

Phân lớp ảnh bằng cây KD-Tree cho bài toán tìm kiếm ảnh tương tự

Nguyễn Thị Định^{1,3}, Văn Thế Thành², Lê Mạnh Thành³

¹ Khoa Công nghệ Thông tin, Trường ĐH Công nghiệp Thực phẩm TP.HCM

² Phòng quản lý khoa học và đào tạo sau Đại học, Trường ĐH Công nghiệp Thực phẩm TP.HCM

³ Trường Đại học Khoa học, Đại học Huế

Tác giả liên hệ: Nguyễn Thị Định, nguyenthidinh.hcm@gmail.com

Ngày nhận bài: 15/04/2021, ngày sửa chữa: 01/06/2021, ngày duyệt đăng: 12/06/2021

Định danh DOI: 10.32913/mic-ict-research-vn.v2021.n1.966

Tóm tắt: Trong bài báo này, một cấu trúc KD-Tree (k - Dimensional Tree) cải tiến được xây dựng nhằm phân lớp dữ liệu hình ảnh và ứng dụng cho bài toán tìm kiếm ảnh tương tự gọi là CKD-Tree (Classification k -Dimensional Tree). Quá trình xây dựng cấu trúc CKD-Tree được thực hiện theo phương pháp học bán giám sát làm cơ sở cho quá trình phân lớp dữ liệu hình ảnh tại các nút trong, đồng thời gom cụm dữ liệu tại các nút lá. Kết quả phân lớp này được áp dụng cho quá trình tìm kiếm tập ảnh tương tự từ một ảnh đầu vào sau khi trích xuất véc-tơ đặc trưng. Để minh chứng cho cơ sở lý thuyết đã đề xuất, chúng tôi tiến hành xây dựng cấu trúc CKD-Tree thực nghiệm trên bộ ảnh COREL (gồm 1000 ảnh, 10 phân lớp) và bộ ảnh Wang (gồm 10800 ảnh, 80 phân lớp). Kết quả thực nghiệm truy vấn ảnh được so sánh với các công trình khác cùng bộ dữ liệu nhằm minh chứng phương pháp đề xuất của chúng tôi là hiệu quả và áp dụng tốt trong các hệ tìm kiếm dữ liệu đa phương tiện.

Từ khóa: CKD-Tree, image classification, image retrieval, similar image.

Title: Image Classification Using Kd-Tree for Image Retrieval Problem

Abstract: In this paper, an improved KD-Tree (k - Dimensional Tree) structure is built to classify image data and applied to the similar image retrieval which called CKD-Tree (Classification k -Dimensional Tree). The CKD-Tree structure is built on the semi-supervised learning method. It is not only the background of the process of classifying image data at the inner nodes but also clustering data at the leaf nodes. This classification result is applied to retrieval the similar image set from an input image after extracting the feature vector. Proving for proposed theorem, we had experimental research of CKD-Tree structure on the COREL (including 1000 images, 10 subclasses) and Wang image set (including 10800 images, 80 subclasses). The experimental results of image retrieval evaluated with the recently published methods on the same dataset. This proof shows that our proposed method is effective and appropriate for multimedia data retrieval systems.

Keywords: CKD-Tree, image classification, image retrieval, similar image.

I. GIỚI THIỆU

Ngày nay, các phương pháp tìm kiếm ảnh được thực hiện bởi nhiều công trình nghiên cứu; trong đó tìm kiếm ảnh dựa trên nội dung CBIR (Content-based Image Retrieval) [3, 6] là phương pháp tìm kiếm dựa trên các đặc trưng cấp thấp như màu sắc, hình dạng, kết cấu, v.v.. Bài toán tìm kiếm ảnh đã trở nên cấp thiết đối với con người vì được ứng dụng trong nhiều lĩnh vực như: y tế, giáo dục, giải trí, địa lý, ứng dụng y sinh, v.v.. Sự phát triển của các thiết bị điện tử như camera, smartphone, v.v. đã làm cho ảnh số gia tăng nhanh và trở nên quen thuộc, gắn gũi với cuộc sống con người. Điều này đã mang lại nhiều cơ hội và thách thức cho lĩnh

vực tra cứu ảnh. Vì vậy, nhiều hệ tìm kiếm ảnh đã công bố [2, 3, 4, 5] nhằm ứng dụng vào nhiều lĩnh vực trong đời sống và nâng cao hiệu suất truy vấn ảnh. Bài toán tìm kiếm ảnh tương tự là một trong những vấn đề quan trọng của hệ thống tra cứu dữ liệu đa phương tiện được nhiều nhóm nghiên cứu quan tâm [6, 7]. Trong cách tiếp cận của chúng tôi, một kỹ thuật phân lớp dữ liệu hình ảnh dựa trên cấu trúc KD-Tree cải tiến nhằm tạo ra một quá trình phân lớp dữ liệu theo mô hình cây; đồng thời ứng dụng cho bài toán tìm kiếm ảnh tương tự.

Đóng góp của bài báo gồm: (1) Đề xuất phương pháp phân lớp dữ liệu dựa trên cấu trúc CKD-Tree; (2) Đề xuất

các thuật toán xây dựng cấu trúc CKD-Tree; gán nhãn cho nút lá; huấn luyện trọng số và thuật toán tìm kiếm tập ảnh; (3) Đề xuất mô hình tìm kiếm ảnh tương tự dựa trên cấu trúc CKD-Tree; (4) Xây dựng thực nghiệm và chứng minh tính đúng đắn của phương pháp đề xuất dựa trên bộ dữ liệu COREL [26], Wang [27].

Phần còn lại của bài báo gồm: Phần II, khảo sát và phân tích ưu, nhược điểm của một số công trình liên quan. Phần III, trình bày thuật toán xây dựng cấu trúc CKD-Tree, gán nhãn nút lá và huấn luyện trọng số. Phần IV, trình bày mô hình truy vấn ảnh và thuật toán tìm kiếm ảnh tương tự dựa trên cấu trúc CKD-Tree. Mô tả dữ liệu, xây dựng thực nghiệm và kết quả được đánh giá trên bộ ảnh COREL (1000 ảnh) và bộ dữ liệu Wang (10800 ảnh) được trình bày trong phần V; Phần VI là kết luận và hướng phát triển.

II. CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

Phân lớp ảnh và ứng dụng cho bài toán tìm kiếm ảnh tương tự đã được nhiều công trình thực hiện với các kỹ thuật học máy khác nhau: phân lớp bằng thuật toán tìm kiếm láng giềng gần nhất k-NN (*k-Nearest Neighbors*) [10, 13]; phân lớp bằng kỹ thuật SVM (*Support Vector Machines*) [9]; phân lớp bằng Naïve Bayes [14]; phân lớp theo cấu trúc cây như cây quyết định (*Decision Tree*), cây tìm kiếm đa chiều (*KD-Tree*) [10, 12], v.v..

Yuqian Zhang và cộng sự (2016) [8] sử dụng phương pháp phân chia vùng ảnh để phác thảo và nhận diện khuôn mặt. Trong công trình này, cấu trúc KD-Tree được xây dựng trên tập ảnh phân vùng nhằm phân lớp và lưu trữ dữ liệu. Cây KD-Tree được xây dựng theo cấu trúc chỉ mục nhằm giảm thời gian tìm kiếm đồng thời

Kết quả nhận diện khuôn mặt với thuật toán tìm kiếm láng giềng k-NN. Thực nghiệm trên bộ ảnh khuôn mặt phác thảo CUFS (*Chinese University Face Sketch*) chứng minh rằng phương pháp đề xuất của tác giả là hiệu quả. Tuy nhiên, cấu trúc KD-Tree xây dựng theo mô hình phân lớp dữ liệu chưa kết hợp các kỹ thuật học máy nhằm tăng hiệu suất nhận diện khuôn mặt.

Dan Gao và cộng sự (2018) [9] thực hiện một so sánh giữa kỹ thuật phân lớp dữ liệu lớn bằng thuật toán SVM kết hợp cấu trúc KD-Tree cho bộ dữ liệu SDSS (*Sloan Digital Sky Survey*) và 2MASS (*Two-Micron All Sky Survey*). Xét tốc độ và độ chính xác thì việc phân lớp theo KD-Tree mang lại hiệu suất cao hơn so với kỹ thuật SVM. Bên cạnh đó, những phần tử phân loại sai của SVM và KD-Tree là trùng khớp nhau. Điều này chứng tỏ rằng phương pháp phân lớp dữ liệu bằng cây KD-Tree là hoàn toàn khả thi và hiệu quả, có thể so sánh với các phương pháp phân lớp trước đây.

Wenfeng Hou và cộng sự (2018) [27] đã thực hiện một phương pháp phân lớp dữ liệu và áp dụng cho bài toán

tìm kiếm ảnh bằng cách kết hợp thuật toán tìm kiếm láng giềng k-NN và cấu trúc KD-Tree. Trong công trình này, tác giả kết hợp thuật toán k-NN và cấu trúc KD-Tree để xây dựng cây k-NN-KDTree theo mô hình phân lớp bằng phương pháp học bán giám sát. Tại mỗi tầng trên cây lần lượt chọn mỗi chiều $x_0, x_1, \dots, x_k$ đại diện trong véc-tơ $f(x_0, x_1, \dots, x_k)$ làm cơ sở cho quá trình so sánh và xây dựng cây nhị phân. Kết quả thực nghiệm so sánh hiệu suất tìm kiếm với phương pháp chỉ dùng thuật toán k-NN trên các bộ dữ liệu ảnh chứng minh rằng sự kết hợp k-NN và cây KD-Tree đã mang lại hiệu quả cao; đồng thời so sánh thời gian huấn luyện trên các bộ ảnh thì kết quả đề xuất phương pháp kết hợp k-NN-KDTree chỉ bằng $\frac{1}{4}$ so với phương pháp chỉ dùng thuật toán k-NN.

