

TRA CỨU ẢNH THEO NGỮ NGHĨA DỰA TRÊN CÂY PHÂN CỤM PHÂN CẤP

Nguyễn Minh Hải¹, Lê Thị Vĩnh Thanh², Văn Thế Thành³, Trần Văn Lãng^{4,*}

¹ Khoa Vật lý, Trường Đại học Sư phạm TP. Hồ Chí Minh

² Viện Công nghệ Thông tin - Điện - Điện tử, Trường Đại học Bà Rịa - Vũng Tàu

³ Phòng Quản lý khoa học và Đào tạo Sau đại học, Trường Đại học Công nghiệp Thực phẩm TP. Hồ Chí Minh

⁴ Viện Cơ học và Tin học ứng dụng, Viện Hàn Lâm Khoa học và Công Nghệ Việt Nam

hainm@hcmue.edu.vn, lethivinhtanh.hcm@gmail.com, vanthethanh@gmail.com, langtv@vast.vn

TÓM TẮT: Trong bài báo này, chúng tôi trình bày một số kết quả về việc xây dựng hệ truy vấn và trích xuất ngữ nghĩa hình ảnh SBIR (Semantic-based Image Retrieval) trên cơ sở cấu trúc cây phân cụm phân cấp H-Tree (Hierarchical-Tree). Cấu trúc dữ liệu này được tạo ra dựa trên phương pháp phân cụm phân cấp các véc-tơ đặc trưng thị giác của hình ảnh. Vì vậy, cấu trúc cây H-Tree được xây dựng trên cơ sở các thao tác như thêm, xóa, cập nhật; đồng thời phương pháp truy vấn và trích xuất ngữ nghĩa được đề xuất. Để minh chứng tính đúng đắn của cơ sở lý thuyết đã được đề nghị, chúng tôi thiết kế mô hình và xây dựng ứng dụng thực nghiệm truy vấn ảnh ngữ nghĩa trên bộ dữ liệu ảnh ImageCLEF. Nhằm đánh giá hiệu quả của phương pháp đề xuất, kết quả thực nghiệm được so sánh với một số kết quả của các công trình đã được công bố gần đây trên cùng bộ dữ liệu. Theo kết quả so sánh, phương pháp của chúng tôi đề xuất có tính khả thi và đạt được hiệu quả cao.

Từ khóa: Hierarchical clustering, Similar images, Similarity measure, Image retrieval.

I. GIỚI THIỆU

Dữ liệu đa phương tiện ngày càng được phát triển nhanh chóng về mặt số lượng và kích thước. Theo WordStream, năm 2019, mỗi ngày có 8.95 triệu ảnh, video được chia sẻ trên Instagram và cho đến nay có hơn 40 tỷ hình ảnh. Theo thống kê của Google, khoảng 93 triệu ảnh được tạo ra mỗi ngày kể từ 2014; theo như Gigaom, năm 2017, một người trung bình tạo ra 630 ảnh trên điện thoại; cũng theo Mylio, năm 2017, mỗi năm một người tạo ra ít nhất 1000 ảnh, mỗi năm tạo ra ít nhất 1 nghìn tỷ hình ảnh,... Từ đó cho thấy kích thước cũng như số lượng ảnh ngày càng được gia tăng nên cần phải có các hệ thống truy vấn ảnh số tự động trên các thiết bị cũng như trong các hệ thống đa phương tiện. Do đó, bài toán tra cứu ảnh để tìm ra tập ảnh tương tự và trích xuất ngữ nghĩa hình ảnh là một bài toán quan trọng trong các hệ thống đa phương tiện và phù hợp với xu thế của xã hội hiện đại. Trong bài báo này, chúng tôi đề xuất một phương pháp tra cứu ảnh trên cơ sở kỹ thuật khai phá dữ liệu phân cụm phân cấp để từ đó xây dựng một hệ truy vấn ảnh theo hướng tiếp cận ngữ nghĩa.

Có hai vấn đề chính yếu khi thực hiện bài toán truy vấn ảnh đó là mô tả nội dung thị giác của hình ảnh và xây dựng dữ liệu mô tả cho nội dung thị giác dưới dạng chỉ mục. Từ đó, bài toán tra cứu ảnh được thực hiện cũng như việc gia tăng tốc độ và độ chính xác khi tìm kiếm hình ảnh trong các hệ thống đa phương tiện có tập dữ liệu ảnh lớn. Đối với bài toán tìm kiếm ảnh theo ngữ nghĩa SBIR, hai vấn đề trên vẫn phải được thực hiện vì bản chất quá trình truy vấn có đầu vào là một hình ảnh tuy nhiên cần bổ sung hai vấn đề bao gồm chuyển đổi ảnh truy vấn trở thành ngữ nghĩa mô tả và dựa trên ngữ nghĩa mô tả này để tìm kiếm các hình ảnh trên một Ontology cho trước cũng như tìm các ngữ nghĩa mô tả cho tập ảnh tương tự.

Theo những phân tích như trên, chúng tôi xây dựng một hệ thống truy vấn ảnh theo tiếp cận ngữ nghĩa trên cơ sở xây dựng một cấu trúc dữ liệu cây phân cụm phân cấp; cấu trúc này được gọi là cây H-Tree. Để tạo ra cấu trúc dữ liệu này, chúng tôi dựa trên kỹ thuật học bán giám sát đồng thời kết hợp phương pháp phân cụm phân hoạch và phân cụm phân cấp. Cây phân cụm phân cấp H-Tree là một mô hình để phân hoạch tự động các bộ dữ liệu đồng thời có thể tăng trưởng số nhánh nhằm đáp ứng nhu cầu gia tăng dữ liệu hình ảnh.

Đóng góp của bài báo gồm: (1) Xây dựng một cấu trúc cây H-Tree đa nhánh nhằm tạo ra một mô hình phân loại dữ liệu cũng như giúp quá trình tìm kiếm được hiệu quả về tốc độ và độ chính xác; (2) Đề xuất một mô hình truy vấn ảnh theo tiếp cận ngữ nghĩa để từ đó ánh xạ đặc trưng thị giác cấp thấp trở thành ngôn ngữ thị giác cấp cao; (3) Đề xuất các thuật toán tạo cây H-Tree cũng như thuật toán tìm kiếm ảnh; (4) Kết hợp phương pháp phân cụm phân hoạch và phân cụm phân cấp trên cơ sở thuật toán K-Mean nhằm tạo ra một mô hình phân hoạch dữ liệu bằng cây H-Tree; (5) Xây dựng một hệ truy vấn ảnh theo ngữ nghĩa mà dữ liệu đầu vào là một ảnh truy vấn.

Phần còn lại của bài báo được tổ chức như sau. Phần 2, giới thiệu các công trình liên quan, nhằm phân tích ưu khuyết điểm những công trình đã có và đưa ra cách khắc phục cũng như phân tích tính khả thi của hướng tiếp cận của bài báo. Phần 3, mô tả quá trình xây dựng cấu trúc cây phân cụm phân cấp H-Tree. Phần 4, xây dựng mô hình cho bài toán truy vấn ảnh theo tiếp cận ngữ nghĩa sử dụng cây H-Tree. Phần 5, xây dựng ứng dụng thực nghiệm và đánh giá kết quả dựa trên cơ sở lý thuyết đã đề nghị. Phần 6, đưa ra kết luận và hướng phát triển.

