TẠP CHÍ KHOA HỌC
TRƯỜNG ĐẠI HỌC SƯ PHẠM TP HỒ CHÍ MINH
Tập 20, Số 5 (2023): 842-854
Website: https://journal.hcmue.edu.vn
ISSN: 2734-9918

HO CHI MINH CITY UNIVERSITY OF EDUCATION
JOURNAL OF SCIENCE
Vol. 20, No. 5 (2023): 842-854
https://doi.org/10.54607/hcmue.js.20.5.3631(2023)

Bài báo nghiên cứu

TRUY VẤN ẢNH SỬ DỤNG R^S-TREE VÀ MẠNG HỌC SÂU R-CNN

Lê Thị Vĩnh Thanh[1], Nguyễn Thị Quỳnh Hương[2], Văn Thế Thành[3*]

[1]Trường Đại học Bà Rịa – Vũng Tàu, Việt Nam

[2]Trường THPT Chuyên Lê Quý Đôn, TP Vũng Tàu, Việt Nam

[3]Trường Đại học Sư phạm Thành phố Hồ Chí Minh, Việt Nam

[*]Tác giả liên hệ: Văn Thế Thành – Email: thanhvt@hcmue.edu.vn

Ngày nhận bài: 11-10-2022; ngày nhận bài sửa: 10-02-2023; ngày duyệt đăng: 21-02-2023

TÓM TẮT

Trong bài báo này, một mô hình tìm kiếm ảnh sử dụng cấu trúc R^S-Tree và mạng học sâu Faster R-CNN được đề xuất nhằm nâng cao hiệu suất truy vấn ảnh. Trong mô hình này, các công việc sau được thực hiện: (1) cấu trúc R^S-Tree được cải tiến thuật toán tách nút để nâng cao hiệu quả gom cụm các véc-tơ đặc trưng của tập ảnh đa đối tượng; (2) mạng học sâu Faster R-CNN được sử dụng để phát hiện và phân loại các đối tượng trên hình ảnh; (3) các hộp giới hạn chứa đối tượng trên ảnh được trích xuất đặc trưng cấp thấp và lưu trữ trên cấu trúc R^S-Tree; Với mỗi ảnh đầu vào, hệ thống phát hiện và phân loại từng đối tượng bằng mạng học sâu Faster R-CNN; trích xuất véc-tơ đặc trưng cấp thấp; thực hiện truy vấn ảnh tương tự dựa trên cấu trúc R^S-Tree. Thực nghiệm được thực hiện trên bộ ảnh đa đối tượng MS-COCO gồm 5000 ảnh với độ chính xác là 77.39%. Kết quả thực nghiệm được so sánh với các công trình khác trên cùng bộ ảnh nhằm đánh giá tính đúng đắn của mô hình đề xuất.

Từ khóa: clustering; image Retrieval; R-CNN; R^S-Tree

1. Giới thiệu

Tra cứu ảnh tương tự và phân lớp ngữ nghĩa hình ảnh là một trong những bài toán quan trọng và phù hợp với xu thế của xã hội hiện đại (Chou et al., 2016; Liu et al., 2015). Vì vậy, các hệ thống tìm kiếm ảnh tương tự được các nhà nghiên cứu quan tâm trong nhiều thập niên gần đây và có nhiều phương pháp khác nhau được đề xuất nhằm nâng hiệu quả tìm kiếm. Mô tả nội dung thị giác của hình ảnh và lập chỉ mục cho nội dung thị giác là hai vấn đề cần thiết khi thực hiện bài toán truy vấn ảnh theo nội dung (Begum & Supreethi, 2018; Sivakumar et al., 2021). Hiện nay, có nhiều phương pháp lập chỉ mục cho dữ liệu đa chiều như KD-Tree, QuarTree, M-Tree, R-Tree... Trong đó, R-Tree là một trong những cấu trúc được sử dụng phổ biến để lưu trữ chỉ mục dựa trên phân vùng dữ liệu (Manolopoulos et al., 2006). Trong những thập niên gần đây, các phương pháp học máy được áp dụng rộng rãi cho bài toán tìm kiếm ảnh nhằm nâng cao chất lượng truy vấn. Dữ liệu đa phương tiện ngày càng

Cite this article as: Le Thi Vinh Thanh, Nguyen Thi Quynh Huong, & Van The Thanh (2023). Image retrieval using R^s-Tree and R-CNN deep learning network. *Ho Chi Minh City University of Education Journal of Science, 20*(5), 842-854.

gia tăng nhanh theo thời gian là thách thức cho việc lưu trữ và tìm kiếm hiệu quả. Do đó, việc kết hợp các phương pháp khác nhau cho bài toán truy vấn ảnh cần được thực hiện nhằm nâng cao hiệu suất, giảm thời gian tìm kiếm cũng như tối ưu hóa không gian lưu trữ là cần thiết (Zhou et al., 2022).

Phát hiện đối tượng là một chủ đề được nhiều nhà khoa học quan tâm trong lĩnh vực thị giác máy tính. Mục đích chính của phát hiện đối tượng là tìm đối tượng quan tâm trong hình ảnh hoặc video, phát hiện vị trí và kích thước của chúng. Trong những năm gần đây, phát hiện đối tượng đã được sử dụng rộng rãi trong trí tuệ nhân tạo, nhận dạng khuôn mặt, lái xe không người lái và các lĩnh vực khác. Các thuật toán phát hiện đối tượng hiện có bao gồm thuật toán phát hiện truyền thống và thuật toán phát hiện dựa trên học sâu. Các thuật toán phát hiện đối tượng truyền thống chủ yếu dựa trên khung cửa sổ trượt hoặc đối sánh dựa trên các điểm đặc trưng. Với sự phát triển của công nghệ học sâu, các thuật toán phát hiện đối tượng đã chuyển từ phương pháp truyền thống dựa trên các đặc trưng được lựa chọn thủ công sang phương pháp phát hiện dựa trên mạng nơ-ron sâu. Các phương pháp phát hiện dựa trên mạng nơ-ron sâu có thể chủ yếu được chia thành hai loại: (1) thuật toán phát hiện đối tượng hai giai đoạn kết hợp mạng đề xuất vùng RPN (Region Proposal Network) và mạng nơ-ron tích chập (CNN), chẳng hạn như R-CNN; (2) thuật toán phát hiện đối tượng một giai đoạn chuyển đổi phát hiện đối tượng thành một bài toán hồi quy (ví dụ: YOLO). Đối với ảnh đa đối tượng, việc phát hiện đối tượng và phân lớp đối tượng trên ảnh là cần thiết để áp dụng cho các bài toán tìm kiếm ảnh nhằm nâng cao độ chính xác. Nhiều phương pháp học hiện đại được sử dụng để phát hiện và phân lớp ảnh đa đối tượng bao gồm: Mạng học sâu Fast R-CNN (Girshick, 2015), Faster R-CNN (Amitha & Narayanan, 2021; Ren et al., 2015) Yolo (Pestana et al., 2021) (Singh et al., 2021)...

