

PHÂN TÍCH MỘT SỐ PHƯƠNG PHÁP CẢI TIẾN DỰA TRÊN R-TREE CHO BÀI TOÁN TÌM KIẾM ẢNH

Lê Thị Vĩnh Thanh^{1,3}, Lê Mạnh Thanh¹, Văn Thế Thành^{2*}

¹Khoa Công nghệ thông tin, Trường Đại học Khoa học, Đại học Huế

²Khoa Công nghệ thông tin, Trường Đại học Sư phạm TP.HCM

³Trường Đại học Bà Rịa Vũng Tàu

Email: thanhhtv@hueuni.edu.vn, lmthanh@hueuni.edu.vn, thanhvt@hcmue.edu.vn

Ngày nhận bài: 13/12/2022; ngày hoàn thành phản biện: 19/12/2022; ngày duyệt đăng: 26/6/2023

TÓM TẮT

Tìm kiếm ảnh dựa trên nội dung là một bài toán đầy thách thức để tìm kiếm các ảnh liên quan từ một không gian lưu trữ lớn. Vấn đề đặt ra là cần phải có một cấu trúc lưu trữ và đánh chỉ mục các đặc trưng của tập dữ liệu ảnh để cải thiện hiệu suất tìm kiếm ảnh. Bài báo này trình bày một số các cải tiến dựa trên cấu trúc R-Tree cho bài toán tìm kiếm ảnh để nâng cao độ chính xác và cải thiện thời gian tìm kiếm. Đầu tiên, một cấu trúc R^S-Tree được cải tiến dựa vào khối cầu không gian nhằm tối ưu hóa không gian lưu trữ và tăng độ chính xác tìm kiếm. Thứ hai, phương pháp k-Means được áp dụng để cải tiến thuật toán tách nút nhằm nâng cao độ chính xác gom cụm. Thứ 3, một cấu trúc đồ thị láng giềng được xây dựng tại tầng nút lá của cây R^S-Tree để nâng cao độ chính xác. Thử nghiệm được tiến hành trên các tập dữ liệu ảnh CUB-200-2011. Bài báo thực hiện so sánh và đánh giá hiệu suất giữa các phương pháp; đồng thời so sánh kết quả với các công trình gần đây trên cùng tập dữ liệu để minh chứng tính đúng đắn của các phương pháp.

Từ khóa: R^S-Tree; Image Retrieval, k-Means, CBIR, Neighbor Graph.

1. GIỚI THIỆU

Trong những năm gần đây, nhờ sự phát triển về công nghệ liên quan đến thiết bị lưu trữ và chụp ảnh kỹ thuật số mà ảnh số đã gia tăng nhanh chóng. Vì vậy, để lưu trữ và quản lý dữ liệu kho dữ liệu hình ảnh lớn một cách hiệu quả là một thách thức cho cộng đồng nghiên cứu. Kích thước của cơ sở dữ liệu lớn cũng rất khó khăn để xử lý và truy xuất hình ảnh. Do đó, trong những thập kỷ qua, có nhiều phương pháp truy xuất hình ảnh từ cơ sở dữ liệu lớn được phát triển. Vấn đề chính của một hệ thống tìm kiếm ảnh gặp khó khăn là làm thế nào để xác định các hình ảnh liên quan với ảnh truy

vấn trong bộ sưu tập hình ảnh kết quả lớn và đa dạng. Để giải quyết vấn đề này, dựa trên văn bản và dựa trên nội dung là hai kỹ thuật được sử dụng phổ biến để tìm kiếm và truy xuất trong cơ sở dữ liệu hình ảnh [1].

Trong bài báo này chúng tôi trình bày các phân tích kết quả cải tiến dựa trên cấu trúc R-Tree cho bài toán truy vấn ảnh theo nội dung. Các cải tiến bao gồm: (1) Cấu trúc R^S -Tree được thiết kế dựa trên R-Tree cho bài toán tìm kiếm ảnh nhằm nâng cao độ chính xác và thời gian tìm kiếm ảnh; (2) Một cấu trúc cải tiến iR^S -Tree được tạo ra bằng việc sử dụng phương pháp k-Means cho thuật toán tách nút để cải thiện độ chính xác gom cụm; (3) Cấu trúc NBGraphRST là sự kết hợp R^S -Tree và đồ thị cụm láng giềng tại tầng nút lá của cây để tăng độ chính xác tìm kiếm ảnh tương tự. Trên cơ sở những nội dung cải tiến, chúng tôi thực hiện phân tích đánh giá kết quả của các phương pháp để minh chứng tính đúng đắn của từng phương pháp cải tiến.

2. CÁC CÔNG TRÌNH LIÊN QUAN

Trong những thập niên gần đây, nhiều công trình đã sử dụng các phương pháp khác nhau cho bài toán truy vấn ảnh theo nội dung để cải thiện hiệu suất hệ thống cụ thể như sau:

Nhiều công trình đã áp dụng các kỹ thuật lập chỉ mục hình ảnh nhằm nâng cao tốc độ tìm kiếm ảnh. Haldurai và cộng sự (2015) đã đề xuất một hệ truy vấn ảnh tương tự theo nội dung sử dụng cấu trúc cây R-Tree [2]. Vanitha và cộng sự (2017) đã đề xuất một cấu trúc chỉ mục SR-Tree ứng dụng cho hệ thống tìm kiếm ảnh tương tự theo nội dung. Hệ thống thực hiện trích xuất đặc trưng màu sắc, đặc trưng không gian và lưu trữ véc-tơ đặc trưng trên cây SR-Tree [3]. Shama và cộng sự (2015) đã đề xuất một hệ thống truy vấn ảnh tương tự sử dụng cấu trúc R^* -Tree cho bộ ảnh thực vật. Nhóm tác giả sử dụng phương pháp ma trận đồng xuất hiện và phép lọc Gabour để trích xuất đặc trưng ảnh [4]. Alfarrarjeh và cộng sự (2020) đã đề xuất một lớp chỉ mục R^* -Tree ứng dụng cho bài toán tìm kiếm ảnh tương tự với dữ liệu ảnh đường phố [5].