Fengquan Zhang và cộng sự (2019) [11] đã thực hiện một phương pháp đối sánh hình ảnh bằng cách xây dựng cấu trúc Vocabulary-KD; ứng dụng cho các bộ dữ liệu tăng trưởng; đồng thời cải thiện thời gian tìm kiếm. Cuối cùng tác giả đề xuất giải pháp đa tiến trình song song cho quá trình xây dựng và tìm kiếm đồng thời dựa trên cấu trúc Vocabulary-KD. Tuy nhiên, tác giả chưa đề cập đến quá trình xây dựng cấu trúc Vocabulary-KD theo phương pháp phân lớp với đa tiến trình nhằm tăng hiệu suất tìm kiếm từ kết quả phân lớp.

Reid Pinkham và cộng sự (2020) [26] đã thực hiện một phương pháp tối ưu hóa bộ nhớ và hiệu suất truy vấn các đối tượng 3D bằng hình ảnh theo phương pháp tìm kiếm láng giềng k-NN kết hợp cấu trúc KD-Tree. Trong công trình này, nhóm tác giả đã thực hiện tối ưu hóa thời gian truy vấn, tăng khả năng lưu trữ bằng cách thực hiện đa tiến trình song song, kết hợp tìm kiếm theo cấu trúc KD-Tree; với kiến trúc này để áp dụng cho các loại dữ liệu tăng trưởng đa chiều.

Từ các công trình nghiên cứu cho thấy phương pháp phân lớp dữ liệu hình ảnh sử dụng các kỹ thuật học máy kết hợp với cấu trúc KD-Tree là hoàn toàn khả thi, hiệu quả. Tuy nhiên, các công trình này hầu hết thực hiện phân lớp dữ liệu hình ảnh dựa cấu trúc KD-Tree chưa kết hợp thêm các kỹ thuật học máy hoặc chưa thực hiện phân lớp nhiều lần cho một đối tượng theo phương pháp học sâu nhằm nâng cao độ chính xác. Trong bài báo này, chúng tôi đề xuất một phương pháp phân lớp dữ liệu hình ảnh theo mô hình dạng cây gọi là cấu trúc CKD-Tree, đồng thời thực hiện tìm kiếm tập ảnh tương tự dựa trên cấu trúc đã xây dựng. Cấu trúc CKD-Tree đề xuất gồm các véc-tơ trọng số được lưu trữ tại các nút trong và dữ liệu hình ảnh được lưu trữ tại nút lá tạo thành các phân cụm. Cấu trúc CKD-Tree là một dạng phân lớp dữ liệu đa tầng nhằm phân lớp nhiều lần cho đối tượng theo mô hình Deep Learning, tại mỗi nhánh trên CKD-Tree thực hiện một lần phân lớp. Việc phân lớp này dùng để phân cụm dữ liệu và có thể phát hiện được các cụm

lân cận trong tìm kiếm tập ảnh tương tự. Điểm mới trong bài báo này là xây dựng cấu trúc CKD-Tree theo phương pháp phân lớp. Nút gốc và các nút trong chỉ lưu một véc-tơ trọng số ban đầu là ngẫu nhiên, sau đó được huấn luyện nhằm thực hiện quá trình phân lớp nhiều lần cho đối tượng tại các tầng trên cây. Nút lá lưu trữ tập ảnh tương tự gọi là các cụm tương đồng. Trong khi đó cấu trúc KD-Tree được công bố bởi các công trình trước đây dữ liệu được lưu tại tất cả các nút trên cây hoặc chưa thực hiện phân lớp cho đối tượng nhiều lần. Đồng thời, quá trình huấn luyện trọng số tại các nút trong theo phương pháp Gradient nhằm giảm sai số phân lớp hình ảnh được thực hiện song song với quá trình xây dựng cấu trúc CKD-Tree.

III. PHƯƠNG PHÁP XÂY DỰNG CẤU TRÚC CKD-TREE

1. Mô tả cấu trúc CKD-Tree

Cấu trúc CKD-Tree được xây dựng dựa trên cây KD-Tree nguyên thủy [1] là một cây nhị phân, cân bằng bao gồm: Một nút gốc (*Root*), tập nút trong $\{iNode\}$ và tập nút lá $\{LNode\}$. Ban đầu, cấu trúc CKD-Tree được tạo ra với một khung cây, nút gốc và nút trong lưu trữ véc-tơ trọng số, nút lá lưu trữ dữ liệu hình ảnh với bộ dữ liệu huấn luyện (70% bộ ảnh thực nghiệm COREL, Wang). Một nút không phải là nút lá chia cây CKD-Tree thành hai phần gọi là cây con trái và cây con phải. Mỗi nút trong ($iNode_i$) lưu trữ một véc-tơ trọng số n chiều $w_i = (x_{i1}, x_{i2}, \dots, x_{in})$. Giá trị đầu ra y_i tại mỗi $iNode_i$ khi truyền vào một véc-tơ $f_j = (x_{j1}, x_{j2}, \dots, x_{jn})$ được xác định bởi công thức (1). Trong đó, hàm truyền $Sigmoid(x)$ được sử dụng nhằm giới hạn miền giá trị đầu ra tại mỗi $iNode_i$ xác định bởi công thức (2); Vì CKD-Tree là cây nhị phân, cân bằng nên tại mỗi nút trong của CKD-Tree giá trị đầu ra được làm cơ sở để xác định đường đi tiếp theo thuộc một trong hai nhánh con trái hoặc phải. Do đó hàm dấu $Sign(y_i)$ được áp dụng ngay tại mỗi $iNode_i$ xác định bởi công thức (3). Đồng thời tại mỗi nút trong $iNode_i$ được ràng buộc bởi một ngưỡng t là điều kiện xác định số nhánh trên CKD-Tree.

$$y_i = Sign(Sigmoid(w_i * f_j - t) - t) \quad (1)$$

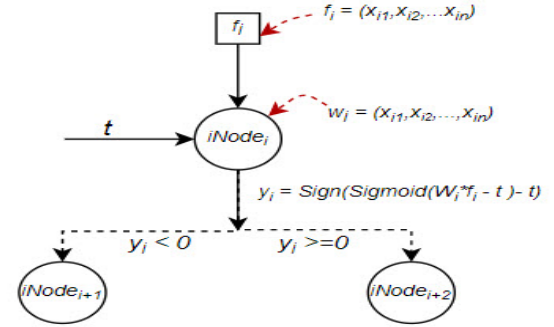
$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

$$Sign(y_i) = \begin{cases} 1, & y_i \geq 0 \\ -1, & y_i < 0 \end{cases} \quad (3)$$

Trong quá trình tạo cây CKD-Tree, hàm $Sigmoid(x)$ được sử dụng nhằm xác định giá trị đầu ra tại mỗi $iNode_i$, sau đó thêm ngưỡng t bằng 0.5 là hằng số giúp tạo cây nhị phân, cân bằng trong miền giá trị $[0..1]$. Bên cạnh đó, cấu trúc KD-Tree trong công trình [18] xây dựng theo phương pháp phân cụm, trong đó nút cha lưu giá trị trung bình trên mỗi

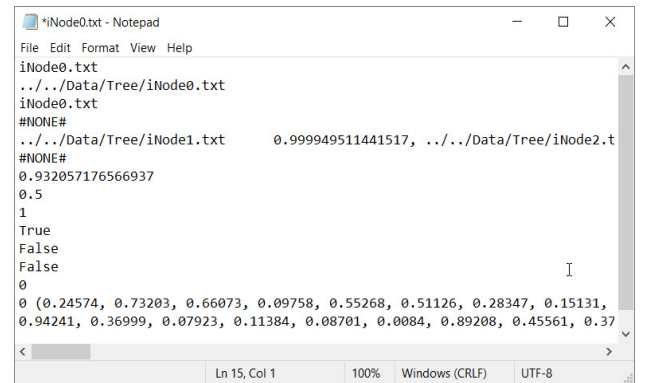
nhánh, điều này dẫn đến hiệu suất gom cụm trên cây chưa cao, đó là điểm tồn tại của công trình này và phương pháp xây dựng CKD-Tree là một cải tiến hiệu quả.

Xét hàm đầu theo công thức (3) nếu $y_i \geq 0$ thì f_j qua nhánh phải; ngược lại f_j qua nhánh trái; xác định đường đi cho véc-tơ f_j tại $iNode_i$ nhằm tìm nhánh con thuộc tầng kế tiếp trên cây, quá trình này được mô tả như Hình 1.



Hình 1. Mô tả xác định nhánh con tại $iNode_i$

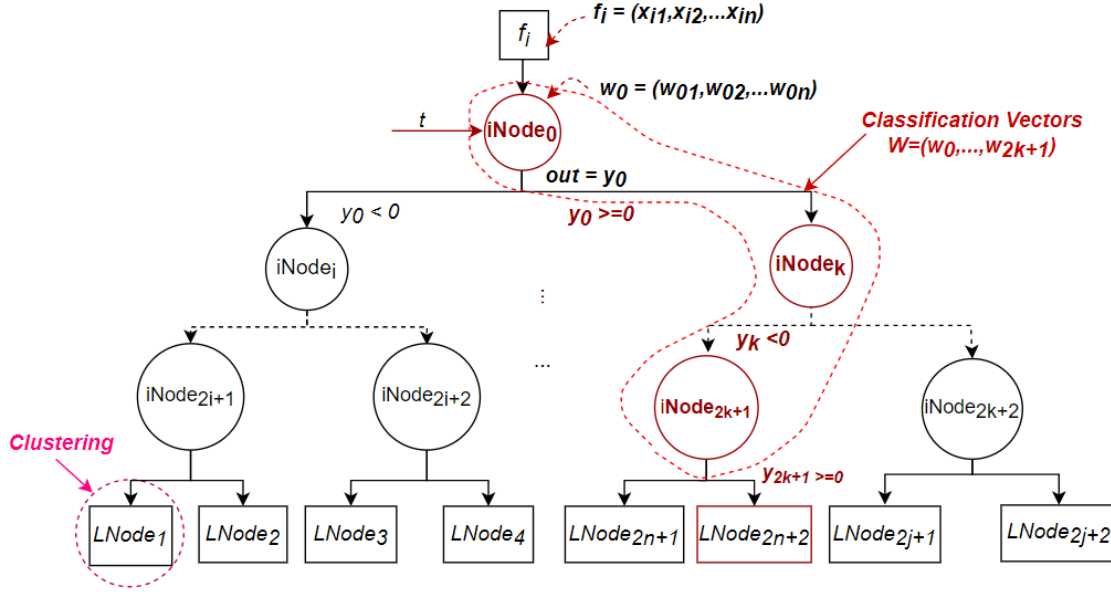
Vì giá trị đầu ra tại các $iNode_i$ thuộc miền $[0..1]$ nên công thức (1) phải giới hạn hai lần giá trị t để miền giá trị của hàm đầu tại công thức (3) phân biệt giá trị âm, dương làm cơ sở xét f_j theo nhánh con trái hoặc con phải tại các nút trong. Hàm truyền $Sigmoid(x) = \sigma(x)$ có một số tính chất: 1) Miền giá trị là $[0..1]$; 2) Hàm $Sigmoid(x)$ tăng liên tục trên miền $[0..1]$; 3) Giá trị đạo hàm tồn tại với mọi điểm thuộc miền $[0..1]$ giúp giảm chi phí huấn luyện trọng số.



