JTE HCMUTE
ISSN: 2615-9740

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

Efficiency Evaluation of C-Tree and KD-Tree in Content-Based Image Retrieval

Thi Dinh Nguyen[1], **Thanh Manh Le**[2], **Thi Uyen Nhi Nguyen**[3*]

[1]*Ho Chi Minh City University of Industry and Trade, Vietnam*
[2]*University of Sciences, Hue University, Vietnam*
[3]*University of Economics, The University of Da Nang, Vietnam*

[*]Corresponding author. *Email: nhintu@due.udn.vn*

ARTICLE INFO

Received:	25/03/2025
Revised:	01/04/2025
Accepted:	28/04/2025
Published:	28/08/2025

KEYWORDS

C-Tree;

KD-Tree;

Image retrieval;

Image classification;

Precision.

ABSTRACT

The image retrieval problem is performed on the C-Tree and KD-Tree structures which have brought many positive results regarding retrieval time and precision. The C-Tree structure is built according to the data clustering method, while the KD-Tree is built according to the multi-layer data classification method. Therefore, the common goal of these two structures applies to the image retrieval problem with quite high efficiency. In this paper, the results obtained from the two C-Tree and KD-Tree structures are evaluated, analyzed, and compared to the image retrieval problem. The experiments are conducted on the same COREL and WANG image data sets to serve as a basis for evaluating the performance of these two structures together; At the same time, the results are also compared with other works to demonstrate the effectiveness of the experimental method. Finally, some disadvantages of each structure are also analyzed for further improvements, and these two structures are combined to propose an image retrieval model to improve accuracy based on the advantages of the C-Tree and KD-Tree.

Doi: https://doi.org/10.54644/jte.2025.1860

1. Introduction

The research on image data has been an intensive problem in multimedia databases in recent decades. Digital images increase over time, which is a challenge for storage and retrieval. To solve this problem, the emergence of tree or graph data structures is necessary to improve storage efficiency and query time [1]. It is necessary to integrate many solutions and machine learning techniques to build a storage structure that satisfies the optimal requirements for memory space and retrieval performance. In this paper, the C-Tree and KD-Tree structures have been constructed and applied to store and retrieve related images, which we need to analyze, compare, and evaluate the effectiveness of these two structures. At the same time, detect the remaining limitations to have improved solutions, combined with some other techniques in the future to enhance the performance of the image retrieval problem.

The works [9] – [11] and [15] – [17] experimented on the COREL image set, and the works [12] – [14] and [18] – [19] experimented on the COREL image set with the results of similar image search accuracy not reaching 70%. Therefore, the solution of building the C-Tree, and KD-Tree for the image search problem is appropriate and needs to be implemented.

The image retrieval problem is performed by many methods, from improvements in content-based image retrieval to semantic image retrieval. In which, machine learning techniques are applied to image retrieval models that are quite effective in terms of accuracy and time. Storage organization and arrangement techniques are important factors contributing to improving the efficiency of image retrieval. In this paper, the C-Tree clustering structure and the KD-Tree layered structure have achieved positive results for the content-based image retrieval problem.

The contributions of the paper are: (1) Comparing the methods of construction, storage, and searching on C-Tree and KD-Tree; (2) Analyzing, evaluating, and comparing the experimental results of searching images by content on two datasets COREL, WANG with two structures C-Tree and KD-Tree; (3) Comparing the experimental results on C-Tree, KD-Tree with other works to confirm the correctness of the experimental method; (4) Detecting the remaining limitations on C-Tree, KD-Tree to have further improvement proposals.

2. Related works

Retrieving similar images in huge image databases is one of the major challenges in computer vision. The need for an efficient data storage structure that can be searched quickly, stably, and with high accuracy without incurring too much computational cost is an issue that has received much research attention. In particular, data clustering is a simple, popular, and effective method to store data without incurring too much complexity. When the size of the image database increases too quickly, the number of clusters also increases rapidly, inserting or searching for data takes a long time, and the computational complexity is high due to having to manage all the data clusters. Therefore, the clustering tree structure is one of the powerful tools to organize and manage image data. Recent studies on clustering methods and clustering tree structures in image retrieval problems show that this is an issue that receives much attention.

Machine learning techniques play a core role in the implementation of related problems such as social network analysis, building application models for social network analysis, image analysis, image retrieval, etc. However, the storage structure is a part that contributes to the implementation of the image retrieval problem, helping to optimize storage and search. In recent decades, tree and graph structures have made many contributions to the image retrieval problem, especially by integrating many semi-supervised machine-learning techniques into the construction process. Therefore, quite a few works applying the C-Tree, and KD-Tree structures have brought certain results, some typical works that need to be mentioned such as:

Zang H. (2018) et al. [2] proposed a method for feature extraction and organizing these vectors on a hierarchical tree structure (2-level) using the K-means. The JSEG method is used for image segmentation, and the features of each region are extracted based on HSV color histograms to form a 48-dimensional vector. The root node represents the entire image, and the child nodes represent the regions segmented from the original image. This method was experimented on the COREL-1k with a precision of 0.5052, the number of clusters needs to be determined in advance so it is not suitable for rapidly changing data, creating child nodes by assigning to the nearest cluster, reducing the discriminative power between different child nodes. However, the dimensionality of the feature vector is low so the average query time is fast (19.32 ms).