Bên cạnh đó nhiều công trình cũng đã tiếp cận các phương pháp học máy để áp dụng cho các bài toán truy vấn ảnh nhằm nâng cao độ chính xác. Trong công trình [6] đề xuất phương pháp tập trung bộ mô tả tích chập chọn lọc SCDA (*Selective Convolutional Descriptor Aggregation*), phương pháp này tự động định vị đối tượng chính trong các hình ảnh chi tiết và trích xuất đặc trưng phân biệt cho chúng. Trong SCDA, chỉ có một mô hình CNN được tiền huấn luyện (từ ImageNet) được sử dụng. Kết quả truy vấn với top-1 đạt 59.62%, top-5 đạt 65.79%. Trong công trình [7] nhóm tác giả đề xuất một mô hình truy vấn ảnh dựa trên mạng CNN. Trong công trình này một phương pháp tổng hợp mới dựa trên đặc trưng tích chập có chọn lọc, được gọi là SGeM. Bên cạnh đó, nhóm tác giả thiết kế một sơ đồ lựa chọn đặc trưng đạt đến kích

hoạt tối đa và đặt ngưỡng thích hợp để chọn các bộ mô tả cục bộ. Kết quả truy vấn với top-1 đạt 60.63%, top-5 đạt 66.57%. Trong [8], Zeng, H. và cộng sự đã phát triển một phương pháp băm để khoanh vùng các đối tượng phân biệt chi tiết và tạo ra các mã nhị phân hiệu quả. Nhóm tác giả đã thiết kế một kiến trúc mới thực hiện đồng thời việc huấn luyện các vùng cục bộ và tạo mã băm. Kết quả thực nghiệm truy vấn trên bộ dữ liệu ảnh CUB-200-2011 với độ chính xác 69.07% (ảnh 16 bits) và 70.10% (ảnh 64 bits). Các đánh giá thực nghiệm trên một số bộ dữ liệu ảnh chi tiết cho thấy rằng phương pháp đề xuất đạt được hiệu suất tốt. Tuy nhiên, phương pháp này đòi hỏi tốn nhiều thời gian để huấn luyện. Đồng thời, trong công trình này chưa đề xuất một cấu trúc dữ liệu để lưu trữ hình ảnh nhằm cải thiện thời gian tìm kiếm ảnh.

Trên cơ sở khảo sát các công trình liên quan, trong bài báo này, chúng tôi trình bày các phương pháp cải tiến trên cấu trúc lưu trữ chỉ mục không gian R^S -Tree áp dụng cho bài toán tìm kiếm ảnh. Phân tích và đánh giá hiệu quả của các phương pháp cải tiến và so sánh với các công trình khác trên cùng bộ dữ liệu CUB-200-2011.

3. CÁC CẢI TIẾN DỰA TRÊN CẤU TRÚC R-TREE

3.1. R-Tree nguyên thủy

R-Tree là một cấu trúc cây cân bằng động dùng để phân cụm dữ liệu không gian đa chiều. Trong cấu trúc R-Tree nguyên thủy [9], mỗi *Node* là một bộ $\langle MBR, p \rangle$, trong đó, *MBR* (Minimun Bounding Rectangle) là một vùng không gian hình chữ nhật chứa các vùng không gian nút con bên trong nó và *p* là con trỏ liên kết đến các nút con. Mỗi Leaf là một bộ $\langle MBR, oid \rangle$, trong đó, *oid* là định danh đối tượng. Mỗi Leaf trên cây có số phần tử tối thiểu là *m* và số phần tử tối đa là *M*.

Các cấu trúc lập chỉ mục đa chiều được sử dụng rộng rãi là R-Tree, R^* -Tree, SS-Tree và đã được chứng minh là hoạt động khá tốt với việc phân đoạn dữ liệu. Kỹ thuật này phân vùng tập dữ liệu đa chiều thành một khối siêu hình chữ nhật thô để giúp xây dựng cấu trúc chỉ mục không bị ảnh hưởng nhiều bởi không gian chết[10]. Ngoài ra, các cấu trúc chỉ mục dựa trên R-Tree có thể hỗ trợ cả dữ liệu điểm và dữ liệu không gian bằng cách sử dụng phương pháp phân vùng dữ liệu với phân đoạn dữ liệu để cung cấp khả năng phân cụm không gian tốt hơn vì nó không yêu cầu bất kỳ phép biến đổi điểm nào để tổ chức dữ liệu không gian [9]. Mặc dù R-Tree, R^* -tree và SS-tree hỗ trợ tổ chức dữ liệu không gian, nhưng việc lựa chọn cấu trúc lập chỉ mục hiệu quả bị ảnh hưởng rất nhiều bởi một số yếu tố là thuộc tính dữ liệu, phương pháp xây dựng dữ liệu và loại truy vấn [10].

