

## ĐÁNH GIÁ MỘT SỐ CẢI TIẾN CẤU TRÚC KD-TREE CHO BÀI TOÁN TÌM KIẾM ẢNH

Nguyễn Thị Định<sup>1,3\*</sup>, Nguyễn Phương Hạc<sup>3</sup>, Văn Thế Thành<sup>2</sup>, Lê Mạnh Thành<sup>1</sup>

<sup>1</sup>Khoa Công nghệ thông tin, Trường Đại học Khoa học, Đại học Huế

<sup>2</sup>Trường Đại học Sư phạm TP. HCM

<sup>3</sup>Khoa Công nghệ thông tin, Trường Đại học Công nghiệp Thực phẩm TP. HCM

\*Email: dinhnt@hueuni.edu.vn

*Ngày nhận bài: 13/12/2022; ngày hoàn thành phần biện: 20/12/2022; ngày duyệt đăng: 26/6/2023*

### TÓM TẮT

Trong bài báo này, kết quả tìm kiếm ảnh dựa trên một số cải tiến cấu trúc KD-Tree được trình bày, phân tích và đánh giá. Trong đó, các cải tiến cấu trúc KD-Tree gồm: (1) cấu trúc KD-Tree nhị phân; (2) cấu trúc KD-Tree đa nhánh cân bằng; (3) cấu trúc iKD-Tree; (4) cấu trúc KD-Tree lồng nhau được đề xuất và thực nghiệm trên các bộ ảnh COREL, Wang, Caltech-101, Caltech-256 với độ chính xác và thời gian tìm kiếm trung bình là khác nhau. Điều này cho thấy, các cải tiến cấu trúc KD-Tree là phù hợp, đúng đắn trên cơ sở cải tiến những hạn chế từ đề xuất trước đó để làm căn cứ cho cải tiến tiếp theo. Cuối cùng, so sánh kết quả tìm kiếm ảnh giữa các phương pháp thực nghiệm trên cùng bộ ảnh; đồng thời so sánh kết quả thực nghiệm của từng phương pháp đề xuất với các phương pháp khác.

**Từ khóa:** iKD-Tree, KD-Tree, KD-Tree lồng nhau, tìm kiếm ảnh.

### 1. MỞ ĐẦU

Một trong những phương pháp để nâng cao hiệu suất tìm kiếm ảnh trong tập dữ liệu lớn là tổ chức dữ liệu hình ảnh theo cấu trúc dạng cây hoặc đồ thị. Hiện nay, có nhiều cấu trúc cây và đồ thị được sử dụng cho bài toán tìm kiếm ảnh với hiệu suất cao như S-Tree [1], KD-Tree [2], đồ thị cụm [3], v.v. Trong bài báo này, cấu trúc KD-Tree áp dụng cho bài toán tìm kiếm ảnh được thực hiện và đề xuất các cải tiến nhằm nâng cao độ chính xác, đồng thời giảm thời gian tìm kiếm để đáp ứng nhu cầu người dùng.

Trong những thập niên gần đây, một số công trình tìm kiếm ảnh đã mang lại hiệu suất cao, trong đó có sự tích hợp của nhiều kỹ thuật học máy vào một cấu trúc dữ liệu nhằm nâng cao hiệu suất về độ chính xác, giảm thời gian tìm kiếm và tăng khả năng lưu trữ để đáp ứng nhu cầu gia tăng dữ liệu ảnh số theo thời gian. Trong bài toán

tìm kiếm ảnh thì cấu trúc dữ liệu lưu trữ là một vấn đề được nhiều nhóm nghiên cứu quan tâm để cải tiến và phát triển. Cấu trúc KD-Tree cùng với những cải tiến KD-Tree đã được công bố trong thời gian qua [4, 5, 6, 7] với những kết quả khả quan. Trong bài báo này, chúng tôi thực hiện một đánh giá, so sánh về tính hiệu quả của các hệ tra cứu ảnh dựa trên các cải tiến cấu trúc KD-Tree nhằm minh chứng, đánh giá tính khả thi bài toán truy vấn ảnh, đồng thời so sánh giữa các cải tiến KD-Tree mà chúng tôi đề xuất trên cùng bộ ảnh thực nghiệm.

Đóng góp của bài báo gồm: (1) Tổng hợp những cải tiến cấu trúc KD-Tree cho tìm kiếm ảnh; (2) thực hiện đánh giá, phân tích những kết quả thu được để xác định tính đúng đắn, hiệu quả trong các cải tiến cấu trúc KD-Tree, từ đó có những định hướng cải tiến tiếp theo; (3) so sánh kết quả giữa các cải tiến đề xuất với một số phương pháp khác trên cùng bộ ảnh thực nghiệm để khẳng định tính đúng đắn trong các kết quả thu được.

## 2. CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

Các nghiên cứu về cấu trúc KD-Tree áp dụng cho bài toán xử lý dữ liệu đa chiều nó chung và bài toán tìm kiếm ảnh nói riêng đã có nhiều công bố với kết quả khả quan trong những thập niên vừa qua, một số công trình đó cụ thể là:

**Logamani. K** và cộng sự (2014) [8] đã đưa ra phương pháp gom cụm dữ liệu dựa trên cây KD-Tree bằng cách phát triển KD-Tree kết hợp kỹ thuật gom cụm k-Means cho lưu trữ và gom cụm dữ liệu. Trong đó, tác giả đã xây dựng cây KD-Tree là một cây tăng trưởng đáp ứng nhu cầu gia tăng dữ liệu. Phương pháp này đã áp dụng cho các bài toán có tập dữ liệu tăng trưởng cho kết quả khá tốt.