Hình 2. Minh họa véc-tơ trọng số tại $iNode_i$

Véc-tơ $w_i = (x_{i1}, x_{i2}, \dots, x_{in})$ là trọng số được lưu trữ tại $iNode_i$ được minh họa như Hình 2. Các thành phần tại một nút và mối quan hệ giữa các nút trên cây CKD-Tree được mô tả như sau:

1) *Root* là nút không có nút cha, lưu trữ một véc-tơ trọng số (w_0), có hai nút con trái (*left*), phải (*right*) và có một mức trên cây (*level*): $Root = \langle w_0, left, right, level \rangle$



Hình 3. Minh họa quá trình phân lớp trên CKD-Tree

2) $iNode_i$ là nút trong có một nút cha (*parent*); lưu trữ véc-tơ trọng số (w_i), hai nút con trái (*left*), phải (*right*) và có một mức (*level*) : $iNode_i = \langle parent, w_i, left, right, level \rangle$

3) $LNode_i$ là nút lá có một nút cha (*parent*), không có nút con; lưu trữ tập véc-tơ đặc trưng hình ảnh $f_1, f_2, \dots, f_k$; có một mức (*level*) và mỗi nút lá $LNode_i$ được gán một nhãn *label* : $LNode_i = \langle parent, f_1, \dots, f_k, level, label \rangle$

4) Hai nút $iNode_i$ và $iNode_j$ gọi là hai nút anh em nếu chúng có cùng một nút cha: $iNode_i.parent = iNode_j.parent$

5) Hai nút $iNode_i$ và $iNode_j$ gọi là cha con nếu $iNode_j.parent = iNode_i$ hoặc $iNode_i.parent = iNode_j$

6) Hai nút $iNode_i$ và $iNode_j$ gọi là đồng cấp nếu $iNode_i.level = iNode_j.level$

2. Nguyên tắc xây dựng cấu trúc CKD-Tree

Trên cơ sở các thành phần của CKD-Tree, quá trình xây dựng được thực hiện theo các nguyên tắc và các ràng buộc sau:

- Khởi tạo chiều cao cây bằng h thì số nút gốc $NumRoot = 1$; số nút trong $Num_iNode = 2^1 + \dots + 2^{h-1}$, số nút lá tối đa trên cây là $NumLNode = 2^h$
- Tại mỗi $iNode_i$ lưu trữ véc-tơ trọng số khởi tạo $iNode.w_i = (w_{i1}, w_{i2}, \dots, w_{in})$. Ban đầu các véc-tơ trọng số w_i được khởi tạo ngẫu nhiên, sau đó được huấn luyện theo phương pháp giảm sai số trung bình tỷ lệ phân lớp tại các nút lá. Quá trình tạo cây cho các bộ dữ liệu ảnh khác

nhau thì bộ trọng số này sẽ khác nhau.

c) Phần tử f_j được thêm vào CKD-Tree bằng cách duyệt từ nút gốc đến các tầng nút trong để tìm vị trí và lưu trữ f_j tại $LNode_k$. Tại mỗi nút $iNode_i$ trên cây, xác định giá trị đầu ra y_i bởi công thức (1)

d) Xác định đường đi cho véc-tơ f_j tại mỗi $iNode_i$: Nếu giá trị $y_i \geq 0$ đường đi của véc-tơ f_j qua nhánh phải; ngược lại đường đi của véc-tơ f_j qua nhánh trái.

e) Các bước (3) và (4) lặp lại đến tầng cuối cùng kiểm tra đúng nút lá thì ghi f_j vào $LNode_k$.

Dựa trên mô tả các thành phần và nguyên tắc xây dựng, cấu trúc CKD-Tree được minh họa như Hình 3, trong đó *Classification Vectors* là tập véc-tơ phân lớp $W = \langle w_0, w_2, \dots, w_m \rangle$, *Clustering* là một cụm nút lá, $out = y_i$ là giá trị đầu ra tại các $iNode_i$. Quá trình phân lớp cho một đối tượng được thực hiện theo nhiều tầng tại các nút trong ($iNode_i$) của CKD-Tree và cuối cùng thực hiện gom cụm tại các nút lá trên CKD-Tree, mỗi nút lá là một cụm chứa các ảnh tương tự.

3. Thuật toán xây dựng CKD-Tree

Thuật toán xây dựng cấu trúc CKD-Tree – CKDT

Để xây dựng cấu trúc CKD-Tree, ban đầu khởi tạo bộ trọng số ngẫu nhiên $W_{kt} = W_{kt0}, W_{kt1}, \dots, W_{ktP} : W_{kti} = (w_{i0}, w_{i1}, \dots, w_{ik}) ; i = 0, \dots, n$. Tập véc-tơ $F = f_i : f_i = (x_{i0}, x_{i1}, \dots, x_{ik}) ; i = 0, \dots, n$ của tập dữ liệu ảnh đã được trích xuất như được mô tả Phần V.1, mỗi véc-tơ đặc trưng gồm 81 chiều, giá trị ngưỡng t bằng 0,5 là hằng số không đổi theo trong công thức (1).

Thuật toán 1: Thuật toán CKDT

```

1 Đầu vào: Tập vector  $F = f_i : f_i = (x_{i0}, x_{i1}, \dots, x_{ik}); i = 0, \dots, n$ 
2 Bộ trọng số khởi tạo  $W_{kt} = W_{kt0}, W_{kt1}, \dots, W_{ktP} : W_{kti} = (w_{i0}, w_{i1}, \dots, w_{ik}); i = 0, \dots, n$ 
3 Đầu ra: Cây CKD-Tree
4 Function CKDT ( $F, W_{kt}, t$ )
5 begin
6   int  $h$ ; CKD-Tree =  $\emptyset$ ; NumiNode = 0;
7   MaxNumLNode =  $2^h$ ;
8   for  $i = 0$  to  $h - 1$  do
9     | NumiNode = NumiNode +  $2^i$ ;
10  end
11  for  $j = 0$  to NumiNode do
12    |  $iNode_j = \emptyset$ ;
13  end
14  for  $k = NumiNode + 1$  to  $2^h - 2$  do
15    |  $LNode_k = \emptyset$ ;
16  end
17  foreach ( $W_{kti}$ ) do
18    |  $iNode_i.w_i = w_{kti}$ ;
19  end
20  foreach ( $f_i \in F$ ) do
21    foreach ( $iNode_i$ ) do
22      |  $T = \text{Sigmoid}(w_{kti} * f_j + t)$ ;
23      if ( $\text{Sign}(T + t) \geq 0$ ) then
24        | CKD - Tree = CKD - Tree  $\cup$ 
25        | CKDT( $f_j$ , CKDT.right,  $t$ );
26      end
27      else
28        | CKD - Tree = CKD - Tree  $\cup$ 
29        | CKDT( $f_j$ , CKDT.left,  $t$ );
30      end
31    end
32    if  $LNode_i.leaf = \text{true}$  then
33      | Insert( $f_j$ ,  $LNode_k$ );
34    end
35  end
36 return CKD-Tree;

```

Mệnh đề 1: Độ phức tạp của thuật toán **CKDT** là $O(n * h)$, n là số véc-tơ xây dựng cây và h là chiều cao cây CKD-Tree.

Chứng minh: Với mỗi véc-tơ thuộc tập n phần tử cần thêm vào cây CKD-Tree, thuật toán **CKDT** lần lượt duyệt qua tất cả các tầng từ gốc đến lá với chiều cao h . Do đó, độ phức tạp của Thuật toán **CKDT** là $O(n * h)$

4. Thuật toán gán nhãn nút lá

Nút lá $LNode_i$ lưu tập véc-tơ đặc trưng thuộc nhiều phân lớp. Vì vậy mỗi nút lá $LNode_i$ cần được gán một nhãn (label) theo số phần tử thuộc phân lớp nhiều nhất.

Bước 1: Mỗi nút lá $LNode_i$ đếm tần số xuất hiện của tất cả các nhãn trong bộ ảnh. Nhãn nào không xuất hiện trong $LNode_i$ thì gán tần số xuất hiện bằng 0.