II. CÁC CÔNG TRÌNH LIÊN QUAN

Trong những năm gần đây, có nhiều nhóm nghiên cứu về nâng cao hiệu quả cho bài toán tìm kiếm ảnh theo nội dung và theo ngữ nghĩa như tìm kiếm ảnh và phân tích ngữ nghĩa dựa trên phương pháp đánh trọng số đặc trưng vùng trên ảnh và ứng dụng kỹ thuật học không giám sát [1], truy vấn ảnh dựa trên kỹ thuật phân lớp bằng máy vectơ hỗ trợ SVM[2], truy vấn ảnh dựa trên cấu trúc dữ liệu cây S-Tree [4], truy vấn ảnh theo ngữ nghĩa dựa trên mô hình phân lớp phân cấp bằng kỹ thuật mạng nơron học sâu [5], truy vấn ảnh theo nội dung và trích xuất ngữ nghĩa sử dụng kỹ thuật máy vectơ hỗ trợ SVM [6], truy vấn theo ngữ nghĩa hình ảnh dựa trên kỹ thuật phân lớp các vùng đặc trưng của ảnh [7], truy vấn ngữ nghĩa dựa trên kỹ thuật phân lớp đặc trưng thị giác của hình ảnh [8],...

Năm 2015, A. Alzu'bi và cộng sự khảo sát và đánh giá tính khả thi của hệ truy vấn ảnh theo nội dung CBIR và theo ngữ nghĩa SBIR. Trong công trình này, nhóm tác giả đã phân tích những kỹ thuật cơ sở cho bài toán CBIR và SBIR đồng thời đề xuất các đặc trưng của hình ảnh cần phải có để thực hiện quá trình truy vấn ảnh. Các kỹ thuật về máy học, khai phá dữ liệu cũng được giới thiệu để làm cơ sở xây dựng bài toán truy vấn ảnh. Theo đó, nhóm tác giả đã giới thiệu nhiều bộ dữ liệu có thể được ứng dụng để xây dựng các hệ thống truy vấn ảnh nhằm đánh giá tính hiệu quả của từng phương pháp đề xuất. Từ công trình này cho thấy bài toán truy vấn ảnh theo nội dung và theo ngữ nghĩa là một bài toán khả thi và có thể áp dụng cho nhiều hệ thống đa phương tiện khác nhau [3].

Năm 2016, Van T.T. và cộng sự đã giới thiệu một phương pháp cải tiến cho hệ truy vấn ảnh theo nội dung. Trong công trình này, nhóm tác giả đã đề xuất cải tiến một cấu trúc dữ liệu cây đa nhánh và thực nghiệm trên bộ ảnh COREL. Tuy nhiên, trong công trình này, nhóm tác giả chưa phân tích ngữ nghĩa hình ảnh, cây S-Tree chưa thể lưu trữ chỉ mục và chú thích của hình ảnh đồng thời nhóm tác giả chưa tạo ra câu truy vấn SPARQL để thực hiện hệ truy vấn theo tiếp cận ngữ nghĩa của hình ảnh [4].

Năm 2016, A. L. Mane xây dựng hệ truy vấn ảnh theo tiếp cận ngữ nghĩa bằng cách trích xuất đặc trưng về nội dung của hình ảnh đồng thời đánh chỉ mục. Từ đó, tác giả thực hiện phân loại chỉ mục thành các phân lớp ngữ nghĩa khác nhau dựa trên kỹ thuật từ điển dữ liệu. Nhóm tác giả đã sử dụng bộ ảnh COREL để đánh giá kết quả thực nghiệm cũng như tính hiệu quả của phương pháp đề xuất. Tuy nhiên, trong công trình này tác giả thực hiện việc đánh chỉ mục ngữ nghĩa trên cơ sở chủ quan, không có một mô hình phân loại tự động nhằm ứng dụng tổng quát cho nhiều bộ ảnh khác nhau [8].

Năm 2017, Hakan Cevikalp và cộng sự đã thực hiện bài toán tìm kiếm ảnh trên các hình ảnh có kích thước lớn. Trong bài báo này, nhóm tác giả đã xây dựng cây phân cấp nhị phân và máy vectơ hỗ trợ để phân loại các lớp đối tượng thị giác của hình ảnh. Thực nghiệm của bài báo được ứng dụng trên bộ ảnh ImageCLEF để đánh giá độ chính xác của phương pháp đề xuất. Tuy nhiên, cấu trúc cây nhị phân trong bài báo này chỉ dùng để phân lớp các hình ảnh, vẫn chưa tạo một cấu trúc lưu trữ cho hình ảnh. Do đó, tốc độ truy xuất và tốc độ tìm kiếm của hình ảnh còn hạn chế. Mặt khác, trong công trình này chưa phân tích cũng như trích xuất ngữ nghĩa thị giác của ảnh truy vấn [2].

Năm 2017, Zahid Medmood và cộng sự thực hiện bài toán tìm kiếm ảnh dựa trên nội dung và phân tích ngữ nghĩa. Từ đó, nhóm tác giả ứng dụng kỹ thuật vectơ từ thị giác BoVW nhằm mô tả ngữ nghĩa thị giác của hình ảnh. Trong công trình này, nhóm tác giả ứng dụng kỹ thuật từ điển dữ liệu để ánh xạ giữa ngữ nghĩa thị giác bậc cao của hình ảnh và đặc trưng cấp thấp. Nhóm tác giả đã thực nghiệm trên bộ ảnh COREL nhằm minh chứng tính hiệu quả về độ chính xác của phương pháp. Tuy nhiên, trong công trình này tác giả chưa xây dựng một mô hình tìm kiếm cũng như chưa tạo câu truy vấn SPARQL nhằm thực hiện bài toán tìm kiếm và tra cứu ngữ nghĩa hình ảnh [6].

Năm 2017, Ritika Hirwane giới thiệu bài báo về truy vấn ảnh theo tiếp cận ngữ nghĩa. Tác giả đã giới thiệu các kỹ thuật về phân hồi liên quan, phân lớp và đánh giá độ đo ngữ nghĩa nhằm xây dựng mô hình truy vấn ngữ nghĩa cho hình ảnh. Trong công trình này, tác giả chỉ áp dụng các kỹ thuật khai phá dữ liệu, không sử dụng các mô hình tìm kiếm để nâng cao hiệu quả cho bài toán tìm kiếm ảnh theo tiếp cận ngữ nghĩa [7].

Năm 2018, Ouiem Bchir và cộng sự đã thực hiện truy vấn ảnh dựa trên việc trích xuất vectơ đặc trưng của đối tượng vùng để thực hiện quá trình phân hoạch nhằm tăng tốc độ tìm kiếm ảnh. Trong phương pháp này, nhóm tác giả xây dựng một ánh xạ ngữ nghĩa giữa đặc trưng thị giác và ngữ nghĩa cấp cao. Nhóm tác giả đã thực nghiệm trên bộ ảnh ImageCLEF và cho thấy tính hiệu quả của phương pháp đề xuất. Tuy nhiên, trong công trình này chưa tạo ra một mô hình tìm kiếm cũng như chưa xây dựng quá trình trích xuất và truy vấn ngữ nghĩa trên một ontology cho trước [1].