**N. Puviarasan** và cộng sự (2015) [9] đã xây dựng cấu trúc Index KD-Tree nhằm lưu trữ tập ảnh có kích thước lớn cho bài toán tìm kiếm ảnh thực nghiệm trên bộ ảnh COREL. Trong đó, cấu trúc Index KD-Tree tác giả quan tâm đến việc chèn nút vào cây; sau đó thực hiện quá trình tìm kiếm trên cây theo vùng không gian gần nhất với điểm truy vấn dựa trên phương pháp lập chỉ mục. Công trình này được đánh giá là hiệu quả cho bài toán lưu trữ tập dữ liệu lớn, thời gian truy vấn nhanh và khả thi cho bài toán tìm kiếm ảnh theo nội dung. **Y.H. Sharath Kumar** và cộng sự (2015) [10] đề xuất mô hình tìm kiếm ảnh theo nội dung dựa trên cấu trúc Indexing KD-Tree. Sau khi xây dựng cấu trúc chỉ mục Indexing KD-Tree là một hệ tri thức để áp dụng cho bài toán truy vấn ảnh, quá trình tìm kiếm và đối sánh được thực hiện thêm một bước lọc ra những kết quả tối ưu nhất. Đây là một ưu điểm so với các công trình tìm kiếm trực tiếp trên cấu trúc KD-Tree giúp nâng cao hiệu suất tìm kiếm. Tuy nhiên, để xây dựng hệ tri thức này thì phải tốn một lượng chi phí nhất định.

**Parikshit Ram** và cộng sự (2019) sử dụng kỹ thuật tìm kiếm láng giềng k-NN kết hợp cấu trúc KD-Tree. Việc kết hợp này nhằm cải tiến hiệu suất tìm kiếm bằng cách xây dựng cây phân vùng không gian ngẫu nhiên để thực hiện các lược đồ tìm kiếm theo cấu trúc KD-Tree [11]. **Yewang Chen** và cộng sự (2019) [12] sử dụng hai kỹ thuật tìm kiếm láng giềng RNN (*Range Nearest Neighbors*) và NN (*Nearest Neighbors*) dựa trên cây KD-Tree. Kỹ thuật RNN nhằm giảm các tính toán khoảng cách không cần thiết bằng cách kiểm tra vị trí của đối tượng đang xét nằm bên trong hay bên ngoài vùng lân cận của điểm cần tìm. Kỹ thuật NN được sử dụng để giảm các nút truy cập dư thừa bằng cách lưu chỉ số truy cập các điểm láng giềng. Thực nghiệm chứng minh tính hiệu quả của việc kết hợp các thuật toán tìm kiếm láng giềng RNN, NN và k-NN trên cây KD-Tree là hiệu quả. **Reid Pinkham** và cộng sự (2020) [13] đã thực hiện một phương pháp tối ưu bộ nhớ và tăng tốc độ tìm kiếm trên KD-Tree. Cấu trúc KD-Tree được xây dựng theo phương pháp phân chia dữ liệu thành các nhóm nút: (1) nhóm nút được lưu vào bộ nhớ đệm nhằm tạo điều kiện tìm kiếm nhanh, các nhóm điểm có thể tái sử dụng thành các phân đoạn liên kề thường xuyên trong bộ nhớ ngoài hỗ trợ cho quá trình tìm kiếm; (2) quá trình đọc ghi trên bộ nhớ đệm được thêm vào để thực hiện truy cập ngẫu nhiên để chuyển thành truy cập tuần tự; (3) xây dựng cây KD-Tree và tìm kiếm được thực hiện xen kẽ với các luồng truy cập dư thừa, băng thông được tối ưu hóa nhằm hỗ trợ quá trình tìm kiếm trên cây.

Trên cơ sở khảo sát các công trình tìm kiếm ảnh sử dụng cấu trúc KD-Tree kết hợp một số thuật toán gom cụm, phân lớp đã thu được những kết quả khả quan, mỗi công trình có những ưu và nhược điểm riêng cần cải tiến. Vì vậy, trong bài báo này một số khuyến nghị được trình bày dựa trên các cải tiến KD-Tree nhằm nâng cao hiệu suất tìm kiếm ảnh.

### 3. MỘT SỐ CẢI TIẾN CẤU TRÚC KD-TREE

Cây KD-Tree nguyên thủy được Bentley (1975) [14] đề xuất là một dạng cây tìm kiếm nhị phân BST (Binary Search Tree). Trên cơ sở phát triển cây BST với mỗi nút trên cây lưu trữ một điểm không gian đa chiều hình thành nên cấu trúc gọi là KD-Tree. KD-Tree cũng là cấu trúc cây tìm kiếm nhị phân bao gồm một nút gốc (*Root*) các nút trong (*Node*) và nút lá (*Leaf*). Tại mỗi nút chỉ có tối đa hai nhánh con, chia cây thành hai phần gồm cây con trái và cây con phải. Cấu trúc KD-Tree được ứng dụng hiệu quả trong nhiều lĩnh vực như biểu diễn cơ sở dữ liệu không gian và thời gian, biểu diễn các hệ cơ sở dữ liệu lớn (*BigData*) nhằm giảm không gian lưu trữ và tìm kiếm nhanh, v.v. Vì vậy, cấu trúc KD-Tree lưu trữ được dữ liệu đa chiều, trong khi đó dữ liệu đa phương tiện như video, hình ảnh, v.v. sau khi trích xuất là véc-tơ đặc trưng đa chiều nên cấu trúc KD-Tree phù hợp cho bài toán tìm kiếm ảnh.

Trên cơ sở KD-Tree được ứng dụng cho lưu trữ dữ liệu đa chiều; trong bài báo này, một số cải tiến cấu trúc KD-Tree được áp dụng cho bài toán tìm kiếm ảnh gồm: Cấu trúc KD-Tree nhị phân; cấu trúc KD-Tree đa nhánh cân bằng; cấu trúc iKD-Tree và cấu trúc KD-Tree lồng nhau. Đồng thời, một số kết quả thực nghiệm được trình bày, so sánh và phân tích trên từng cải tiến.