Ghodratnama S. et al. (2021) [3] introduced a novel content-based image retrieval (CBIR) method that uses feature weights and C-means clustering in a multi-label classification framework. Images are clustered using a supervised C-means algorithm and feature weights are assigned in each cluster to reduce the classification error. A query image is matched to the closest cluster and retrieval is performed using the weighted features of that cluster. Experimental results on Corel1000, Corel5000, and Scene datasets demonstrate that this method outperforms conventional CBIR methods by effectively utilizing feature weights in a multi-label context, thus achieving high precision and recall rates.

Joseph et al. (2021) [4], The authors proposed a content-based image retrieval (CBIR) system that uses a hybrid clustering method that combines K-means with the Moth Flame Optimization (MFO) algorithm, aiming to improve retrieval efficiency by reducing the search space. Low-level features such as color moments, HSV color histogram, color correlogram, GLCM, and wavelet transform are extracted to represent image content. The hybrid approach first uses MFO to optimize the clustering of centers and then applies K-means clustering to partition the dataset, which reduces the computational cost and improves the retrieval speed. When tested on the Corel-1K dataset, the hybrid KMFO method shows higher accuracy and faster response time than traditional CBIR techniques.

Varnish, N. (2022) [5] presented an improved content-based image retrieval (CBIR) approach by merging color, texture, and shape features into a low-dimensional feature descriptor using histogram-based color moments, texture features using Gray-Level Co-occurrence Matrix (GLCM) in the Discrete Cosine Transform (DCT) domain, and shape features through multi-resolution sub-images. These low-level features are clustered with K-means and organized in an R-tree structure. The approach was evaluated on Corel-1k datasets with an average accuracy of 70.87%. Advantages of this approach include its simplicity and reduced computational cost, as it does not require complex processing.

However, it has limitations in structuring element size, which needs customization per dataset, limiting scalability.

The image classification problem using the KD-Tree structure was implemented by the group of authors Y Narasimhulu and colleagues in 2021 [6] with positive results, starting a series of subsequent results when applying KD-Tree. First, a large unclassified data set was used to build the Coresets data set based on the classification algorithm proposed by the author. Then, based on the KD-Tree construction algorithm, a KD-Tree tree containing the classified data set was formed. After building the KD-Tree, each input image was searched according to the largest number of neighbors as a basis for determining the classification of the image. Finally, the author used the distance scale to classify the training image data sets. In this work, the KD-Tree tree is used directly to store data and classify an input image with good results without much additional cost. This is a proposed work for an image classification problem based on the KD-Tree structure which is considered quite good.

A survey result on the application of KD-Tree conducted by author Sumeet Gill and colleagues in 2021 [1] surveyed and summarized quite complete KD-Tree structural variations applicable to many cases in different problems, specifically: Movies (2009) [1] is a multidimensional index structure for storing objects with moving properties over time such as videos, requiring indexing, storage and query techniques on large data sets. Movies meet the need to update data over time, with low query time costs. Therefore, Movies handle moving objects well by considering images taken around the moving object to predict the properties and actions of the object to be surveyed. This work is developed and applied well to real-time video data. Randomly Projected K-d Trees (2011) [1] proposed a technique to build multiple KD-Tree trees into random forests for application to multi-object image retrieval problems. Buffer k-d Trees (2014) [1] is a KD-Tree structure built based on combining neighbor search techniques and focusing on GPU (Graphics Processing Units) memory in fast image processing. Progressive k-d Tree (2017) [1] is an improved structure from the classic KD-Tree by using the k-NN neighbor search algorithm in data retrieval. Sumeet Gill has conducted a fairly complete survey to help readers visualize the most general view possible while bringing great scientific value to the KD-Tree structure and its application to specific types of problems [1].

3. The structure of C-Tree and KD-Tree for image retrieval problem

In this section, the C-Tree structure is constructed using the data clustering method, and the KD-Tree is built using the multi-layer data classification method from the root to the leaf. Each of these structures has its own characteristics presented in sections 3.1 and 3.2.