Trên cơ sở kế thừa cấu trúc R-Tree và các biến thể của nó, một cấu trúc cải tiến R^S -Tree được trình bày trong phần 3.2.

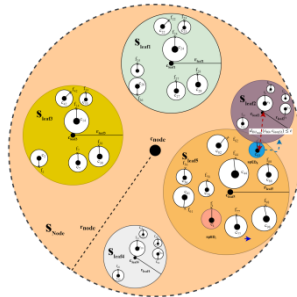
3.2. Cấu trúc R^S-Tree

R^S-Tree được đề xuất trong công trình [14] nhằm lưu trữ các phần tử hình ảnh trong không gian d chiều, là cây đa nhánh cân bằng ứng dụng cho bài toán tìm kiếm ảnh. Việc gom nhóm dữ liệu được thực hiện trên từng nút của cây R^S-Tree dựa vào độ đo tương tự giữa các véc-tơ đặc trưng ảnh ngưỡng θ cho trước nhằm tạo ra một cây đa nhánh cân bằng để nâng cao hiệu suất truy vấn. Nút trong S_{node} là một bộ $\langle MBS, p \rangle$, trong đó MBS là một khối cầu có tâm \vec{c}_{node} và bán kính r_{node} , p là con trỏ liên kết đến các nút con. Khối cầu này bao phủ các khối cầu của các nút thuộc nhánh cây con. Mỗi nút trong S_{node} có số phần tử tối đa là N và tối thiểu là 2. Nút lá S_{leaf} là bộ $\langle MBS, entity \rangle$, trong đó MBS là một khối cầu có tâm \vec{c}_{leaf} và bán kính r_{leaf} chứa một tập thực thể $entity$ với mỗi thực thể $spED$ là một bộ $\langle MBS, oid \rangle$ trong đó MBS là khối cầu có tâm \vec{c}_{sp} và bán kính r_{sp} chứa không gian đối tượng, oid là định danh đối tượng $\vec{f} = (v_1, v_2, v_3, \dots, v_d)$. Mỗi nút lá S_{leaf} có số phần tử tối đa là M và số phần tử tối thiểu là $m(1 < m \leq M/2)$.

3.3. Cấu trúc cải tiến iR^S-Tree

Việc tách nút xảy ra thường xuyên sẽ làm chậm quá trình tạo cây. Do đó, để nâng cao hiệu quả gom cụm và cải thiện thời gian tạo cây, chúng tôi xây dựng cấu trúc iR^S-Tree, được cải tiến từ R^S-Tree. Các cải tiến trong cấu trúc iR^S-Tree bao gồm:

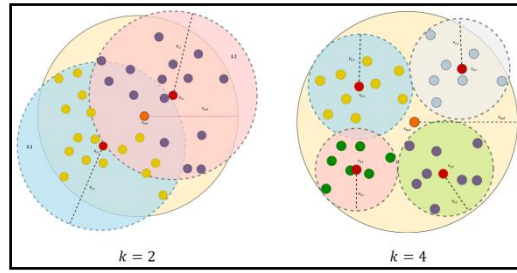
Thứ nhất, việc thêm một phần tử $spED_i$ vào iR^S-Tree được thực hiện theo các nguyên tắc tương tự R^S-Tree. Tuy nhiên, khi gặp một nút lá đầy, thay vì thực hiện tách nút, một phần tử xa tâm nhất của nút lá hiện hành nhất ký hiệu $spED_k$ được chọn để thêm vào một nút lá khác có cùng cha. Nút lá được chọn để thêm vào thỏa mãn hai tiêu chí sau: (1) nút lá được chọn có số phần tử nhỏ hơn M ; (2) khoảng cách Euclidean từ tâm của phần tử $spED_k$ đến tâm của phần tử nút lá được chọn nhỏ hơn hoặc bằng θ . Trường hợp có nhiều nút lá thỏa mãn hai tiêu chí trên nút lá có khoảng cách Euclidean ngắn nhất sẽ được chọn. Ngược lại, nếu không tồn tại nút lá nào thỏa mãn hai tiêu chí này thì thực hiện tách nút. Quá trình thêm phần tử iR^S-Tree vào được minh họa như Hình 1.



Hình 1. Minh họa việc thêm một phần tử vào cây.

Trong **Hình 1**, mô tả quá trình thêm phần tử $spED_i$ vào cây iR^S -Tree, S_{leaf5} là nút lá được lựa chọn để thêm phần tử $spED_i$. Trong trường hợp nút lá S_{leaf5} đầy, phần tử $spED_k$ là phần tử xa nhất được chọn để thêm vào nút lá gần nhất S_{leaf2} .

Thứ hai, khi một nút lá bị tràn, phương pháp k-Means được sử dụng để thực hiện tách nút thành k cụm ($k \geq 2$). Việc này nhằm tối ưu hóa độ tương tự của các đối tượng không gian trong một cụm, đồng thời giảm thiểu độ tương tự giữa các cụm khác nhau. Các phần tử trong nút tràn sẽ được phân hoạch thành k cụm theo độ đo tương đồng được minh họa như **Hình 2**.



