

MỘT CẢI TIẾN CỦA CÂY KD-TREE CHO BÀI TOÁN TÌM KIẾM ẢNH

Nguyễn Thị Định^{1*}, Lê Thị Vĩnh Thanh²,

Nguyễn Thế Hữu¹, Nguyễn Văn Thịnh¹

¹Trường Đại học Công nghiệp Thực phẩm TP.HCM

²Trường Đại học Bà Rịa - Vũng Tàu

*Email: dinhnt@hufi.edu.vn

Ngày nhận bài: 10/10/2019; Ngày chấp nhận đăng: 06/12/2019

TÓM TẮT

Tìm kiếm ảnh là một bài toán được quan tâm và đã có nhiều phương pháp được công bố trong thời gian gần đây. Trong nghiên cứu này, nhóm tác giả xây dựng cây BKD-Tree, là một cải tiến của cây KD-Tree, bao gồm: (1) lưu trữ các đối tượng đa chiều tại nút lá của cây để tạo ra một mô hình phân cụm trên cơ sở phương pháp học bán giám sát; (2) tạo ra một cấu trúc cây nhị phân cân bằng nhằm tăng hiệu suất cho bài toán tìm kiếm ảnh. Dựa trên cơ sở lý thuyết đã đề nghị, nhóm tác giả đề xuất mô hình truy vấn ảnh trên cây BKD-Tree đồng thời thực nghiệm trên bộ ảnh ImageCLEF (gồm 20.000 ảnh). Kết quả thực nghiệm được so sánh với một số công trình gần đây trên cùng bộ dữ liệu để minh chứng tính hiệu quả của phương pháp đã được đề xuất. Kết quả thực nghiệm cho thấy, phương pháp của nhóm tác giả là hiệu quả và có thể áp dụng được cho các hệ thống tìm kiếm ảnh tương tự theo nội dung.

Từ khóa: KD-Tree, độ đo tương tự, phân cụm, ảnh tương tự, truy vấn ảnh.

1. TỔNG QUAN

Dữ liệu đa phương tiện tăng nhanh theo thời gian đã thúc đẩy việc nghiên cứu và triển khai các phương pháp tìm kiếm ảnh [1]. Trong những thập niên gần đây, bài toán tìm kiếm ảnh đã được thực hiện bởi khá nhiều phương pháp như tìm theo từ khóa TBIR (Text-based Image Retrieval), tìm theo nội dung CBIR (Content-based Image Retrieval) hay tìm theo ngữ nghĩa SBIR (Semantic-based Image Retrieval) và nhiều công trình nghiên cứu đã công bố [1, 2]. Việc tìm kiếm ảnh cần thực hiện trên tập dữ liệu lớn, do đó việc gom cụm dữ liệu theo các chủ đề trong vấn đề tìm kiếm là một yêu cầu quan trọng của bài toán truy vấn ảnh. Ngày nay, nhiều phương pháp gom cụm dữ liệu được thực hiện bằng nhiều thuật toán khác nhau như: gom cụm dữ liệu sử dụng thuật toán Bees và cấu trúc cây KD-Tree [3], gom cụm dùng liên kết động sử dụng cây KD-Tree [4],...

Từ thập niên 1980 đã có nhiều phương pháp truy vấn ảnh theo nội dung được giới thiệu như QBIC, Photobook, Visual-Seek, MARS, El Nino, CIRES, PicSOM, PicHunter, MIRROR, Virage, Netra, SIMPLicity [5]. Hầu hết các công trình này tập trung vào kỹ thuật trích xuất đặc trưng ảnh mà chưa quan tâm đến xây dựng mô hình dữ liệu nhằm giảm không gian lưu trữ và tăng tốc độ truy vấn. Bài báo này đề xuất một mô hình phân cụm dữ liệu dựa trên cây BKD-Tree để áp dụng cho bài toán tìm kiếm ảnh tương tự theo nội dung.

Cây KD-Tree là một cấu trúc dữ liệu được giới thiệu từ những năm 1970 để đánh chỉ mục đa chiều, là một cấu trúc dữ liệu phân vùng không gian tổ chức thành những điểm trong không gian k -chiều [6]. Cây KD-Tree thuộc dạng cây nhị phân tìm kiếm mà mỗi nút là một

véc-tơ k -chiều. Mỗi nút không phải là nút lá thì chia không gian dữ liệu thành hai phần trên mặt phẳng k -chiều. Dựa trên cây KD-Tree nguyên thủy này, nhóm tác giả xây dựng cây BKD-Tree cải tiến là cây nhị phân cân bằng để ứng dụng cho bài toán tìm kiếm ảnh và thực nghiệm trên bộ ảnh ImageCLEF. Cây BKD-Tree cải tiến được dùng để lưu trữ các véc-tơ đặc trưng thị giác của hình ảnh đã phân đoạn. Việc phân lớp dữ liệu được thực hiện trên từng nút của cây BKD-Tree để tạo ra một cây cân bằng nhằm hỗ trợ cho quá trình tìm kiếm nhanh và tăng độ chính xác.

Trong nghiên cứu này, nhóm tác giả hướng tới một số nội dung, bao gồm: (1) Cải tiến cây BKD-Tree trở thành cây nhị phân cân bằng nhằm lưu trữ véc-tơ đặc trưng thị giác cấp thấp của hình ảnh; (2) Đề xuất thuật toán tạo cây; (3) Đề xuất mô hình tìm kiếm ảnh tương tự dựa trên cây BKD-Tree; (4) Đề xuất thuật toán truy vấn ảnh tương tự theo nội dung trên cây BKD-Tree; (5) Xây dựng thực nghiệm và so sánh kết quả với một số phương pháp gần đây trên bộ ảnh ImageCLEF.

Truy vấn ảnh dựa trên cấu trúc cây là một trong những mô hình được nghiên cứu và ứng dụng rộng rãi hiện nay [2]. Năm 2002, Otair đã thực hiện một khảo sát về tính hiệu quả của việc sử dụng cây KD-Tree để nâng cao hiệu quả tìm kiếm ảnh [6]. Trong nghiên cứu này, nhóm tác giả đã đề xuất mô hình VAM KD-Tree để tăng hiệu quả tìm kiếm ảnh đặc biệt là giảm thiểu thời gian tìm kiếm trên cây. Kết quả thực nghiệm trên tập dữ liệu gồm 10.115 ảnh với lược đồ màu có 4096 chiều. Với cây VAM KD-Tree có 4096 chiều, kết quả của công trình đã cải thiện được thời gian truy vấn ảnh nhanh gấp 3 lần so với cách tìm kiếm tuyến tính [7].