Bước 2: Biểu diễn tần số xuất hiện các nhãn $Label_i$

Thuật toán 2: Thuật toán gán nhãn - SL2L.

```

1 Đầu vào: Cây CKD-Tree, ListLabels, ListLNode
2 Đầu ra: Tập nút lá LNode đã gán nhãn {ListLNodeAddedLabel}
3 Function SL2L (CKD-Tree, ListLabels, ListLNode)
4 begin
5   ListLNodeAddedLabel = null;
6   foreach  $LNode_i$  in ListLNode do
7     foreach  $Label_j$  in ListLabels do
8       |  $TS = \text{Count}(Label_j, LNode_i)$ ;
9     end
10  end
11  for  $i = 0$  to  $n$  do
12    for  $j = 0$  to  $m$  do
13      |  $MT(n, m) =$ 
14      | CreateMatrix(NumberLabel, NumberLNode);
15    end
16    for  $i = 0$  to  $n$  do
17      for  $j = 0$  to  $m$  do
18        repeat
19          SearchMax =  $VT_{i,j}$ ;
20          MaxLabel = Value ( $VT_{i,j}$ );
21           $LNode_j.Label =$ 
22          SetLabel( $LNode_j$ ,  $Label_i$ , MaxLabel);
23          ListLNodeNotAdd =
24          ListLNode - { $LNode_j$ };
25          ListLabelNotAdd = ListLabel - { $Label_j$ };
26          foreach ( $Label_k$  in  $LNode_k$ ) do
27            |  $VT(k, j) = 0$ ;
28          end
29          foreach ( $LNode_k$  in ListLNodeNotAdd) do
30            |  $VT(i, k) = 0$ ;
31          end
32          until ( $VT_{i,j} = 0$ )
33        end
34      end
35    end
36  return ListLNodeAddedLabel

```

theo từng nút lá $LNode_j$ bởi ma trận $MT(n, m) : \{n = \text{NumberLabel}; m = \text{NumberLNode}\}$

Bước 3: Tìm giá trị lớn nhất của ma trận $MT(n, m)$ là $VT_{i,j}$ tại dòng $Label_i$ và cột $LNode_j$. Gán nhãn $Label_i$ cho nút lá $LNode_j$ là giá trị $LNode_j.Label = \text{Value}(VT_{i,j})$

Bước 4: Chuyển các giá trị thuộc dòng i và cột j thành 0

Bước 5: Lặp lại **Bước 3** và **Bước 4** cho đến khi gán hết tập các nhãn $Label_i$ cho tập nút lá $LNode_j$

Thuật toán gán nhãn nút lá - SL2L: Để thực hiện gán nhãn cho nút lá tối ưu nhất, kỹ thuật dùng ma trận lưu trữ tần số xuất hiện của ảnh theo label tại các nút lá.

Mệnh đề 2: Độ phức tạp của thuật toán SL2L là $O(l * k)$ với l là số nhãn của tập dữ liệu ảnh, k số nút lá.

Chứng minh: Thuật toán SL2L duyệt qua k nút lá. Mỗi nhãn l trong tập dữ liệu cần được duyệt để gán cho các lá. Do đó, độ phức tạp của Thuật toán SL2L là $O(l * k)$.

5. Huấn luyện trọng số

Sau khi xây dựng cây CKD-Tree ban đầu với bộ trọng số ngẫu nhiên thì hiệu suất phân lớp ảnh chưa cao, vì vậy cần huấn luyện bộ trọng số tại các nút trong nhằm tăng hiệu suất phân lớp cho một ảnh đầu vào. Quá trình huấn luyện trọng số theo mô hình cây CKD-Tree sử dụng đạo hàm tại mỗi nút trong của cây. Công thức đạo hàm của hàm Sigmoid(x) xác định tại công thức (4).

$$\begin{aligned} \text{Sigmoid}'(x) &= \frac{e^{-x}}{(1 + e^{-x})^2} \\ &= \frac{1}{1 + e^{-x}} \frac{e^{-x}}{1 + e^{-x}} \\ &= \sigma(x)(1 - \sigma(x)) \end{aligned} \quad (4)$$

$W_i = (w_{i1}, w_{i2}, \dots, w_{in})$ là véc-tơ trọng số tại $iNode_i$, thực hiện cập nhật giá trị véc-tơ trọng số bằng hàm $\sigma(x) = \sigma(W_i * f_i - t)$ và $\eta = const$ theo công thức (5).

$$\begin{aligned} W_i &= W_i - \text{Sign}(\sigma(W_i * f_i - t) - t) * \eta * \sigma(W_i * f_i - t) \\ &\quad * (1 - \sigma(W_i * f_i - t)) \end{aligned} \quad (5)$$

Sau khi thực hiện gán nhãn cho tất cả các nút lá, tính tổng sai số phân lớp theo công thức (6).

$$P_1 = AVG\left(\frac{\sum(f_{ki})}{\sum(f_{mi})}\right) \quad (6)$$

Trong đó:

$\sum(f_{ki})$ là số véc-tơ sai nhãn thuộc nút $LNode_i$, $\sum(f_{mi})$ là số véc-tơ thuộc nút $LNode_i$. Mỗi lần cập nhật véc-tơ trọng số theo công thức (5), tính tổng sai số phân lớp theo công thức (6). Nếu $P_2 < P_1$ thì chọn phương án cập nhật véc-tơ trọng số mới; ngược lại giữ nguyên trọng số và chọn véc-tơ khác để thực hiện điều chỉnh trọng số. Kết quả huấn luyện bộ trọng số với bộ ảnh COREL, chiều cao cây bằng 4 minh họa bởi Hình 4.

Hình 4. Bộ véc-tơ trọng số đã huấn luyện trên tập ảnh COREL

Để tối ưu quá trình huấn luyện trọng số, cần theo một số quy tắc sau:

Quy tắc 1: Ưu tiên chọn véc-tơ có vị trí sai gần tầng lá nhất để điều chỉnh.

Quy tắc 2: Muốn điều chỉnh một véc-tơ từ f_j từ nhánh trái sang nhánh phải (hoặc ngược lại) tại vị trí thì chọn véc-tơ có giá trị $|\text{Sigmoid}(W_i * f_j - t)|$ nhỏ nhất trong số các véc-tơ sai đường đi tại vị trí này để thực hiện điều chỉnh.

IV. MÔ HÌNH TÌM KIẾM ẢNH DỰA TRÊN CẤU TRÚC CKD-TREE

1. Mô hình truy vấn ảnh

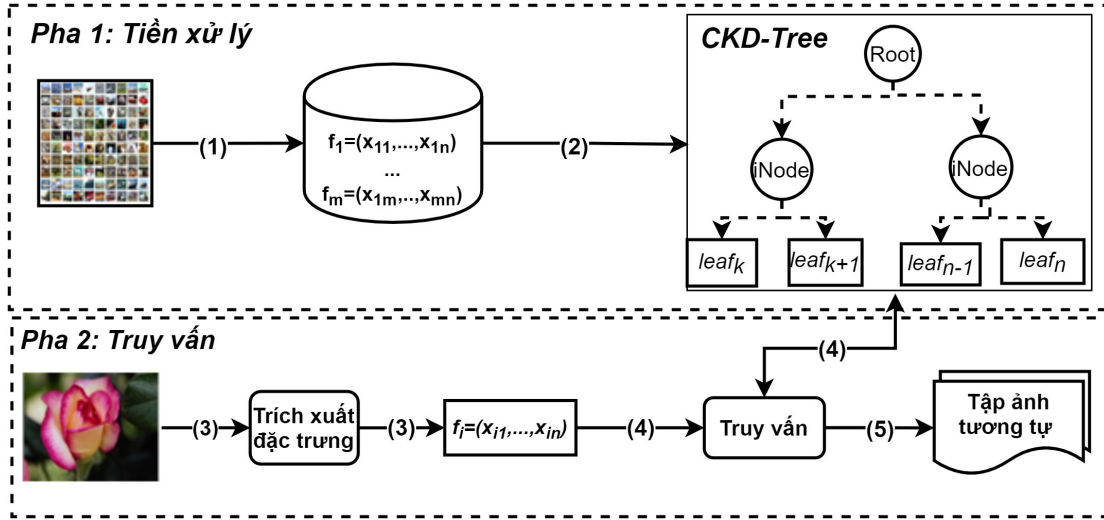
Mô hình truy vấn ảnh dựa trên cấu trúc CKD-Tree được mô tả trong Hình 5, gọi là CBIR-CKDT (*Content Based Image Retrieval - Classification KD-Tree*). Hệ truy vấn gồm hai pha: 1) Pha tiền xử lý trích xuất đặc trưng bộ ảnh và xây dựng cấu trúc CKD-Tree; 2) Pha truy vấn thực hiện trích xuất đặc trưng ảnh đầu vào, sau đó thực hiện tìm kiếm trên cấu trúc CKD-Tree để trích xuất tập ảnh tương tự.

- **Pha tiền xử lý:** Xây dựng cấu trúc CKD-Tree
 - a) Trích xuất véc-tơ đặc trưng của tập dữ liệu ảnh tạo cơ sở dữ liệu ban đầu, việc trích xuất đặc trưng dựa trên các đặc trưng cấp thấp của ảnh như màu sắc, diện tích, vị trí, bề mặt, đường biên của đối tượng ảnh, v.v.. Kết quả của quá trình trích xuất đặc trưng ảnh là tập các véc-tơ đặc trưng có 81 chiều và nhãn lớp.
 - b) Xây dựng cấu trúc CKD-Tree nhằm lưu trữ dữ liệu hình ảnh tại nút lá. Phân lớp dữ liệu tại các nút trong.
- **Pha truy vấn:** Thực hiện tìm kiếm tập ảnh tương tự
 - c) Trích xuất véc-tơ đặc trưng ảnh cần truy vấn.
 - d) Truy vấn trên cây CKD-Tree để tìm tập ảnh tương tự bằng thuật toán tìm kiếm **SCKDT**.
 - e) Kết xuất tập ảnh tương tự với ảnh truy vấn là tập ảnh thuộc nút lá chứa ảnh đầu vào.

2. Thuật toán truy vấn ảnh dựa trên cấu trúc CKD-Tree

Từ một ảnh đầu vào I_i sau khi thực hiện trích xuất véc-tơ đặc trưng f_i , quá trình thực hiện tìm kiếm tập ảnh tương tự của ảnh I_i theo các bước sau: 1) Duyệt từ nút gốc cho véc-tơ f_i đi qua nút trong tại mỗi tầng của cây CKD-Tree, tính giá trị đầu ra y_i theo công thức (3) tại mỗi $iNode_i$; 2) kiểm tra nếu $y_i \geq 0$ thì thực hiện tìm kiếm f_i qua nhánh phải của $iNode_i$; 3) ngược lại $y_i < 0$ thì thực hiện tìm kiếm f_i qua nhánh trái của $iNode_i$; 4) Khi duyệt hết các tầng nút trong, kiểm tra đến nút lá và trích xuất tập ảnh tương tự chính là tập ảnh lưu trữ tại nút lá chứa f_i .

Mệnh đề 3: Độ phức tạp của thuật toán **SCKDT** là $O(h)$; trong đó h là chiều cao của cây CKD-Tree.