Năm 2019, Bjorn Barz và cộng sự ứng dụng mạng nơron học sâu để phân lớp hình ảnh và đánh giá độ tương tự giữa hai ảnh trên cơ sở độ đo ngữ nghĩa. Trong công trình này, nhóm tác giả cũng đã thực hiện trên bộ ảnh ImageCLEF nhằm đánh giá tính hiệu quả độ chính xác của phương pháp đề xuất. Tuy nhiên, trong công trình này không xây dựng mô hình tìm kiếm nhằm tăng tốc độ cho bài toán truy vấn ảnh tương tự theo ngữ nghĩa [5].

Từ các công trình đã công bố cho thấy bài toán tra cứu ảnh có nhiều quan tâm của các nhóm tác giả. Hơn nữa, việc áp dụng cây phân cụm phân cấp để thực hiện tìm kiếm ảnh tương tự theo ngữ nghĩa là một hướng tiếp cận có tính khả thi và nhiều thách thức.

Trên cơ sở kế thừa từ các công trình đã có và khắc phục những hạn chế của các phương pháp liên quan đã công bố, đồng thời tạo ra một hệ truy vấn ảnh theo ngữ nghĩa nhằm nâng cao hiệu quả tìm kiếm, chúng tôi đề xuất một mô

hình truy vấn ảnh bằng cách tạo cây phân cụm phân cấp các vectơ đặc trưng thị giác cấp thấp với các thao tác thêm, xóa và quan hệ không gian của hình ảnh nhằm nâng cao hiệu suất cho quá trình tìm kiếm ảnh tương tự. Ngoài ra, chúng tôi cũng đề xuất một phương pháp tra cứu ảnh theo hướng tiếp cận ngữ nghĩa. Phương pháp này dựa trên cơ sở kỹ thuật khai phá dữ liệu dưới dạng phân cụm phân cấp và phân cụm phân hoạch để từ đó tạo ra một mô hình tìm kiếm dưới dạng cấu trúc cây H-Tree.

III. CÂY PHÂN CỤM H-TREE

A. Cấu trúc cây phân cụm H-Tree

H-Tree là một cây đa nhánh gồm một nút gốc, một tập các nút trong và một tập các nút lá. Mỗi nút trong cây H-Tree chứa một tập vectơ đặc trưng của ảnh. Việc tạo cây H-Tree dựa trên thao tác thêm, xóa, sửa các nút. Quá trình tìm kiếm ảnh được thực hiện từ nút gốc và chọn một nhánh trong cây, sau đó tổng hợp các vectơ đặc trưng từ nút gốc đến nút lá. Trên cơ sở kết xuất định danh URI của hình ảnh, cùng với meta-data của hình ảnh đó, tập ảnh tương tự và ngữ nghĩa của hình ảnh được trích xuất. Cây H-Tree lưu trữ các vectơ đặc trưng v của một vùng trên một ảnh và thuộc về một phần tử E của một nút trên cây. Phần tử E được định nghĩa như sau:

Định nghĩa 1. Gọi v, l, id lần lượt là vectơ đặc trưng, phân lớp ngữ nghĩa của một vùng trên ảnh và định danh của hình ảnh, ta có: $E = \langle v, l, id \rangle$

Với $v = (v_1, \dots, v_n)$ là vectơ đặc trưng của một vùng trên ảnh; $l \in \{l_1, l_2, \dots, l_m\}$; m, n lần lượt là số phân lớp nhãn ngữ nghĩa và số đặc trưng của một vùng trên ảnh, id là định danh của hình ảnh tương ứng.

Cây H-Tree được tổ chức theo cấu trúc phân cụm phân cấp dựa trên khoảng cách Euclide để gom cụm tập các vectơ đặc trưng vùng của hình ảnh. Do đó, cây H-Tree lưu trữ tập các phần tử mô tả vùng ảnh $T = \{E_i = \langle v_i, l_i, id \rangle \mid i = 1, \dots, N\}$, trong đó N là số lượng các vùng trong tập ảnh.

Cây H-Tree tạo ra một mô hình phân cụm tập các vectơ đặc trưng nhằm phục vụ cho bài toán tìm kiếm ảnh tương tự. Kết quả của quá trình tạo cây là một tập các nút sao cho mỗi nút chứa tập phần tử μ mô tả vùng ảnh của tập ảnh ban đầu $\mathfrak{S} = \{\mu_k \mid k = 1..K\}$ với $\mu_k = \langle E_k, cent_k, links_k \rangle$, trong đó $K, cent_k, links_k$ lần lượt là số lượng nút, tâm và liên kết tại nút thứ k trong cây H-Tree. Các nút trong cây H-Tree được định nghĩa như sau:

Định nghĩa 2. Gọi H-Tree là một cây phân cụm phân cấp, ta có:

- Nút gốc gồm một tập các liên kết đến các nhánh kế cận: $root = \{links_i \mid i = 1..t\}$
- Nút trong: $Node = \{\mu_k = \langle E_k, cent_k, links_k \rangle \mid k = 1..K\}$, $links_k \neq null$;
- Nút lá: $Node = \{\mu_k = \langle E_k, cent_k, links_k \rangle \mid k = 1..K\}$, $links_k = null$;
- Hai nút được gọi là đồng cấp nếu có cùng 1 nút cha.
- Nút p_Node gọi là cha của nút c_Node nếu nút p_Node có một thành phần liên kết đến c_Node .

Tại thời điểm ban đầu, cây H-Tree là rỗng, tức là chỉ gồm một nút gốc chứa các liên kết là rỗng. Sau đó, từng phần tử E_i được thêm vào cây để tạo ra các nhánh tương ứng với các nút trong cây. Trên cơ sở độ đo Euclide, phần tử E_i được phân bố vào các nút trong cây. Gọi $d(v_i, cent_k)$ là khoảng cách Euclide giữa vectơ đặc trưng vùng v_i với tâm $cent_k$, khoảng cách này được chuẩn hóa trong miền giá trị $[0,1]$ bởi vì từng thành phần của vectơ đặc trưng được chuẩn hóa trong miền giá trị $[0,1]$. Gọi hai ngưỡng khoảng cách giữa hai vectơ đặc trưng lần lượt là ϵ, σ , với $0 < \epsilon < \sigma < 1$. Quy tắc phân bố được định nghĩa như sau:

Định nghĩa 3. Quy tắc phân bố phần tử trong cây H-Tree, bắt đầu thực hiện từ nút gốc và lần lượt thực hiện theo các quy tắc sau:

- Chọn hướng đi từ nút hiện hành đến các nút của nhánh kế cận và chọn nhánh có khoảng cách $d(v_i, v_{c_j})$ ngắn nhất.
- Nếu $d(v_i, v_{c_j}) < \epsilon$ thì v_i được thêm vào nút hiện tại, tức là nút có tâm là v_{c_j} .
- Nếu $\epsilon \leq d(v_i, v_{c_j}) \leq \sigma$ thì v_i được thêm vào nút con của nút có tâm là v_{c_j} . Quá trình tìm kiếm nút con phù hợp được thực hiện lại từ quy tắc (a).
- Nếu $d(v_i, v_{c_j}) > \sigma$ thì khởi tạo một nút mới đồng cấp với nút có tâm v_{c_j} .

Vì dữ liệu ảnh được gia tăng nhanh chóng, do đó cây H-Tree phải có khả năng tăng trưởng để phù hợp cho việc lưu trữ dữ liệu ảnh. Định lý sau đây minh chứng tính tăng trưởng của cây H-Tree.