Các thành phần trên KD-Tree cải tiến được mô tả như sau:

- 1) Nút gốc (*Root*) là nút không có nút cha, lưu trữ một vector trọng số ( $w_0$ ), có tập nút con trái  $\{left\}$ , tập nút con phải  $\{right\}$  và thuộc mức trên cây là 0 ( $level = 0$ ). Ký hiệu:  $Root = \langle w_0, \{left\}, \{right\}, 0 \rangle$
- 2) Nút trong (*Node<sub>i</sub>*) là nút trong có một nút cha (*parent*), lưu trữ một vector trọng số ( $w_i$ ), có tập nút con trái  $\{left\}$ , tập nút con phải  $\{right\}$  và có một mức trên cây ( $level$ ). Ký hiệu:  $Node_i = \langle parent, w_i, \{left\}, \{right\}, level \rangle$
- 3) Nút lá (*Leaf<sub>i</sub>*) là nút có một nút cha (*parent*), không có nút con, lưu trữ tập vector đặc trưng hình ảnh  $\{f_1, f_2, \dots, f_k\}$ , có một mức trên cây ( $level$ ) và được gán một nhãn ( $label$ ). Ký hiệu:  $Leaf_i = \langle parent, \{f_1, f_2, \dots, f_k\}, level, label \rangle$

Quy tắc chung để xây dựng cấu trúc KD-Tree cho tìm kiếm ảnh gồm các bước:

**Bước 1.** Khởi tạo chiều cao KD-Tree bằng  $h$ , khởi tạo số nhánh tối đa tại mỗi nút trong là  $b$ . Khi đó số nút lá tối đa trên KD-Tree được tạo ra là  $b^h$ .

**Bước 2.** Khởi tạo tập véc-tơ trọng số ngẫu nhiên vừa đủ để phân bố vào các nút trong  $W_{kt} = \langle w_0, \dots, w_{h-1} \rangle$ ,  $w_i = (w_{i0}, \dots, w_{in})$ ; mỗi véc-tơ  $w_i$  lưu trữ tại  $Node_i$ .

**Bước 3.** Tại  $Node_i$  khởi tạo các ngưỡng trái, phải  $w_i.left = w_i.right = 0.5$  để KD-Tree cân bằng.

**Bước 4.** Lần lượt cho véc-tơ  $f_j$  duyệt từ nút gốc; giá trị đầu ra của véc-tơ  $f_j$  tại  $Node_i$  được tính theo hàm truyền Sigmoid() và xác định đường đi cho  $f_j$  đến nhánh thuộc tầng kế tiếp.

**Bước 5.** Lặp lại Bước 3, Bước 4 cho đến khi gặp nút lá  $leaf_k$  thì ghi  $f_j$  vào  $leaf_k$

Dữ liệu thực nghiệm trên các hệ tìm kiếm ảnh **CB-KDT**, **SB-KDT**, **SB-iKDT** và **SB-NKDT** được mô tả trong **Bảng 1**.

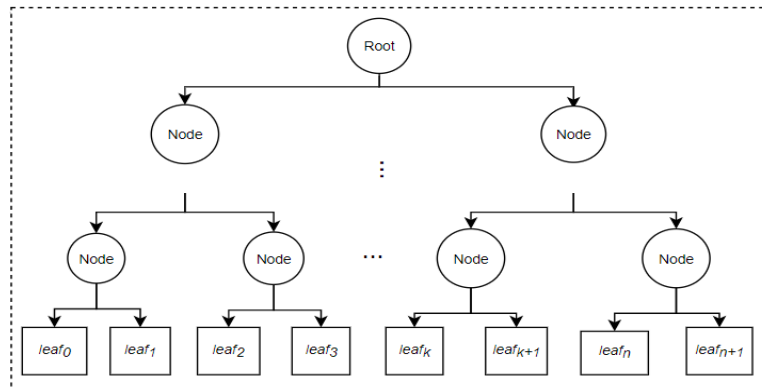
**Bảng 1.** Mô tả các tập dữ liệu ảnh thực nghiệm

TT	Tập ảnh	Số lượng ảnh	Số phân lớp
1	COREL	1,000	10
2	Wang	10,800	80
3	Caltech-101	1,344	102
4	Caltech-256	30,607	257

Môi trường thực nghiệm các hệ tìm kiếm ảnh tương tự **CB-KDT**, **SB-KDT**, **SB-iKDT** và **SB-NKDT** được xây dựng trên nền tảng dotNET Framework 4.5, ngôn ngữ lập trình C#. Cấu hình máy tính: Intel(R) Core™ i5-5200U, CPU 2.7GHz, RAM 16GB và hệ điều hành Windows 10 Professional.

### 3.1. Cấu trúc KD-Tree nhị phân

Cấu trúc KD-Tree nhị phân được xây dựng gồm: Một nút gốc (*Root*), tập nút trong  $\{Node\}$  và tập nút lá  $\{Leaf\}$ . Ban đầu, cấu trúc KD-Tree được tạo ra với một khung cây, nút gốc và nút trong lưu trữ véc-tơ trọng số khởi tạo ngẫu nhiên, nút lá lưu trữ dữ liệu hình ảnh thuộc bộ dữ liệu thực nghiệm. Một nút không phải là nút lá có hai nhánh con gọi là nhánh con trái và nhánh con phải [4]. Cấu trúc KD-Tree nhị phân được minh họa như **Hình 1**.



**Hình 1.** Cấu trúc KD-Tree nhị phân

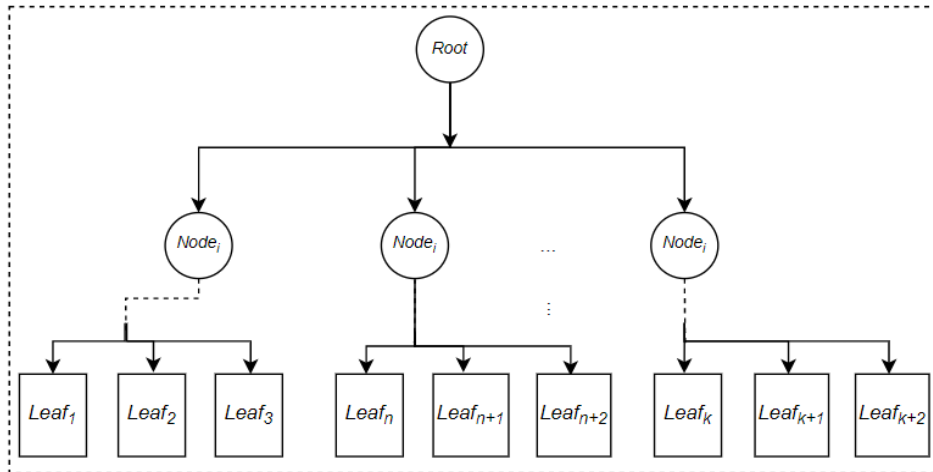
Kết quả thực nghiệm tìm kiếm ảnh trên KD-Tree nhị phân được trình bày trong **Bảng 2** [4].