3.1. The C-Tree structure for image retrieval problem

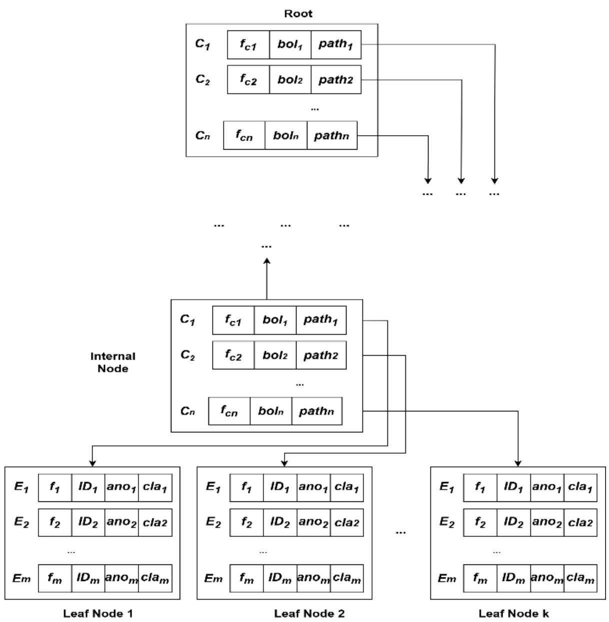

Figure 1. *Illustration of the C-Tree structure*

C-Tree [7] is a balanced clustering tree structure that uses hierarchical and partition-based clustering methods to store low-level feature vectors of images. C-Tree can store large amounts of data on external memory and supports fast searching. The balanced clustering tree structure of the C-Tree is illustrated in Figure 1.

The balanced clustering tree, C-Tree, is described as follows:

- The C-Tree consists of a root node, internal nodes I, and leaf nodes L. The nodes are connected by paths (links).
- Nodes contain elements that are similar to each other and are clustered based on the Euclidean distance measure.
- All leaf nodes L are at the same depth (balancing condition).
- The height of the C-Tree increases from the root node.
- The leaf node L is a node without child nodes and contains a maximum of M data elements E: $L = \{E_i, 1 \le i \le M\}$, E_i is the i-th data element in the leaf node L with:

$$E_i =< f, ID, ano, cla >$$

where f is low-level features vectors of images, ID is the image identifier, ano is the file containing image annotations, and cla represents the image subclasses, which are mapped from the set of image labels.

- A internal node I has at least 2 child nodes and contains a maximum of N centroid elements C_j: $I = \{C_j, 2 \le j \le N\}$, with:

$$C =< f_c, bol, path >$$

where f_c is the centroid vector of the low-level features vectors f at a child node that has a $path$ to EC and bol is the value to check whether the next sub-cluster is leaf or not (Leaf – 1; Not Leaf – 0);

Image data is continuously increasing, so the C-Tree requires scalability to accommodate increasing image data. The storage capacity of the C-Tree must correspond to the increase in the number of images, ensure proper distribution, and enable fast searching of similar image datasets. Therefore, data elements will be inserted, and split nodes on C-Tree to expand data storage. Initially, when the root node is empty $root = \emptyset$, the root acts as a leaf node $root = Lroot$, and data elements E are added to the root node: $Lroot = \{E_i | i = 1..M\}$. If $i > M$ then $Lroot$ splits the node into two nodes, creating a new root: $root = \{C_j | j = 2..N\}$. At this time, the $root$ plays the role of an internal node, containing at least two elements C. Thus, the C-Tree is formed from the root node and branches are created through the node splitting process. Data elements E are added to the tree according to the rule of choosing the branch with the closest measure to the cluster center based on the Euclidean distance, until a suitable leaf node is found to add; data elements E are clustered into clusters with similar features and measures. After each data addition, recursive updates are performed from the leaves to the root of the C-Tree.

3.2. The KD-Tree structure for image retrieval problem

The KD-Tree structure is built according to the image classification method from the root node to the leaf node, the nodes in the weight vector storage have been trained by the supervised learning method to obtain the classification results at the leaf node. Classification using the KD-Tree structure is the process of classifying an object when going through each layer on the tree, performing one classification. At each node on the KD-Tree is a single-node neural network used to classify the image object once. Therefore, the KD-Tree structure has brought high image classification results after training. However, the cost of training the classification weight set on the KD tree is quite large. In the classification process, each leaf node is an image classification illustrated in Figure 2 [8].