Hình 5. Mô hình tìm kiếm ảnh dựa trên cấu trúc CKD-Tree

Thuật toán 3: Thuật toán truy vấn ảnh_SCKDT.

```

1 Đầu vào: Véc-tơ đặc trưng  $f_i$  của ảnh  $I_i$ , CKD - Tree
2 Đầu ra: Tập ảnh tương tự SI của ảnh  $I_i$ 
3 Function SCKDT ( $f_i$ , CKD-Tree)
4 begin
5   Result =  $\phi$ ; double t = 0.5
6   foreach  $f_i$  do
7     foreach  $iNode_i$  do
8        $S = \text{Sign}(\text{Sigmoid}(\text{Product}(iNode_i.w_i, f_i) - t) - t)$ 
9       if  $S \geq 0$  then
10        | SCKDT  $f_i$ , CKD - Tree.right ;
11      end
12      else
13        | SCKDT  $f_i$ , CKD - Tree.left ;
14      end
15    end
16  end
17  if  $Lnode_i.leaf = \text{true}$  then
18    | Result = Result  $\cup$   $LNode_i.f_i$  ;
19  end
20
21  return Result;
22 end

```

Chứng minh: Thuật toán SCKDT duyệt qua các nút từ gốc đến lá, cây CKD-Tree là cây cân bằng nên thuật toán SCKDT duyệt qua chiều cao h của cây. Do đó, độ phức tạp của thuật toán SCKDT là $O(h)$

V. THỰC NGHIỆM

1. Trích xuất đặc trưng hình ảnh

Trong bài báo này, chúng tôi tiến hành thực nghiệm với bộ dữ liệu ảnh COREL, Wang. Đầu tiên, các hình ảnh được trích xuất thành véc-tơ đặc trưng 81 chiều làm cơ sở xây dựng cấu trúc CKD-Tree cũng như làm dữ liệu đầu vào cho

ảnh truy vấn. Các đặc trưng cấp thấp chúng tôi sử dụng gồm màu sắc, hình dạng, kết cấu, v.v.. Để trích xuất đặc trưng hình ảnh, đầu tiên chúng tôi thực hiện phân đoạn ảnh nhằm chia một ảnh thành nhiều vùng nhỏ giúp dễ trích xuất đặc trưng. Sau đó các phương pháp trích xuất đặc trưng màu sắc, hình dạng, kết cấu [25] được thực hiện.

Phân đoạn ảnh (Image Segmentation): Phân đoạn ảnh là công việc đầu tiên trong trích xuất đặc trưng. Việc phân đoạn ảnh màu được thực hiện bằng cách phân chia hình ảnh thành các vùng riêng biệt nhau để từ đó trích xuất đặc trưng trên mỗi vùng. Để thực hiện được điều này, chúng tôi đề xuất phương pháp phân vùng dựa trên độ tương phản, nghĩa là vùng nào có độ tương phản thấp là hình nền, vùng nào có độ tương phản cao là hình đối tượng. Tuy nhiên, việc phân biệt hình nền và hình đối tượng sẽ bị nhập nhằng trong một số hình ảnh có xu hướng ngược lại nên chúng tôi sử dụng hai loại ảnh đối xứng nhau giữa hình nền và hình đối tượng. Đồng thời, để làm giảm độ nhiễu giữa các vùng quá sáng hoặc quá tối, một số điểm ảnh nằm trong vùng lân cận của giá trị lớn nhất và giá trị nhỏ nhất của độ tương phản thì được quy về giá trị tương đương [25].

Đặc trưng màu sắc (Color): Màu sắc là một trong những đặc điểm quan trọng nhất trong trích xuất đặc trưng cấp thấp hình ảnh. Màu sắc có mối quan hệ chặt chẽ với các ảnh đối tượng ảnh và ảnh nền. Đặc điểm của màu sắc là không thay đổi đối với kích thước và hướng của đối tượng [23]. Do đó, màu sắc dễ dàng phân tích và trích xuất bằng cách sử dụng mô men màu (Color Moment), biểu đồ màu (Color Histogram), không gian màu (Color Space), v.v.. Trong bài báo này, chúng tôi thực hiện trích xuất đặc trưng màu sắc theo MPEG-7 gồm 25 thuộc tính.

Đặc trưng hình dạng (Shape): Hình dạng đối tượng là một trong những đặc trưng cơ bản trong đặc trưng cấp thấp

của hình ảnh. Đặc trưng hình dạng được sử dụng để phát hiện các đối tượng tương tự từ cơ sở dữ liệu mà không phụ thuộc vào vị trí, góc quay hay sự co giãn của đối tượng ảnh [22, 24]. Các phương pháp trích xuất đặc trưng hình dạng thường được chia thành hai loại: dựa theo đường biên và dựa theo vùng ảnh. Kỹ thuật xác định đường biên đối tượng có thể dùng Gradient và phương pháp Laplacian. Vì vậy, để trích xuất các đặc trưng theo hình dạng, trước hết các đối tượng ảnh cần phải được phân đoạn thành những thành phần có cùng tính chất tương đồng dựa trên đường biên hay các vùng lân cận. Sau đó sử dụng các kỹ thuật phát hiện đường biên ảnh để xác định hình dạng cho đối tượng đã được tách ra khỏi hình nền [21]. Trong bài báo này, chúng tôi sử dụng phương pháp phát hiện biên đối tượng bằng phép biến đổi LoG (*Laplacian of Gaussian*) [19] cho ảnh màu. Phép biến đổi LoG bất biến đối với sự biến đổi cường độ ảnh cũng như sự biến đổi tỉ lệ, phép quay, phép biến đổi affine. Vì vậy, giá trị Gaussian được xác định bởi công thức (7).

$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \times \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (7)$$

Với σ là đạo hàm chuẩn, biểu diễn không gian tỷ lệ Gaussian $L(x, y, \sigma)$ của ảnh $f(x, y)$ theo công thức (8).

$$L(x, y, \sigma) = f(x, y) * G(x, y, \sigma) \quad (8)$$

Trong công thức (8), phép toán $*$ là phép tích chập (*convolution*); (x, y) là tọa độ điểm ảnh. Toán tử Laplacian ∇^2 được tính toán theo công thức (9).

$$\nabla^2 = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (9)$$

Toán tử LoG được tính toán đầu tiên và sau đó được đối sánh với ảnh để tạo ra biểu diễn không gian tỷ lệ LoG theo công thức (10).

$$\nabla^2 G(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{\pi\sigma^4} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (10)$$

Phương pháp LoG nhằm xác định đường biên, đặc trưng của đối tượng được trích xuất gồm chu vi đối tượng, vị trí tương đối của các đường viền.

Ngoài ra, để nhận dạng đối tượng dựa trên biên và làm mịn bề mặt, phép lọc Sobel [20, 21] được sử dụng. Phép lọc Sobel là toán tử Gradient dựa trên vùng láng giềng 3x3. Mặt nạ tích chập cho toán tử Sobel trên ảnh số tỷ lệ xám $Sobel_x$ và $Sobel_y$ được xác định trong công thức (11).

$$Sobel_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad Sobel_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (11)$$

Các vị trí trên hai mặt nạ được áp dụng riêng biệt trên ảnh đầu vào để tạo hai thành phần Gradient G_x và G_y theo hướng ngang và dọc tương ứng theo công thức (12), (13).

$$G_x = \sum_{i=1}^3 \sum_{j=1}^3 Sobel_{x_{i,j}} \cdot f_{x+1-2, y+j-2} \quad (12)$$

$$G_y = \sum_{i=1}^3 \sum_{j=1}^3 Sobel_{y_{i,j}} \cdot f_{x+1-2, y+j-2} \quad (13)$$

Lúc này, vị trí của ảnh đầu vào f được thể hiện trong công thức (14), trong đó trục là các vị trí nằm ngang và trục là các vị trí thẳng đứng.

$$\begin{bmatrix} (x-1, y-1) & (x-1, y) & (x-1, y+1) \\ (x, y-1) & (x, y) & (x, y+1) \\ (x+1, y+1) & (x+1, y) & (x+1, y+1) \end{bmatrix} \quad (14)$$

Độ lớn Gradient được tính bởi công thức (15).

$$G[f(x, y)] = \sqrt{G_x^2 + G_y^2} \quad (15)$$

Bảng I
CÁC GIÁ TRỊ TRÍCH XUẤT VEC-TƠ ĐẶC TRƯNG CỦA HÌNH ẢNH

Mô tả đặc trưng	Số giá trị
Đặc trưng màu sắc theo MPEG-7	25
Phép lọc tần số cao để lấy ảnh đường nét	9
Phép lọc Gaussian để nâng cao cường độ ảnh	9
Đặc trưng cường độ các điểm ảnh theo láng giềng	9
Đặc trưng cường độ của đối tượng	9
Đặc trưng cường độ của hình nền	9
Đặc trưng diện tích đối tượng	1
Đặc trưng hình dạng của đường biên ảnh	1
Đặc trưng vị trí tương đối của đối tượng theo trục X, Y	2
Đặc trưng vị trí tương đối của hình nền theo trục X, Y	2
Đặc trưng chu vi của đối tượng	1
Đặc trưng chu vi của đối tượng theo phép lọc Sobel	1
Đặc trưng cường độ các điểm ảnh theo láng giềng dựa theo phép lọc Sobel	1
Đặc trưng chu vi của đối tượng theo phép lọc Laplacian	1
Đặc trưng đường nét ảnh theo phép lọc Laplacian	1

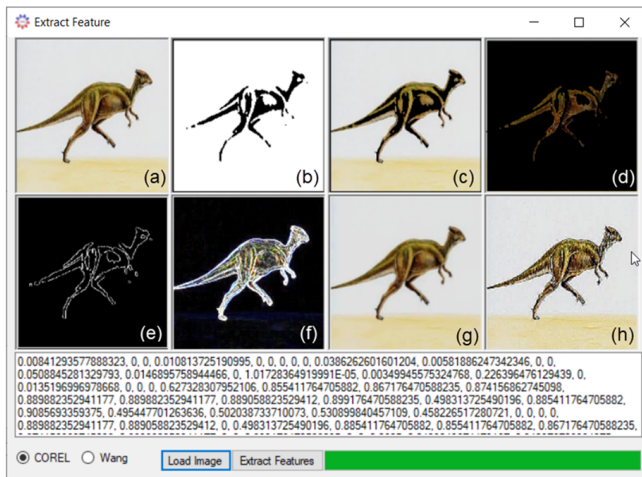
Để xác định vị trí tương đối của đối tượng theo trục X và trục Y, hàm khoảng cách trọng tâm CDF (*Centroid Distance Function*) được sử dụng. CDF tính toán khoảng cách từ tâm (x_0, y_0) đến các điểm đường viền của một hình dạng và được biểu diễn bằng công thức (16).

$$r(n) = \sqrt{(x(n) - x_0)^2 + (y(n) - y_0)^2} \quad (16)$$

Từ cơ sở lý thuyết, mỗi ảnh được trích xuất thành vec-tơ đặc trưng 81 chiều với các thành phần và số giá trị được mô tả trong Bảng I.