**Bảng 2.** Hiệu suất tìm kiếm của hệ truy vấn ảnh **CB-KDT** [4]

Tập ảnh	Độ chính xác trung bình	Độ phủ trung bình	Độ dung hòa trung bình	Thời gian truy vấn trung bình (ms)
COREL	<b>0.7640</b>	0.6878	0.7188	38.00
Wang	<b>0.7327</b>	0.6508	0.6869	87.00

### 3.2. Cấu trúc KD-Tree đa nhánh cân bằng

Cấu trúc KD-Tree đa nhánh cân bằng được minh họa như **Hình 2**.



Hình 2. Cấu trúc KD-Tree đa nhánh cân bằng

Phát triển KD-Tree từ nhị phân thành đa nhánh là nhằm giảm chiều cao cây nhưng vẫn đảm bảo được số nút lá cần thiết cho tập ảnh thực nghiệm. Vì vậy, trên cơ sở cấu trúc KD-Tree nhị phân đã xây dựng [4], cấu trúc KD-Tree cải tiến tiếp theo là cây đa nhánh cân bằng gồm một nút gốc, tập nút trong và nhiều nút lá. Mỗi nút khác nút lá có nhiều nhánh con trái và nhánh con phải. Số nhánh con trái mỗi nút trong và nút gốc là lớn hơn 2.

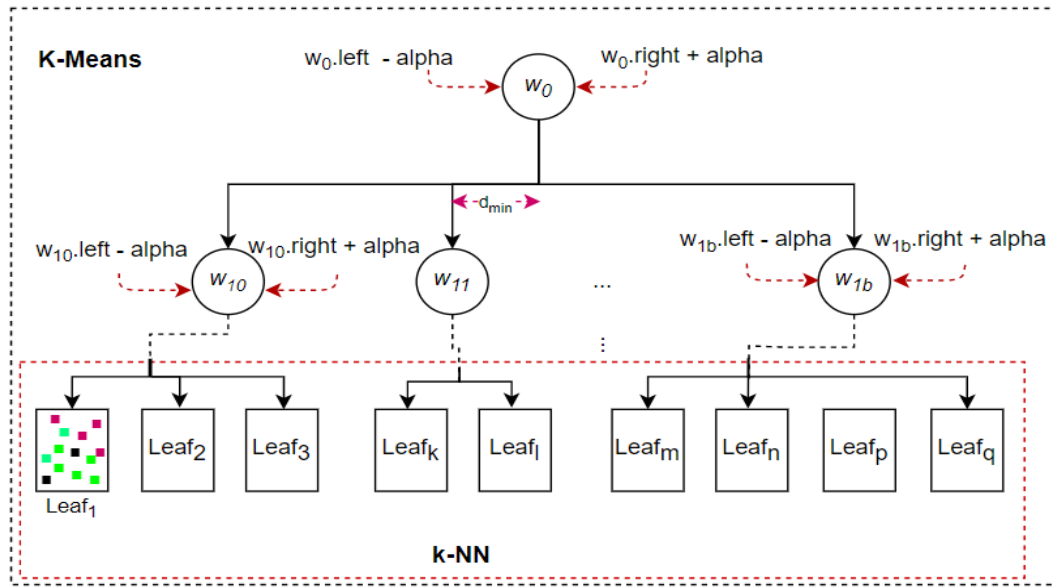
Kết quả tìm kiếm ảnh trên các bộ ảnh thực nghiệm dựa trên cấu trúc KD-Tree đa nhánh cân bằng được trình bày trong Bảng 3.

Bảng 3. Hiệu suất tìm kiếm ảnh của hệ SB-KDT [5]

Tập ảnh	Độ chính xác trung bình	Độ phủ trung bình	Độ dung hòa trung bình	Thời gian truy vấn trung bình (ms)
COREL	0.7998	0.7043	0.7484	96.77
Caltech-101	0.7101	0.6609	0.6846	122.25
Caltech-256	0.7020	0.6291	0.6627	136.89

### 3.3. Cấu trúc iKD-Tree

Cấu trúc iKD-Tree được minh họa như Hình 3. Hệ số tăng khoảng cách giữa hai nhánh con là  $\alpha$ , thực nghiệm được chọn là 0.001 [7].



Hình 3. Cấu trúc iKD-Tree

Mục đích tích hợp thuật toán K-Means vào KD-Tree nhằm tăng khoảng cách giữa hai nhánh con tại một nút cha, để từ đó tăng độ chính xác gom cụm tại mỗi nút lá. Dựa trên cấu trúc KD-Tree đã xây dựng trong công trình [5] là một cây đa nhánh cân bằng. Cấu trúc iKD-Tree được xây dựng bằng cách kết hợp thuật toán k-Means trong quá trình chọn nhánh đi cho mỗi véc-tơ đầu vào. Cấu trúc iKD-Tree gồm các thành phần giống như cấu trúc KD-Tree đa nhánh cân bằng [6]. Tuy nhiên, quá trình xây dựng iKD-Tree kết hợp với thuật toán K-Means để xác định đường đi thuộc nhánh tiếp theo, thuật toán k-NN được thực hiện phân lớp tại nút lá.

