSIFT [4]. 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

<sup>1</sup>Khoa Kỹ thuật và Công nghệ, Trường Đại học Trà Vinh  
Ngày nhận bài: 22/4/2016, ngày nhận kết quả bình duyệt: 22/11/2016, ngày chấp nhận đăng: 20/12/2016

chiếu sáng, nhiều, sự che khuất một phần. Ngoài ra, việc biểu diễn ảnh theo các đặc trưng cục bộ sẽ cô đọng hơn vì thế sẽ giảm kích thước trong không gian tìm kiếm hoặc so sánh. Cách tiếp cận này sẽ sử dụng mô hình máy học để nhận dạng logo như mạng nơron [5], mạng bayesian ,SVM và mô hình “Bag of Words” [3]. Hướng tiếp cận thứ hai này cho độ chính xác cao hơn hướng tiếp cận thứ nhất. Tuy nhiên, việc sử dụng đặc trưng cục bộ có thể làm mất thông tin về bố trí không gian đối với những ảnh có kích thước nhỏ nên có thể thiếu khả năng mô tả ảnh. Ngoài ra, việc sử dụng mô hình máy học thường phức tạp, tốn thời gian huấn luyện và phải huấn luyện lại nếu có thay đổi về tập dữ liệu huấn luyện.

Trong bài báo này, chúng tôi đề xuất sử dụng giải thuật NBNN với đặc trưng DSIFT để nhận dạng logo. Giải thuật NBNN là một giải thuật đơn giản, không cần quá trình học nên rất thích hợp với các ứng dụng nhận dạng logo trực tuyến bởi vì không phải huấn luyện lại mô hình máy học khi thay đổi tập dữ liệu huấn luyện mà vẫn có thể cập nhật trực tiếp vào cơ sở dữ liệu đối tượng khi có một logo mới được nhận dạng thành công. Kết quả thực nghiệm cũng cho thấy phương pháp đề xuất đạt được độ chính xác cao và thời gian xử lý nhanh, đáp ứng yêu cầu thời gian thực.

Phần tiếp theo của bài báo được tổ chức như sau: phần 2 trình bày về phương pháp nhận dạng logo sử dụng NBNN và đặc trưng DSIFT; phần 3 trình bày về kết quả thực nghiệm; cuối cùng kết luận sẽ được trình bày trong phần 4.

## II. KIẾN TRÚC CỦA HỆ THỐNG NHẬN DẠNG LOGO

### A. Kiến trúc tổng quát

Một hệ thống nhận dạng logo tự động bao gồm 2 bộ phận chính (Hình 1): (1) Phát hiện và rút trích logo từ ảnh đầu vào; (2) Nhận dạng logo được rút trích từ (1) và trả về kết quả nhận dạng.

- *Bộ phát hiện và rút trích logo*: Để nhận dạng logo, đầu tiên hệ thống phải phát hiện và định vị được các vùng chứa logo trong ảnh truy vấn. Sau đó, hệ thống tự động rút trích vùng chứa logo và chuyển cho bộ nhận dạng logo để xử lý. Như vậy, một hệ thống nhận dạng logo tốt trước tiên phải đảm bảo phát hiện chính xác vùng có chứa logo. Độ chính xác của bộ phát hiện và định vị logo

có ảnh hưởng rất lớn đến kết quả nhận dạng của toàn hệ thống. Nếu bộ phát hiện và định vị logo xác định sai vùng chứa logo thì kết quả nhận dạng cũng sai.

- *Bộ nhận dạng logo*: Có chức năng nhận dạng logo từ vùng chứa logo đã được rút trích và trả về kết quả nhận dạng. Ngoài yêu cầu về độ chính xác cao, thời gian xử lý nhanh cũng là một tiêu chí quan trọng đối với các hệ thống nhận dạng logo.

### B. Định vị logo với giải thuật Cascades of Boosted sử dụng đặc trưng Haar-like

#### 1) Đặc trưng Haar-like

Đặc trưng Haar-like [6] là đặc trưng được sử dụng phổ biến trong việc nhận dạng đối tượng. Đặc trưng Haar-like biểu diễn mối liên hệ tương quan giữa các bộ phận trong ảnh mà bản thân giá trị từng pixel không diễn đạt được. Đặc trưng Haar-like được tạo thành bằng việc kết hợp các hình chữ nhật đen, trắng với nhau theo một trật tự, một kích thước nào đó. Có 4 đặc trưng Haar-like cơ bản.