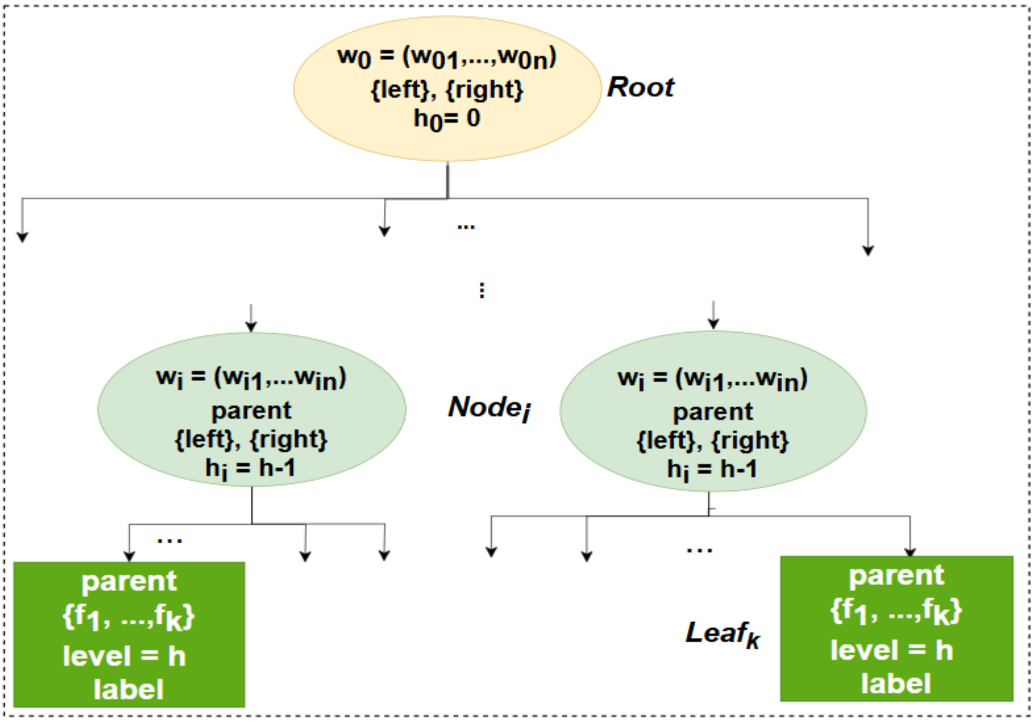

Figure 2. *Description of the components on KD-Tree*

- The $Root$ is a node with no parent, stores a weight vector (w_0), has a left child node set $\{left\}$, a right child node set $\{right\}$ and a level in the tree ofh_0 ($level = h_0$). Notation: $Root = < w_0, \{left\}, \{right\}, h_0 >$;
- An internal node ($Node_i$) is a node that has a parent node, stores a weight vector (w_i), has a left child node set $\{left\}$, a right child node set $\{right\}$, and has a level in the tree ($level = h_i$). Notation: $Node_i \ = < parent, w_i, \{left\}, \{right\}, h_i >$;
- A leaf node ($Leaf_k$) is a node with one parent node, no children, stores the image feature vector set $\{f_1, \dots, f_m\}$, has one level in the tree ($level = h$) and is assigned a label ($label$). Notation: $leaf_k \ = < parent, \{f_1, \dots, f_m\}, level, label >$;

After building the KD-Tree for image classification, the process of clustering similar elements at a leaf, forming similar data clusters that can be applied to the image retrieval problem. The method of building KD-Tree classification before clustering into similar groups according to Euclidean measure has contributed to improving the accuracy of searching images by content. Some results of content based image retrieval on KD-Tree are presented in section 4.2.

4. Evaluation of experimental results

4.1. Experimental results on C-Tree

The results show that the image retrieval accuracy of image sets such as COREL (0,7372) and WANG (0,6621) is not too high (Table 1). However, the average image retrieval time of each image dataset is small, COREL (25,47 ms) and WANG (38,76 ms), which proves that the balanced structure of C-Tree is effective in image retrieval speed.

Table 1. *Image retrieval performance on C-Tree structure of image datasets*

Data sets	No. Images	Avg. Precision	Time query (ms)
COREL	1.000	0,7372	25,47
WANG	80.000	0,6621	38,76

4.2. Experimental results on KD-Tree

The results of applying KD-Tree for image classification on COREL and WANG data sets are presented in Table 2.

Table 2. *The results of applying KD-Tree for image classification*

Data sets	No. Images	No. lacsses	Accuracy
COREL	1.000	10	0,8110
WANG	80.000	80	0,7586

Some experimental results of image retrieval by content performed on KD-Tree with two image data sets COREL, WANG are presented in table 3.

Table 3. *The results of applying KD-Tree for image retrieval*

Data sets	No. Images	Avg. Precision	Time query (ms)
COREL	10.000	0,7981	41,77
WANG	80.000	0,7268	59,62

4.3. Analyzing, comparing and evaluating experimental results

To evaluate the precision and efficiency of the image retrieval method based on C-Tree, the performance obtained from experiments is compared with other methods of other research works on the same image datasets.