Trong bài báo này, cấu trúc chỉ mục *Improved-RST*, một cải tiến của cấu trúc R^S-Tree (Le et al., 2022) được đề xuất. Trong cấu trúc *Improved-RST*, các véc-tơ đặc trưng hình ảnh được biểu diễn dưới dạng các khối cầu và được lưu trữ tại các nút lá của cây tương tự như R^S-Tree. Nhằm nâng cao hiệu quả lưu trữ và truy vấn trên cấu trúc *Improved-RST*, một phương pháp thêm phần tử vào cây được đề xuất để cải thiện thời gian tạo cây và giúp cân bằng dữ liệu tại các nút lá. Việc cải tiến bao gồm các nội dung sau: (1) Khi một nút đầy, một nút tràn cho nút đó được tạo ra và tất cả các nút tràn được lưu trong một bảng băm; (2) Nếu nút đó tiếp tục được thêm dữ liệu, dữ liệu sẽ được thêm vào nút tràn của nó; (3) Khi một nút tràn bị đầy, quá trình tách nút được thực hiện. Trên cơ sở đó, một mô hình truy vấn ảnh sử dụng cấu trúc *Improved-RST* và mạng học sâu Faster R-CNN được đề xuất để nâng cao hiệu quả tìm kiếm ảnh. Thực nghiệm truy vấn ảnh tương tự được thực hiện trên bộ ảnh MS-COCO gồm 5000 ảnh.

Các công trình nghiên cứu liên quan

Gần đây, mạng nơ-ron tích chập (*Convolution Neural Network - CNN*) đã được chứng minh là đạt được hiệu suất cao trong nhiều nhiệm vụ thị giác máy tính như phân loại hình ảnh (Simonyan & Zisserman, 2014), phát hiện đối tượng (Ren et al., 2015) hoặc phân đoạn

ngữ nghĩa (Long et al., 2015). Các mạng CNN được huấn luyện với lượng lớn dữ liệu đã được chứng minh là học được các biểu diễn đặc trưng tổng quát để sử dụng khi giải quyết các nhiệm vụ mà chúng chưa được huấn luyện (Wang et al., 2019). Đặc biệt đối với việc truy xuất hình ảnh, nhiều công trình đã áp dụng các giải pháp dựa trên các đặc trưng vượt trội được trích xuất từ một CNN được huấn luyện trước cho nhiệm vụ phân loại hình ảnh (Babenko & Lempitsky, 2015; Tolias et al., 2015).

Wenze Li (2021) và (Li, 2021) đã thực hiện phân tích hiệu suất của các mô hình Faster R-CNN dựa trên các mô hình tiền huấn luyện khác nhau và tiến hành đánh giá toàn diện về hiệu suất của Faster R-CNN. Kết quả thử nghiệm cho thấy độ chính xác và tốc độ phát hiện của R-CNN, Fast R-CNN và Faster R-CNN nhanh hơn dựa trên ba tập dữ liệu khác nhau. Thuật toán Faster R-CNN được thực hiện dựa trên Pytorch, VGG16 và ResNet101 được sử dụng làm mô hình tiền huấn luyện để huấn luyện và ghi lại thời gian trên hai tập dữ liệu Pascal VOC và COCO tương ứng. Hiệu suất của Faster R-CNN được phân tích theo các mô hình tiền huấn luyện và các tập dữ liệu khác nhau. Kết quả thử nghiệm chứng minh rằng độ chính xác và tốc độ phát hiện của Faster R-CNN được cải thiện rất nhiều so với R-CNN và Fast RCNN vì nó sử dụng RPN để thay thế bộ tìm kiếm chọn lọc.

Wenmei Wang và cộng sự (2019) (Wang et al., 2019) đã đề xuất một phương pháp dựa trên học sâu để truy xuất ảnh nhãn hiệu. Faster R-CNN được áp dụng trong việc truy xuất hình ảnh nhãn hiệu để trích xuất các đặc điểm ngữ nghĩa cấp cao của hình ảnh nhãn hiệu. Bộ mô tả đặc điểm toàn cục của hình ảnh được Faster R-CNN trích xuất và đặc điểm cục bộ của hình ảnh được trích xuất thông qua các vùng đề xuất đối tượng bởi mạng đề xuất khu vực (RPN). Để có được hiệu quả truy xuất tốt hơn, nhóm tác giả áp dụng kết hợp chiến lược truy xuất xếp hạng và sắp xếp lại theo không gian.

Amaia Salvador và cộng sự (2016) (Salvador et al., 2016) đã đề xuất sử dụng CNN tiền huấn luyện phát hiện đối tượng để trích xuất các đặc trưng toàn cục và cục bộ của hình ảnh. Nhóm tác giả sử dụng phương pháp xếp hạng lại và vận dụng các vị trí mà mạng đề xuất khu vực (RPN) học được để cung cấp định vị đối tượng cho các hình ảnh được truy xuất.