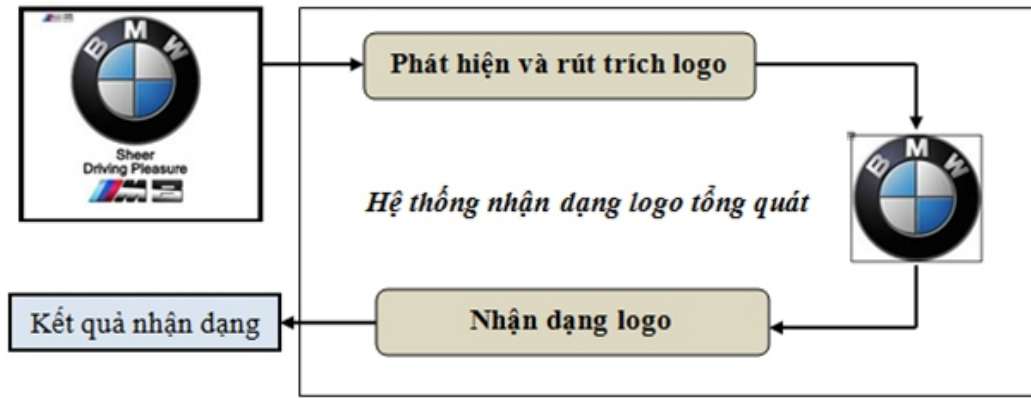
Năm 2002, Rainer Lienhart và Jochen Maydt đã mở rộng các đặc trưng Haar-like cơ bản lên 14 đặc trưng và chia thành các nhóm đặc trưng cạnh, đặc trưng đường và đặc trưng xung quanh tâm [7].

Để tính giá trị các đặc trưng Haar-like trong ảnh chứa logo, cho một cửa sổ hình chữ nhật có kích thước  $24 \times 24$  pixel lướt qua toàn bộ ảnh, trong khi di chuyển cửa sổ trượt chậm mặt nạ lên vùng cần tính. Giá trị của đặc trưng Haar-like là sự chênh lệch tổng các điểm ảnh giữa các vùng đen và các vùng trắng như trong công thức sau:

$$F(x) = \text{Tổng}_{\text{vùng đen}}(\text{pixel}) - \text{Tổng}_{\text{vùng trắng}}(\text{pixel}) \quad (2.1)$$

#### 2) AdaBoost

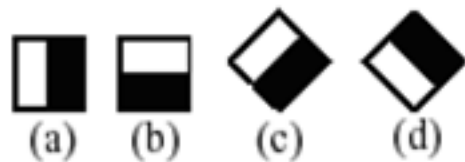
AdaBoost được giới thiệu bởi Yoav Freund và Schapire [8] vào năm 1995, là một bộ phân loại mạnh phi tuyến phức, hoạt động trên nguyên tắc kết hợp tuyến tính các bộ phân loại yếu để tạo nên một bộ phân loại mạnh. AdaBoost sử dụng trọng số để đánh dấu các mẫu khó nhận dạng. Trong quá trình huấn luyện, cứ mỗi bộ phân loại yếu được xây dựng thì thuật toán sẽ tiến hành cập nhật lại trọng số để chuẩn bị cho việc xây dựng bộ phân loại tiếp theo. Cập nhật bằng cách



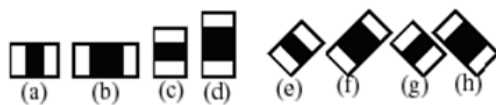
Hình 1: Hệ thống nhận dạng logo tổng quát



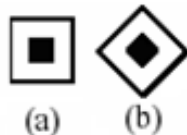
Hình 2: Các đặc trưng Haar-like cơ bản



Đặc trưng cạnh



Đặc trưng đường



Đặc trưng xung quanh tâm

Hình 3: Các đặc trưng Haar-like mở rộng

tăng trọng số của các mẫu nhận dạng sai và giảm trọng số của các mẫu được nhận dạng đúng bởi bộ phân loại yếu vừa xây dựng. Bằng cách này thì bộ phân loại sau sẽ tập trung vào các mẫu mà bộ phân loại trước nó nhận dạng sai. Cuối cùng các bộ phân loại yếu sẽ được kết hợp lại tùy theo mức độ tốt của chúng để tạo nên một bộ phân loại mạnh.