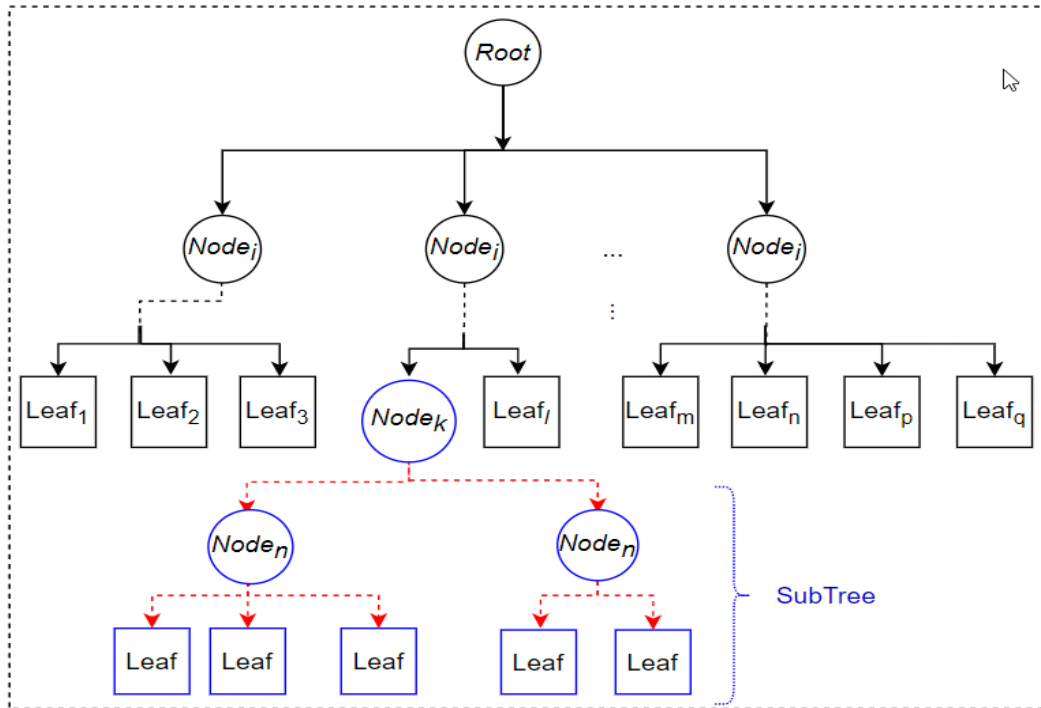
Kết quả tìm kiếm ảnh trên các bộ ảnh thực nghiệm dựa trên cấu trúc iKD-Tree được trình bày trong **Bảng 4**.

Bảng 4. Hiệu suất tìm kiếm ảnh của hệ SB-iKDT trên các bộ dữ liệu [6]

Tập ảnh	Độ chính xác trung bình	Độ phủ trung bình	Độ dung hòa trung bình	Thời gian truy vấn trung bình (ms)
Wang	0.8010	0.7404	0.7695	101.36
Caltech-101	0.7769	0.6915	0.7317	109.84
Caltech-256	0.7228	0.5828	0.6453	134.91

### 3.4. Cấu trúc KD-Tree lồng nhau

Thay vì phải thêm chi phí huấn luyện cấu trúc iKD-Tree thì một giải pháp là cải tiến iKD-Tree thành KD-Tree lồng nhau bằng cách phát triển một số nút lá thành một cây iKD-Tree tương tự. Cấu trúc KD-Tree lồng nhau minh họa như **Hình 4**.



Hình 4. Cấu trúc KD-Tree lồng nhau

Cấu trúc KD-Tree lồng nhau được xây dựng gồm hai giai đoạn: (1) Xây dựng cấu trúc iKD-Tree từ tập dữ liệu ban đầu; (2) phát triển một số nút lá trên iKD-Tree thành cây con SubTree cùng chiều cao và tính chất với iKD-Tree để hình thành cấu trúc KD-Tree lồng nhau [7]. Mục đích của việc phát triển một nút lá thành SubTree nhằm tạo ra quá trình phân lớp mịn hơn và hiệu suất phân lớp tốt hơn so với cấu trúc iKD-Tree. Điều kiện để phát triển một nút lá thành cây con SubTree là:

- 1) Số phần tử  $F = \{f_1, f_2, \dots, f_k\}$  của nút  $Leaf_j$  lớn hơn ngưỡng  $M$  ( $M = 200, 250, 400$ )
- 2) Nút lá  $Leaf_j$  chứa số phân lớp lớn hơn ngưỡng  $N$  ( $N > 2$ )
- 3) Cây con SubTree tại  $Leaf_j$  có chiều cao  $h$  và số nhánh tối đa là  $b$ .

Kết quả tìm kiếm ảnh trên các bộ ảnh thực nghiệm dựa trên cấu trúc KD-Tree lồng nhau được trình bày trong **Bảng 5**.

Bảng 5. Kết quả hiệu suất tìm kiếm ảnh của hệ SB-NKDT trên các bộ dữ liệu [7]

Tập ảnh	Precision	Recall	F-measure	Time query (ms)
COREL	<b>0.8120</b>	0.7164	0.7612	109.49
Wang	<b>0.8029</b>	0.7083	0.7526	121.67
Caltech-101	<b>0.7889</b>	0.6688	0.7239	151.92



#### 4. ĐÁNH GIÁ VÀ PHÂN TÍCH CÁC KẾT QUẢ THỰC NGHIỆM

*Bảng 6* mô tả các thành phần và số chiều của véc-tơ đặc trưng được trích xuất trên các hệ thực nghiệm ứng với các cải tiến cấu trúc KD-Tree.

*Bảng 6.* Tổng hợp các loại đặc trưng trên các thực nghiệm

Hệ tìm kiếm ảnh	Loại đặc trưng	Số giá trị
CB-KDT	MPEG-7, Sobel, Shape, Texture, vị trí tương đối, chu vi đối tượng, diện tích đối tượng	81
SB-KDT	MPEG-7, Sobel, Shape, Texture, vị trí tương đối, chu vi đối tượng, diện tích đối tượng	81
SB-iKDT	SIFT MPEG-7, Shi MPEG-7, Color MPEG-7, HOG, Sobel, Laplace	183
SB-NKDT	MPEG-7, Sobel, Shape, Texture, vị trí tương đối, chu vi đối tượng, diện tích đối tượng	81

Để minh chứng những đề xuất cải tiến trên cấu trúc KD-Tree là đúng đắn là hiệu quả cho bài toán tìm kiếm ảnh. *Bảng 7 - 10* trình bày so sánh độ chính xác và thời gian tìm kiếm trung bình trên các bộ ảnh COREL, Wang, Caltech-101, Caltech-256 tương ứng với các cấu trúc KD-Tree nhị phân, KD-Tree đa nhánh, iKD-Tree và KD-Tree lồng nhau.