Định lý 1. Cây H-Tree là cây tăng trưởng theo hướng từ gốc tới lá.

Chứng minh: theo định nghĩa 3, mỗi khi thêm một phần tử (theo định nghĩa 1) thì phần tử này được thêm vào nút hiện tại, hoặc nút con, hoặc tại nút mới đồng cấp với một nút phù hợp. Do đó, cây H-Tree là cây tăng trưởng ■

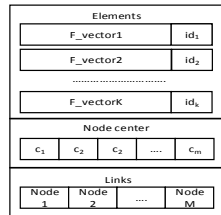
B. Cài đặt cấu trúc cây

Trên cơ sở **định nghĩa 1** một nút trên cây H-Tree được mô tả như trong **Hình 1** trong đó:

Elements là tập các thành phần E_i trong một Node.

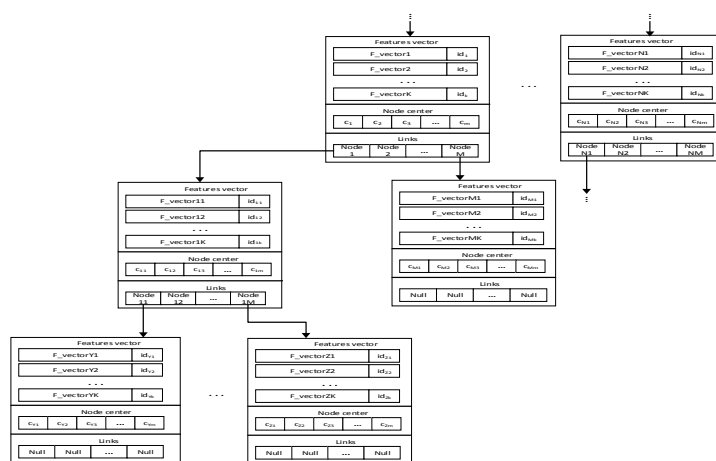
Node center là tâm của Node.

Links là tập các liên kết đến các nút con của nút hiện hành, nếu là nút lá thì các liên kết sẽ trở đến *null*.



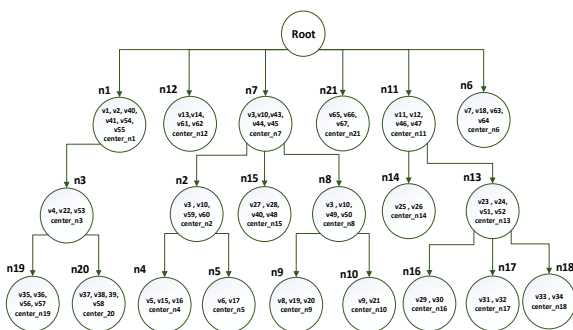
Hình 1. Mô tả cấu trúc của một nút trên cây H-Tree

Theo **định nghĩa 2**, cấu trúc của cây H-Tree sẽ được mô tả như trong **Hình 2**.

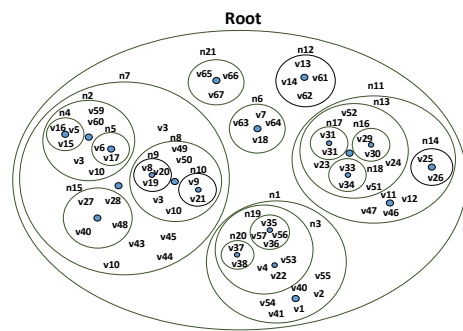


Hình 2. Mô tả cấu trúc của cây H-Tree

C. Quá trình xây dựng cây



Hình 3. Một mô hình cây phân cụm phân cấp H-Tree



Hình 4. Một mô hình dạng phẳng của cây H-Tree

Tại **Hình 3** và **Hình 4** mô tả về cấu trúc cây phân cụm phân cấp H-Tree dưới dạng phân cấp và dạng phẳng. Phương pháp tạo cây phân cụm phân cấp H-Tree được thực hiện theo các bước như sau:

- Bước 1:** khởi tạo nút gốc gồm các liên kết là rỗng nhằm liên kết các nhánh trong cây H-Tree.
- Bước 2:** lần lượt chọn từng vectơ đặc trưng trong bộ dữ liệu ảnh để đưa vào cây H-Tree theo định nghĩa 1 và 2.
- Bước 3:** với mỗi vectơ đặc trưng v thêm vào cây H-Tree, ta thực hiện theo quy tắc định nghĩa 3.

D. Các thuật toán tạo cây H-Tree

1. Thêm vectơ đặc trưng trên cây

Với mỗi véc-tơ đặc trưng v , ta tạo một phần tử E để thêm vào cây theo định nghĩa 1 và 2. Việc thêm phần tử E vào một nút phù hợp dựa trên một độ đo cho trước theo định nghĩa 3.

Mỗi phần tử được lần lượt thêm vào cây H-Tree, do đó cần phải tồn tại một nút để chứa phần tử này. Định lý sau đây chứng minh tính tồn tại và duy nhất của một nút trên cây H-Tree lưu trữ các phần tử.

Định lý 2. Cho 1 véctor đặc trưng v thì véctor này thuộc vào 1 nút trên cây.

Chứng minh: gọi v là véctor cần thêm vào một nút trên cây H-Tree. Theo định nghĩa 3, thực hiện quy tắc tạo cây thì phân tử v này thuộc về một nút hiện hành hoặc một nút con hoặc một nút mới đồng cấp với nút phù hợp. Do đó, ta luôn tìm được một nút để lưu trữ véctor v ■

Định lý 3. Một véctor đặc trưng v chỉ được lưu trữ trong một nút duy nhất trên cây H-Tree.

Chứng minh: giả sử có 2 véctor v_1, v_2 cùng thuộc một nút. Khi đó, v_1, v_2 là hai phân tử thuộc cụm của một nút. Suy ra: $d(v_1, c) = d(v_2, c)$. Vì một nút là một phân cụm phẳng, do đó $v_1 \equiv v_2$. Nghĩa là, một véctor đặc trưng v chỉ được lưu trữ trong một nút duy nhất trên cây H-Tree ■

Khi thực hiện phân bố từng phân tử thì phân tử này phải thuộc về một cụm phù hợp nhất. Tức là, các phân tử trong cùng một cụm sẽ có độ tương tự nhiều nhất theo độ đo đã được chọn trước. Định lý sau đây chứng minh tính phân bố phù hợp một phân tử trên cây H-Tree.

Định lý 4. Một véctor đặc trưng v được phân bố vào cụm phù hợp nhất theo độ đo Euclide.

Chứng minh:

Trường hợp 1: véctor đặc trưng thuộc vào nút con của nút hiện hành, nghĩa là $d(v, v_{cj}) < \varepsilon$ hay nói cách khác ta tìm được một cụm tại một nút con có độ tương tự gần nhất.

Trường hợp 2: véctor đặc trưng thuộc về một nút con, nghĩa là $\varepsilon \leq d(v, v_{cj}) \leq \sigma$ hay nói cách khác ta đã tìm được nút con sao cho $d(v, v_c) < \varepsilon$, v_c là tâm của nút con phù hợp của nút hiện hành, nghĩa là ta đã tìm được một nút con phù hợp nhất theo quy tắc tại định nghĩa 2 và định nghĩa 3.