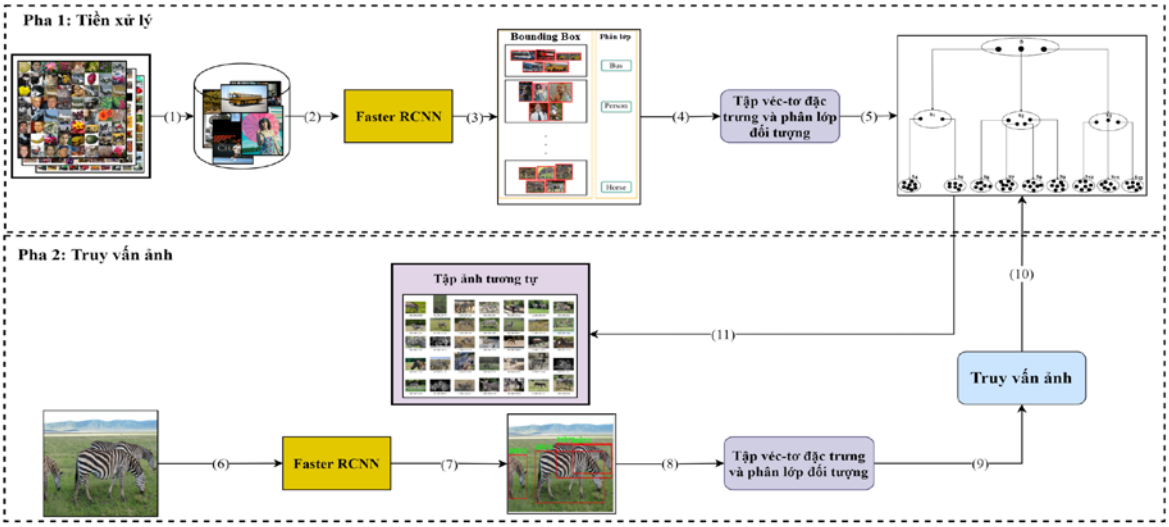
Bên cạnh đó, nhiều công trình đã áp dụng các kĩ thuật lập chỉ mục hình ảnh nhằm nâng cao tốc độ tìm kiếm ảnh. Haldurai và cộng sự (2015) đã đề xuất một hệ truy vấn ảnh tương tự theo nội dung sử dụng cấu trúc cây R-Tree (Haldurai & Vinodhini, 2015). Vanitha và cộng sự (2017) đã đề xuất một cấu trúc chỉ 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 (Vanitha & SenthilMurugan, 2017). 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 (Shama et al., 2015). 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 (Alfarrarjeh et al., 2020).

Từ việc phân tích các nghiên cứu liên quan ở trên cho thấy mô hình truy xuất ảnh dựa trên cấu trúc cây được đánh giá là đáng tin cậy và hiệu quả. Bên cạnh đó, những cách tiếp cận gần đây đã tập trung vào các phương pháp học máy cho bài toán truy vấn ảnh nhằm nâng cao độ chính xác. Kết quả của những công trình đó cho thấy việc áp dụng mạng học sâu Faster R-CNN cho bài toán truy xuất hình ảnh là khả thi. Trên cơ sở kế thừa và khắc phục những hạn chế của các công trình liên quan, một mô hình truy xuất hình ảnh sử dụng cấu trúc cây kết hợp mạng học sâu Faster R-CNN được giới thiệu để cải thiện hiệu suất truy xuất hình ảnh. Thử nghiệm được thực hiện xây dựng trên bộ dữ liệu ảnh MS-COCO (bao gồm 5000 hình ảnh, 80 lớp), để chứng minh hiệu quả truy vấn ảnh của phương pháp đề xuất.

2. Nội dung

2.1. Mô hình truy vấn ảnh dựa trên cấu trúc Improved-RST và mạng Faster R-CNN

Một ảnh cần truy vấn được phát hiện đối tượng và trích xuất véc-tơ đặc trưng bằng mạng học sâu Faster R-CNN và thực hiện truy vấn trên cấu trúc cây *Improved-RST*. Quá trình truy vấn trên cây cho đến khi gặp được nút lá phù hợp thì tập hợp tất cả các phần tử dữ liệu trong nút lá đó được gọi là một tập ảnh tương tự của ảnh truy vấn. Sau đó, tập ảnh này được sắp xếp theo độ đo tương tự để tìm ra các ảnh tương tự gần nhất. Mô hình truy vấn ảnh tương tự theo nội dung với một ảnh truy vấn đầu vào cho trước dựa trên cây *Improved-RST* được minh họa như Hình 1.



Hình 1. *Mô hình tìm kiếm ảnh dựa trên Improved-RST và mạng học sâu Faster R-CNN*

Quá trình tìm kiếm ảnh được thực hiện gồm hai pha, pha thứ nhất thực hiện gom cụm và lưu trữ dữ liệu hình ảnh trên cây *Improved-RST*, pha thứ hai thực hiện tìm kiếm các hình ảnh tương tự và phân lớp ngữ nghĩa 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 *Improved-RST* dựa trên véc-tơ đặc trưng của tập dữ liệu ảnh gồm 3 bước như sau:

Bước 1. Trích xuất véc-tơ đặc trưng f_i của tập dữ liệu ảnh sử dụng mạng Faster R-CNN.

Bước 2. Biểu diễn véc-tơ đặc trưng của tập dữ liệu ảnh thành khối cầu không gian.