Hình 2. Phép tách nút trong iR^S -Tree bằng phương pháp k-Means

3.4. Cấu trúc đồ thị láng giềng NBGraphRST

Trong công trình [11] trình bày về các định nghĩa cơ bản làm cơ sở nền tảng cho việc xây dựng một cấu trúc đồ thị láng giềng trên cây R^S -Tree. Với hai khối cầu nút lá $S_{leaf1}(\vec{c}_1, r_1), S_{leaf2}(\vec{c}_2, r_2)$ có tâm lần lượt là \vec{c}_1, \vec{c}_2 và bán kính là r_1, r_2 . Hai khối cầu S_{leaf1}, S_{leaf2} gọi là chồng lấp không gian, ký hiệu là $Overlap(S_{leaf1}, S_{leaf2})$ khi:

$$dist = d_E(S_{leaf1} \cdot \vec{c}_1, S_{leaf2} \cdot \vec{c}_2) < S_{leaf1} \cdot r_1 + S_{leaf2} \cdot r_2, \quad (1)$$

với d_E là hàm khoảng cách Euclidean.

Giả sử hai vùng S_{leaf1}, S_{leaf2} không chồng lấp không gian, khoảng cách của hai vùng không gian S_{leaf1}, S_{leaf2} , ký hiệu là:

$$dist_{eps} = d_E(S_{leaf1} \cdot \vec{c}_1, S_{leaf2} \cdot \vec{c}_2) - S_{leaf1} \cdot r_1 + S_{leaf2} \cdot r_2. \quad (2)$$

Gọi tập các phần tử đối tượng dữ liệu hình ảnh có trong nút là:

$$C_{spE} = \{spED_i \in S_{leaf}, i = 1..S_{leaf} \cdot count\} \quad (3)$$

Độ đo phân lớp của nút lá S_{leaf} là:

$$classification(S_{leaf}) = \{lable_k | \max\{count(spED_i)\}, spED_i.lable = lable_k\} \quad (4)$$

Các loại láng giềng của một nút là bao gồm:

(1) láng giềng *overlap* của nút lá S_{leafk} là tập hợp:

$$N_o(S_{leafk}) = \{S_{leaf} \in S_L \setminus \{S_{leafk}\} | Overlap(S_{leafk}, S_{leaf})\} \quad (5)$$

(2) láng giềng ϵ của nút lá S_{leafk} là tập hợp:

$$N_e(S_{leafk}) = \{S_{leaf} \in S_L \setminus \{S_{leafk}\} | dist_{esp}(S_{leafk}, S_{leaf}) \leq \epsilon\} \quad (6)$$

(3) láng giềng $classes$ của nút lá S_{leafk} là tập hợp:

$$N_c(S_{leafk}) = \{S_{leaf} \in S_L | classification(S_{leafk}) = classification(S_{leaf})\} \quad (7)$$

(4) láng giềng của một nút lá S_{leafk} , ký hiệu $N_n(S_{leafk})$, là hội của tất cả các phần tử láng giềng $overlap, \epsilon$ và $classes$, nghĩa là:

$$N_n(S_{leafk}) = N_o(S_{leafk}) \cup N_e(S_{leafk}) \cup N_c(S_{leafk}) \quad (8)$$

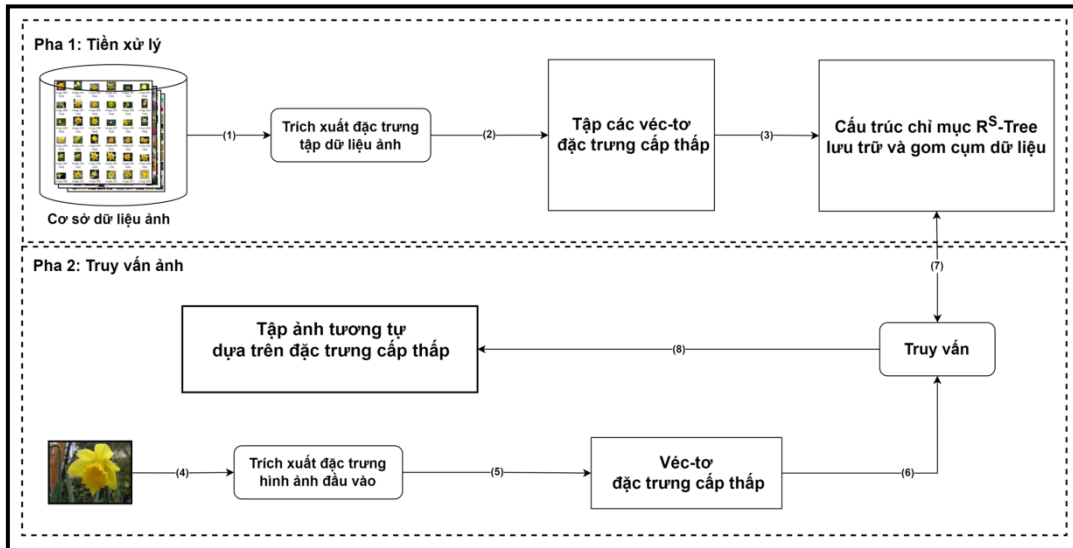
Trên cơ sở đó, một đồ thị láng giềng của các nút lá được tạo ra trong quá trình tạo cây nhằm nâng cao độ chính xác truy vấn và được mô tả như sau: đồ thị láng giềng của nút lá S_{leafk} là một đồ thị ký hiệu là $NBGraph(S_{leafk})t = \{(V, E)\}$, trong đó:

$$V = \{S_{leafk}\} \cup N_n(S_{leafk}), E = \{(v, v_i) | v = S_{leafk}, v_i \in V \setminus \{S_{leafk}\}\} \quad (9)$$