Table 4. *Comparison of precision on C-Tree with other methods*

Data sets	Method	Avg. Precision
COREL	Multi-feature and Decision tree, (2021) [9]	0,6680
COREL	GLCM + DCT domain + shape features, (2022) [10]	0,7350
COREL	Dominant Color Descriptor (DCD) with weighted informative pixels, (2022) [11]	0,7087
COREL	**C-Tree**	**0,7372**
	Color Difference Histogram + HSV+entropy, 2019 [12]	0,5067
WANG	Combined feature HSV+LBP+Canny, 2018 [13]	0,5998
WANG	Image signature + BoSW, (2021) [14]	0,6000
WANG	**C-Tree**	**0,6621**

Thus, comparing the precision of the proposed method (C-Tree-based image retrieval) with other modern image retrieval methods shows that, with similar approaches such as extracting low-level features based on color, shape, texture and using traditional data clustering and classification algorithms, this method gives better or equivalent results.

The results of image retrieval by the content of COREL and WANG image data sets on KD-Tree are compared with some other works using different methods presented in Table 5.

Table 5. *Comparison of precision on KD-Tree with other methods*

Data sets	Method	Avg. Precision
COREL	Multi-feature and SVM, 2018 [15]	0,7657
COREL	Multi-feature with neural network, 2020 [16]	0,7941
COREL	ORB 8-dimensions with MPL, 2020 [17]	0,6656

JTE HCMUTE
ISSN: 2615-9740

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

COREL	**KD-Tree**	**0,7981**
WANG	8D-GLCM+GSF+ HSVCM, 2019 [18]	0,5970
WANG	ORB and SIFT features, 2020 [17]	0,6320
WANG	DSFH (low feature + VGG-16), 2021 [19]	0,7894
WANG	**KD-Tree**	**0,7268**

The visual illustration of the image retrieval precision comparison between C-Tree and KD-Tree on the image data sets COREL and WANG is shown in Figures 3 and 4.

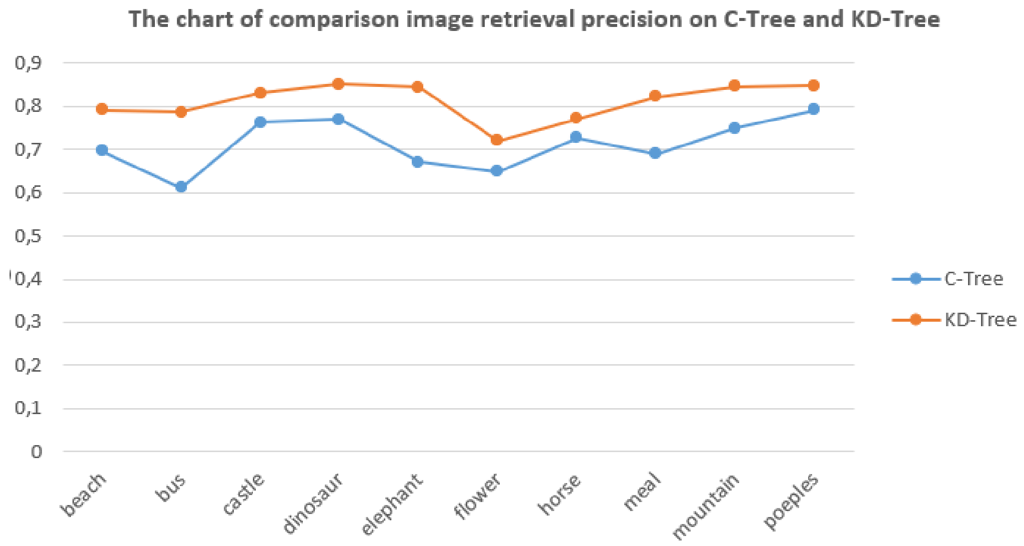

Figure 3. *The chart of comparison precision on C-Tree and KD-Tree on the COREL data set*