Bước 3. Dựa trên độ đo tương tự đề xuất và phương pháp phân cụm K-mean, cây gom cụm chỉ mục không gian *Improved-RST* được tạo ra với mỗi nút lá của cây là tập các thực thể khối cầu chứa các véc-tơ f_i mô tả đặc trưng thị giác 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 và phân lớp ngữ nghĩa hình ảnh dựa trên cây gom cụm chỉ mục *Improved-RST*. Quá trình tìm kiếm ảnh tương tự được thực hiện theo các bước như sau:

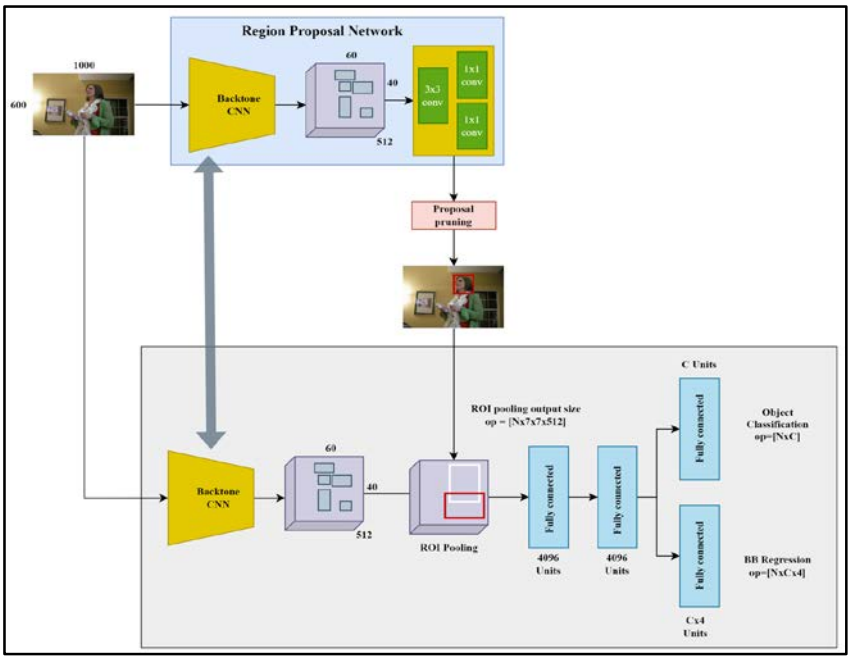
Bước 1. Trích xuất véc-tơ đặc trưng của ảnh cần truy vấn dựa trên mạng Faster R-CNN và chuyển đổi thành dạng khối cầu không gian.

Bước 2. Thực hiện truy vấn ảnh tương tự dựa trên cấu trúc không gian *Improved-RST*.

Bước 3. Tra cứu tập ảnh tương tự dựa trên tập chỉ mục đã được truy vấn.

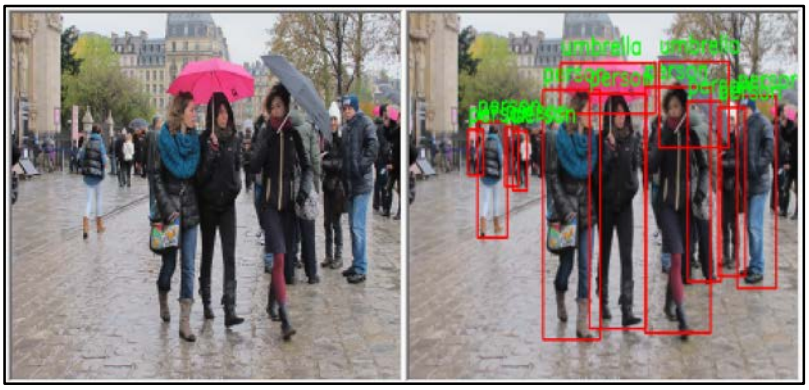
2.2. Mạng học sâu Faster R-CNN

Faster R-CNN là phiên bản hiện đại được sử dụng rộng rãi nhất của họ R-CNN. Các nhiệm vụ của mạng này bao gồm: (1) Thuật toán đề xuất vùng để tạo ra các "hộp giới hạn" hoặc vị trí của các đối tượng có thể có trong hình ảnh; (2) Giai đoạn tạo đặc trưng để có được các đặc trưng của các đối tượng này, thường sử dụng CNN; (3) Lớp phân loại để dự đoán đối tượng này thuộc lớp nào; (4) Lớp hồi quy để làm cho tọa độ của hộp giới hạn đối tượng chính xác hơn. Fast R-CNN đã sử dụng thuật toán tìm kiếm chọn lọc dựa trên CPU cho đề xuất vùng, mất khoảng 2 giây cho mỗi hình ảnh và chạy trên tính toán của CPU. Faster R-CNN khắc phục điều này bằng cách sử dụng mạng đề xuất khu vực (RPN) để tạo các đề xuất khu vực. Điều này làm giảm thời gian đề xuất khu vực từ 2 giây xuống 10 mili giây cho mỗi hình ảnh. Kiến trúc của Faster R-CNN được thể hiện trong Hình 2.



Hình 2. *Kiến trúc Faster R-CNN*

Trong bài báo này, mạng Faster R-CNN như trong công trình (Ren et al., 2015) được sử dụng để phát hiện và phân lớp đối tượng trong hình ảnh. Kết quả của quá trình này là một bộ các véc-tơ đặc trưng của các đối tượng trên hình ảnh và các phân lớp của chúng. Tập các thông tin này được sử dụng để gom cụm trên cấu trúc *Improved-RST*. Một ví dụ của việc phân lớp ảnh đầu vào dựa trên mạng Faster R-CNN được minh họa như Hình 3.



Hình 3. *Phân lớp đối tượng bằng mạng Faster R-CNN*

Mỗi đối tượng sau khi được trích xuất bằng mạng Faster R-CNN, các đặc trưng cấp thấp được trích xuất riêng biệt dựa trên khối chữ nhật chứa đối tượng đó. Các đặc trưng này được đưa vào lưu trữ trên cấu trúc R^S-Tree cải tiến.

2.3. Cấu trúc cây RS-Tree cải tiến

Quá trình gom cụm các véc-tơ đặc trưng dựa trên cấu trúc R^S-Tree (Le et al., 2022). Các nút trên R-Tree bao gồm: (1) 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 thiểu là 2 và tối đa là N; (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 $m(1 < m \leq {}^{M}/_{2})$.