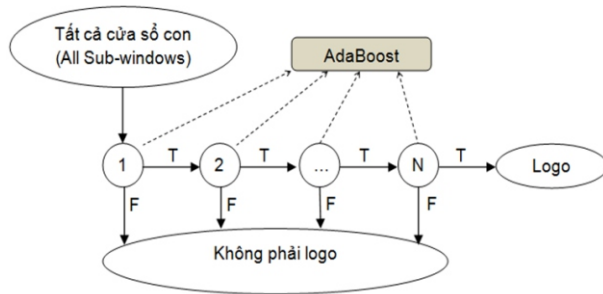
Bộ phân loại yếu  $h_j(x)$  được biểu diễn như sau:

$$h_j(x) = \begin{cases} 1 & \text{Nếu } p_j f_j(x) < p_j \theta_j \\ 0 & \text{Nếu ngược lại} \end{cases} \quad (2)$$

Với  $x$  là cửa sổ con cần xét,  $\theta_j$  là ngưỡng,  $f_j$  là giá trị đặc trưng Haar-like và  $p_j$  là hệ số quyết định chiều của phương trình.

### 3) Mô hình Cascades of Boosted

Mô hình Cascades of Boosted được xây dựng nhằm rút ngắn thời gian xử lý, giảm thiểu nhận dạng lầm cho bộ phân loại [8]. Cascades of Boosted gồm nhiều tầng, mỗi tầng là một mô hình AdaBoost với bộ phân lớp yếu là các cây quyết định. Một mẫu để được phân loại là logo thì nó cần phải đi qua hết tất cả các tầng. Tầng sau được huấn luyện bằng những mẫu mà các tầng trước nó chấp nhận, tức là nó sẽ tập trung học từ các mẫu khó hơn. Do đó sự kết hợp các tầng này lại sẽ giúp bộ phân loại giảm thiểu nhận dạng lầm. Với cấu trúc này, mô hình Cascades of Boosted cố gắng loại bỏ nhiều mẫu không phải là logo ngay từ những tầng đầu tiên, giúp đáp ứng tốt nhất thời gian xử lý và vẫn duy trì được hiệu quả phát hiện logo.



Hình 4: Mô hình Cascades of Boosted

### C. Nhận dạng logo với giải thuật NBNN sử dụng đặc trưng DSIFT

#### 1) Giới thiệu về SIFT

SIFT (ScaleInvariant Feature Transform) [9] là đặc trưng cục bộ bất biến đối với những phép biến đổi tỷ lệ, tịnh tiến, phép quay và không đổi một phần đối với những thay đổi về góc nhìn, đồng thời nó cũng rất mạnh với những thay đổi về độ sáng, sự che khuất, nhiễu. Phương pháp trích rút đặc trưng SIFT có thể được tóm tắt như sau: (1) *Phát hiện các điểm cực trị trong không gian tỷ lệ (Scale-Space extrema detection)*: sử dụng hàm sai khác Gaussian (Different of Gaussian - DoG) để xác định tất cả các điểm hấp dẫn tiềm năng mà bất biến với tỷ lệ và hướng của ảnh; (2) *Định vị các điểm hấp dẫn (Keypoint localization)*: ứng với mỗi vị trí tiềm năng, hàm kiểm tra sẽ được đưa ra để quyết định xem các điểm hấp dẫn tiềm năng có được lựa chọn hay không. Các điểm hấp dẫn được lựa chọn dựa trên việc đo lường tính ổn định của chúng; (3) *Xác định hướng cho các điểm hấp dẫn (Orientation assignment)*: một hoặc nhiều hướng được gán cho mỗi vị trí điểm hấp dẫn dựa trên hướng gradient cục bộ của ảnh; (4) *Mô tả các điểm hấp dẫn (Keypoint descriptor)*: các gradient ảnh cục bộ được xác định ở tỷ lệ được chọn trong vùng bao quanh mỗi điểm hấp dẫn. Các gradient được biểu diễn sang một dạng mà cho phép bất biến với sự thay đổi về hình dạng và điều kiện chiếu sáng.