4. MÔ HÌNH TỔNG QUÁT CHO BÀI TOÁN TÌM KIẾM ẢNH

4.1. Mô hình tìm kiếm ảnh

Mô hình truy vấn ảnh theo nội dung trong các phương pháp đề xuất được thực hiện gồm hai pha: Thứ nhất là pha tiền xử lý, các hình ảnh trong cơ sở dữ liệu hình ảnh được trích xuất véc-tơ đặc trưng và thực hiện đánh chỉ mục dựa trên một cấu trúc dữ liệu. Thứ 2 là pha truy vấn ảnh, từ ảnh đầu vào hệ thống thực hiện tìm kiếm trên một cấu trúc dữ liệu để trả về tập ảnh tương tự. Mô hình được minh họa như **Hình 3**.



Hình 3. Mô hình tổng quát cho bài toán truy vấn ảnh theo nội dung

Quá trình tìm kiếm ảnh được thực hiện gồm hai pha, pha thứ nhất thực hiện gom cụm và lưu trữ dữ liệu ảnh trên cấu trúc cây chỉ mục R^S -Tree, pha thứ hai thực hiện tìm kiếm các hình ảnh tương tự cho ảnh đầu vào. Quá trình thực hiện được mô tả như sau:

Xây dựng cây phân cụm. Quá trình xây dựng cây gom cụm dữ liệu không gian R^S -Tree dựa trên véc-tơ đặc trưng của tập dữ liệu ảnh gồm các bước như sau:

- (1) Trích xuất tập véc-tơ đặc trưng f_i của tập dữ liệu ảnh;
- (2) Cấu trúc gom cụm chỉ mục được tạo ra để lưu trữ các véc-tơ f_i mô tả đặc trưng của hình ảnh.

Tìm kiếm ảnh. Việc tìm kiếm ảnh tương tự được thực hiện với đầu vào là một hình ảnh truy vấn và đầu ra là tập ảnh tương tự dựa trên cây chỉ mục. Quá trình tìm kiếm ảnh tương tự được thực hiện theo các bước như sau:

- (1) Trích xuất véc-tơ đặc trưng của ảnh cần truy vấn;
- (2) Thực hiện truy vấn ảnh tương tự dựa trên cấu trúc chỉ mục R^S -Tree;
- (3) Tra cứu tập ảnh tương tự dựa trên tập chỉ mục đã được truy vấn.

4.2. Thuật toán tìm kiếm ảnh

Từ cây phân cụm dữ liệu không gian R^S -Tree đã tạo, một thuật toán tra cứu ảnh tương tự theo nội dung dựa trên cây R^S -Tree được đề xuất. Quá trình tìm kiếm ảnh tương tự được thực hiện trên cây R^S -Tree và được mô tả như sau:

Thuật toán 1. Thuật toán tìm kiếm ảnh trên R^S -Tree

Input: véc-tơ đặc trưng $spED$ của ảnh truy vấn, cây R^S -Tree

Output: tập ảnh tương tự SI ;

Function: **RSIR** (S_{Nr} , $spED$)

begin

$S_{Node} = S_{Nr}$;

if (S_{Node} trở tới phần tử *null*) **then**

return null;

else

if (S_N không phải lá nút lá) **then**

Đi theo nhánh gần nhất với phần tử truy vấn ký hiệu S_{Nk} ;

RSIR (S_{Nk} , $spED$);

```

else
     $SI = \{\text{Các phần tử ảnh tương tự trong nút lá hiện hành } S_{Lcrt}\};$ 
endif
endif
return  $\{SI\}$ ;
end.
    
```

Gọi n là số phần tử của tập dữ liệu, M là số phần tử tối đa trong một nút của R^S-Tree. Thuật toán RSIR lần lượt duyệt qua các nút từ gốc đến lá, hơn nữa vì cây R^S-Tree là cây cân bằng nên thuật toán RSIR duyệt qua chiều cao h của cây. Mỗi lần duyệt, thuật toán RSIR phải so sánh với M phần tử của mỗi nút. M là hằng số. Do đó, độ phức tạp của thuật toán RSIR là $O(\log n)$.