2. Môi trường xây dựng thực nghiệm

Thực nghiệm trích xuất đặc trưng và hệ truy vấn CBIR-CKDT 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: Intel(R) Core™ i5-5200U, CPU 2.2GHz, RAM 16GB và hệ điều hành Windows 10 Professional. Chúng tôi tiến hành thực nghiệm trích xuất véc-tơ đặc trưng cho bộ dữ liệu COREL và bộ ảnh Wang được minh họa bởi Hình 6. Quá trình xây dựng cây CKD-Tree được minh họa ở Hình 7. Cây CKD-Tree được xây dựng (*Create CKD-Tree*) xác định bởi chiều cao (*Height of CKD-Tree*), số nhánh (*Number of Brand*) và một ngưỡng (*Bias*). Hình 8 là giao diện tìm kiếm ảnh tương tự cho ảnh 450.jpg bộ COREL, Hình 9 là kết quả trích xuất tập ảnh tương tự của ảnh 450.jpg đầu vào.

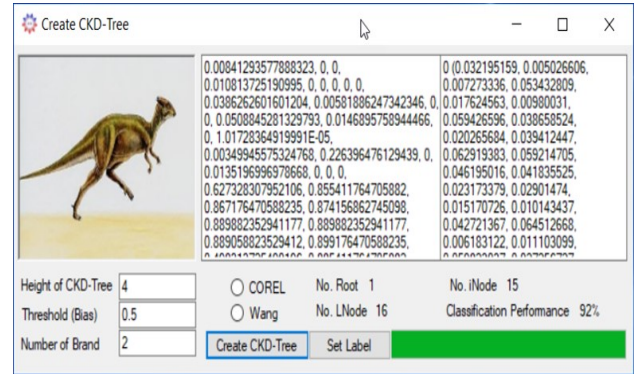


Hình 6. Véc-tơ đặc trưng ảnh 450.jpg bộ COREL

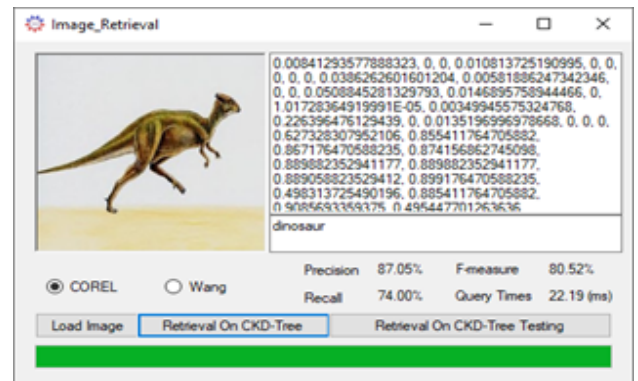
Kết quả trích xuất véc-tơ đặc trưng của ảnh 405.jpg bộ ảnh COREL được thể hiện bằng 8 ảnh được mô tả trong Hình 6 gồm: (a) ảnh gốc; (b) ảnh mặt nạ đối tượng (*ForeGround*); (c) ảnh đối tượng; (d) hình nền ảnh gốc; (e) ảnh đường biên đối tượng theo phép lọc Laplace cho ảnh mặt nạ đối tượng; (f) ảnh đường biên đối tượng và vân ảnh bề mặt đối tượng theo phép lọc Sobel; (g) Ảnh đường nét của ảnh gốc theo phép lọc Laplace và Gaussian; (h) ảnh phép lọc thông cao cho ảnh gốc.

Bảng II
HIỆU SUẤT TÌM KIẾM CỦA HỆ
TRUY VẤN ẢNH CBIR-CKDT

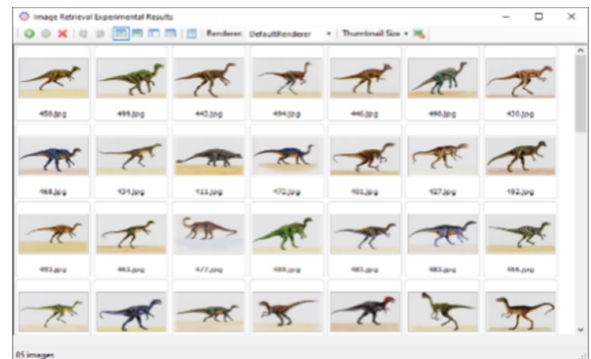
Tập ảnh	Độ chính xác trung bình	Độ phủ trung bình	Độ dung hòa trung bình	Thời gian truy vấn trung bình (ms)
COREL	0,7640	0,6878	0,7188	38
Wang	0,7327	0,6508	0,6869	87



Hình 7. Xây dựng cấu trúc CKD-Tree

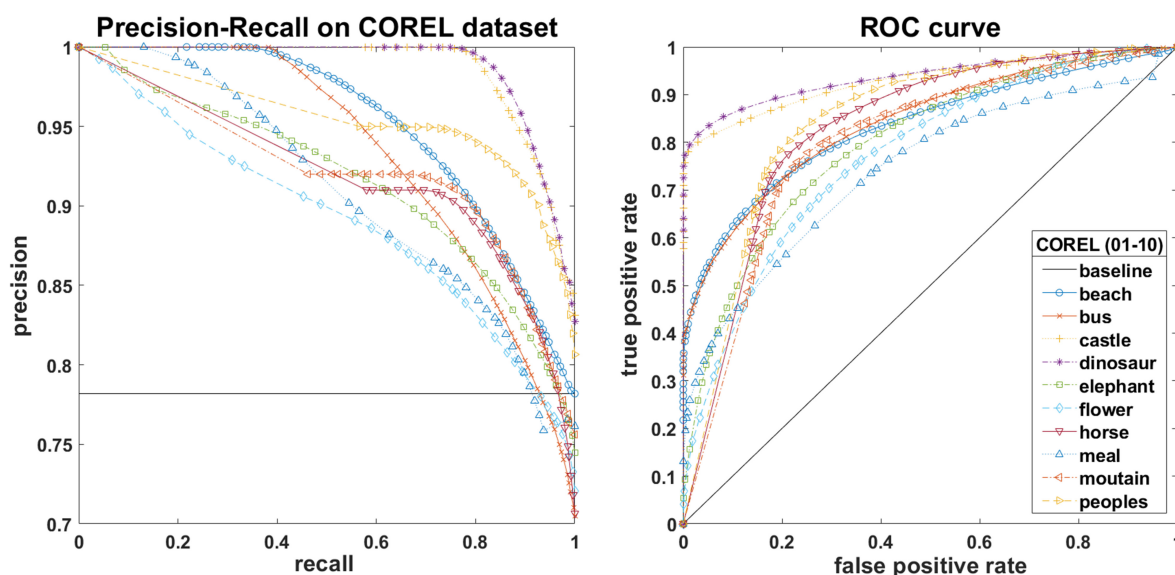


Hình 8. Tìm kiếm ảnh tương tự cho ảnh 450.jpg bộ COREL



Hình 9. Tập ảnh tương tự của ảnh 450.jpg bộ COREL

Hệ truy vấn ảnh CBIR-CKDT thực hiện truy vấn (*Retrieval On CKD-Tree*) ảnh đầu vào (*Load Image*) và kết quả tập ảnh tương tự; đồng thời thực nghiệm toàn bộ dữ liệu (*Retrieval On CKD-Tree Testing*). Kết quả thực nghiệm trên bộ ảnh COREL và bộ ảnh Wang được trình bày trong Bảng II cho thấy hiệu suất truy vấn trên bộ COREL cao hơn bộ dữ liệu Wang và thời gian truy vấn trên bộ ảnh COREL nhanh hơn bộ ảnh Wang là vì có sự khác nhau chiều cao cây CKD-Tree.