Trường hợp 3: nếu tất cả các khoảng cách giữa tâm và véctor v của các nút đều vượt ngưỡng cho trước, tức là các nút hiện tại trên cây đều không phù hợp hay nói cách khác không tồn tại nút nào trên cây có đặc tính giống với véctor v , thì ta tạo nút mới và áp dụng quy tắc tại định nghĩa 3, nút mới này sẽ chứa các phân tử khác gần nhau và gần véctor v .

Từ 3 trường hợp trên ta có một véctor đặc trưng v được phân bố vào cụm phù hợp nhất theo độ đo Euclide ■

Đối với cấu trúc cây H-Tree, khi phân loại một véctor đặc trưng, nếu véctor này thuộc về một nhánh, thì nhánh đó tiếp tục phân hoạch mịn hơn tại các nhánh kế cận. Nếu không thuộc về một nhánh hiện hành thì tạo ra một nhánh mới, điều này làm cho phân hoạch tại các nút đồng cấp trở nên mịn hơn, nghĩa là tính đúng đắn của phân hoạch cao hơn.

Trong cây H-Tree, một nút có thể vừa là lá vừa là nút trong, nút lá có thể trở thành nút trong nếu thêm nhánh mới. Cây H-Tree tạo ra một phân hoạch đa tầng, do đó khi tìm một cụm có bán kính ε thì độ chính xác đạt cao nhất. Nếu trong phạm vi bán kính σ , cây sẽ phân hoạch véctor đặc trưng về một nhánh, và do đó các phân hoạch sẽ có xu hướng đều đặn vì các phân tử quá khác biệt nhau sẽ không thuộc một nhánh của cây, và như vậy cây H-Tree có xu hướng tự nhiên trở thành đa nhánh cân bằng.

Thuật toán thêm một nút trên cây được thực hiện như sau:

Thuật toán 1: INE

Đầu vào: phân tử E , nút gốc $root$, giá trị ngưỡng ε , σ

Đầu ra: cây H-Tree sau khi thêm phân tử E

Function $INE(E, root, \varepsilon, \sigma)$

Begin

$NODE = root;$

If $NODE = null$ **then**

Initialize $root = \{links_k | links_k = null; k = 1..n_k\};$

Create new $node = \langle E, cent, links \rangle, links = null;$

$root.links_0 = node;$

Else

$i = \operatorname{argmin}\{euclide(NODE.links.cent_k, E.v), k = 1..|NODE.links|\};$

$d = Euclide(NODE.links_i.cent, E.v);$

If $d \leq \varepsilon$ **then**

$NODE.links_i.e = NODE.links_i.e \cup E;$

ElseIf $d > \varepsilon$ and $d \leq \sigma$ **then**

$INE(E, NODE.links_i, \varepsilon, \sigma);$

ElseIf $d > \sigma$ **then**

Create new $node = \langle E, cent, links \rangle, links = null;$

$NODE.links_{|NODE.links|+1} = node;$

```

EndIf
Return H-Tree;

```

```

EndIf

```

```

End

```

Mệnh đề 1. Thuật toán *INE* có độ phức tạp là $O(NxK)$, với N, K lần lượt là số nút và số phần tử tối đa của một nút trong cây *H-Tree*.

Chứng minh: Vì thuật toán *INE* thực hiện bằng cách đệ quy từ nút gốc đến nút lá, mỗi lần thực hiện sẽ tạo ra một nút hoặc bổ sung một phần tử vào nút trong cây, trong trường hợp xấu nhất là duyệt hết các nút của cây. Mỗi lần duyệt tại một nút, thuật toán *INE* duyệt qua K phần tử để tìm nhánh kế cận. Do đó độ phức tạp của thuật toán *INE* là $O(NxK)$ ■

2. Xóa một phần tử trên cây *H-Tree*

Trong trường hợp một phần tử bị phân loại vào một nút nhưng bị nhầm lẫn, thì ta thực hiện thao tác xóa phần tử này trong cây *H-Tree*. Nếu cụm này chỉ có một phần tử, tức là theo định nghĩa 3 về nguyên tắc tạo cây, phần tử này vẫn thỏa mãn điều kiện của cây *H-Tree*, ta có thể không xóa phần tử này. Hoặc nếu trong trường hợp xóa thì ta phân bố lại tất cả các phần tử của các nút con vào trong cây *H-Tree*. Việc phân bố lại các phần tử được thực hiện giống thuật toán thêm như trên. Do đó, chúng tôi không trình bày vấn đề này. Vì vậy, thuật toán xóa một phần tử được thực hiện như sau:

Thuật toán 2: DNE

Đầu vào: phần tử E , nút gốc $root$, giá trị ngưỡng ε, σ

Đầu ra: cây *H-Tree* sau khi xóa phần tử E

Function $DNE(E, root, \varepsilon, \sigma)$

```

Begin

```

```

     $NODE = root;$ 

```

```

    If  $NODE = null$  then

```

```

        Return  $null;$ 

```

```

    Else

```

```

         $i = \operatorname{argmin}\{euclide(NODE.links.cent_k, E.v), k = 1..|NODE.links|\};$ 

```

```

         $d = Euclide(NODE.links_i.cent, E.v);$ 

```

```

        If  $d \leq \varepsilon$  and  $|NODE| > 1$  then

```

```

             $NODE.links_i = NODE.links_i \setminus E;$ 

```

```

        Elseif  $d > \varepsilon$  and  $d \leq \sigma$  then

```

```

             $DNE(E, NODE.links_i, \varepsilon, \sigma);$ 

```

```

        EndIf

```

```

    EndIf

```

```

    Return H-Tree;

```

```

End.

```

Mệnh đề 2. Thuật toán *DNE* có độ phức tạp là $O(hxK)$, với h, K lần lượt là chiều cao và số phần tử tối đa của một nút trong cây *H-Tree*.

Chứng minh: Thuật toán *DNE* thực hiện gọi đệ quy theo một nhánh của cây, trong trường hợp xấu nhất là duyệt nhánh có chiều cao của cây *H-Tree*, mỗi lần duyệt một nút sẽ kiểm tra K phần tử của nút đó. Vì vậy, thuật toán *DNE* có độ phức tạp là $O(hxK)$ ■

3. Cập nhật một phần tử trên cây *H-Tree*

Trong trường hợp một phần tử có thông tin bị sai lệch, ta có thể thực hiện cập nhật thông tin cho phần tử đó. Quá trình cập nhật này được thực hiện bằng cách tìm kiếm phần tử cần cập nhật và thay đổi thông tin cho phần tử đó. Vì vậy, thuật toán cập nhật một phần tử được thực hiện như sau:

Thuật toán 3: UNE

Đầu vào: phần tử E_{old} , phần tử E_{new} , nút gốc $root$, giá trị ngưỡng ε, σ

Đầu ra: cây *H-Tree* sau khi cập nhật phần tử E_{old} thành E_{new}

Function $UNE(E_{old}, E_{new}, root, \varepsilon, \sigma)$

```

Begin

```

```

     $NODE = root;$ 

```

```

    If  $NODE = null$  then

```

```

        Return  $null;$ 

```

```

    Else

```

```

         $i = \operatorname{argmin}\{euclide(NODE.links.cent_k, E.v), k = 1..|NODE.links|\};$ 

```

```

         $d = Euclide(NODE.links_i.cent, E.v);$ 

```

```

        If  $d \leq \varepsilon$  then

```

```

             $NODE.links_i.E_{old} = NODE.links_i.E_{new};$ 

```

```

ElseIf  $d > \varepsilon$  and  $d \leq \sigma$  then
     $UNE(E_{old}, E_{new}, NODE.links_i, \varepsilon, \sigma)$ ;

```

```

EndIf

```

```

EndIf

```

```

Return H-Tree;

```

End.