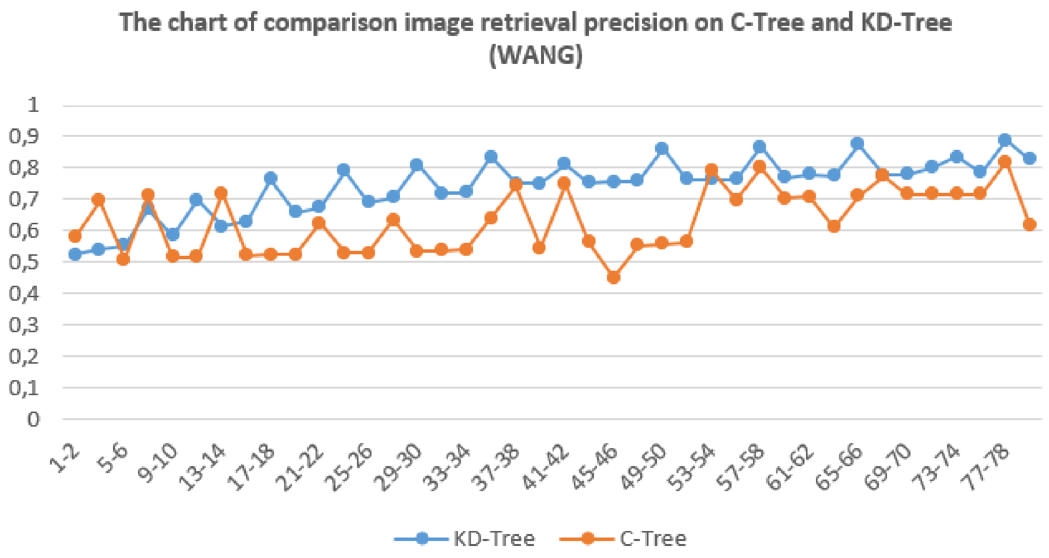

Figure 4. *The chart of comparison precision on C-Tree and KD-Tree on the WANG data set*

Comparing the average query time on C-Tree and KD-Tree, we conduct experiments on the same computer configuration with Intel(R) Core i7-5200U processors, CPU 2.70GHz, RAM 16GB, and Windows 10 Professional operating systems. Server configuration for training image retrieval models using Knowledge Graph: Xeon(R) Gold 6258R CPU 2.70Ghz CPU, 1024GB SSD, 16GB RAM, Server Datacenter 2019 operating system. The visual illustration of the time query comparison between C-Tree and KD-Tree on the image data sets COREL and WANG is shown in Figures 5 and 6.

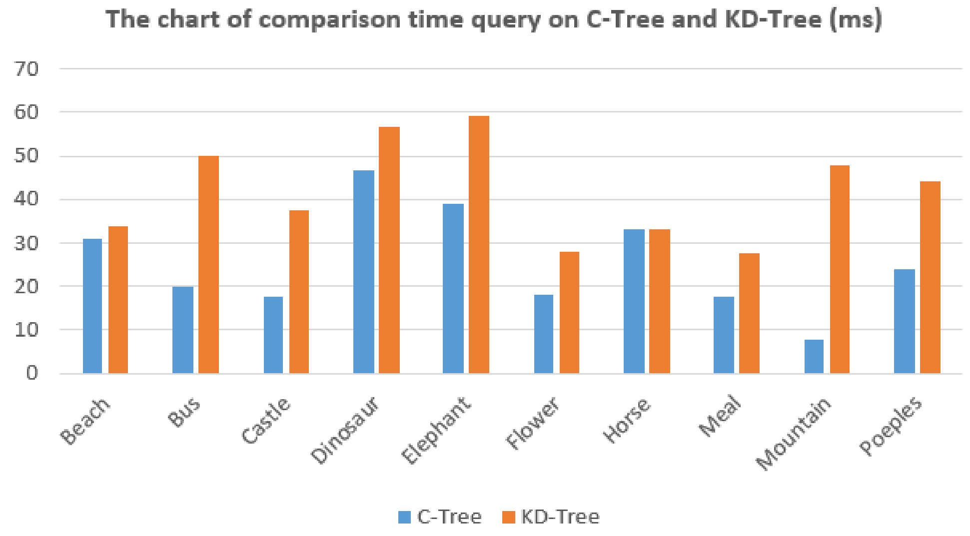

Figure 5. *The chart of comparison time query on C-Tree and KD-Tree on the COREL data set*

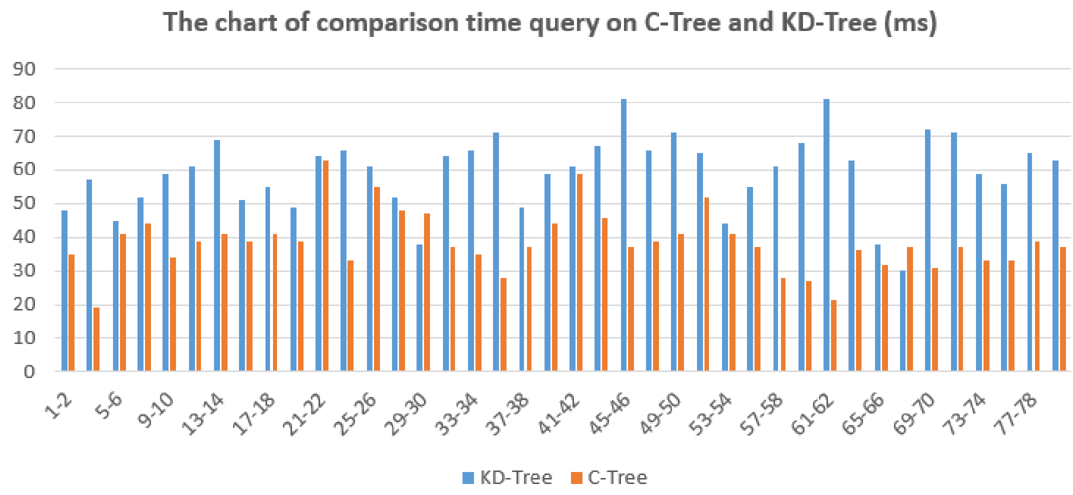

Figure 6. *The chart of comparison time query on C-Tree and KD-Tree on the WANG data set*