Năm 2007, Redmond and Heneghan sử dụng thuật toán k -Means kết hợp với thuật toán Katsavounidis được thực nghiệm trên 36 bộ dữ liệu tổng hợp và 2 bộ dữ liệu của UCI (UC Irvine Machine Learning Repository) [8]. Kết quả thực nghiệm đã cải thiện hiệu suất phân loại dữ liệu trên cây KD-Tree đồng thời nâng cao hiệu quả và giảm độ phức tạp của thuật toán tìm kiếm. Thuật toán đề xuất cải thiện quá trình phân cụm, là cơ sở cho việc nghiên cứu mở rộng cây R^+ -Tree và cây Bkd-Tree [8].

Năm 2009, tác giả H. Al-Jabbouli đã đề xuất thuật toán Bees thực hiện gom cụm trên tập dữ liệu mở dựa vào cấu trúc cây KD-Tree [3]. Trong công trình này, nhóm tác giả đã khắc phục những nhược điểm của thuật toán K-means, đồng thời kết hợp thuật toán K-means và C-means để tìm ra phương pháp gom cụm tối ưu gọi là thuật toán Bees. Kết quả thực nghiệm trên nhiều bộ dữ liệu cho thấy thuật toán Bees đã có nhiều cải tiến so với thuật toán k -Means. Năm 2011, Velmurugan & Baboo đã đề xuất thuật toán tìm kiếm ảnh tương tự theo nội dung dựa trên cây KD-Tree. Trong nghiên cứu này, tác giả đã kết hợp đặc trưng SURF (Speed up robust feature) với đặc trưng màu sắc để nâng cao độ chính xác cho hệ tìm kiếm ảnh với kết quả đạt được 88% [9].

Năm 2011, Zouaki & Abdelkhalak dựa trên cấu trúc cây KD-Tree đã đề xuất một mô hình truy vấn ảnh bằng cách lập chỉ mục cùng với vị trí của chúng trong ảnh nhằm giảm kích thước của đối tượng, giảm thời gian tính toán với độ đo tương tự sử dụng khoảng cách EMD. Trong phương pháp này tác giả đã thực nghiệm mang lại kết quả cao về độ chính xác [10]. Năm 2016, Jayashree Das và Minakshi Gogoi đã đề xuất một phương pháp lập chỉ mục và xây dựng hệ thống truy vấn ảnh dựa trên cây KD-Tree k -chiều. Nhóm tác giả đã trích xuất đặc trưng màu sắc của đối tượng ảnh, sau đó cây KD-Tree được xây dựng dựa trên đặc trưng màu sắc trích xuất để lập chỉ mục. Phương pháp này đã giảm đáng kể thời gian truy vấn ảnh trên cây KD-Tree. Kết quả thực nghiệm được so sánh với phương pháp lập chỉ mục hình ảnh để tìm kiếm mà không sử dụng cây KD-Tree thì thời gian tìm kiếm giảm còn $O(\log(n))$. Kết luận cho thấy cây KD-Tree có thời gian tìm kiếm nhanh hơn cây tứ phân QuadTree và kết quả tìm kiếm trên cây KD-Tree có độ chính xác cao hơn [11].

Năm 2014, Logamani & Punitha đã đề xuất một phương pháp gom cụm dữ liệu dựa trên cây KD-Tree với thuật toán k -NN [12]. Phương pháp này so sánh với các phương pháp trước

đó thì thời gian tìm kiếm trên cây giảm đáng kể [6]. Theo Logamani & Punitha năm 2014 đã đưa ra mô hình phân cụm dữ liệu dựa trên cây KD-Tree. Nhóm tác giả thực hiện gom cụm dữ liệu bằng phương pháp k-Means đồng thời xây dựng cây KD-Tree là một cây tăng trưởng nên việc xây dựng cây mất nhiều thời gian nhưng thích hợp cho bài toán có dữ liệu gia tăng [12].

Năm 2015, Kumar & Pavithra đã giới thiệu một mô hình biểu diễn và lập chỉ mục các đối tượng cần truy vấn. Trong nghiên cứu này, cơ sở dữ liệu đầu vào rất lớn nên thời gian truy xuất lâu. Một giải pháp cho tăng tốc quá trình truy xuất là thiết kế mô hình lập chỉ mục. Tác giả đã nghiên cứu cơ chế lập chỉ mục của cây KD-Tree cho hệ thống truy xuất dữ liệu dựa trên đặc trưng SIFT (Scale Invariant Feature Transform), biểu đồ phân lớp HOG (Histogram of Gradients), biểu đồ hướng cạnh EOH (Edge orientation histograms) và hình dạng SC (Shape context) [13].

Năm 2017, Cevikalpa *et al.* đã đề xuất một phương pháp truy vấn ảnh bằng cách sử dụng cây nhị phân phân cấp kết hợp với máy vectơ hỗ trợ chuyển đổi (TSVM). Hệ thống đã được triển khai và thực nghiệm đánh giá trên bộ dữ liệu ImageCLEF, hiệu suất truy vấn thu được với độ chính xác là 46,78% [14]. Cùng thời điểm đó, nhóm tác giả Jiu & Sahbi đã đề xuất một phương pháp mới để truy vấn ảnh. Trong phương pháp này, tác giả đã kết hợp mạng nhiều lớp học sâu đồng thời kết hợp kỹ thuật máy hỗ trợ vec-tơ (SVM) để phân lớp hình ảnh. Phương pháp này được thực nghiệm trên bộ dữ liệu ImageCLEF với hiệu suất truy vấn độ chính xác là 59,70% [15].