*Bảng 7.* So sánh hiệu suất giữa các cấu trúc đề xuất trên bộ ảnh COREL

Cấu trúc	Độ chính xác trung bình	Độ phủ trung bình	Độ dung hòa trung bình	Thời gian truy vấn trung bình (ms)
KD-Tree nhị phân	0.7640	0.6878	0.7188	38.00
KD-Tree đa nhánh	0.7998	0.7043	0.7484	96.77
KD-Tree lồng nhau	<b>0.8120</b>	0.7164	0.7612	109.49

*Bảng 8.* So sánh hiệu suất giữa các cấu trúc đề xuất trên bộ ảnh Wang

Cấu trúc	Độ chính xác trung bình	Độ phủ trung bình	Độ dung hòa trung bình	Thời gian truy vấn trung bình (ms)
KD-Tree nhị phân	0.7327	0.6508	0.6869	87.00
iKD-Tree	0.8010	0.7404	0.7695	101.36
KD-Tree lồng nhau	<b>0.8029</b>	0.7083	0.7526	121.67

**Bảng 9.** So sánh hiệu suất giữa các cấu trúc đề xuất trên bộ ảnh Caltech-101

Cấu trúc	Độ chính xác trung bình	Độ phủ trung bình	Độ dung hòa trung bình	Thời gian truy vấn trung bình (ms)
KD-Tree đa nhánh	0.7101	0.6609	0.6846	122.25
iKD-Tree	0.7769	0.6915	0.7317	<b>109.84</b>
KD-Tree lồng nhau	<b>0.7889</b>	0.6688	0.7239	151.92

**Bảng 10.** So sánh hiệu suất giữa các phương pháp đề xuất trên bộ ảnh Caltech-256

Cấu trúc	Độ chính xác trung bình	Độ phủ trung bình	Độ dung hòa trung bình	Thời gian truy vấn trung bình (ms)
KD-Tree đa nhánh	0.7020	0.6291	0.6627	136.89
iKD-Tree	<b>0.7228</b>	0.5828	0.6453	<b>134.91</b>

Kết quả so sánh từ các **Bảng 7 -10**, cho thấy các định hướng cải tiến trên cấu trúc KD-Tree là khả thi và hiệu quả, hiệu suất truy vấn ảnh trên cấu trúc KD-Tree lồng nhau cao hơn thì thời gian tìm kiếm tăng lên do chiều cao của KD-Tree lồng nhau tăng lên khi thực hiện phát triển một số nút lá thành cây con, chiều cao KD-Tree lồng nhau tăng lên nên ảnh hưởng đến thời gian tìm kiếm trên cấu trúc này. Cấu trúc KD-Tree đa nhánh và iKD-Tree giúp giảm chiều cao cây nên thời gian xây tìm kiếm trên hệ SB-KDT cũng giảm so với tìm kiếm trên KD-Tree nhị phân. Các giá trị so sánh thời gian tìm kiếm trung bình trên một bộ ảnh giữa các phương pháp đề xuất còn chênh lệch nhau do chiều cao cây được xây dựng trong các thực nghiệm là khác nhau, dẫn đến thời gian tìm kiếm cũng khác nhau.

Gọi  $b$  là số nhánh tại mỗi nút cần xây dựng;  $h$  là chiều cao cây;  $Time$  là thời gian xây dựng và huấn luyện từng cấu trúc. **Bảng 11** là các tham số của các cải tiến KD-Tree.

**Bảng 11.** Bảng tổng hợp tham số xây dựng và huấn luyện các cải tiến KD-Tree

Bộ ảnh	KD-Tree nhị phân			KD-Tree đa nhánh			iKD-Tree			KD-Tree lồng nhau		
	$b$	$h$	$Time$	$b$	$h$	$Time$	$b$	$h$	$Time$	$b$	$h$	$Time$
COREL	2	4	0:25:45	3	2	0:38:54	-	-	-	2	2	0:35:56
Wang	2	7	0:50:56	-	-	-	3	4	1:10:54	3	3	0:55:43
Caltech-101	-	-	-	5	3	0:44:45	5	3	1:58:43	-	-	-
Caltech-256	-	-	-	4	4	0:56:32	4	4	2:11:21	4	3	1:44:39

Để minh chứng tính đúng đắn trong các cải tiến KD-Tree, *Bảng 12 – 15* thực hiện so sánh độ chính xác tìm kiếm ảnh giữa các đề xuất với các phương pháp khác.

*Bảng 12.* So sánh độ chính xác với các công trình khác trên bộ ảnh COREL

Phương pháp	Độ chính xác trung bình
B_SHIFT, (Douik, Ali. et al.), 2016 [15]	0.7200
Shape and Texture Features, (Abdulkadhem, et al.), 2019 [16]	0.7286
ORB 8-dimensions with MPL (P. Chhabra et al.), 2020 [17]	0.6656
C-Tree (Nhi, N. T. U. et al. ), 2021 [18]	0.6777
<b>CB-KDT</b> (KD-Tree nhị phân)	<b>0.7640</b>
<b>SB-KDT</b> (KD-Tree đa nhánh)	<b>0.7998</b>
<b>SB-NKDT</b> (KD-Tree lồng nhau)	<b>0.8120</b>

*Bảng 13.* So sánh độ chính xác với các công trình khác trên bộ ảnh Wang

Phương pháp	Độ chính xác trung bình
ORB and SIFT features (P. Chhabra et al.), 2020 [17]	0.6320
SBIR-CT (Nhi N.T.U et al.), 2021 [18]	0.6072
SVM (Das et al.), 2017 [19]	0.5590
8D-GLCM+GSF+ HSVCM (Bella & Vasuki), 2019 [20]	0.5970
<b>CB-KDT</b> (KD-Tree nhị phân)	0.7327
<b>SB-KDT</b> (KD-Tree đa nhánh)	<b>0.7810</b>
<b>SB-iKDT</b> (iKD-Tree)	<b>0.8010</b>
<b>SB-NKDT</b> (KD-Tree lồng nhau)	<b>0.8029</b>

*Bảng 14.* So sánh độ chính xác với các công trình khác trên bộ ảnh Caltech-101

Phương pháp	Độ chính xác trung bình
CE and Valleys with SVM (Rajesh, M. B., et al.), 2020 [21]	0.6314
CBIR Multi-trend Structure (Sathiamoorthy, S., et al.), 2020 [22]	0.6942
Feature Detector (Ahmed, K. T., et al.), 2021 [23]	0.6200