Through the comparison charts of experimental results, C-Tree has a lower query time than KD-Tree because C-Tree is a hierarchical clustering, the height is smaller than KD-Tree, so the query process on C-Tree is faster. On the contrary, the accuracy of retrieving similar images on the KD-Tree is higher than on the C-Tree. However, the cost of building the KD-Tree is more than the C-Tree for the same set of experimental images.

During the experiment, we found that single-object image sets such as COREL and WANG are suitable for the two structures KD-Tree and C-Tree in the problem of content-based image retrieval. Therefore, to extend the experiment to other image sets should also be performed on single-object image sets.

The technique of building C-Tree and KD-Tree by clustering and classification can be applied to the problem of image retrieval by content quite well. The comparison results of image retrievalaccuracy show that KD-Tree has better results than C-Tree. However, the query time on C-Tree is faster than retrieving on KD-Tree, at the same time, the cost of training KD-Tree is more costly in terms of resources and time than C-Tree.

Clustered and balanced C-Tree can store large data, effective for image retrieval. C-Tree implements partition clustering and hierarchical clustering methods, so it can quickly search along the branch with the most similar measure to find similar elements at the leaf node cluster, so C-Tree has fast search time and relatively high accuracy. However, the main disadvantage of C-Tree is that each time a node is split, similar elements may be split into different nodes, in the worst case, these nodes may be on separate branches. Therefore, the image search process on C-Tree may miss similar elements that have been branched. This affects the retrieval performance on C-Tree.

In this evaluation, the extension to time-growing image sets is feasible on KD-Tree because KD-Tree is built using a layered approach, training weights at internal nodes. For C-Tree, this task is more difficult to implement.

5. Conclusions

In this paper, we have performed a comparison table, evaluating the performance of image retrieval on two structures C-Tree and KD-Tree; and at the same time, analyzing the advantages and disadvantages that affect the image retrieval results. Each structure is built according to a different method but both achieve positive results in terms of accuracy. However, with these structures applied to image retrieval by content with low accuracy, it is necessary to develop the form of image retrieval by semantic approach to solve the remaining problems in these experiments. Therefore, following these results, in the future, we will combine the C-Tree, and KD-Tree with an image knowledge graph to develop in the direction of semantic image retrieval in different domains or develop the problem of extracting image captions and analyzing image semantics.

Acknowledgments

The authors would like to thank the Faculty of Information Technology, Ho Chi Minh City University of Industry and Trade; University of Sciences, Hue University; University of Economics, The University of Da Nang which are sponsors of this research. We also thank anonymous reviewers for their helpful comments on this paper.

Conflict of Interest

The authors declare no conflict of interest in this article.

REFERENCES

[1] S. Gill and M. Hooda, "The design perspective of the structures based on kd tree," in *Rising Threats in Expert Applications and Solutions: Proc. FICR-TEAS 2020*, Springer, vol. 1187, pp. 515–524, 2021, doi: 10.1007/978-981-15-6014-9_61.

[2] H. Zhang *et al.*, "Tree2Vector: learning a vectorial representation for tree-structured data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5304–5318, 2018, doi: 10.1109/TNNLS.2018.2876826.

[3] S. Ghodratnama and H. A. Moghaddam, "Content-based image retrieval using feature weighting and C-means clustering in a multi-label classification framework," *Pattern Anal. Appl.*, vol. 24, no. 1, pp. 1–10, 2021, doi: 10.1007/s10044-020-00925-4.

[4] A. Joseph *et al.*, "Content-based image retrieval using hybrid k-means moth flame optimization algorithm," *Arab. J. Geosci.*, vol. 14, no. 8, p. 687, 2021, doi: 10.1007/s12517-021-06836-4.

[5] N. Varish, "A modified similarity measurement for image retrieval scheme using fusion of color, texture and shape moments," *Multimed. Tools Appl.*, vol. 81, no. 15, pp. 20373–20405, 2022, doi: 10.1007/s11042-021-11800-x.

[6] Y. Narasimhulu *et al.*, "CKD-Tree: An improved KD-Tree construction algorithm," in *ISIC*, pp. 211–218, 2021. [Online]. Available: https://ceur-ws.org/Vol-2786/Paper28.pdf

[7] N. T. U. Nhi, "A self-balanced clustering tree for semantic-based image retrieval," *J. Comput. Sci. Cybern.*, vol. 36, no. 1, pp. 49–67, 2020, doi: 10.15625/1813-9663/36/1/15041.