Trên cơ sở cấu trúc R^S-Tree đã được đề xuất, trong bài báo này một cấu trúc cải tiến được trình bày như sau nhằm nâng cao hiệu quả lưu trữ và tìm kiếm ảnh. Để giảm sự chồng chéo giữa các nút trung gian và nâng cao chất lượng của R^S-Tree, cần phải cải thiện hai quá trình của R^S-Tree đó là: (1) quá trình thêm dữ liệu vào cây; (2) quá trình tách nút trên cây. Một phương pháp thêm phần tử vào cây dựa trên việc kết hợp bảng băm nhằm nâng cao chất lượng truy vấn và cân bằng dữ liệu trên cây được đề xuất trong phần 3.3.

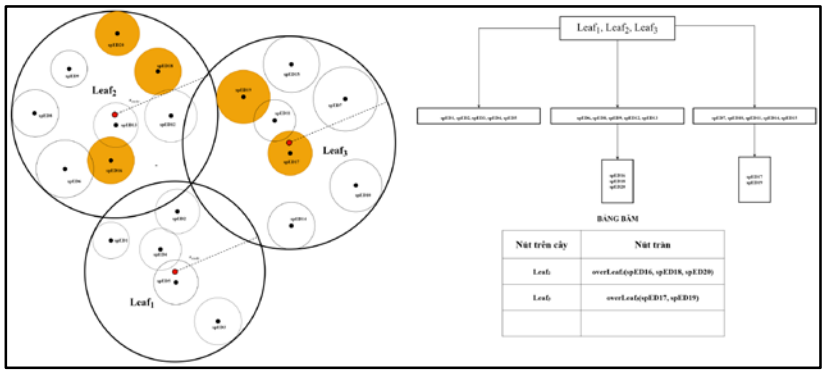
2.4. Các thuật toán cải tiến trên cấu trúc cây Improved-RST

Việc tách nút thường xuyên dẫn đến làm tăng chiều cao của cây và gia tăng vùng chồng lắp không gian, ảnh hưởng đến hiệu quả truy vấn. Để cải thiện hai vấn đề này, bài báo đề xuất chiến lược thuật toán tách nút có độ trễ để tối ưu hóa quy trình tạo cây R^S-Tree.

Thuật toán được chia thành hai phần: thêm dữ liệu và tách nút. Trong quá trình thêm, việc phân tách có độ trễ được áp dụng. Trong quá trình phân tách nút, một thuật toán tách tốt hơn được áp dụng để thực hiện phân tách có độ trễ. Thuật toán hoàn chỉnh làm giảm số lần phân tách xảy ra trong quá trình xây dựng cây R^S-Tree và tăng sự phân chia không gian của cây R^S-Tree, do đó cải thiện hiệu quả tìm kiếm.

Trong quá trình thêm dữ liệu vào cây, để chọn một nút lá phù hợp để thêm vào, tiêu chuẩn là giảm thiểu diện tích của các khối cầu của các nút khi dữ liệu được thêm vào. Khi một nút đã đầy, tức là số lượng dữ liệu mà nút chứa đã đạt đến M, thuật toán tách nút không thực hiện ngay mà thay vào đó sẽ thêm dữ liệu vào một nút tràn. Khi quá trình thêm kết thúc, nút tràn được xem xét đã đầy hay không; nếu đầy quá trình hợp nhất và tách nút được thực hiện: nút hiện hành và nút tràn của nó được hợp nhất dữ liệu gồm có 2M phần tử. Sau đó, thuật toán tách nút được thực hiện tương tự như trong thuật toán tách nút của cây R^S-Tree.

Để thực hiện thao tác thêm, một bảng băm được sử dụng, kí hiệu *HM*, chứa các nút tràn để ghi lại các dữ liệu. Khi thêm dữ liệu vào nút chứa dữ liệu, thuật toán đầu tiên xác định xem nút có đầy hay không, nếu không, nó sẽ thêm dữ liệu trực tiếp và cập nhật khối cầu của nút đó. Nếu nút đầy, thuật toán sẽ kiểm tra xem nút đó có nút tràn trong bảng băm hay không. Nếu không có nút tràn nào tồn tại trong bảng băm, một nút tràn sẽ được tạo cho nút đó và dữ liệu cần thêm sẽ được thêm vào nút tràn. Sau đó, nó cập nhật khối cầu của nút hiện hành và khối cầu của nút tràn tương ứng. Khi nút đã có nút tràn, thuật toán sẽ thêm dữ liệu trực tiếp vào nút tràn, cập nhật khối cầu của nút và khối cầu của nút tràn và khi quá trình thêm hoàn tất, sẽ kiểm tra xem nút tràn có đầy hay không. Khi nút tràn của nút hiện tại đã đầy, bước tiếp theo là thực hiện thao tác tách-gộp bằng cách chia nhỏ dữ liệu của nút và nút tràn thành hai nút, mỗi nút có khối lượng dữ liệu M, ghi chúng trở lại cây *Improved-RST* và loại bỏ nút khỏi các nút tràn trong bảng băm *HM*. Hình 4 minh họa cấu trúc của nút tràn và bảng băm.



Hình 4. *Một minh họa của cấu trúc cây Improved-RST và nút tràn sử dụng bảng băm*

Khi một phần tử $spED$ được thêm vào cây R^S-Tree, việc thêm được thực hiện bắt đầu từ nút gốc, lần lượt duyệt qua tất cả các nút con của nút gốc và đi theo gần nhất cho đến khi tìm được nút lá. Sau đó, phần tử $spED_i$ được phân phối vào nút lá hiện hành thỏa điều kiện $d_E(spED.\vec{c}_{sp}, S_{leaf}.\vec{c}_{leaf}) < \theta$. Ngược lại, một nút lá mới $S_{newleaf}$ được tạo cùng cha với nút lá hiện hành để lưu phần tử $spED$. Khi gặp một nút đầy thì một nút tràn được tạo trong bảng băm để lưu trữ phần tử này. Thuật toán thêm phần tử vào cây được trình bày như trong

Thuật toán 1.