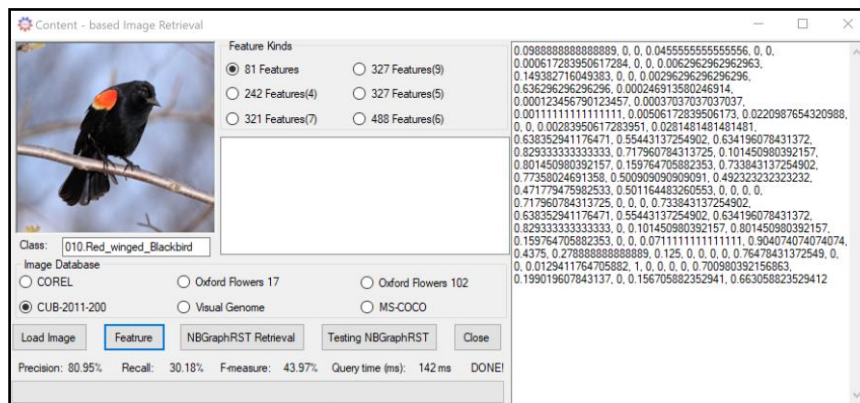
5. THỰC NGHIỆM

5.1. Môi trường thực nghiệm

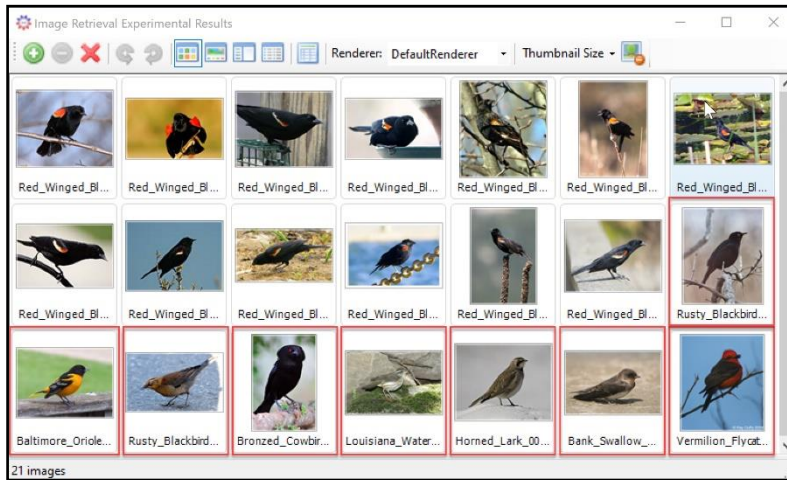
Pha tiền xử lý được thực hiện trên máy PC CPU 2.3GHz 8-core 9th-generation Intel Core i9, 16GB 2666MHz memory, 1TB flash storage. Pha tìm kiếm được thực nghiệm trên máy PC CPU Intel Core i7-6500U CPU @ 2.50GHz, 8.0GB RAM, hệ điều hành Windows 10 Pro 64 bit.

5.2. Ứng dụng và kết quả thực nghiệm

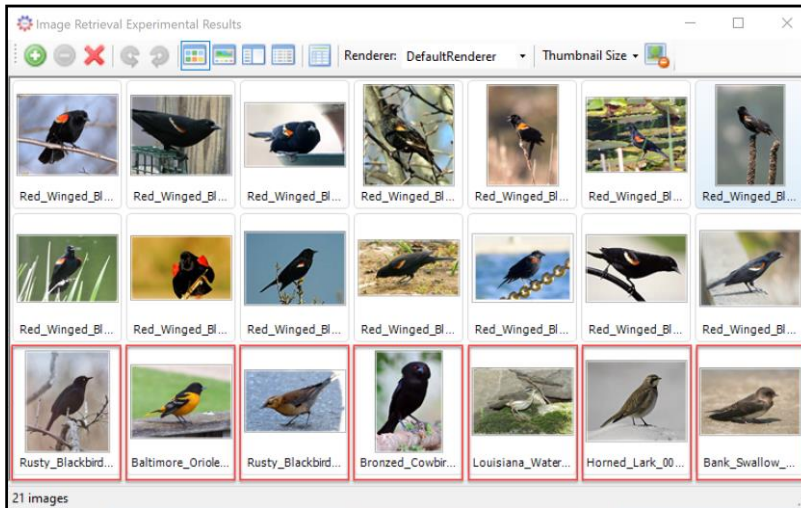
Trong bài báo này, chúng tôi tiến hành thực nghiệm trên bộ ảnh CUB-200-2011 gồm 11.788 ảnh được chia thành 200 phân lớp. Tuy nhiên, trong bài báo này chúng tôi chỉ thực nghiệm truy vấn ảnh và tính độ chính xác cho tập ảnh truy vấn gồm 5000 hình ảnh có độ chính xác tốt nhất. Đồng thời $topk = 21$ ảnh tương tự sẽ được trả về. Ứng dụng thực nghiệm được minh họa trong **Hình 4-7**.



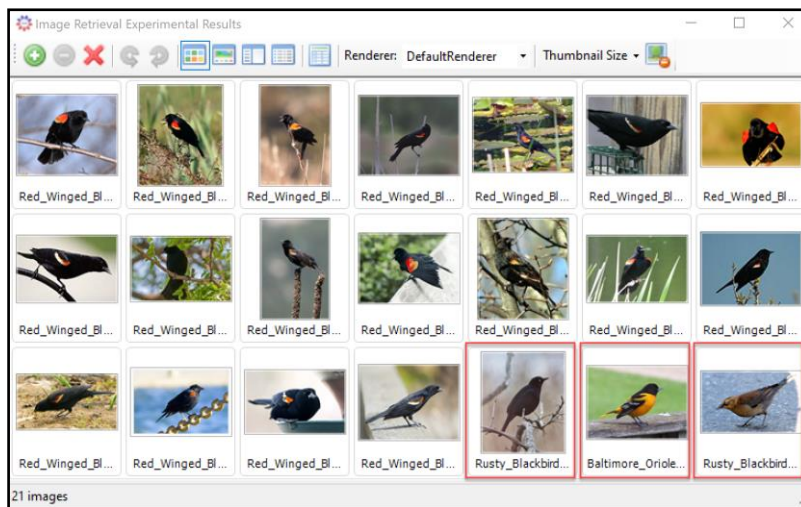
Hình 4. Một giao diện truy vấn ảnh trên cấu trúc NBGraphRST