Mệnh đề 3. Thuật toán UNE có độ phức tạp là $O(hxK)$, với h, K lần lượt là chiều cao và số phần tử tối đa của một nút trong cây H -Tree.

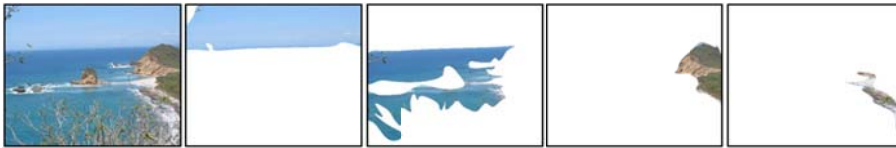
Chứng minh: Tương tự thuật toán DNE , thuật toán UNE thực hiện gọi đệ quy theo một nhánh của cây, mỗi lần duyệt một nút sẽ kiểm tra K phần tử của nút đó. Vì vậy, thuật toán UNE có độ phức tạp là $O(hxK)$ ■

IV. MÔ HÌNH TRUY VẤN ẢNH

A. Dữ liệu ảnh

Mỗi ảnh sẽ được chia thành nhiều vùng nhau theo phương pháp của Hugo Jair Escalante, mỗi vùng được trích xuất một vectơ đặc trưng bao gồm đặc trưng vùng: diện tích, chiều rộng và chiều cao; đặc trưng về vị trí: giá trị trung bình và độ lệch chuẩn theo trục x và trục y ; đặc trưng về hình dạng; đặc trưng màu sắc trong không gian RGB và CIE-Lab.

Ví dụ: Cho một ảnh được chia thành các vùng như sau:



Hình 6. Ảnh gốc và các ảnh phân vùng (5001.jpg)

Hình 6 mô tả một ảnh gốc và 4 ảnh của các vùng thuộc về các lớp ảnh: *sky-light*, *ocean-animal*, *horn*, *sidewalk* của ảnh 5001.jpg.

B. Thuật toán trích xuất vectơ từ và tập ảnh tương tự

Từ cây phân cụm phân cấp H -Tree đã tạo, chúng tôi đề xuất thuật toán trích xuất vectơ từ và tập ảnh tương tự làm cơ sở cho việc trích xuất ngữ nghĩa hình ảnh. Với mỗi ảnh truy vấn, thuật toán thực hiện tìm kiếm tập ảnh tương tự và trích xuất vectơ từ dựa trên tập ảnh tương tự này. Quá trình trích xuất vectơ từ và tập ảnh tương tự cần tìm ra cụm trong cây H -Tree có độ tương tự gần với ảnh truy vấn nhất.

Thuật toán 4. IRHT

Đầu vào: vectơ đặc trưng v của ảnh truy vấn I_q , nút gốc $root$ của cây H -Tree, giá trị ngưỡng ε , giá trị ngưỡng σ .

Đầu ra: tập ảnh tương tự SI

Function $IRHT(v, root, \varepsilon, \sigma)$

Begin

```

     $NODE = root$ ;

```

```

If  $NODE = null$  then

```

```

    Return  $null$ ;

```

```

Else

```

```

     $i = \operatorname{argmin}\{euclide(NODE.links.cent_k, E.v), k = 1..|NODE.links|\}$ ;

```

```

     $d = Euclide(NODE.links_i.cent, E.v)$ ;

```

```

If  $d \leq \varepsilon$  then

```

```

     $SI = NODE.links_i.E$ ;

```

```

ElseIf  $d > \varepsilon$  and  $d \leq \sigma$  then

```

```

     $IRHT(v, NODE.links_i, \varepsilon, \sigma)$ ;

```

```

EndIf

```

```

EndIf

```

```

Return  $SI$ ;

```

End.

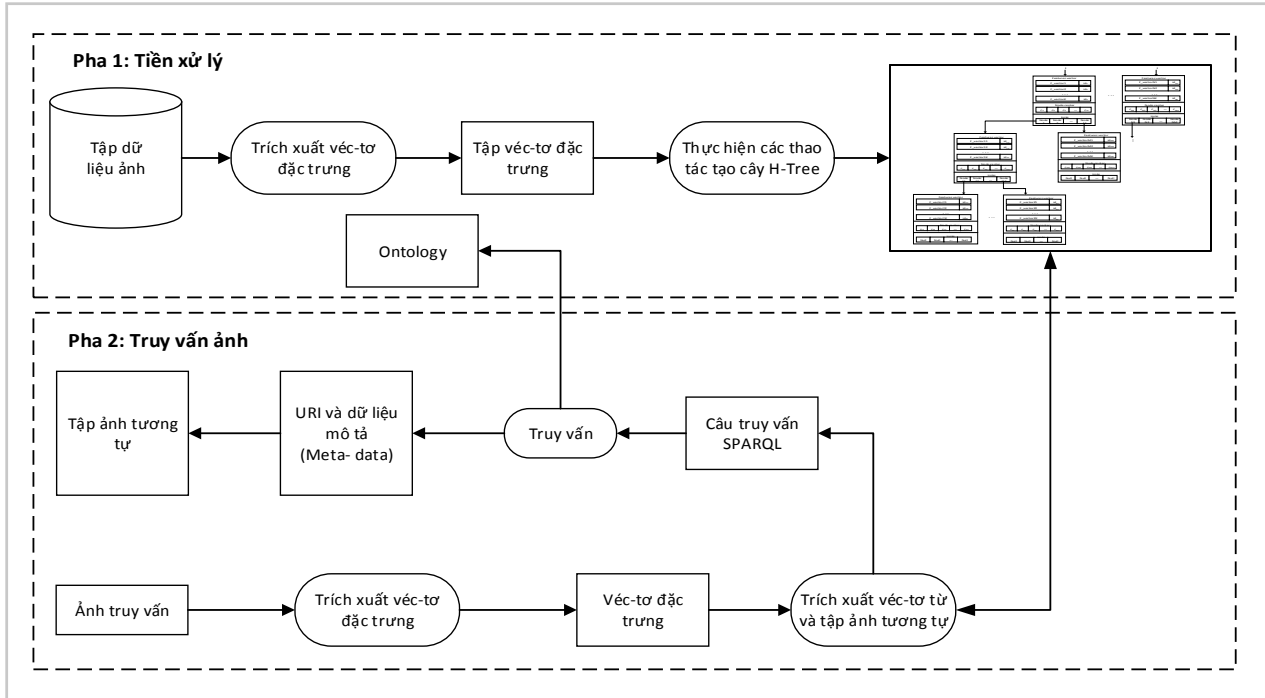
Mệnh đề 4. Thuật toán $IRHT$ có độ phức tạp là $O(hxK)$, với h, K lần lượt là chiều cao và số phần tử tối đa của một nút trong cây H -Tree.