Thuật toán 1: Thêm phần tử vào cây

Đầu vào: cây *Improved-RST*, nút S_N và *spED* là phần tử sẽ được thêm vào

Đầu ra: cây *Improved-RST* được cập nhật

Function. InsertspED (*Improved-RST*, *spED*)

Begin

1. **Nếu** S_N không phải nút lá **thì**

Chọn nhánh gần với phần tử được thêm vào theo độ đo Euclidean cho đến khi gặp được nút lá hiện hành S_{Lcrt};

2. **Nếu** S_{Lcrt} có số phần tử nhỏ hơn M **thì**

Tính khoảng cách Euclidean $d = Euclidean(S_{spED}.\vec{c}, S_{Lcrt}.\vec{c}) + S_{spED}.r$;

3. **Nếu** $d < \theta$ **thì**

Thêm phần tử $spED$ vào nút lá S_{Lcrt};

Cập nhật tâm cụm;

4. **Ngược lại**

Tạo một nút lá mới S_{Lnew} cùng cha với nút lá hiện hành;

Thêm phần tử S_{spED} vào nút lá mới S_{Lnew};

Cập nhật tâm cụm;

5. **Nếu** S_N có số phần tử bằng M **thì**

Gán O ← S_N vào trong bảng nút tràn *HM*;

6. **Nếu** O khác null **thì**

Thêm *spED* vào O;

Cập nhật tâm cụm cho nút S_N và nút O;

7. **Nếu** O đầy **thì**

Gọi split(S_N) tách nút S_N và nút tràn O thành hai nút;

Gán hai nút mới vào cây *Improved-RST*;

Loại bỏ O khỏi HM;

8. **Ngược lại**

Tạo một nút tràn O mới cho nút S_N trong bảng nút tràn *HM*;

Thêm phần tử *spED* vào O;

Cập nhật tâm cụm cho nút S_N và nút O;

9. **Trả về** *Improved-RST*;

End

Gọi n là số phần tử của tập dữ liệu, M là số phần tử tối đa trong một nút của *Improved-RST*. Thuật toán **InsertspED** lần lượt thực hiện duyệt từ nút gốc đến nút lá, mỗi lần duyệt qua M phần tử, mỗi lần duyệt thuật toán **InsertspED** thực hiện phép cập nhật tâm và tách nút từ nút lá đến nút gốc. M là hằng số. Do đó, thuật toán **InsertspED** có độ phức tạp là $O(logn)^3$.

Việc nút O bị tràn được xử lí bằng cách trộn các phần tử của nút O và nút S_N lại gồm có 2*M phần tử, sau đó thực hiện việc tách nút này thành hai nút lá mới S_{L1}, S_{L2}, cùng mức với S_N, phân bố $2 * M$ phần tử vào hai nút. Thuật toán tách nút được trình bày như trong

Thuật toán 2.

Thuật toán 2: Tách nút trên cây *Improved-RST*

Đầu vào: S_N, nút được tách.

Đầu ra: S_{L1} và S_{L2}, hai nút mới.

Function: IRSTsplit (S_N)

Begin

1. Tạo nút W;
2. Gán O ← S_N vào trong bảng nút tràn *HM;*
3. Gán W = O ∪ S_N;
4. Tạo nút lá S_{L1};
5. Tạo nút lá S_{L2};
6. Chọn hai phần tử xa nhau nhất trong nút W giả sử là $spED_k$ và $spED_t$;
7. Đưa $spED_k$ vào nút lá S_{L1}; Cập nhật tâm và bán kính;
8. Đưa $spED_t$ vào nút lá S_{L2}; Cập nhật tâm và bán kính;
9. Xóa hai phần tử này ra khỏi W;
10. Gọi hàm phân phối các phần tử vào hai nút;
11. $DistributeSpED(W, S_{L1}, S_{L2})$;
12. S_{Lpr}=$S_N.parent$;
13. **Nếu (**Số phần tử nút cha S_{Lpr} nhỏ hơn N**) thì**
14. Cập nhật tâm nút cha;
15. **Ngược lại**
16. Tách nút cha;
17. endif

End.

Gọi n là số phần tử của tập dữ liệu, M là số phần tử tối đa trong một nút *Improved-RST*. Khi thực hiện tách nút, trong trường hợp xấu nhất, thuật toán **IRSTsplit** phải tách từ nút lá đến nút gốc. Mỗi lần tách nút, thuật toán **IRSTsplit** phải thực hiện M phép so sánh để phân bố về k-cụm. Mặt khác, trong trường hợp xấu nhất thuật toán **IRSTsplit** phải gọi đệ quy từ nút lá đến nút gốc, M là hằng số. Do đó, độ phức tạp của thuật toán **IRSTsplit** là $O(logn)^2$.

2.5. Thuật toán tìm kiếm ảnh tương tự

Từ cây phân cụm dữ liệu không gian *Improved-RST* đã tạo, một thuật toán tra cứu ảnh tương tự theo nội dung dựa trên cây *Improved-RST* được đề xuất. Quá trình tìm kiếm ảnh tương tự được thực hiện trên cây *Improved-RST* và được mô tả như trong **Thuật toán 3**.

Thuật toán 3. Truy vấn ảnh tương tự

Đầu vào: vec-tơ đặc trưng S_{spED} của ảnh truy vấn, cây *Improved-RST*.

Đầu ra: tập ảnh tương tự SI;

Function: **IRSTIR** (S_{Nr}, S_{spED})

Begin

1. $S_{Node} = S_{Nr}$;
2. **Nếu** S_{Node} trỏ tới phần tử *null*) **thì**
3. **Trả về** *null*;
4. **Ngược lại**
5. **Nếu** S_N không phải lá nút lá **thì**
6. Đi theo nhánh gần nhất với phần tử truy vấn kí hiệu S_{Nk};
7. IRSTIR (S_{Nk}, $spED$);
8. **Ngược lại**
9. $SI = \{$*Các phần tử ảnh tương tự trong nút lá hiện hành* $S_{Lcrt}\}$;
10. *Trả về tập ảnh tương tự* $\{SI\}$;