Năm 2019 đã có một số công trình truy vấn ảnh trên cây và thực nghiệm trên bộ ảnh ImageCLEF, cụ thể là mô hình truy vấn ảnh dựa trên cây phân cụm tự cân bằng C-Tree [16], với cây C-Tree nhóm tác giả đã thực nghiệm trên bộ ảnh ImageCLEF gồm 7092 hình ảnh truy vấn thì kết quả độ chính xác là 65%, thời gian truy vấn trung bình 73.0605 ms. Bên cạnh đó, một mô hình truy vấn ảnh dựa trên cây phân cụm phân cấp H-Tree cũng thực nghiệm trên bộ ảnh này với tổng số ảnh truy vấn là 7000 thì kết quả độ chính xác là 67% [17]. Ngoài ra, nhiều công trình nghiên cứu về tìm kiếm ảnh tương tự dựa trên cấu trúc dữ liệu dạng cây như tạo chỉ mục và cây chữ ký [18]; truy vấn ảnh dựa trên độ đo EMD và cây S-Tree [2]. Mô hình truy vấn ảnh dựa trên cây phân cụm đa nhánh cân bằng [19] v.v. cũng thu được những kết quả khả quan.

Từ phân tích các công trình liên quan cho thấy mô hình truy vấn ảnh dựa trên cấu trúc cây là một mô hình được đánh giá khả thi và có tính hiệu quả. Do đó trong bài báo này, nhóm tác giả tiếp cận theo phương pháp tổ chức và cải tiến cây KD-Tree để trở thành cây nhị phân cân bằng BKD-Tree nhằm lưu trữ dữ liệu đặc trưng cấp thấp của tập dữ liệu ảnh, đồng thời truy vấn nhanh các hình ảnh tương tự. Cuối cùng, nhóm tác giả thực nghiệm trên bộ ảnh ImageCLEF để chứng minh cho mô hình truy vấn ảnh đề xuất dựa trên cấu trúc cây BKD-Tree là hiệu quả.

2. CƠ SỞ LÝ THUYẾT

2.1. Mô tả cây BKD-Tree cải tiến

Cây BKD-Tree được xây dựng là một cấu trúc dữ liệu lưu trữ tập dữ liệu ảnh, phân cụm tự động các phần tử trên nút lá và tìm kiếm phần tử trên cây đã xây dựng, từ đó thực hiện việc cải tiến quá trình tìm kiếm ảnh tương tự bằng cách tăng tốc độ truy xuất mà vẫn đảm bảo độ chính xác.

Cây KD-Tree nguyên thủy lưu trữ dữ liệu tại tất cả các nút mà mỗi nút là một véc-tơ k -chiều. Trong trường hợp này, áp dụng cho bài toán có tập dữ liệu tăng trưởng thì cây KD-Tree sẽ tăng trưởng nhanh theo chiều sâu và làm cây mất cân bằng. Do đó, cần xây dựng cây BKD-Tree cân bằng để thời gian tìm kiếm trên các nút lá là gần như nhau. Nhóm tác giả cải tiến thành cây BKD-Tree với các nút lưu trữ dữ liệu là một thành phần x_i của véc-tơ

$V_{tb} = (x_0, x_1, \dots, x_{k-1})$ với k là chiều cao của cây, đồng thời xây dựng mô hình tìm kiếm ảnh tương tự theo nội dung dựa trên kỹ thuật tổ chức và lưu trữ dữ liệu trên cây BKD-Tree cải tiến đã đề xuất.

Cây BKD-Tree cải tiến là một cây nhị phân cân bằng, dữ liệu lưu trữ tại nút lá dùng để phân cụm, các nút trong không dùng lưu dữ liệu mà chỉ để phân lớp. Tại mỗi mức của nút trong là một giá trị phân lớp, đi từ mức thứ nhất đến mức thứ k sẽ hình thành véc-tơ k -chiều để phân lớp dữ liệu. Cấu trúc cây BKD-Tree được xây dựng dùng để lưu trữ và truy vấn tập ảnh tương tự của bộ ảnh ImageCLEF. Trong bộ ảnh này, mỗi ảnh chia thành nhiều vùng, mỗi vùng được trích xuất một véc-tơ đặc trưng có m thuộc tính. Vì vậy, nếu xây dựng cây BKD-Tree tương ứng m chiều thì số nút lá trên cây là 2^m , trong khi đó bộ ảnh ImageCLEF được chia thành 276 phân lớp khác nhau. Vì vậy, chiều cao của cây cần thiết để lưu trữ tập dữ liệu là $\log_2 276 \approx 8.1$. Do đó, nhóm tác giả chọn 9 thuộc tính của véc-tơ đặc trưng trên bộ ảnh ImageCLEF để tiến hành xây dựng cây BKD-Tree có chiều cao $h = 9$ là đủ để lưu tập dữ liệu thực nghiệm.

2.2. Xây dựng cấu trúc cây BKD-Tree

Cây BKD-Tree được xây dựng gồm một nút gốc (*root*) và nhiều nút trong (*iNode*), nút gốc và nút trong luôn có 2 nút con. Nút lá (*lvNode*) là nơi lưu trữ dữ liệu đồng thời đóng vai trò để phân cụm dữ liệu. Gọi $V_{tb} = (x_0, x_1, \dots, x_{k-1})$ là véc-tơ trung bình của bộ dữ liệu ban đầu, cấu trúc nút gốc, nút trong và nút lá được định nghĩa như sau:

Định nghĩa: Cây BKD-Tree là một cây nhị phân cân bằng thỏa mãn:

- Nút gốc là nút không có nút cha và có tối đa 2 nút con *left, right*. Thành phần của nút gốc là $root = \langle x, left, right \rangle$. Trong đó, $x = x_0$; *left, right* lần lượt là liên kết đến cây con trái và cây con phải.
- Nút trong *iNode* = $\langle x, ileft, iright \rangle$, với $x = x_l$ là một giá trị tại mức thứ l .
- Nút lá *lvNode* = $\langle f_i, id_i \rangle$ là nút không có nút con, trong đó f_i, id_i lần lượt là véc-tơ đặc trưng và định danh của ảnh thứ i .

Trên cơ sở **Định nghĩa** các nút trên cây, quá trình tạo cây BKD-Tree được mô tả như sau:

Bước 1: Tại thời điểm ban đầu cây BKD-Tree chỉ có một nút gốc $root = \emptyset$, với mức $l_0 = 0$

Bước 2: Tính véc-tơ trung bình của bộ dữ liệu ban đầu $V_{tb} = (x_0, x_1, \dots, x_{k-1})$ và gán giá trị nút gốc là $root.x = x_0$

Bước 3: Gán giá trị tại mỗi nút trong mức l_i là x_i

Bước 4: Với mỗi véc-tơ $f_j = (x_{j0}, x_{j1}, \dots, x_{j(k-1)})$ lần lượt so sánh giá trị x_i ($i = 0..m-1$) với các giá trị x_{jl} ($l = 1..m-1$) trong cây. Nếu $x_{jl} > x_i$ thì véc-tơ f_j thuộc cây con bên phải; ngược lại f_j thuộc cây con bên trái. Quá trình này lặp lại cho đến khi tìm được nút lá và f_j được lưu vào nút lá đó.