Hình 10. Precision-Recall và đường cong ROC bộ ảnh COREL

Bảng III
SO SÁNH HIỆU SUẤT TRUY VẤN GIỮA CÁC
PHƯƠNG PHÁP TRÊN CÁC BỘ DỮ LIỆU

Phương pháp	Bộ dữ liệu	Độ chính xác trung bình
B_SHIFT, 2016 [15]	COREL	72%
CBIR-CKDT	COREL	76,4%
P. CHHABRA, 2020 [16]	Wang	63,2%
CBIR-CKDT	Wang	73,27%

Bảng IV
SO SÁNH THỜI GIAN TRUY VẤN GIỮA CÁC
PHƯƠNG PHÁP TRÊN CÁC BỘ DỮ LIỆU

Phương pháp	Bộ dữ liệu	Thời gian truy vấn trung bình (ms)
B_SHIFT, 2016 [15]	COREL	5524,4
CBIR-CKDT	COREL	37
P. CHHABRA, 2020 [16]	Wang	144
CBIR-CKDT	Wang	87

3. Đánh giá kết quả thực nghiệm

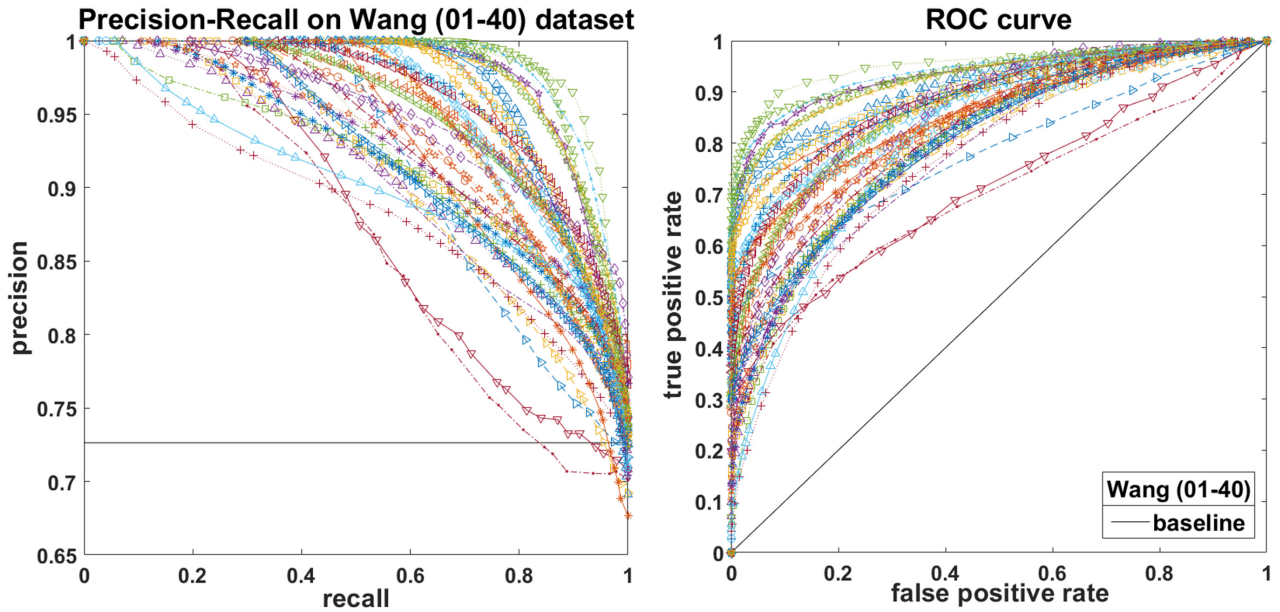
Kết quả thực nghiệm truy vấn ảnh của hệ CBIR-CKDT được minh họa bởi Hình 10, Hình 11 và Hình 12. Mỗi đường cong trên đồ thị mô tả kết quả độ chính xác (*Precision*) và độ phủ (*Recall*) thuộc một chủ đề trong bộ dữ liệu. Đường cong ROC (*ROC Curve*) cho biết tỷ lệ kết quả truy vấn đúng (*True positive rate*) và kết quả truy vấn sai (*False positive rate*). So sánh hiệu suất truy vấn giữa các bộ dữ liệu trên từng bộ ảnh COREL, Wang minh họa bởi Hình 13, Hình 15. Thời gian truy vấn trên hệ CBIR-CKDT được minh họa bởi đồ thị Hình 14, Hình 16.

Hiệu suất và thời gian truy vấn trên hệ CBIR-CKDT được so sánh với công trình [15, 16] cho thấy hệ truy vấn ảnh CBIR-CKDT cao hơn về độ chính xác và độ phủ. Độ phủ của hệ B_SHIFT thấp là do tác giả chỉ lấy top 20 ảnh trong bộ dữ liệu truy vấn. Đồng thời độ chính xác trên hệ CBIR-CKDT cao hơn phương pháp B_SHIFT và P. Chhabra do: 1) CBIR-CKDT thực hiện phân lớp ảnh trước khi gom cụm để tìm tập ảnh tương tự; 2) CBIR-CKDT phân lớp cho đối

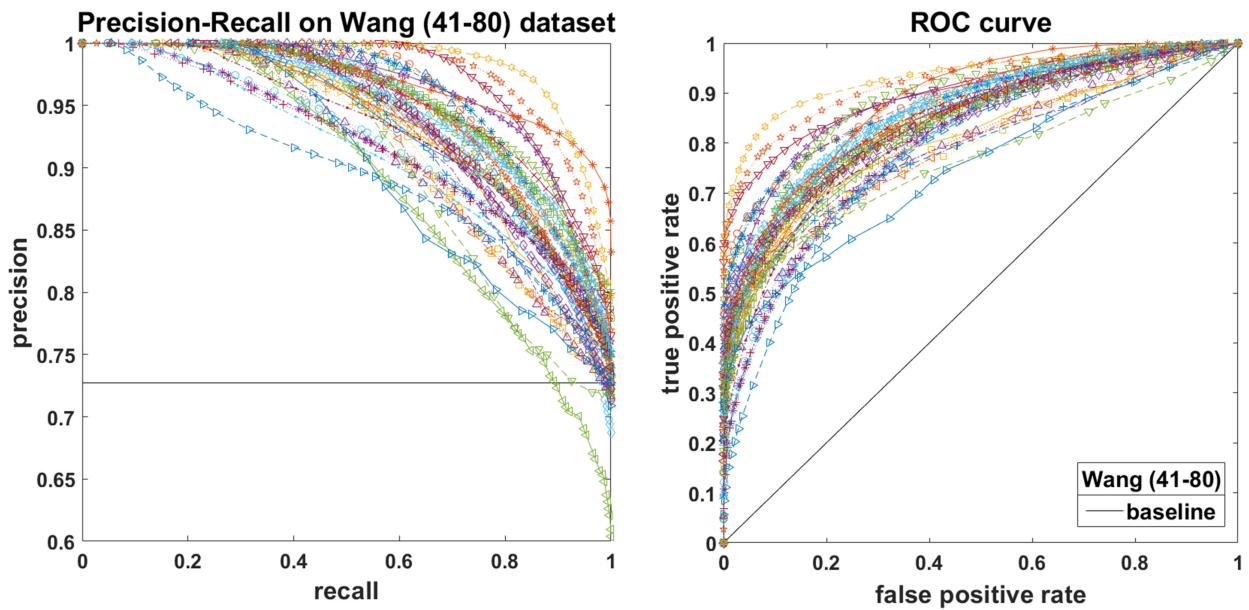
tượng nhiều lần tại các nút trong của CKD-Tree theo mô hình Deep Learning; 3) CBIR-CKDT xây dựng cấu trúc dữ liệu lưu trữ ảnh số. Tuy nhiên, công trình [16] khi thực hiện trích xuất đặc trưng véc-tơ với đặc trưng SIFT, độ dài 8 thì độ chính xác là 86,2%. Do đó, kết quả trích xuất đặc trưng hình ảnh đóng một vai trò quan trọng trong nâng cao hiệu suất truy vấn ảnh. Bên cạnh đó, thời gian truy vấn trung bình của CBIR-CKDT thấp hơn là do quá trình tìm kiếm thực hiện theo sơ đồ hình cây KD-Tree được rẽ nhánh tại các tầng nên thời gian tối ưu.

VI. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong bài báo này, chúng tôi đã đề xuất một phương pháp phân lớp dữ liệu hình ảnh bằng cây KD-Tree cải tiến và áp dụng cho bài toán tìm kiếm ảnh tương tự là hoàn toàn khả thi và hiệu quả. Đây là một tính mới trong quá trình xây dựng cấu trúc CKD-Tree theo phương pháp phân lớp dữ liệu để hình thành các cụm tương đồng, quá trình này đã tích hợp phương pháp học có giám sát (huấn luyện trọng



Hình 11. Precision-Recall và đường ROC bộ ảnh Wang (1..40)



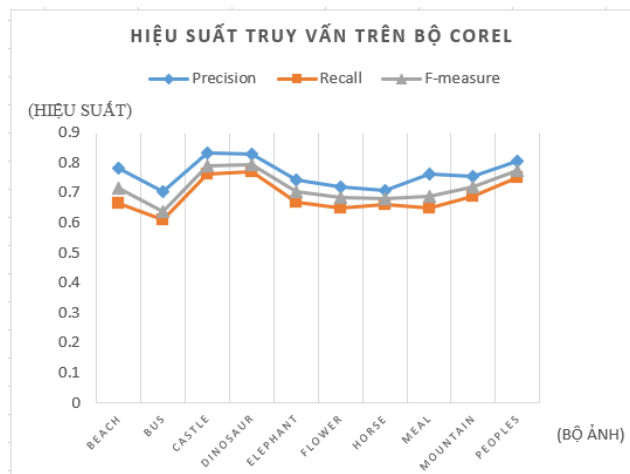
Hình 12. Precision-Recall, đường ROC bộ ảnh Wang (41..80)

số trong xây dựng mô hình cây), bán giám sát (từ mô hình KD-Tree, thực hiện tìm kiếm ảnh theo nút lá để ra tập ảnh tương tự) và không giám sát (gom cụm theo trọng số tại nút lá sau khi huấn luyện trọng số). Hệ truy vấn ảnh CBIR-CKDT đã thực nghiệm trên bộ ảnh COREL, Wang được đánh giá dựa trên độ chính xác, độ phủ và độ dung hòa với bộ ảnh COREL lần lượt là: 76,04%, 68,78%, 71,88% và bộ ảnh Wang là: 73,27%, 65,08%, 68,69%. Hướng phát triển tiếp theo của chúng tôi là xây dựng một cấu trúc KD-

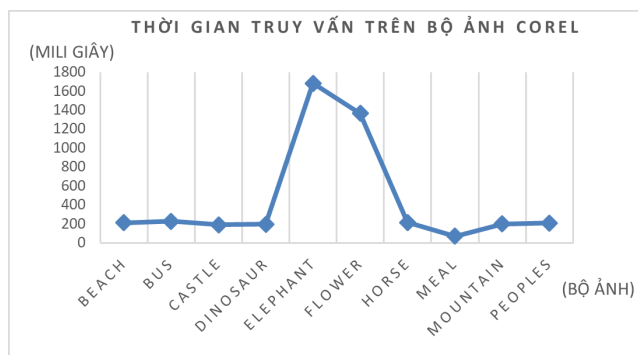
Tree đa nhánh, cân bằng áp dụng cho các tập dữ liệu tăng trưởng theo phân lớp và không giới hạn số phân lớp ban đầu. Đồng thời thực hiện bài toán tìm kiếm ảnh theo ngữ nghĩa.