End

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

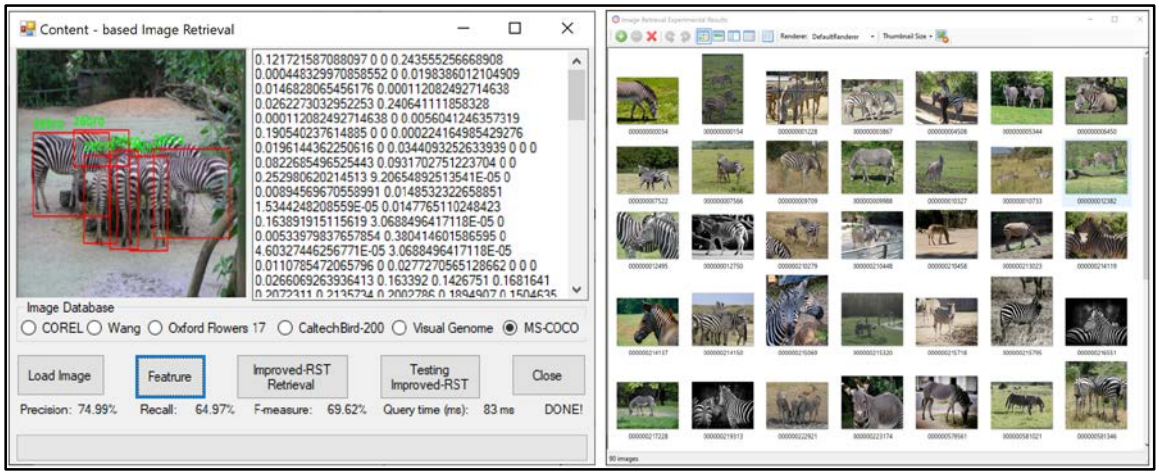
3. Kết quả thực nghiệm và đánh giá

3.1. Môi trường thực nghiệm

Pha tiền xử lí 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.

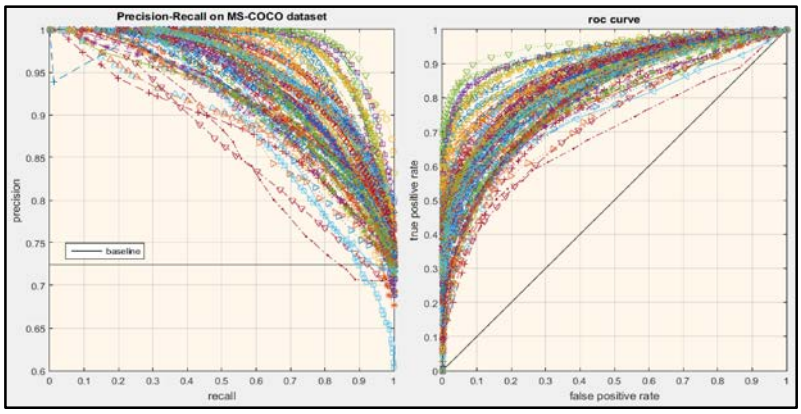
3.2. Ứng dụng và kết quả thực nghiệm

Trong bài báo này, chúng tôi tiến hành thực nghiệm trên bộ ảnh MS-COCO gồm 5000 ảnh được chia thành 80 phân lớp. Ứng dụng thực nghiệm được minh họa trong Hình 5.



Hình 5. *Giao diện truy vấn ảnh trên cấu trúc Improved-RST*

Để đánh giá hiệu quả của phương pháp tìm kiếm ảnh, phần thực nghiệm được đánh giá các giá trị gồm: độ chính xác (precision), độ phủ (recall) và độ đo dung hòa F-measure. Kết quả thực nghiệm được thể hiện như trong Hình 6.



Hình 6. *Độ chính xác, độ phủ và đường cong ROC của bộ dữ liệu MS-COCO*

Mỗi đường cong trên đồ thị mô tả kết quả truy vấn từ một chủ đề ảnh trong bộ dữ MS-COCO, mỗi điểm trên đường cong là một hình ảnh theo từng chủ đề. Đồng thời, đường cong tương ứng trong đồ thị ROC cho biết tỷ lệ kết quả truy vấn đúng và sai, nghĩa là diện tích dưới đường cong này đánh giá tính đúng đắn của các kết quả truy vấn. Diện tích AUC dưới đường cong của đồ thị ROC nằm trên đường baseline, cho thấy kết quả phân loại trong bài báo của chúng tôi là đúng.

Để đánh giá hiệu suất của phương pháp đề xuất, chúng tôi so sánh kết quả thực nghiệm với các công trình trước đây trên bộ ảnh MS-COCO được trình bày trong bảng sau

Bảng so sánh độ chính xác trung bình với các phương pháp khác

Methods	**MAP**	**Dataset**
Cao, Y., 2018 (Cao et al., 2018)	70.13	MS-COCO
Cao, Z., 2017 (Cao et al., 2017)	73.62	MS-COCO
CBIR-iRST	**77.39**	**MS-COCO**

4. Kết luận và kiến nghị

Trong bài báo này, chúng tôi đã xây dựng một cấu trúc *Improved-RST*, một cải tiến của cấu trúc R^S-Tree, áp dụng cho bài toán tìm kiếm ảnh. Trong cấu trúc này, thuật toán thêm phần tử được cải tiến để cân bằng dữ liệu trong cây, đồng thời hạn chế thời gian tách nút trên cây giúp tăng thời gian tạo cây; thuật toán thêm phần tử vào cây được cải tiến bằng cách sử dụng bảng băm để lưu trữ các phần tử thuộc nút tràn nhằm tạo độ trễ cho quá trình tách nút. Từ đó, một mô hình truy vấn ảnh tương tự được đề xuất kết hợp mạng học sâu Faster R-CNN và cấu trúc *Improved-RST* để nâng cao hiệu suất truy vấn. Trong mô hình này, mạng học sâu Faster R-CNN được sử dụng để phân lớp cho tập dữ liệu ảnh và ảnh đầu vào. Các đối tượng được trích xuất đặc trưng và được lưu trữ trên cấu trúc cây *Improved-RST* nhằm nâng cao hiệu quả cho việc tìm kiếm tập ảnh tương tự. Kết quả thực nghiệm trên bộ dữ liệu ảnh MS-COCO có độ chính xác là 77.39%. Theo kết quả thực nghiệm cho thấy tính hiệu quả so với các công trình khác trên cùng một tập dữ liệu ảnh.