Trên cơ sở **Định nghĩa** các thành phần trên cây, nhóm tác giả đã xây dựng cây BKD-Tree có chiều cao $h = m$ tại mỗi nút trong luôn tồn tại nút con trái và nút con phải, mức $(m-1)$ của cây tại mỗi nút trong lưu trữ 2 nút lá. Bên cạnh đó, chiều cao của cây con trái và chiều cao cây con phải $h_l = h_r = m$ Vậy BKD-Tree được xây dựng là một cây nhị phân cân bằng.

Định lý: Với mỗi véc-tơ đặc trưng f_i của ảnh I , ta có:

- Tồn tại duy nhất một nút lá trên cây BKD-Tree để lưu trữ véc-tơ f_i . Véc-tơ đặc trưng f_i được lưu trữ trên nút lá phải phù hợp về độ đo tương tự.
- Véc-tơ đặc trưng f_i được lưu trữ trên nút lá phải phù hợp về độ đo tương tự.

Chứng minh:

- a) **Tính tồn tại:** Vì BKD-Tree là cây nhị phân cân bằng, dữ liệu chỉ lưu trữ tại nút lá. Do đó, tại mỗi nút *trong* của cây BKD-Tree luôn tồn tại một nhánh con trái hoặc một nhánh con phải để véc-tơ f_i tìm đến vị trí của nút lá phù hợp. Do đó, luôn tồn tại một nút lá để lưu trữ véc-tơ f_i .

Tính duy nhất: Duyệt từ nút gốc của cây, tại mỗi nút *trong* của cây BKD-Tree, ta chỉ chọn được duy nhất một hướng (trái hoặc phải) để tìm vị trí lưu trữ véc-tơ đặc trưng f_i . Do đó, ta chỉ chọn được một nút lá phù hợp nhất để lưu trữ véc-tơ f_i , nghĩa là véc-tơ f_i chỉ thuộc về một nút lá duy nhất.

- b) Vì mỗi lần thực hiện thêm phần tử f_i vào nút lá trên cây BKD-Tree, ta phải duyệt từ nút gốc và tìm nhánh gần nhất nên việc chọn nhánh kế cận có độ tương tự gần nhất (phù hợp nhất). Do đó, kết quả quá trình này là tìm được một nút lá phù hợp nhất để thêm phần tử f_i vào cây BKD-Tree.

2.3. Thuật toán tạo cây BKD-Tree

Thuật toán tạo cây - IEBKT

Input: Tập phần tử f_v cần thêm vào cây BKD-Tree

Output: Cây BKD-Tree

Function IEBKT ($f_v, BKD - Tree, m$)

Begin

$BKD - Tree = \emptyset;$

$V_{tb} = (x_0, x_1 \dots, x_l) = avg \{x_{ij}: i = 0..m - 1; j = 0..m - 1\}$

$root.x = x_0; l_0 = 0;$

If ($f_v.x_m < V_{tb}.x_m$) **then** $BKD-Tree = BKD-Tree \cup$ **IEBKT** ($f_v.x_l, BKD-Tree.left, m+1$)

Else $BKD-Tree = BKD-Tree \cup$ **IEBKT** ($f_v.x_l, BKD-Tree.right, m+1$)

EndIf

Return $BKD-Tree;$

End.

Mệnh đề 1: Độ phức tạp của thuật toán **IEBKT** là $O(n)$. Với n số phần tử cần thêm vào cây BKD-Tree.

Chứng minh:

Gọi m, n lần lượt là chiều cao và số phần tử cần chèn vào cây. Với mỗi phần tử cần chèn vào cây, **Thuật toán IEBKT** duyệt qua m mức của cây nhằm tìm được nút lá phù hợp để chèn phần tử vào. Do đó, số phép toán so sánh của thuật toán là $m*n$. Mặt khác, n thường lớn hơn nhiều so với m , vì vậy độ phức tạp của **Thuật toán IEBKT** là $O(n)$ ■

Nhằm giảm không gian lưu trữ và tăng tốc độ truy vấn ảnh, bài báo này xây dựng cấu trúc cây BKD-Tree tổ chức lưu trữ dữ liệu tại các nút lá, các nút *trong* đóng vai trò trung gian. Dữ liệu được lưu trữ ở đây là các véc-tơ đặc trưng f_i của ảnh I đã phân đoạn. Trên mỗi vùng ảnh đã phân đoạn, véc-tơ đặc trưng vùng được trích xuất dựa trên các đặc trưng thị giác như màu sắc, vị trí, hình dạng, kết cấu v.v. Mỗi véc-tơ đặc trưng thị giác được gán nhãn để mô tả nội dung thị giác cho từng vùng ảnh tương ứng. Như vậy mỗi hình ảnh được trích xuất nhiều véc-tơ đặc trưng k -chiều. Sau khi tạo cây BKD-Tree theo quy tắc trên, bài toán tìm kiếm ảnh tương tự theo nội dung được nâng cao hiệu quả về hiệu suất truy vấn và tốc độ tìm kiếm.

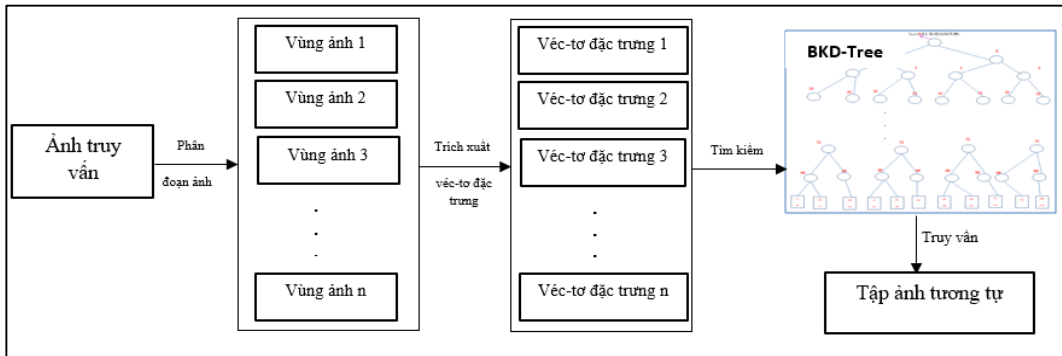
2.4. Truy vấn ảnh trên cây BKD-Tree

Mỗi ảnh trên bộ ImageCLEF được chia thành nhiều vùng và các vùng của ảnh có thể thuộc nhiều phân lớp khác nhau. Các vùng ảnh phân đoạn của ảnh 18161.JPG gồm 8 ảnh phân đoạn được minh họa như **Hình 1**.