[8] N. T. Dinh, T. T. Van, and T. M. Le, "Image classification using KD-Tree for similar image retrieval problem," *Res. Dev. Inf. Commun. Technol.*, pp. 40–52, 2021, doi: 10.32913/mic-ict-research-vn.v2021.n1.966.

[9] M. Garg and G. Dhiman, "A novel content-based image retrieval approach for classification using GLCM features and texture fused LBP variants," *Neural Comput. Appl.*, vol. 33, no. 4, pp. 1311–1328, 2021, doi: 10.1007/s00521-020-05132-8.

[10] N. Varish and P. Singh, "Image retrieval scheme using efficient fusion of color and shape moments," in *Proc. Int. Conf. Big Data, Mach. Learn. Appl. (BigDML 2019)*, Springer, vol. 180, pp. 193–206, 2021, doi: 10.1007/978-981-33-4788-5_16.

[11] S. Fadaei, "New dominant color descriptor features based on weighting of more informative pixels using suitable masks for content-based image retrieval," *Int. J. Eng.*, vol. 35, no. 8, pp. 1457–1467, 2022, doi: 10.5829/IJE.2022.35.08B.15.

[12] H. Qazanfari, H. Hassanpour, and K. Qazanfari, "Content-based image retrieval using HSV color space features," *Int. J. Comput. Inf. Eng.*, vol. 13, no. 10, pp. 533–541, 2019. [Online]. Available: https://publications.waset.org/10010804/content-based-image-retrieval-using-hsv-color-space-features

[13] L. Pavithra and T. S. Sharmila, "An efficient seed points selection approach in dominant color descriptors (DCD)," *Cluster Comput.*, vol. 22, no. 4, pp. 1225–1240, 2019, doi: 10.1007/s10586-018-2486-4.

[14] A. Ouni *et al.*, "Leveraging semantic segmentation for hybrid image retrieval methods," *Neural Comput. Appl.*, vol. 34, no. 24, pp. 21519–21537, 2022, doi: 10.1007/s00521-021-06369-3.

[15] F. M. F. Gonçalves, I. R. Guilherme, and D. C. G. Pedronette, "Semantic guided interactive image retrieval for plant identification," *Expert Syst. Appl.*, vol. 91, pp. 12–26, 2018, doi: 10.1016/j.eswa.2017.08.017.

[16] R. Raja, S. Kumar, and M. R. Mahmood, "Color object detection based image retrieval using ROI segmentation with multi-feature method," *Wireless Pers. Commun.*, vol. 112, no. 1, pp. 169–192, 2020, doi: 10.1007/s11277-019-06890-8.

[17] P. Chhabra, N. K. Garg, and M. Kumar, "Content-based image retrieval system using ORB and SIFT features," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 2725–2733, 2020, doi: 10.1007/s00521-019-04086-5.

[18] M. I. T. Bella and A. Vasuki, “An efficient image retrieval framework using fused information feature,” *Comput. Electr. Eng.*, vol. 75, pp. 46–60, 2019, doi: 10.1016/j.compeleceng.2019.01.013.

[19] G. H. Liu and J. Y. Yang, “Deep-seated features histogram: a novel image retrieval method,” *Pattern Recognit.*, vol. 116, p. 107926, 2021, doi: 10.1016/j.patcog.2021.107926.

Thi Dinh Nguyen was born in 1983, graduated in Pedagogy Informatics Ho Chi Minh City University of Education in 2006, and received a Master's degree in industry Data transmission and computer network at Ho Chi Minh City Institute of Post and Telecommunications Technology Ho Chi Minh City in 2011. In 2023, she received a PhD degree in Computer Science from the University of Science, Hue, Vietnam. Field research: image processing, image retrieval, and information system.
Email: dinhnt@huit.edu.vn. ORCID: https://orcid.org/0000-0003-3428-3101

Thanh Manh Le was born in 1953. He received a Ph.D. degree in computer science from Budapest University (ELTE), Hungary, in 1994. He became an associate professor at Hue University, Vietnam, in 2004. His research interests include databases, knowledge bases and logic programming.
Email: lmthanh@hueuni.edu.vn. ORCID: https://orcid.org/0000-0002-0949-222X

Thi Uyen Nhi Nguyen received bachelor's and master's degrees in Computer Science and Computational Engineering, Volgagrad State Technical University, Russian Federation, in 2008 and 2010, respectively. In 2022, received a PhD in Computer Science from the University of Science, Hue University. Currently, teaching at the Faculty of Statistics and Informatics, University of Economics, The University of Danang. Field research: image processing, image retrieval, and information system.
Email: nhintu@due.udn.vn. ORCID: https://orcid.org/0009-0000-6685-025X