CBIR with features (Soumya Prakash Rana, et al.), 2019 [24]	0.6450
<b>SB-KDT</b> (KD-Tree đa nhánh)	<b>0.7101</b>
<b>SB-iKDT</b> (iKD-Tree)	<b>0.7769</b>
<b>SB-NKDT</b> (KD-Tree lồng nhau)	<b>0.7889</b>

**Bảng 15.** So sánh độ chính xác với các công trình khác trên bộ ảnh Caltech-256

Phương pháp	Độ chính xác trung bình
Visual Words FVFF (Mehmood, Z., et al.), 2018 [25]	0.4630
SVM (Uzma Sharif, et al.), 2019 [26]	0.4752
Ensembles of deep learning (Safa Hamreras), 2020 [27] (50 classes)	0.5726
CBIR (Khawaja Tehseen Ahmed), 2019 [28] (Topk = 25)	0.6985
<b>SB-KDT</b> (KD-Tree đa nhánh)	<b>0.7020</b>
<b>SB-iKDT</b> (iKD-Tree)	<b>0.7228</b>

Kết quả so sánh độ chính xác giữa các cải tiến trên cấu trúc KD-Tree với một số phương pháp khác trên cùng bộ ảnh thực nghiệm COREL, Wang, Caltech-101, Caltech-256 cho thấy các đề xuất cải tiến KD-Tree cho bài toán tìm kiếm ảnh có kết quả độ chính xác cao hơn một số công trình đã công bố. Điều này cho thấy một số đề xuất cải tiến cấu trúc KD-Tree áp dụng trên các bộ ảnh này là khả thi và hiệu quả.

## 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong bài báo này, một số cải tiến trên cấu trúc KD-Tree nhằm nâng cao độ chính xác tìm kiếm ảnh được trình bày, đánh giá và phân tích. Kết quả phân tích ưu, nhược điểm của một số hệ tìm kiếm ảnh **CB-KDT**, **SB-KDT**, **SB-iKDT**, **SB-NKDT** được thực hiện. Các phương pháp cải tiến KD-Tree được so sánh trên cùng một bộ ảnh ảnh; đồng thời so sánh giữa phương pháp đề xuất với các phương pháp khác. Định hướng tiếp theo của chúng tôi là tích hợp thêm các kỹ thuật phân lớp hình ảnh như SVM, DNN, CNN để nâng cao hiệu suất phân lớp ảnh đầu vào, từ đó cải thiện độ chính xác tìm kiếm ảnh dựa trên cấu trúc KD-Tree.

## LỜI CẢM ƠN

Trân trọng cảm ơn Khoa Công nghệ thông tin, Trường Đại học Khoa học - Đại học Huế, Trường Đại học Sư phạm TP.HCM và Trường Đại học Công nghiệp Thực

phẩm Thành phố Hồ Chí Minh đã hỗ trợ về chuyên môn và cơ sở vật chất để nhóm tác giả hoàn thành nghiên cứu này.

### TÀI LIỆU THAM KHẢO

- [1] Le, Thanh Manh, et al (2013). Image retrieval system based on EMD similarity measure and S-tree, *In: Intelligent Technologies and Engineering Systems, Springer, New York, NY*, p. 139-146
- [2] Zhang, Fengquan, Yahui Gao, and Liuqing Xu (2019). An adaptive image feature matching method using mixed Vocabulary-KD tree, *Multimedia Tools and Applications*, pp. 1-19
- [3] Van T. T., Le T. M. (2018). Content-based image retrieval based on binary signatures cluster graph, *Expert Systems*, 35(1), 2220
- [4] Nguyễn Thị Định, Thế Thành Văn, Mạnh Thạnh Lê (2021). Phân lớp ảnh bằng cây KD-Tree cho bài toán tìm kiếm ảnh tương tự, *Các công trình nghiên cứu, phát triển và ứng dụng Công nghệ Thông tin và Truyền thông, Tạp chí Thông tin và Truyền thông*, tr. 40-52
- [5] Nguyễn Thị Định, Thế Thành Văn, Mạnh Thạnh Lê (2021), Một phương pháp phân lớp dựa trên cấu trúc KD-Tree cho bài toán tìm kiếm ảnh theo ngữ nghĩa, *Kỷ yếu hội nghị quốc gia lần thứ 14 về nghiên cứu cơ bản và ứng dụng công nghệ thông tin*, Fair2021, TP. Hồ Chí Minh, 23/12/2021. DOI: [10.15625/vap.2021.0075](https://doi.org/10.15625/vap.2021.0075)
- [6] Dinh, N. T., and Le, T. M. (2022). An Improvement Method of Kd-Tree Using k-Means and k-NN for Semantic-Based Image Retrieval System. *In World Conference on Information Systems and Technologies, Springer, Cham*, 177-187
- [7] Dinh, N. T., V. T. Thanh, Thanh, L. M. (2022). A semantic-based image retrieval system using Nested KD-Tree Structure, *Annales Univ. Sci. Budapest. Sect. Comp.*, Vol. 53, p. 3 – 28.
- [8] Logamani, K.; Punitha, S. C (2014). Density Based Clustering using Enhanced KD Tree, *International Journal of Science, Engineering and Computer Technology*, 4.11: 314
- [9] Puviarasan, N., and R. Bhavani (2015). A new indexing technique to retrieve images using integration of colour-size, texture and shape features, *International Journal of Applied Pattern Recognition*, 2,3, pp. 280-302.
- [10] Kumar, YH Sharath, and N. Pavithra (2015). KD-Tree approach in sketch based image retrieval, *International Conference on Mining Intelligence and Knowledge Exploration, Springer, Cham*.
- [11] Ram, Parikshit; Sinha, Kaushik (2019). Revisiting kd-tree for nearest neighbor search. *In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, p. 1378-1388
- [12] Chen, Yewang, et al (2019). Fast neighbor search by using revised kd tree, *Information Sciences*, 472: 145-162
- [13] Pinkham, Reid, Shuqing Zeng, and Zhengya Zhang (2020). Quick-NN: Memory and Performance Optimization of kd Tree Based Nearest Neighbor Search for 3D Point Clouds, *International Symposium on High Performance Computer Architecture (HPCA). IEEE*.