Ảnh giữa là mô phỏng biên độ gradient và hướng tại mỗi mẫu ảnh trong một vùng lân cận với điểm hấp dẫn. Mỗi điểm hấp dẫn sau khi được xác định hướng sẽ được mô tả dưới dạng một vectơ đặc trưng có  $4 \times 4 \times 8 = 128$  chiều.

#### 2) Giới thiệu về DSIFT

DSIFT là một thuật toán có thời gian xử lý nhanh và được kế thừa từ SIFT [10]. Thay vì chỉ tính toán bộ mô tả xung quanh điểm hấp dẫn, DSIFT sẽ chia ảnh thành D phần bằng nhau và rút trích đặc trưng từ các phần của ảnh. DSIFT đưa ra một số giả thuyết mới: (a) Vị trí của các điểm cần rút trích đặc trưng được xác định trước; (b) Tỷ lệ để tính bộ mô tả đặc trưng tại tất cả các điểm thì bằng nhau và cũng được xác định trước; (c) Hướng của mỗi điểm cần tính bộ mô tả giống nhau và được xác định trước.

Như vậy, cách tiếp cận DSIFT sẽ tính toán bộ mô tả xung quanh tất cả các điểm đầy đặc trên một lưới đồng nhất. Nguyên lý của DSIFT là mỗi phần của ảnh đều chứa đựng các thông tin có giá trị để biểu diễn nội dung của ảnh. Với giả thuyết này, DSIFT sẽ rút trích được nhiều đặc trưng hơn trong thời gian ít hơn so với thuật toán SIFT.

#### 3) Giải thuật nhận dạng logo với NBNN và đặc trưng DSIFT

Giải thuật NBNN được Boiman và các cộng sự đề xuất vào năm 2008. NBNN thực hiện tính toán trực tiếp khoảng cách từ “ảnh đến lớp” để thực hiện phân lớp ảnh [11]. Ưu điểm của NBNN là đơn giản, không cần quá trình học nên rất thích hợp với các ứng dụng nhận dạng logo trực tuyến bởi vì không phải huấn luyện lại mô hình máy học khi thay đổi tập dữ liệu huấn luyện mà vẫn có thể cập nhật trực tiếp vào cơ sở dữ liệu đối tượng khi có một logo mới được nhận dạng thành công. NBNN là phương pháp phân loại ảnh rất thành công và được mở rộng trong các ứng dụng tương tự.

Giải thuật nhận dạng logo với NBNN và đặc trưng DSIFT có thể được tóm tắt như sau:

#### Đầu vào:

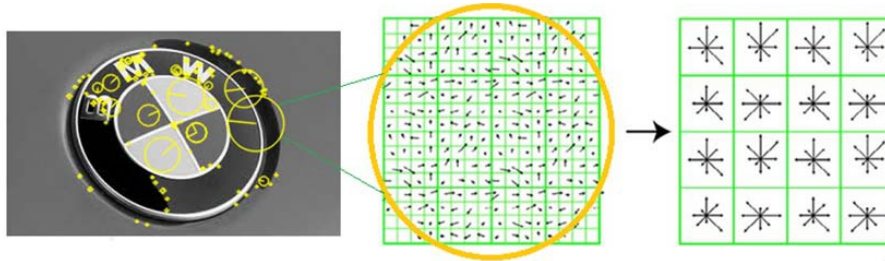
- Logo cần nhận dạng

#### Các bước thực hiện:

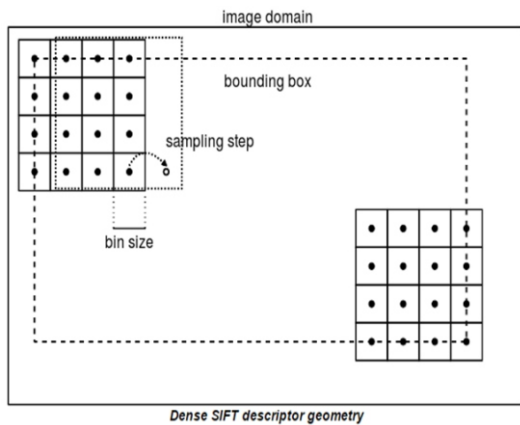
- Bước 1: Tính tất cả đặc trưng DSIFT của logo cần nhận dạng. Mỗi đặc trưng DSIFT được biểu diễn dưới dạng một vectơ có số chiều là 128.