Hình 1. Ảnh gốc và 5 vùng của ảnh đã phân đoạn

Trên cơ sở thuật toán tạo cây BKD-Tree (**IEBKT**), quá trình truy vấn ảnh trên cây được thực hiện bằng cách: Một ảnh gồm nhiều vùng ảnh tương ứng, mỗi vùng ảnh được trích xuất một véc-tơ đặc trưng, quá trình truy vấn để tìm ra tập ảnh tương tự của hình ảnh cần truy vấn được thực hiện dựa vào véc-tơ đặc trưng của từng vùng. Hệ thống thực hiện tìm kiếm ảnh tương tự bằng cách so sánh véc-tơ đặc trưng $f_i = (x_{i0}, x_{i1}, \dots, x_{i(m-1)})$ của mỗi vùng ảnh truy vấn với các thành phần tương ứng của véc-tơ trung bình $V_{tb} = (x_0, x_1, \dots, x_l)$ trên cây BKD-Tree theo hướng từ nút gốc đến nút lá và theo quy tắc xây dựng cây BKD-Tree đã đề xuất. Kết quả của quá trình này là tập hợp các nút lá chứa véc-tơ đặc trưng của từng vùng thuộc ảnh cần truy vấn. Quá trình này minh họa bằng cấu trúc của mô hình truy vấn ảnh trên cây BKD-Tree trong Hình 2.

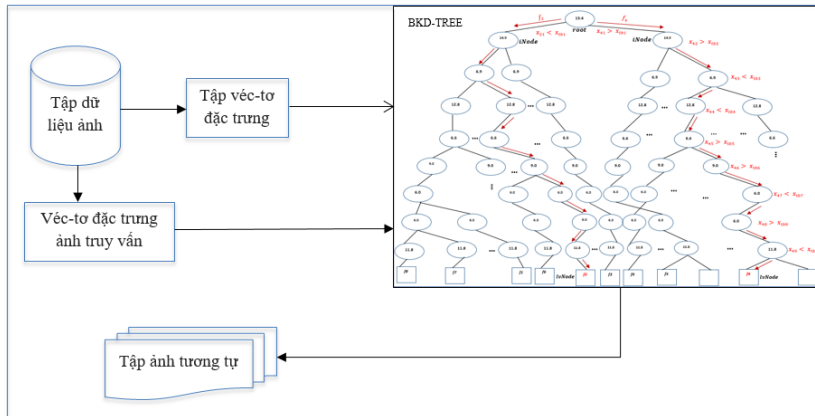


Hình 2. Cấu trúc của mô hình truy vấn ảnh trên cây BKD-Tree

3. XÂY DỰNG THỰC NGHIỆM

3.1. Mô hình truy vấn ảnh

Tập ảnh tương tự với ảnh cần truy vấn I được tra cứu bằng cách lần lượt duyệt qua các nút từ gốc đến lá dựa vào giá trị $f_i \cdot x_{il}$ của véc-tơ đặc trưng vùng ảnh cần truy vấn so sánh với giá trị tại mỗi nút trung gian, nếu $f_i \cdot x_{il} < V_{tb} \cdot x_l$ thì thực hiện tìm kiếm sang nhánh trái và ngược lại. Quá trình này đệ quy đến khi tìm đến nút lá thì dừng. Tập các nút lá có chứa các véc-tơ đặc trưng vùng của ảnh I là tập ảnh tương tự cần tìm.



Hình 3. Mô hình truy vấn ảnh theo nội dung dựa trên cây BKD-Tree

3.2. Quá trình truy vấn ảnh được mô tả như sau:

- Bước 1: Trích xuất tập tập véc-tơ đặc trưng f_i của tập dữ liệu ảnh ban đầu.
- Bước 2: Từ tập véc-tơ đặc trưng này, xây dựng cây BKD-Tree theo thuật toán đã đề xuất.
- Bước 3: Trích xuất véc-tơ đặc trưng ảnh cần truy vấn và truy vấn trực tiếp trên cây BKD-Tree.
- Bước 4: Sắp xếp tập các ảnh tương tự với ảnh cần truy vấn.

4. KẾT QUẢ THỰC NGHIỆM

4.1. Cài đặt thực nghiệm

Hệ truy vấn ảnh theo nội dung CBIR_BKD-Tree thực nghiệm trên bộ dữ liệu ImageCLEF lưu trữ trong 41 thư mục. Mỗi ảnh được phân đoạn thành các vùng được gán nhãn, có một véc-tơ đặc trưng và thuộc một phân lớp. Bộ dữ liệu ảnh có 99.535 vùng, được phân thành 276 lớp. Thực nghiệm truy vấn ảnh CBIR-BKD-Tree được xây dựng trên nền tảng dotNET Framework 4.5, ngôn ngữ lập trình C#. Các đồ thị được xây dựng trên Matlab 2015. Cấu hình máy tính thực nghiệm: Intel(R) Core™ i5-5200U, CPU 2.2GHz, RAM 8GB và hệ điều hành Windows 10 Professional.