Chứng minh: Tương tự thuật toán UNE , thuật toán $IRHT$ đi theo một nhánh của cây và tìm kiếm nút phù hợp và cho ra kết quả là một tập các phần tử trong cụm. Vì vậy, thuật toán $IRHT$ có độ phức tạp là $O(hxK)$ ■

V. THỰC NGHIỆM

A. Mô hình thực nghiệm

Mô hình kiến trúc hệ thống truy vấn ảnh theo ngữ nghĩa dựa trên cây phân cụm phân cấp H-Tree được trình bày như **Hình 7**. Hệ thống được thực hiện bằng cách trích xuất các vector đặc trưng của tập dữ liệu ảnh và xây dựng cây phân cụm phân cấp từ tập các vectơ đặc trưng. Từ đó dựa trên cây phân cụm phân cấp trích xuất vectơ từ và tập ảnh tương tự. Thực hiện truy vấn hình ảnh dựa trên các vectơ từ, truy vấn ngữ nghĩa trên Ontology bằng ngôn ngữ SPARQL để truy xuất tập các hình ảnh tương tự cùng ngữ nghĩa.



Hình 7. Mô hình tìm kiếm ảnh theo ngữ nghĩa dựa trên Ontology và cây phân cụm phân cấp

Pha tiền xử lý

Kết quả của pha tiền xử lý là xây dựng được cây phân cụm phân cấp H-Tree dựa trên vectơ đặc trưng của tập dữ liệu ảnh. Quá trình thực hiện pha tiền xử lý gồm 2 bước như sau:

- ❖ **Bước 1:** trích xuất các vectơ đặc trưng từ tập dữ liệu ảnh.
- ❖ **Bước 2:** dựa trên độ đo tương tự đề xuất và tạo cấu trúc cây phân cụm phân cấp với mỗi nút của cây là tập các vectơ đặc trưng thị giác của một hay nhiều ảnh.

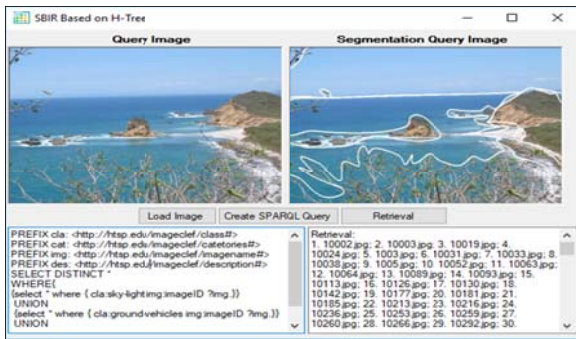
Pha tìm kiếm ảnh tương tự

Việc tìm kiếm ảnh tương tự được thực hiện với đầu vào là vectơ đặc trưng của ảnh truy vấn và tạo ra vectơ từ và tập ảnh tương tự dựa trên cây phân cụm phân cấp. Sau đó hệ thống truy vấn dựa trên Ontology để tạo ra tập URI và meta-data của tập ảnh tương tự. Quá trình thực hiện phase tìm kiếm ảnh tương tự theo các bước sau:

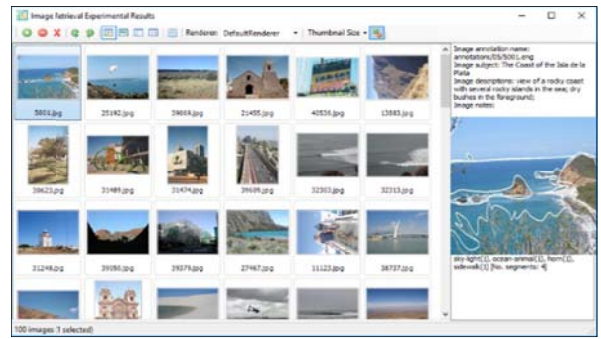
- ❖ **Bước 1:** trích xuất vectơ đặc trưng từ ảnh truy vấn và trích xuất vectơ từ thị giác theo tập ảnh tương tự dựa trên cây phân cụm phân cấp.
- ❖ **Bước 2:** tạo câu truy vấn SPARQL để tìm ra tập URI và meta-data của các hình ảnh.
- ❖ **Bước 3:** kết xuất các hình ảnh từ tập URI đã có và sắp xếp theo độ đo tương tự với ảnh truy vấn.

B. Kết quả thực nghiệm

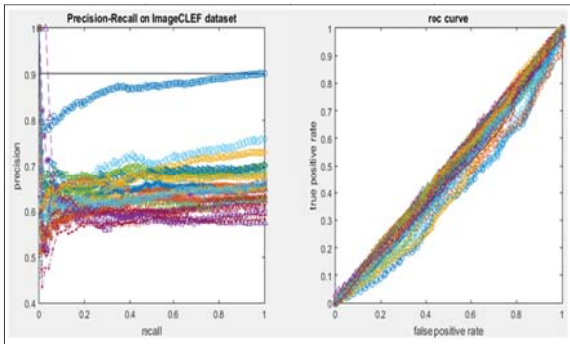
Thực nghiệm trên máy PC CPU Intel Core i5-7200U CPU @ 2.50GHz, 8.0GB RAM, hệ điều hành Windows 10 Pro 64 bit. Kết quả thực nghiệm được đánh giá trên bộ dữ liệu imageCLEF chứa 20,000 ảnh, được chia thành 276 lớp và lưu trữ trong 41 thư mục (từ thư mục 0 đến thư mục 40); bộ dữ liệu của kích thước 1.64 GB. Để đánh giá hiệu quả của phương pháp tìm kiếm ảnh, phần thực nghiệm được đánh giá các giá trị gồm: độ chính xác (precision), độ phủ (recall) và độ đo dung hòa F-measure.



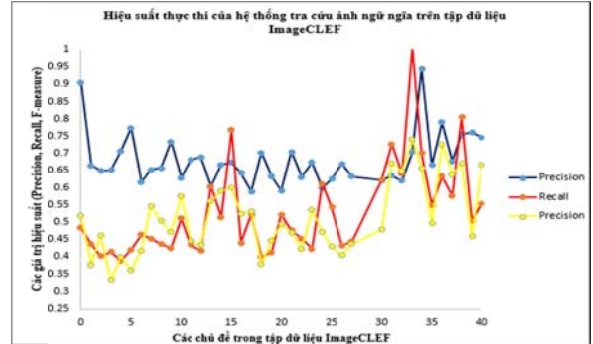
Hình 8. Giao diện của hệ truy vấn ảnh cây H-Tree



Hình 9. Một kết quả của truy vấn ảnh trên cây H-Tree



Hình 10. Biểu đồ Precision-Recall và ROC của hệ truy vấn trên cây H-Tree



Hình 11. Giá trị trung bình của Precision, Recall, F-measure

Bảng 1. Hiệu suất truy vấn ảnh của phương pháp đề xuất trên tập ảnh ImageCLEF

Tập ảnh	Số ảnh	Độ chính xác trung bình	Độ phủ trung bình	Độ đo dung hòa trung bình
00-10	2200	0.674152	0.438917	0.531677
11-20	1500	0.638752	0.454628	0.531187
21-30	1200	0.628562	0.506845	0.561179
31-40	2100	0.759824	0.482568	0.551288

Bảng 2. So sánh độ chính xác giữa các phương pháp trên bộ dữ liệu ImageCLEF

Phương pháp	Mean Average Precision (MAP)
C.A. Hernández-Gracidas, 2013 [10]	0.5826
Hakan Cevikalp, 2017 [2]	0.4678
Vijayarajan, 2016 [11]	0.4618
Y. Cao, 2016 [13]	0.7236
S. Fakhfakh, 2015 [12]	0.5400
Phương pháp đề xuất của chúng tôi	0.6753