- Bước 2: Ứng với mỗi đặc trưng  $d_i$  của logo cần nhận dạng, tìm láng giềng gần nhất của  $d_i$  trong từng lớp của cơ sở dữ liệu đối tượng:  $NN_C(d_i)$ .

- Bước 3: Ứng với mỗi lớp C trong cơ sở dữ liệu đối tượng, tính tổng khoảng cách từ các đặc trưng di của logo cần nhận dạng với láng giềng



Hình 5: Đặc trưng SIFT được tính toán từ vùng xung quanh điểm đặc biệt



Hình 6: Xác định các điểm cần rút trích đặc trưng DSIFT

gần nhất của nó trong lớp C:

$$\sum_{n=1}^n \|d_n - NN_C(d_i)\|^2$$

- Bước 4: Tìm lớp có tổng khoảng cách tính được ở bước 3 là nhỏ nhất. Lớp tìm được chính là lớp mà logo thuộc về.

**Đầu ra:**

- Phân lớp C mà logo thuộc về

**Trong đó:**

-  $d_i$ : Đặc trưng DSIFT thứ  $i$  của logo. Mỗi  $d_i$  là một vectơ có 128 chiều.

-  $NN_C(d_i)$ : Láng giềng gần nhất của  $d_i$  trong lớp C.

**III. THỰC NGHIỆM**

Để đánh giá hiệu quả của phương pháp đề xuất, chúng tôi đã cài đặt hệ thống bằng ngôn ngữ lập trình C/C++, sử dụng thư viện OpenCV để trích đặc trưng DSIFT. Trong giải thuật NBNN cần tìm láng giềng gần nhất của mỗi đặc trưng

Bảng 1: Công cụ dùng trong thực nghiệm

Loại	Tên và kích thước sử dụng
Hiệu máy tính	Laptop HP EliteBook 8540w
Bộ xử lý (CPU)	Intel Core i5 CPU M - 520@2.40GHz x 4
Bộ nhớ (RAM)	4096MB-DDR3
Hệ điều hành	Ubuntu 14.04 OS
Ngôn ngữ lập trình	C/C++, trình biên dịch g++ Linux
IDE	Qt Creator 5.3.1
Thư viện	OpenCV 2.4.9

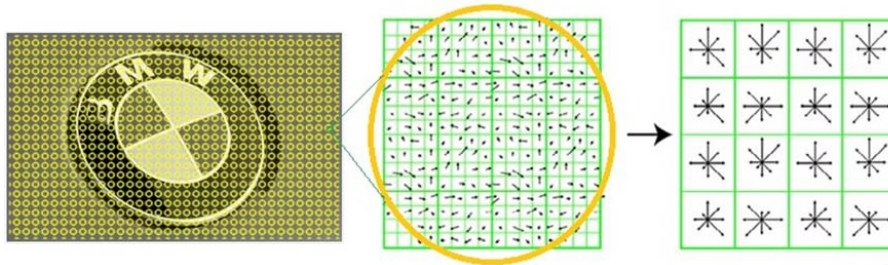
DSIFT của logo cần nhận dạng đến các đặc trưng DSIFT trong cơ sở dữ liệu đối tượng, quá trình này thường mất nhiều thời gian. Để tăng tốc quá trình này, chúng tôi sử dụng cấu trúc chỉ mục KD-Tree đã được cài đặt trong thư viện FLANN - Fast Library for Approximate Nearest Neighbors (Muja 2009), hỗ trợ cho NBNN có thể tính toán nhanh hơn khi phân lớp.

Các thực nghiệm được thực hiện trên một máy tính cá nhân chạy hệ điều hành Linux với bản phân phối Ubuntu 14.04. Thông tin máy tính và các ứng dụng được cài đặt như trong bảng 1.