4.2. Đánh giá kết quả thực nghiệm

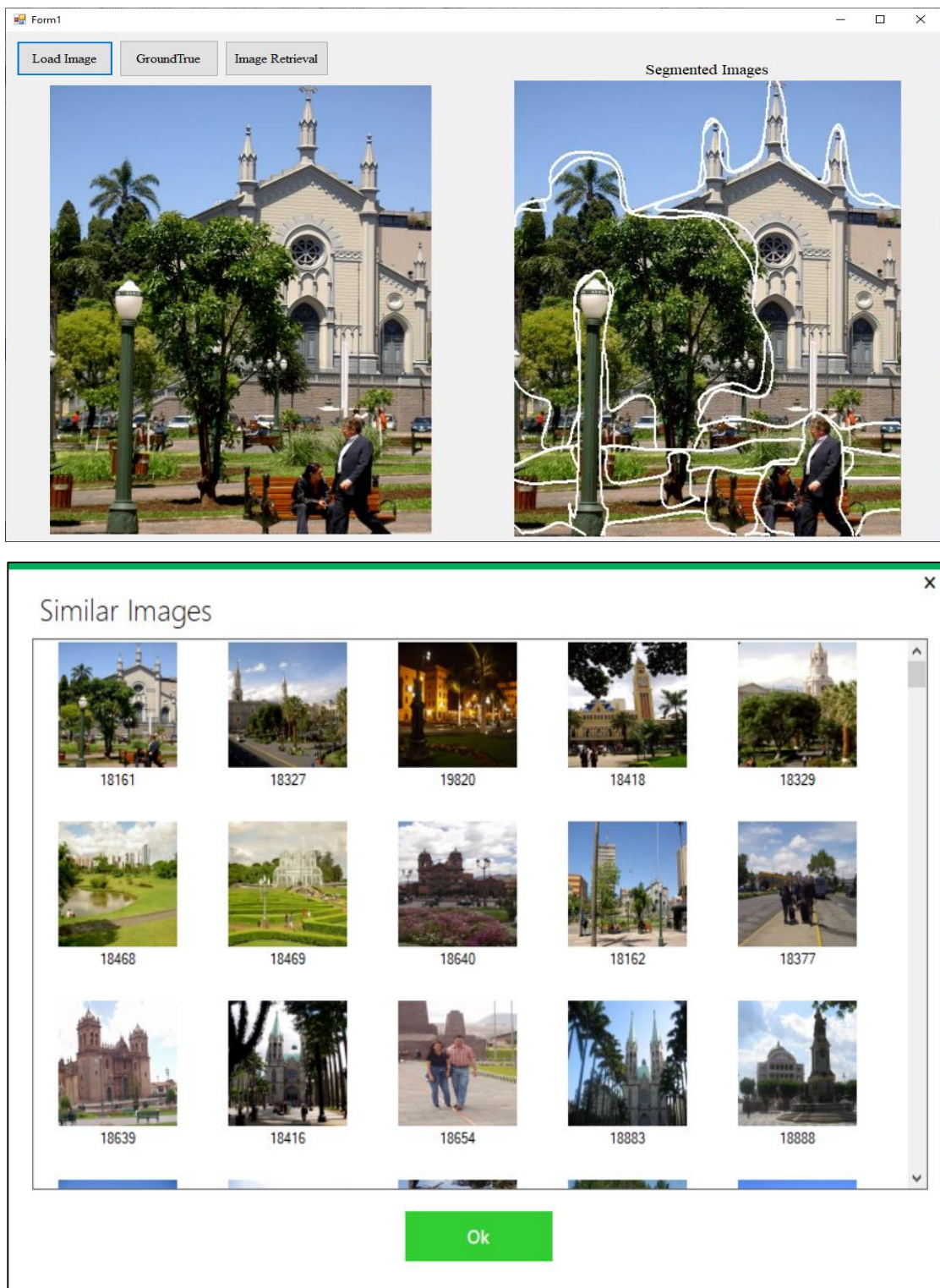
Kết quả thực nghiệm được đánh giá trên bộ dữ liệu imageCLEF chứa 20.000 ảnh. Để đánh giá hiệu suất của phương pháp tìm kiếm ảnh, phần thực nghiệm được đánh giá các giá trị gồm: độ chính xác (precision), độ phủ (recall) và độ đo dung hòa F-measure. Công thức tính các giá trị này như sau:

$$precision = \frac{|relevant\ images \cap retrieved\ images|}{|retrieved\ images|}$$

$$recall = \frac{|relevant\ images \cap retrieved\ images|}{|relevant\ images|}$$

$$F - measure = 2 \times \frac{(precision \times recall)}{(precision + recall)}$$

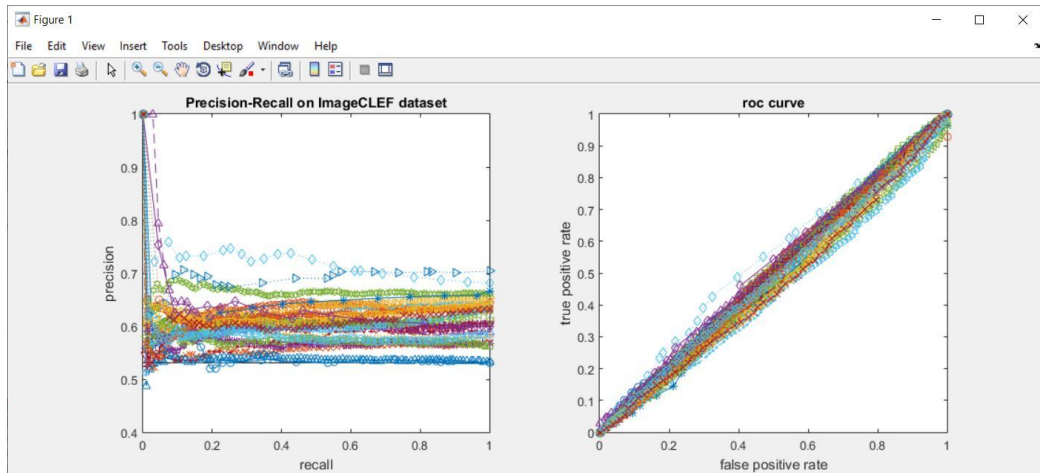
Trong đó, relevant images là tập ảnh tương tự với ảnh truy vấn và có trong tập dữ liệu ảnh, retrieved images là tập ảnh đã tìm kiếm được. Các giá trị độ chính xác, độ phủ và độ dung hòa được tính theo tỷ lệ % và được quy đổi thành giá trị trên đoạn [0, 1]. Nhóm tác giả đã thu được kết quả thực nghiệm cho hiệu suất truy xuất hình ảnh của phương pháp đề xuất trên tập dữ liệu ImageCLEF trong Bảng 1, có 7079 hình ảnh truy vấn; trung bình của hiệu suất độ chính xác 60,92%, độ phủ 49,84%, độ dung hòa 53,49%, và thời gian truy vấn trung bình là 28,15(ms).