LỜI CẢM ƠN

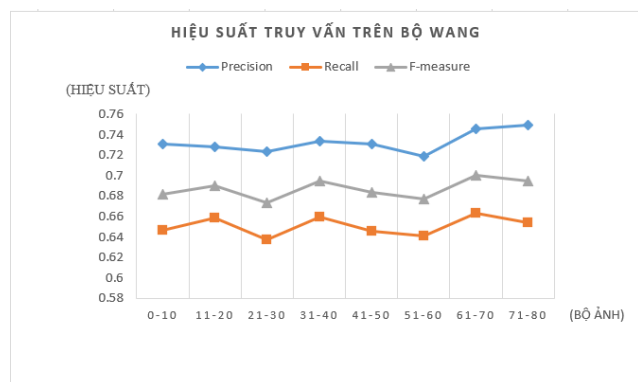
Chúng tôi xin trân trọng cảm ơn Khoa Công nghệ thông tin – Trường Đại học Khoa học - Đại học Huế, nhóm nghiên cứu SBIR-HCM đã góp ý chuyên môn cho nghiên cứu này.



Hình 13. Hiệu suất truy vấn trên bộ ảnh COREL



Hình 14. Thời gian truy vấn trên bộ ảnh COREL

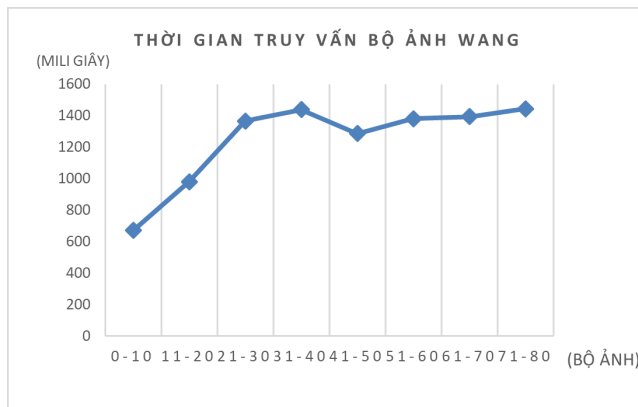


Hình 15. Hiệu suất truy vấn trên bộ ảnh Wang

Chúng tôi xin trân trọng cảm ơn Trường Đại học Công nghiệp Thực phẩm TP. HCM, Trường Đại học Sư phạm TP.HCM đã tạo điều kiện về cơ sở vật chất giúp chúng tôi hoàn thành bài nghiên cứu này.

TÀI LIỆU THAM KHẢO

- [1] Bentley, Jon Louis, "Multidimensional binary search trees used for associative searching", *Communications of the*



Hình 16. Thời gian truy vấn trên bộ ảnh Wang

ACM, vol. 18.9, pp. 509-517, 1975.

- [2] Shen, Xiaohui, et al., "Spatially-constrained similarity measure for large-scale object retrieval", *IEEE transactions on pattern analysis and machine intelligence*, Vol. 36.6, pp.1229-1241, 2013.
- [3] P. Muneesawang, N. Zhang, L. Guan, "Multimedia Database Retrieval: Technology and Applications", *Springer, New York Dordrecht London*, 2014.
- [4] Alzu'bi A, Amira A, Ramzan N, "Semantic content-based image retrieval: A comprehensive study", *J Vis Commun Image Represent*, Vol. 32, pp.20-54, 2015.
- [5] Deloitte, "Photo sharing: trillions and rising", *Deloitte Touche Tohmatsu Limited, Deloitte Global*, 2016.
- [6] A Patrizio, "Data center explorer", *Network World*, 03/12/2018, <https://www.networkworld.com/article/3325397/idc-expect-175-zettabytes-of-data-worldwide-by-2025.html>.
- [7] David Reinsel, John Gantz, John Rydning, "The Digitization of the World: From Edge to Core sponsored by Seagate", *IDC Technical Report*, 2018. <https://www.seagate.com/as/en/our-story/data-age-2025/>.
- [8] Zhang, Yuqian, et al., "Fast face sketch synthesis via kd-tree search." *European Conference on Computer Vision*. Springer, Cham, pp. 64-77, 2016.
- [9] Gao, Dan, Yan-Xia Zhang, and Yong-Heng Zhao, "Support vector machines and kd-tree for separating quasars from large survey data bases." *Monthly Notices of the Royal Astronomical Society*, Vol. 386.3, pp.1417-1425, 2008.
- [10] Hou, Wenfeng, et al., "An advanced k nearest neighbor classification algorithm based on KD-tree", *IEEE International Conference of Safety Produce Informatization (IICSPI) IEEE*, pp. 902-905, 2018.
- [11] Zhang, Fengquan, Yahui Gao and Liuqing Xu, "An adaptive image feature matching method using mixed Vocabulary-KD tree." *Multimedia Tools and Applications*, Vol. 1-19, pp. 16421 - 16439, 2019.
- [12] Hemmer, Michael, and Ondrej Stava, "KD tree encoding for point clouds using deviations", *U.S. Patent*, No. 10, 496, 336, 3-Dec-2019.
- [13] Pinkham, Reid, Shuqing Zeng, and Zhengya Zhang, "Quick-NN: Memory and Performance Optimization of kd Tree Based Nearest Neighbor Search for 3D Point Clouds", *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA) IEEE*, pp. 180-192, 2020.
- [14] McCann, Sancho, David G. Lowe, "Local naive bayes nearest neighbor for image classification", *2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE*, pp. 3650-

- 3658, 2012.
- [15] Douik, Ali, Mehrez Abdellaoui, and Leila Kabbai, "Content based image retrieval using local and global features descriptor", *2016 2nd international conference on advanced Technologies for Signal and Image Processing (ATSIP) IEEE*, pp. 151-154, 2016.
 - [16] Chhabra, Payal, Naresh Kumar Garg, and Munish Kumar, "Content-based image retrieval system using ORB and SIFT features", *Neural Computing and Applications*, Vol. 32.7, pp. 2725-2733, 2020.
 - [17] Das, Rik, Sudeep Thepade, and Saurav Ghosh. "Novel feature extraction technique for content-based image recognition with query classification", *International Journal of Computational Vision and Robotics*, Vol. 7.1-2, pp. 123-147, 2017.
 - [18] Nguyễn Thị Định, Lê Thị Vinh Thanh, Nguyễn Văn Thịnh, Văn Thế Thành, "Một phương pháp phân cụm dựa trên cây KD-Tree cho bài toán tìm kiếm ảnh", *Tạp chí Khoa học Đại học Huế, Chuyên san Kỹ thuật và Công nghệ*, ISSN: 2615-9732, Tập 129, số 2A, 2020.
 - [19] Kong, H. A., "A generalized Laplacian of Gaussian filter for blob detection and its applications", *IEEE transactions on cybernetics*, Vol. 43(6), pp. 1719-1733, 2013.
 - [20] Bora, D. J., "A novel approach for color image edge detection using multidirectional Sobel filter on HSV color space", *Int. J. Comput. Sci. Eng.*, Vol. 5(2), pp. 154-159, 2017.
 - [21] Gonzalez, C. I., "Edge detection methods based on generalized type-2 fuzzy logic", *Springer International Publishing*, pp. 21-35, 2017.
 - [22] He, L. F., "Fast basic shape feature computation", *In Computer Science and Artificial Intelligence Proceedings of the International Conference on Computer Science and Artificial Intelligence (CSAI 2016)*, pp. 22-48, 2017
 - [23] Vinayak, V., "CBIR system using color moment and color auto-Correlogram with block truncation coding", *International Journal of Computer Applications*, Vol. 161(9), pp. 1-7, 2017.
 - [24] Chaki, J., "A beginner's guide to image shape feature extraction techniques", *CRC Press*, pp. 89-131, pp. 15-68, 2019.
 - [25] Chaki, J., "Image Color Feature Extraction Techniques: Fundamentals and Applications", *Singapore: Springer*, pp. 29-80, 2021.
 - [26] Corel 1k Database, Available online: <http://wang.ist.psu.edu/docs/related/>. (n.d.).
 - [27] Wang, J. Z. (n.d.). James Z. Wang Group. Available: <http://wang.ist.psu.edu/docs/home.shtml>.

SƠ LƯỢC VỀ TÁC GIẢ

Nguyễn Thị Định



Sinh năm 1983, tốt nghiệp ngành Sư phạm tin học Trường Đại học Sư phạm TP.HCM vào năm 2006, nhận bằng Thạc sĩ ngành Truyền số liệu và mạng máy tính tại Học viện Công nghệ Bưu chính viễn thông TP.HCM vào năm 2011. Hiện là nghiên cứu sinh ngành Khoa học máy tính Trường Đại học khoa học, Đại học Huế. Lĩnh vực nghiên cứu: xử lý ảnh, tìm kiếm ảnh và cơ sở dữ liệu.

Email: dinhnt@hufi.edu.vn

Văn Thế Thành



Sinh năm 1979, tốt nghiệp chuyên ngành Toán tin tại Đại học Khoa học Tự nhiên - Đại học Quốc gia TP.HCM vào năm 2001, nhận bằng Thạc sĩ Khoa học Máy tính tại Đại học Quốc gia TP.HCM vào năm 2008. Năm 2016, nhận bằng Tiến sĩ Khoa học Máy tính tại Đại học Khoa học, Đại học Huế. Lĩnh vực nghiên cứu: xử lý ảnh, khai thác dữ liệu ảnh và tìm kiếm ảnh.

Email: thanhvt@hufi.edu.vn

Lê Mạnh Thạnh



Sinh năm 1953, nhận bằng Tiến sĩ ngành khoa học máy tính tại Đại học Budapest (ELTE), Hungary vào năm 1994. Nhận hàm Phó giáo sư tại trường Đại học Khoa học, Đại học Huế, Việt Nam vào năm 2004. Lĩnh vực nghiên cứu: cơ sở dữ liệu, cơ sở tri thức và lập trình logic.

Email: lmthanh@hueuni.edu.vn