- [14] Bentley, Jon Louis (1975). Multidimensional binary search trees used for associative searching, *Communications of the ACM*, 18,9, pp. 509-517
- [15] Douik, Ali, Mehrez Abdellaoui, and Leila Kabbai (2016). "Content based image retrieval using local and global features descriptor." *2nd international conference on advanced Technologies for Signal and Image Processing (ATSIP)*. IEEE
- [16] Abdulkadhem, Abdulkadhem Abdulkareem, and Tawfiq A. Al-Assadi (2019). Proposed a Content-Based Image Retrieval System Based on the Shape and Texture Features, *Int. J. Innovat. Technol. Explor. Eng* 8, pp. 2189
- [17] Chhabra, P., Garg, N. K., & Kumar, M. (2020). Content-based image retrieval system using ORB and SIFT features. *Neural Computing and Applications*, 32(7), 2725-2733
- [18] Nhi, N. T. U., & Le, T. M. (2021, March). Semantic-Based Image Retrieval Using Balanced Clustering Tree. In *World Conference on Information Systems and Technologies Springer, Cham*, (pp. 416-427).
- [19] Das, R., Thepade, S., & Ghosh, S. (2017). Novel feature extraction technique for content-based image recognition with query classification. *International Journal of Computational Vision and Robotics*, 7(1-2), 123-147
- [20] Bella, M. I. T., & Vasuki, A. (2019). An efficient image retrieval framework using fused information feature. *Computers & Electrical Engineering*, 75, 46-60
- [21] Rajesh, M. B., & Sathiamoorthy, S. (2020, March). Co-occurrence of Edges and Valleys with Support Véc-to Machine for Content Based Image retrieval. In *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, pp. 465-470.
- [22] Sathiamoorthy, S., & Natarajan, M. (2020). An efficient content based image retrieval using enhanced multi-trend structure descriptor. *SN Applied Sciences*, 2(2), 1-20.
- [23] Ahmed, K. T., Aslam, S., Afzal, H., Iqbal, S., Mehmood, A., & Choi, G. S. (2021). Symmetric Image Contents Analysis and Retrieval Using Decimation, Pattern Analysis, Orientation, and Features Fusion. *IEEE Access*, 9, 57215-57242.
- [24] Rana, S. P., Dey, M., & Siarry, P. (2019). Boosting content based image retrieval performance through integration of parametric & nonparametric approaches. *Journal of Visual Communication and Image Representation*, 58, 205-219
- [25] Mehmood, Z., Abbas, F., Mahmood, T., Javid, M. A., Rehman, A., & Nawaz, T. (2018). Content-Based Image Retrieval Based on Visual Words Fusion Versus Features Fusion of Local and Global Features. *Arabian Journal for Science & Engineering (Springer Science & Business Media BV)*, 43(12).
- [26] Sharif, U., Mehmood, Z., Mahmood, T., Javid, M. A., Rehman, A., & Saba, T. (2019). Scene analysis and search using local features and support véc-to machine for effective content-based image retrieval. *Artificial Intelligence Review*, 52(2), 901-925
- [27] Hamreras, S., Boucheham, B., Molina-Cabello, M. A., Benitez-Rochel, R., & Lopez-Rubio, E. (2020). Content based image retrieval by ensembles of deep learning object classifiers. *Integrated Computer-Aided Engineering*, 27(3), 317-331.
- [28] Ahmed, K. T., Ummesafi, S., & Iqbal, A. (2019). Content based image retrieval using image features information fusion. *Information Fusion*, 51, 76-99

## ASSESSMENT OF SOME IMPROVEMENTS IN KD-TREE STRUCTURE FOR IMAGE RETRIEVAL PROBLEM

Nguyen Thi Dinh<sup>1,3\*</sup>, Nguyen Phuong Hac<sup>3</sup>, Van The Thanh<sup>2</sup>, Le Manh Thanh<sup>1</sup>

<sup>1</sup> Faculty of Information Technology, University of Sciences, Hue University

<sup>2</sup> HCMC University of Education

<sup>3</sup> Faculty of Information Technology, HCMC University of Food Industry

\*Email: dinhnt@hufi.edu.vn

### ABSTRACT

In this paper, image retrieval results based on some KD-Tree structural improvements are presented, analyzed, and evaluated. Of which, the upgrades to the KD-Tree structure include (1) binary KD-Tree structure; (2) balanced multi-branch KD-Tree structure; (3) iKD-Tree structure; (4) The proposed and tested nested KD-Tree structure on COREL, Wang, Caltech-101, Caltech-256 image sets with different accuracy and average search time. This shows that the structural improvements of KD-Tree are appropriate and correct on the basis of improving the limitations from the previous proposal to serve as a basis for the next advance. Finally, compare the results of image search between experimental methods on the same set of images; at the same time, compare the experimental results of the proposed method with other methods.

**Keywords:** KD-Tree, iKD-Tree, Nested KD-Tree, Image retrieval.



**Nguyễn Thị Định** sinh năm 1983. Bà 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.



**Nguyễn Phương Hạc** sinh năm 1979. Bà tốt nghiệp ngành Toán – Tin tại trường Đại học Khoa học Tự Nhiên TP HCM vào năm 2002; tốt nghiệp thạc sĩ Công nghệ Thông tin tại trường Đại học Bách Khoa Hà Nội năm 2010. Hiện nay bà giảng dạy tại trường Đại học Công nghiệp Thực phẩm TP Hồ Chí Minh.

*Lĩnh vực nghiên cứu:* cơ sở dữ liệu, xử lý ảnh.



**Văn Thế Thành** sinh năm 1979. Ông 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 trường Đại học Khoa học, Đại học Huế.

*Lĩnh vực nghiên cứu:* xử lý ảnh, khai thác dữ liệu ảnh và tìm kiếm ảnh.



**Lê Mạnh Thạnh** sinh năm 1953. Ông nhận bằng Tiến sĩ ngành khoa học máy tính tại Đại học Budapest (ELTE), Hungary vào năm 1994. Nhận học 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.