Hình 4. Hệ truy vấn ảnh CBIR_BKD-Tree

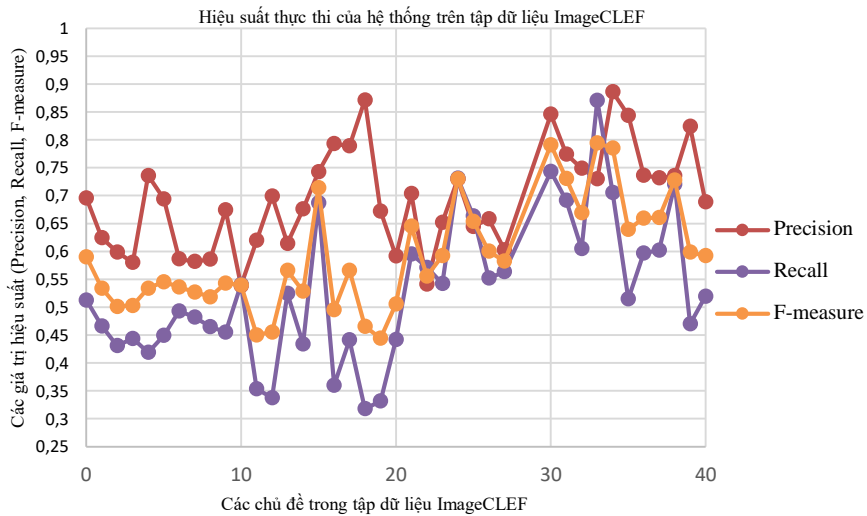
Bảng 1. Hiệu suất truy xuất hình ảnh theo phương pháp đề xuất trên bộ dữ liệu ImageCLEF

Chủ đề	Số ảnh	Trung bình độ chính xác	Trung bình độ phủ	Trung bình độ dung hòa	Thời gian truy vấn trung bình (ms)
00-10	1675	0,591724	0,461117	0,511076	27,565764
11-20	1674	0,638282	0,454543	0,522616	30,768464
21-30	1746	0,612864	0,483218	0,528895	27,575644
31-40	1984	0,594311	0,594797	0,577076	26,676857
AVG	7079	0,609295	0,498418	0,534916	28,146682

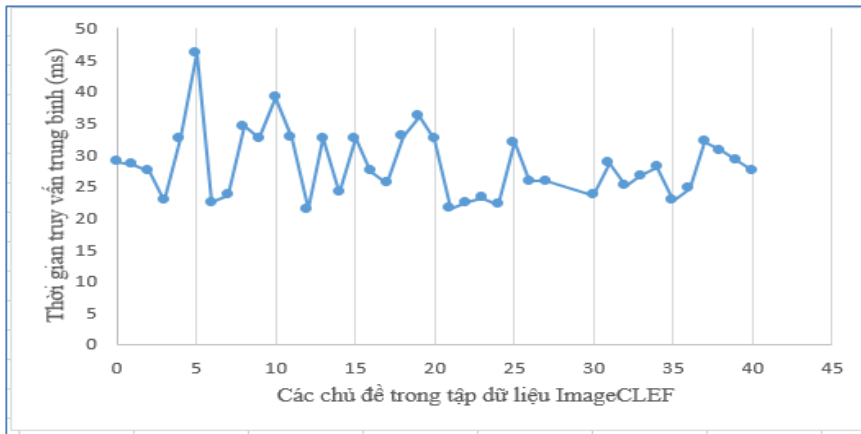


Hình 5. Đồ thị Precision – Recall và đường cong ROC của hệ CBIR_BKD-Tree trên dữ liệu ImageCLEF

Đồ thị của giá trị Precision-Recall và đường cong ROC cho bộ dữ liệu ImageCLEF được minh họa bởi Hình 5. Đồ thị này cho thấy diện tích nằm dưới đường cong Precision-Recall chưa cao, do tính chính xác của hệ truy vấn nằm tập trung ở vùng 0.5 đến 0.7. Tuy nhiên, có vài tập ảnh cho độ chính xác cao hơn nằm trong vùng [0.8, 1.0]. Đồ thị đường cong ROC biểu diễn các giá trị true positive và false positive, các giá trị chủ yếu nằm tập trung trên đường baseline.



Hình 6. Trung bình Precision, Recall, F-measure hệ CBIR_BKD-Tree trên tập dữ liệu ImageCLEF



Hình 7. Thời gian truy vấn trung bình hệ CBIR_BKD-Tree trên tập dữ liệu ImageCLEF

Các đồ thị trong Hình 6 và 7 mô tả giá trị trung bình của độ chính xác, độ phủ, độ dung hoà và thời gian truy vấn trung bình của bộ dữ liệu ImageCLEF. Từ các đồ thị này cho thấy, tính chính xác của truy vấn nằm ở mức trung bình, cần cải thiện thêm để nâng cao hiệu suất truy vấn. Độ phủ của truy vấn khá thấp nên độ dung hoà của truy vấn chưa cao. Tuy nhiên, tốc độ truy vấn khá nhanh cho thấy, hệ thống truy vấn ảnh CBIR_BKD-Tree được đánh giá là khá tốt về thời gian truy vấn.

Giá trị trung bình độ chính xác MAP của hệ truy vấn CBIR_BKD-Tree được so sánh với các phương pháp khác trên cùng bộ dữ liệu ImageCLEF, thể hiện trong Bảng 2.

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

Phương pháp	Giá trị trung bình độ chính xác (MAP)
H. Cevikalp, 2017 [14]	0,4678
M. Jiu, 2017 [15]	0,5970
CBIR_BKD-Tree	0,6092

Từ kết quả so sánh ở Bảng 2 cho thấy, hệ truy vấn hình ảnh CBIR-BKD-Tree có độ chính xác khá tốt so với các nghiên cứu gần đây trong cùng lĩnh vực trên cùng bộ dữ liệu. Kết quả này chứng minh rằng, phương pháp đề xuất của nhóm tác giả là hiệu quả và có khả năng phát triển.