Hình 5. Một kết quả truy vấn ảnh trên cấu trúc R^S-Tree



Hình 6. Một kết quả truy vấn ảnh trên cấu trúc iR^S-Tree



Hình 7. Một kết quả truy vấn ảnh trên cấu trúc NBGraphRST

Hình 5, Hình 6, Hình 7 mô tả kết quả thực nghiệm trên bộ ảnh CUB-200-2011 ứng với ba phương pháp đề xuất. Kết quả cụ thể như sau: đối với cấu trúc R^S -Tree kết quả trả về 13 ảnh tương tự so với ảnh truy vấn trên tổng số 21 ảnh, đạt độ chính xác 61.90%; đối với cấu trúc iR^S -Tree kết quả trả về 14 ảnh tương tự đạt độ chính xác 66.67%; cấu trúc NBGraphRST kết quả trả về 17 ảnh tương tự đạt độ chính xác 80.95%. Từ kết quả cho thấy việc cải tiến dựa trên cấu trúc R^S -Tree đã nâng cao được độ chính xác truy vấn ảnh. Đặc biệt cấu trúc kết hợp R^S -Tree và đồ thị láng giềng đã cho kết quả vượt trội so với hai phương pháp còn lại. Tuy nhiên để trả giá cho điều này, thời gian truy vấn trên cấu trúc NBGraphRST chậm hơn so với hai phương pháp còn lại, nhưng vẫn đảm bảo được thời gian truy vấn là khả thi.

Cấu trúc iR^S -Tree đạt độ chính xác cao hơn cấu trúc R^S -Tree nhờ cải tiến thuật toán tách nút bằng phương pháp k-Means. Cả cấu trúc R^S -Tree và iR^S -Tree trong quá trình tạo cây, thuật toán tách nút xảy ra thường xuyên. Do đó, có một số phần tử đáng lý thuộc cụm này nhưng do sai số trong quá trình tính toán đã thuộc vào một cụm khác. Điều này dẫn đến không tìm kiếm hết các ảnh tương tự của ảnh truy vấn. Trong cấu trúc NBGraphRST, một cấu trúc đồ thị láng giềng được tạo ra tại tầng nút lá nhằm tạo điều kiện để tập ảnh tương tự trả về có độ bao phủ cao hơn đồng thời nâng cao được độ chính xác truy vấn.

Hiệu suất truy vấn trung bình trên tập dữ liệu CUB-2011-0200 của mô hình tìm kiếm ảnh tương tự dựa trên cấu trúc R^S -Tree, iR^S -Tree, NBGraphRST được biểu diễn trong **Bảng 1**. Để đánh giá hiệu suất mô hình đề xuất, chúng tôi so sánh với kết quả của các công trình liên quan trước đó trên cùng tập dữ liệu được mô tả trong **Bảng 2**.

Bảng 1. Hiệu suất tìm kiếm của các phương pháp truy vấn trên bộ dữ liệu CUB-200-2011

Phương pháp	Avg. precision	Avg. recall	Avg. F-measure	Avg. query time (ms)
CBIR-RST	65.45	29.45	40.62	49.27
CBIR-iRST	67.76	30.15	41.73	54.16
CBIR-NBGraphRST	78.66	36.75	50.10	114.89

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

Phương pháp	Mean Average Precision (MAP)
Wei, X. S., 2016 [6]	65.80
Wang, Z., 2018 [7]	66.57
Zeng, H., 2019 [8]	70.10%
CBIR-RST	65.45
CBIR-iRST	67.76
CBIR-NBGraphRST	78.66

Trong công trình, nhóm tác giả đề xuất phương pháp SCDA cho bài toán truy vấn ảnh. Kết quả truy vấn với top-1 đạt 59.62%, top-5 đạt 65.79%. Trong công trình [7] nhóm tác giả đề xuất một mô hình truy vấn ảnh dựa trên mạng CNN. Trong công trình này một phương pháp tổng hợp mới dựa trên đặc trưng tích chập có chọn lọc, được gọi là SGeM. Kết quả truy vấn với top-1 đạt 60.63%, top-5 đạt 66.57%. Nhóm nghiên cứu Zeng H. (2019) [8] sử dụng phương pháp tìm kiếm ảnh trên mạng CNN. Trong công trình này, một phương pháp băm mới để truy xuất hình ảnh chi tiết, đồng thời khoanh vùng các vùng phân biệt và tạo mã nhị phân. Kết quả truy vấn đạt độ chính xác đạt 70.10%. Tuy nhiên, các công trình này tốn nhiều chi phí để huấn luyện, đồng thời nhóm tác giả chưa đề xuất một cấu trúc dữ liệu để lưu trữ chỉ mục đa chiều của hình ảnh nhằm nâng cao tốc độ truy vấn ảnh. Đối với các phương pháp đề xuất cải tiến dựa trên R^S -Tree đã đạt độ chính xác khá cao với top21, đặc biệt phương pháp kết hợp đồ thị láng giềng và R^S -Tree đã đạt được độ chính xác vượt trội so với các phương pháp còn lại. Điều đó cho thấy áp dụng đồ thị láng giềng cho mô hình tìm kiếm ảnh là hiệu quả cho bài toán tìm kiếm ảnh tương tự theo nội dung.