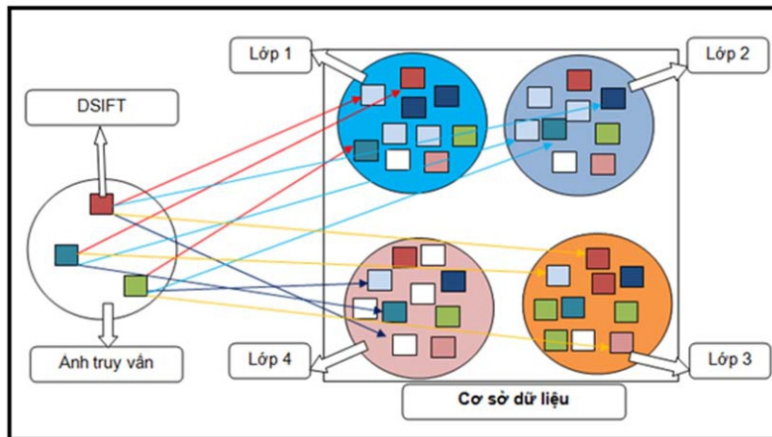
**A. Tập dữ liệu**

Tập dữ liệu gồm các ảnh có chứa logo được thu thập từ Internet. Các ảnh có độ phân giải khác nhau, bao gồm các ảnh chứa các logo có





Hình 7: Đặc trưng DSIFT được tính toán từ vùng xung quanh điểm trên lưới (vòng tròn)



Hình 8: Láng giềng gần nhất từ một đặc trưng DSIFT của ảnh truy vấn đến lớp đối tượng

kích thước rất nhỏ ( $24 \times 24$  pixel) đến các logo có kích thước rất lớn (lớn hơn  $1000 \times 1000$  pixel). Tập dữ liệu thực nghiệm gồm có hai tập: (1) *Tập dữ liệu thứ nhất*: Tự thu thập từ Internet, gồm có 3 lớp dương: BMW, Paulaner, Ford và 1 lớp âm; (2) *Tập dữ liệu thứ hai*: FlickrLogos-32 do cộng đồng nhận dạng logo tạo ra, gồm có 32 lớp dương và 1 lớp âm. Mỗi lớp dương chứa 70 ảnh logo, lớp âm chứa 6000 ảnh không phải logo.

### B. Kết quả thực nghiệm phát hiện và định vị logo

Mô hình phát hiện và định vị logo được thực nghiệm trên tập dữ liệu do chúng tôi tự thu thập từ Internet. Để phát hiện và định vị logo, chúng tôi đã huấn luyện mô hình Cascades of Boosted gồm 19 tầng dựa trên các đặc trưng Haar-like và mô hình máy học AdaBoost. Với kích thước khởi đầu của bộ phát hiện là  $24 \times 24$  pixel, cửa sổ con sẽ được quét qua toàn bộ ảnh để tìm các vùng có chứa logo trong ảnh. Sau đó tăng kích thước của cửa sổ lên theo tỷ lệ 1,05. Chúng tôi đã thực

thực nghiệm trên 295 ảnh có độ phân giải khác nhau, mỗi ảnh có thể chứa một hoặc nhiều logo BMW. Kết quả thực nghiệm cho thấy độ chính xác của mô hình phát hiện và định vị logo đạt được độ chính xác 98,98%.

### C. Kết quả thực nghiệm nhận dạng logo

Để đánh giá kết quả nhận dạng logo với giải thuật NBNN và đặc trưng DSIFT, chúng tôi đã thực nghiệm trên tập dữ liệu FlickrLogos-32 như sau:

- Tập huấn luyện: Gồm 1280 logo từ 32 lớp dương (mỗi lớp 40 logo) và 3000 ảnh không chứa logo trong lớp âm đại diện cho cơ sở dữ liệu đối tượng.

- Tập kiểm tra: Gồm 960 logo từ 32 lớp dương (mỗi lớp 30 logo) và 3000 ảnh không chứa logo trong lớp âm.