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong bài báo này, nhóm tác giả đã xây dựng một cấu trúc cây BKD-Tree nhị phân cân bằng áp dụng cho bài toán tìm kiếm ảnh tương tự. Nhóm tác giả đã đề xuất mô hình truy vấn ảnh dựa trên cây BKD-Tree và thực nghiệm trên bộ ảnh ImageCLEF có độ chính xác là 60,92%, độ phủ 49,84%, độ dung hoà 53,49%, và thời gian truy vấn trung bình là 28,15 (ms). Kết quả thực nghiệm đã được so sánh với các công trình khác trên cùng một tập dữ liệu ảnh, đồng thời so sánh với các phương pháp dựa trên cấu trúc lưu trữ cây KD-Tree. Thực nghiệm cho thấy tính đúng đắn và hiệu quả của phương pháp đề xuất. Phương pháp đề xuất của nhóm tác giả đã làm tăng đáng kể hiệu suất truy vấn ảnh, đáp ứng yêu cầu người dùng. Tuy nhiên, việc xây dựng cây BKD-Tree phụ thuộc nhiều vào vec-tơ trung bình của tập dữ liệu ban đầu. Do đó, hướng phát triển tiếp theo của nhóm tác giả là tạo một cây chỉ mục với mỗi nút liên kết tới một phần tử của bảng tra cứu, với bảng tra cứu này được xây dựng một cách độc lập với cây BKD-Tree để từ đó tăng tính hiệu quả trong việc phân lớp trên cây BKD-Tree.

Lời cảm ơn: Trân trọng cảm ơn nhóm nghiên cứu SBIR-HCM và Trường Đại học Sư phạm Thành phố Hồ Chí Minh đã hỗ trợ về chuyên môn và cơ sở vật chất giúp nhóm tác giả hoàn thành nghiên cứu này.

TÀI LIỆU THAM KHẢO

1. Thanh The Van, Nguyen Van Thinh, Thanh Manh Le - The method proposal of image retrieval based on K-means algorithm, *Advances in Intelligent Systems and Computing* **746** (2) (2018) 481-490.
2. Thanh The Van, Manh Thanh Le - Image Retrieval system based on EMD similarity measure and S-Tree, *ICITES-2012, Springer Verlag, LNEE* 234 (2013) 139-146.
3. AI-Jabbouli H. - Data clustering using the bee algorithm and the KD-Tree structure, PhD Thesis, Cardiff University, United Kingdom, 2009.
4. Abudalfa S., Mikki M. - A dynamic linkage clustering using KD-Tree, *The International Arab Journal of Information Technology* **10** (3) (2013) 283-289.
5. Văn Thế Thành, Lê Mạnh Thạnh - Truy vấn ảnh tri thức dựa trên chữ ký nhị phân, *Tạp chí Khoa học Đại học Huế* **97** (9) (2014) 1-20.
6. Otair M. - Approximate K-nearest neighbour based spital clustering using K-D Tree, *International Journal of Database Management Systems* **5** (1) (2013) 97-108.
7. He Y., Lu G. and Teng S. - An investigation of using K-d Tree to improve image retrieval efficiency, *DICTA2002: Digital Image Computing Techniques and Application, Melbourne, Australia* (2002).
8. Redmond S.J., Heneghan C. - A method for initialising the K-Means clustering algorithm using KD-Tree, *Patter Recognition Letters* **28** (8) (2007) 965-973.
9. Velmurugan K., Baboo L.D - Content-based image retrieval using surf and color moment, *Global Journal of Computer Science and Technology* **11** (10) (2011).
10. Zouaki H., Abdelkhalak B. - Indexing and content based image retrieval, *International Conference on Multimedia Computing and System* (2011) 1-5.
11. Jayashree Das, Minakshi Gogoi - Indexing of voluminous data using K-DTree with reference to CBIR, *International Journal of Computer Sciences and Engineering* **4** (7) (2016) 117-124.
12. Logamani. K, Punitha. S. C - Density base clustering using enhanced KD Tree, *International Journal of Computer Science Engineering and Technology* **4** (11) (2014) 314-318.
13. Kumar Y.H.S, Pavithra N. - KD-Tree approach in sketch based image retrieval, *MIKE 2015: Proceedings of the Third International Conference on Mining Intelligence and Knowledge Exploration* **9468** (2015) 247-258.
14. Cevikalpa H., Elmas M., Ozkan S. - Large-scale image retrieval using transductive support vector machines, *Computer Vision and Image Understanding* (2017) 1-11.
15. Jiu M., Sahbi H. - Nonlinear deep kernel learning for image annotation, *IEEE Transation on Image Processing* **26** (2017) 1820-1832.
16. Nguyen Thi Uyen Nhi, Van The Thanh, Le Manh Thanh - A self balanced clustering tree apply for semantic-based image retrieval, *Fundamental and Applied IT Research (FAIR), Hue University*, 2019.

17. Nguyễn Minh Hải, Lê Thị Vĩnh Thanh, Văn Thế Thành, Trần Văn Lăng - Tra cứu ảnh theo ngữ nghĩa dựa trên cây phân cụm phân cấp, Kỷ yếu hội thảo khoa học quốc gia về nghiên cứu cơ bản và ứng dụng, 2019.
18. Nascimento M.A., Tousidou E., Chitkara V., Manolopoulos Y. - Image indexing and retrieval using signature tree, *Data and Knowledge Engineering* **43** (1) (2002) 57-77.
19. Văn Thế Thành, Nguyễn Phương Hạc - Một phương pháp tra cứu dữ liệu ảnh dựa trên cây phân cụm phân cấp đa nhánh cân bằng, *Tạp chí Khoa học Công nghệ và Thực phẩm* **18** (1) (2019) 140-153.

ABSTRACT

AN IMPROVEMENT OF KD-TREE FOR THE IMAGE RETRIEVAL PROBLEM

Nguyen Thi Dinh^{1*}, Le Thi Vinh Thanh²

Nguyen The Huu¹, Nguyen Van Thinh¹

¹*Ho Chi Minh City University of Food Industry*

²*Baria Vungtau University*

*Email: *dinhnt@hufi.edu.vn*

Searching for images is a concerned problem and many methods have been published recently. In this paper, we build a data clustering model based on the KD-Tree to apply to the image search problem, in which the improved BKD Tree tree includes: (1) storing multi-dimensional data objects at the leaf nodes of trees to create a clustering model on the basis of semi-supervised learning method; (2) creating a balanced tree structure to increase the efficiency of image searching. Based on the proposed theory, we proposed the image query model based on KD-Tree and simultaneous experiment on ImageCLEF dataset (including 20,000 images). Our empirical results were compared with several recent works on the same dataset to demonstrate the effectiveness of the proposed method. The experimental results show that our method is effective and can be applied to content-based similar image search systems.

Keywords: KD-Tree, clustering, similar image, similar measure, image retrieval.