6. KẾT LUẬN

Trong bài báo này, chúng tôi đã thực hiện phân tích các phương pháp cải tiến của cấu trúc R^S -Tree áp dụng cho bài toán tìm kiếm ảnh nhằm nâng cao hiệu quả truy vấn. Các phương pháp cải tiến được tiến hành thực nghiệm trên bộ ảnh CUB-200-2011 với độ chính xác lần lượt là 65.45%, 67,76% và 78.66%. Từ các kết quả phân tích cho thấy rằng các cải tiến trên R^S -Tree đã mang lại độ chính xác tốt hơn cho hệ thống truy vấn ảnh. Đặc biệt cấu trúc NBGraphRST là sự kết hợp giữa cấu trúc R^S -Tree và đồ thị cụm láng giềng đã nâng cao đáng kể độ chính xác truy vấn. Đồng thời chúng tôi cũng thực hiện so sánh với các công trình liên gần đây nhằm minh chứng tính hiệu quả của các phương pháp đề xuất.

LỜI CẢM ƠN

Chúng tôi xin trân trọng cảm ơn Khoa Công nghệ thông tin – Đại học Khoa học, Đại học Huế đã góp ý chuyên môn cho nghiên cứu này. Chúng tôi xin trân trọng cảm ơn Trường Đại học Bà Rịa Vũng Tàu, Trường ĐH 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]. Karthikeyan, T., P. Manikandaprabhu, and S. Nithya, *A survey on text and content based image retrieval system for image mining*. International Journal of Engineering, 2014. 3.

- [2]. Haldurai, L. and V. Vinodhini, *Parallel Indexing on Color and Texture Feature Extraction using R-Tree for Content Based Image Retrieval*. International Journal of Computer Sciences and Engineering, 2015. **3**: p. 11-15.
- [3]. Vanitha, J. and M. SenthilMurugan, *An efficient content based image retrieval using block color histogram and color co-occurrence matrix*. Int. J. Appl. Eng. Res, 2017. **12**(24): p. 15966-15971.
- [4]. Shama, P., K. Badrinath, and A. Tilugul, *An Efficient Indexing Approach for Content based Image Retrieval*. International Journal of Computer Applications, 2015. **117**(15).
- [5]. Alfarrarjeh, A., et al. *A Class of R-tree Indexes for Spatial-Visual Search of Geo-tagged Street Images*. in *2020 IEEE 36th international conference on data engineering (ICDE)*. 2020. IEEE.
- [6]. Wei, X.-S., et al., *Selective convolutional descriptor aggregation for fine-grained image retrieval*. IEEE Transactions on Image Processing, 2017. **26**(6): p. 2868-2881.
- [7]. Wang, Z., et al. *Selective convolutional features based generalized-mean pooling for fine-grained image retrieval*. in *2018 IEEE Visual Communications and Image Processing (VCIP)*. 2018. IEEE.
- [8]. Zeng, H., H. Lai, and J. Yin, *Simultaneous region localization and hash coding for fine-grained image retrieval*. arXiv preprint arXiv:1911.08028, 2019.
- [9]. Guttman, A. *R-trees: A dynamic index structure for spatial searching*. in *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*. 1984.
- [10]. Nam, B. and A. Sussman. *A comparative study of spatial indexing techniques for multidimensional scientific datasets*. in *Proceedings. 16th International Conference on Scientific and Statistical Database Management*, 2004. 2004. IEEE.
- [11]. Thanh, L.T.V., T.M. Le, and T.T. Van. *Semantic-Based Image Retrieval Using RS-Tree and Neighbor Graph*. in *WorldCIST (2)*. 2022.
- [12]. Lê Thị Vĩnh Thanh, Văn Thế Thành, Lê Mạnh Thành (2021). "Một phương pháp tìm kiếm ảnh hiệu quả dựa trên cấu trúc R-Tree", *Kỷ yếu Hội thảo Quốc gia về Công nghệ thông tin và ứng dụng trong các lĩnh vực (CITA2021)*, Đại học Đà Nẵng, Nhà xuất bản Đà Nẵng, ISBN: 978-604-84-5998-7, trang 259-271.

ANALYSIS OF SOME R-TREE-BASED IMPROVEMENT METHODS FOR IMAGE RETRIEVAL PROBLEM

Le Thi Vinh Thanh^{1,3}, Le Manh Thanh¹, Van The Thanh^{2*}

¹ Faculty of Information Technology, University of Sciences, Hue University

² Faculty of Information Technology, HCMC University of Education

³ Ba Ria Vung Tau University

Email: thanhltv@hueuni.edu.vn, lmthanh@hueuni.edu.vn, thanhvt@hcmue.edu.vn

ABSTRACT

Content-based image retrieval is a challenging task to find related images from a large storage space. The problem is that it is necessary to have a structure to store and index the features of the image data set to improve the image retrieval performance. This paper presents a number of improvements based on the R-Tree for the image retrieval problem to enhance the accuracy and improve the retrieval time. First, an improved structure named R^S -Tree, based on space spheres, aims to optimize storage space and increase accuracy. Second, the k-Means method is applied to improve the node splitting algorithm to enhance clustering accuracy. Third, a neighbor graph structure is built at the leaf layer of the R^S -Tree to retrieve with higher accuracy. Experiments were conducted on the image data set CUB-200-2011. The paper compares and evaluates the performance among the methods; and compares the results with recent works on the same data set to demonstrate the correctness of the methods.

Keywords: CBIR, k-Means, Image Retrieval, Neighbor Graph, R^S -Tree.



Lê Thị Vĩnh Thanh 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 Hệ thống thông tin tại Trường Đại học Khoa học Tự Nhiên TP.HCM vào năm 2014. 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.



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.



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.