❖ ***Tuyên bố về quyền lợi:*** *Các tác giả xác nhận hoàn toàn không có xung đột về quyền lợi.*

TÀI LIỆU THAM KHẢO

Alfarrarjeh, A., Kim, S. H., Hegde, V., Shahabi, C., Xie, Q., & Ravada, S. (2020). A Class of R-tree Indexes for Spatial-Visual Search of Geo-tagged Street Images. *2020 IEEE 36th international conference on data engineering (ICDE).*

Amitha, I., & Narayanan, N. (2021). Collaborative MSER and Faster R-CNN Model for Retrieval of Objects in Images. In *Soft Computing for Problem Solving* (pp. 673-682). Springer.

Babenko, A., & Lempitsky, V. (2015). Aggregating local deep features for image retrieval. *Proceedings of the IEEE international conference on computer vision*.

Begum, S. A. N., & Supreethi, K. (2018). A survey on spatial indexing. *Journal of Web Development and Web Designing*, *3*(1).

Cao, Y., Long, M., Liu, B., & Wang, J. (2018). Deep cauchy hashing for hamming space retrieval. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

Cao, Z., Long, M., Wang, J., & Yu, P. S. (2017). Hashnet: Deep learning to hash by continuation. *Proceedings of the IEEE international conference on computer vision.*

Chou, Y., Lee, D. J., & Zhang, D. (2016). Semantic-Based Brain MRI Image Segmentation Using Convolutional Neural Network. *International Symposium on Visual Computing*.

Girshick, R. (2015). Fast r-cnn. *Proceedings of the IEEE international conference on computer vision.*

Haldurai, L., & Vinodhini, V. (2015). Parallel indexing on color and texture feature extraction using r-tree for content based image retrieval. *International Journal of Computer Sciences and Engineering*, *3*, 11-15.

Le, T. V. T., Le, M. T. & Van, T. T. (2022). Semantic-Based Image Retrieval Using RS-Tree and Neighbor Graph. *WorldCIST*, (2).

Li, W. (2021). Analysis of object detection performance based on Faster R-CNN. *Journal of Physics: Conference Series.*

Liu, G.-H., Yang, J.-Y., & Li, Z. (2015). Content-based image retrieval using computational visual attention model. *Pattern Recognition*, *48*(8), 2554-2566.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. P*roceedings of the IEEE conference on computer vision and pattern recognition.*

Manolopoulos, Y., Papadopoulos, A. N., Papadopoulos, A. N., & Theodoridis, Y. (2006). *R-Trees: Theory and Applications: Theory and Applications.* Springer Science & Business Media.

Pestana, D., Miranda, P. R., Lopes, J. D., Duarte, R. P., Véstias, M. P., Neto, H. C., & De Sousa, J. T. (2021). A full featured configurable accelerator for object detection with YOLO. *IEEE Access*, *9*, 75864-75877.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems, 28.*

Salvador, A., Giró-i-Nieto, X., Marqués, F., & Satoh, S. i. (2016). Faster r-cnn features for instance search. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*.

Shama, P., Badrinath, K., & Tilugul, A. (2015). An Efficient Indexing Approach for Content based Image Retrieval. *International Journal of Computer Applications*, *117*(15).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh, S., Ahuja, U., Kumar, M., Kumar, K., & Sachdeva, M. (2021). Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment. *Multimedia tools and applications*, *80*(13), 19753-19768.

Sivakumar, M., Kumar, N. S., & Karthikeyan, N. (2021). Content-Based Image Retrieval Techniques: A Survey. *Journal of Physics: Conference Series.*

Tolias, G., Sicre, R., & Jégou, H. (2015). Particular object retrieval with integral max-pooling of CNN activations. *arXiv preprint arXiv:1511.05879*.

Vanitha, J., & SenthilMurugan, M. (2017). An efficient content based image retrieval using block color histogram and color co-occurrence matrix. *Int. J. Appl. Eng. Res*, *12*(24), 15966-15971.

Wang, W., Xu, X., Zhang, J., Yang, L., Song, G., & Huang, X. (2019). Trademark Image Retrieval Based on Faster R-CNN. *Journal of Physics: Conference Series.*

Zhou, X., Han, X., Li, H., Wang, J., & Liang, X. (2022). Cross-domain image retrieval: methods and applications. *International Journal of Multimedia Information Retrieval*, 1-20.

IMAGE RETRIEVAL USING R^S-TREE AND R-CNN DEEP LEARNING NETWORK

Le Thi Vinh Thanh[1], Nguyen Thi Quynh Huong[2], Van The Thanh[3*]

[1]Ba Ria – Vung Tau University, Vietnam

[2]Le Quy Don High School For The Gifted Students, Ba Ria – Vung Tau, Vietnam

[3]Ho Chi Minh City University of Education, Vietnam

[*]Corresponding author: Van The Thanh – Email: thanhvt@hcmue.edu.vn

Received: October 11, 2022; Revised: February 10, 2023; Accepted: February 21, 2023

ABSTRACT

In this paper, an image retrieval model using the R^S-Tree and the Faster R-CNN deep learning network is proposed to enhance query performance. In this model, the following tasks are performed: (1) the R^S-Tree is improved by the node separation algorithm to enhance the efficiency of the clustering vectors feature of the multi-object image set; (2) the R-CNN deep learning network is used to detect and classify objects in images; (3) bounding boxes containing objects in an image is extracted with low-level features and stored on the R^S-Tree. For each input image, the system detects and classifies each object using the Faster R-CNN deep learning network, extracts low-level features of the image, and performs a process of image retrieval based on the R^S-Tree. The experiment is performed on the MS-COCO multi-object image set of 5000 images with an accuracy of 77.39%. The experimental results are compared with related works to demonstrate the effectiveness of the proposed model.

Keywords: clustering; image Retrieval; R-CNN; R^S-Tree