Kết quả thực nghiệm của hệ truy vấn dựa trên cây H-Tree được mô tả tại Hình 10, Hình 11; hiệu suất của hệ truy vấn được trình bày trong Bảng 1 có tổng số ảnh truy vấn là 7000 và các giá trị hiệu suất trung bình lần lượt là 0.674152; 0.638752; 0.628562; 0.759824. Hình 10 mô tả đường cong Precision-Recall và ROC, mỗi đường cong mô tả một bộ dữ liệu được truy vấn; diện tích dưới các đường cong này cho thấy độ chính xác của việc truy vấn ảnh. Hình 11 mô tả giá trị trung bình của precision, recall, và F-measure của 41 bộ ảnh trên tập ảnh ImageCLEF; Giá trị MAP của phương pháp đề xuất được so sánh với các phương pháp khác trên cùng một bộ dữ liệu imageCLEF được mô tả trong Bảng 2 và cho thấy kết quả truy vấn của phương pháp đề xuất tương đối chính xác so với các hệ truy vấn ảnh theo ngữ nghĩa.

VI. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Bài báo đã xây dựng được phương pháp tìm kiếm ảnh theo ngữ nghĩa dựa trên cây phân cụm phân cấp các đặc trưng thị giác của hình ảnh. Cây phân cụm H-Tree là một mô hình phân hoạch các vectơ đặc trưng để tìm ra tập ảnh tương tự. Đây là mô hình cải tiến của phương pháp khai phá dữ liệu phân cụm phân cấp kết hợp với kỹ thuật học bán giám sát. Trên cơ sở lý thuyết đã được đề xuất, chúng tôi xây dựng hệ truy vấn ảnh theo ngữ nghĩa dựa trên nội dung của hình ảnh. Kết quả thực nghiệm trên bộ dữ liệu ảnh ImageCLEF được đánh giá và so sánh với các công trình khác trên cùng một tập dữ liệu ảnh đã cho thấy phương pháp đề xuất là hiệu quả. Vì vậy, phương pháp đề xuất của chúng tôi có thể được sử dụng cho các hệ thống tra cứu dữ liệu đa phương tiện. Việc xây dựng cây phân cụm dựa trên các đặc trưng thị giác của ảnh đã tạo ra một cấu trúc tìm kiếm ảnh nhanh chóng. Hướng phát triển tiếp theo, chúng tôi xây dựng

mỗi nhánh của cây H-Tree có thể thêm giá trị trọng số để phân hoạch dữ liệu, khi đó bài toán trở thành tìm đường đi trên cây tăng trưởng, tức là không gian ẩn. Ngoài ra, giữa các nút lại có thể liên thông để tạo ra một cấu trúc hệ lai dạng đồ thị và đặc biệt tại mỗi nút có thể ứng dụng một mô hình mạng để phân hoạch.

VII. LỜI CẢM ƠN

Nhóm tác giả chân thành cảm ơn Trường Đại học Sư phạm TP. Hồ Chí Minh, Trường Đại học Công nghiệp Thực phẩm TP. Hồ Chí Minh, Trường Đại học Bà Rịa - Vũng Tàu là những nơi bảo trợ cho nghiên cứu này. Chúng tôi trân trọng cảm ơn nhóm nghiên cứu SBIR-HCM và nhiệm vụ NVCC14.01/19-19 của Viện Hàn lâm Khoa học và Công nghệ Việt Nam đã hỗ trợ chúng tôi hoàn thành bài nghiên cứu này.

TÀI LIỆU THAM KHẢO

- [1] Ouiem Bchir, Mohamed Maher, Hadeel Aljam(2018) “*Region-Based Image Retrieval Using Relevance Feature Weights*” International journal of fuzzy logic and intelligent systems international journal of fuzzy logic and intelligent systems vol.18 no.1, 2018.03, 65-77 (13 pages).
- [2] H. Cevikalp, M. Elmas, S. Ozkan (2017), “*Large-scale image retrieval using transductive support vector machines*”, Computer Vision and Image Understanding, vol. no. pp.1-11, 2017.
- [3] Alzu’bi A, Amira A, Ramzan N (2015), “*Semantic content-based image retrieval: A comprehensive study*”. J Vis Commun Image Represent 32:20–54.
- [4] Van T. T. , Le M. T. (2017), “*Mot so cai tien cho he truy van anh dua tren cay S-Tree*”, proceeding of Publishing House for Science and Technology.
- [5] Björn Barz, Joachim Denzler (2019), “*Hierarchy-based Image Embeddings for Semantic Image Retrieval*”, IEEE Winter Conference on Applications of Computer Vision (WACV), number: 18493437, ISSN: 1550-5790.
- [6] Mehmood, Z., Mahmood, T., & Javid, M. A. (2017), “*Content-based image retrieval and semantic automatic image annotation based on the weighted average of triangular histograms using support vector machine*”, Applied Intelligence, 48(1), 166-181.
- [7] Ritika Hirwane (2017), “*Semantic based Image Retrieval*”, International Journal of Advanced Research in Computer and Communication Engineering, issue 4, ISSN (Print) 2319 5940.
- [8] Amruta Laxman Mane, A.N. Bhute (2016), “*Semantic based image retrieval*”, Indian Journal of Computer Science and Engineering (IJCSSE)
- [9] Wei Wang, Yuqing Song, Aidong Zhang (2002), “*Semantics-Based Image Retrieval by Region Saliency*”, International Conference on Image and Video Retrieval, pp 29-37.
- [10] C.A. Hernández-Gracidas, Sucar, L.E. & Montes-y-Gómez (2013), “*Improving image retrieval by using spatial relations*”, Multimed Tools Application, vol. 62, no. 2, pp. 479–505.
- [11] M.D. V. Vijayarajan, P. Tejaswin, M. Lohani (2016), “*A generic framework for ontology based information retrieval and image retrieval in web data*”, Human-centric Computing and Information Sciences, vol. 6, no. 18, pp. 1-30.
- [12] S. Fakhfakh, M. Tmar, W. Mahdi (2015), “*Image Retrieval Based on Using Hamming Distance*”, Procedia Computer Science vol. 73, no. pp.320-327.
- [13] Y. Cao, M. Long, J. Wang, Q. Yang, P.S. Yu (2016), “*Deep Visual-Semantic Hashing for Cross-Modal Retrieval*”, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA, pp.1445-1454.

SEMANTIC-BASED IMAGE RETRIEVAL USING HIERARCHICAL CLUSTERING TREE

Nguyen Minh Hai, Le Thi Vinh Thanh, Van The Thanh , Tran Van Lang

ABSTRACT: In this paper, we present some results of the Semantic-based Image Retrieval (SBIR) which query and extract semantic of image based on the hierarchical tree structure of H-Tree (Hierarchical-Tree). This data structure is created rely on the hierarchical clustering method of visual featured vectors of image. Therefore, H-Tree structure is built based on the operations such as adding, deleting, and updating; at that time the method of semantics querying and extracting is also proposed. In order to demonstrate the accurateness of the proposed theoretical basis, we designed the model and developed a semantic image retrieval application on the ImageCLEF image data set. To evaluate the effectiveness of the proposed method, experimental results are compared with some of the results of recently published works on the same data set. According to the comparison results, our proposed method is highly feasible and effective.