Bên cạnh đó, chúng tôi cũng cài đặt hệ thống nhận dạng logo với giải thuật NBNN và đặc trưng SIFT. Để so sánh kết quả nhận dạng khi dùng DSIFT với SIFT, chúng tôi sử dụng các độ đo

Precision, Recall và F1. Kết quả thực nghiệm được ghi nhận trong bảng 2. Trong đó:

Bảng 2: So sánh kết quả nhận dạng khi dùng đặc trưng DSIFT và SIFT

	Precision	Recall	F1	Thời gian trung bình nhận dạng 1 logo
NBNN + DSIFT	99,83%	82,12%	90,11%	0,28 giây
NBNN + SIFT	99,82%	80,48%	89,11%	0,24 giây

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Kết quả trên cho thấy khi phân lớp trên tập dữ liệu lớn gồm nhiều lớp thì DSIFT cho kết quả trung bình tốt hơn SIFT. Về thời gian xử lý, thuật toán NBNN + DSIFT phân lớp 3960 logo (32 lớp dương + 1 lớp âm) với tổng thời gian là 18,37 phút. Như vậy thời gian trung bình để nhận dạng một logo đã được rút trích là 0,28 giây. Với kết quả thu được, có thể thấy rằng phương pháp đề xuất vẫn có thời gian xử lý nhanh và đạt được độ chính xác cao. Từ đó, chúng ta có thể áp dụng trong thực tế.

#### IV. KẾT LUẬN

Bài báo đã trình bày một giải pháp nhận dạng logo sử dụng giải thuật NBNN và đặc trưng DSIFT. Một giải thuật máy học Cascades of Boosted dựa trên AdaBoost với tập các đặc trưng Haar-like được áp dụng để định vị logo trong ảnh. Kết quả thực nghiệm cho thấy phương pháp đề xuất đạt được độ chính xác cao với thời gian xử lý nhanh, có thể đáp ứng yêu cầu về tính thời gian thực. Ngoài ra, phương pháp đề xuất cũng giải quyết được vấn đề liên quan đến sự thay đổi góc nhìn, tính không đối xứng của logo cần nhận dạng.

Trong tương lai, chúng tôi sẽ ứng dụng kết quả nghiên cứu trong các hệ thống nhận dạng đối tượng nhằm đáp ứng nhu cầu thực tế như:

hệ thống nhận diện thương hiệu thông qua logo, hệ thống tìm kiếm sản phẩm thông qua logo, hệ thống hỗ trợ đăng ký thương hiệu,... Bên cạnh đó, chúng tôi tiếp tục nghiên cứu các giải thuật máy học khác cũng như việc kết hợp thêm các đặc trưng về màu sắc, hình dáng, kết cấu nhằm nâng cao khả năng nhận dạng.

#### TÀI LIỆU THAM KHẢO

- [1] Hammad Naeem, Maria Minhas, Jameel Ahmed. A comparative study about object classification based on global and local features. *IJCSI International Journal of Computer Science Issues*. 2013;10(2):154–158.
- [2] Souvik Ghosh, Ranjan Parekh. Automated Color Logo Recognition System based on Shape and Color Features. *International Journal of Computer Applications*. 2015;118(12):14–20.
- [3] Romberg S, Pueyo LG, Lienhart R, Zwol R Van. Scalable logo recognition in real-world images. *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*. 2011;Trento, Italy.
- [4] Karthikeyan S, Kalaiyarasi C, Saranya P, V Aishwarya Lakshmi. Enhancing Logo Matching and Recognition Using Local Features. *Journal of Engineering Research and Applications*. 2014;4(2):27–32.
- [5] Gayathri Akashaya. Logo Detection and Recognition Using Artificial Neural Network. *International Journal of Advanced Research in Computer Engineering & Technology*. 2015;4(5):2045–2048.
- [6] Paul Viola, Michael J Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2001;1:511–518. Kauai, Hawaii, USA.
- [7] Rainer Lienhart, Alexander Kuranov, Pisarevsky V. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. *Proceedings of the 25<sup>th</sup> DAGM Symposium on Pattern Recognition*. 2003;Magdeburg, Germany.
- [8] Yoav Freund, Robert E Schapire. A Short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence*. 1999;14(5):771–780.
- [9] David G Lowe. Distinctive image features from Scale-Invariant keypoints. *International Journal of Computer Vision*. 2004;60(2):91–110.
- [10] Vedaldi A, Fulkerson. VLFeat – An open and portable library of computer vision algorithms. *Proceedings of the 18th ACM International Conference on Multimedia*. 2010;p. 1469–1472. Firenze, Italy.
- [11] On Boiman, E Shechtman, Ireani M. In Defense of Nearest-Neighbor Based Image Classification. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008*. 2008;p. 1–8. IEEE.