DEVELOPING A MODEL SEMANTIC-BASED IMAGE RETRIEVAL BY COMBINING KD-TREE STRUCTURE WITH ONTOLOGY

**Thanh Manh Le**

University of Science, Hue University, Hue, Vietnam

*Email:* [*lmthanh@hueuni.edu.vn*](mailto:lmthanh@hueuni.edu.vn)

**Nguyen Thi Dinh**

University of Science, Hue University, Hue, Vietnam  
HCMC University of Food Industry, HoChiMinh city, Vietnam

*Email:* [*dinhnt@hufi.edu.vn*](mailto:dinhnt@hufi.edu.vn)*; dinhnt@hueuni.edu.vn*

**Thanh The Van**HCMC University of Education, Vietnam

*Email:*[*thanhvt@hcmue.edu.vn*](mailto:thanhvt@hcmue.edu.vn)

**Abstract**

The paper proposes an alternative approach to improve the performance of image retrieval. In this work, a framework for image retrieval based on machine learning and semantic retrieval is proposed. In the preprocessing phase, the image is segmented objects by using Graph-cut, and the feature vectors of objects presented in the image and their visual relationships are extracted using R-CNN. The feature vectors, visual relationships, and their symbolic labels are stored in KD-Tree data structures which can be used to predict the label of objects and visual relationships later. To facilitate semantic query, the images use the RDF data model and create an ontology for the symbolic labels annotated. For each query image, after extracting their feature vectors, the KD-Tree is used to classify the objects and predict their relationship. After that, a SPARQL query is built to extract a set of similar images. The SPARQL query consists of triple statements describing the objects and their relationship which were previously predicted. The evaluation of the framework with the MS-COCO dataset and Flickr showed that the precision achieved scores of 0.9218 and 0.9370 respectively.

**Keywords** *KD-Tree, Relationship KD-Tree, Ontology, Image Retrieval, Similar Images*

# Introduction

In recent times, the rapid increase of digital images has posed numerous challenges forimage retrieval systems. For instance, searching for images that depict a dog playing with a cat (as illustrated in Fig 2). To enhance the performance in terms of accuracy and speed of such image searching, using semantic-based image retrieval is a promising approach. Several works have been undertaken to improve the accuracy of semantic-based image searching including using ontology to represent the semantic for images (Asim, 2019), leverage the current advances of machine learning techniques (Mehmood Zahid, 2017) and many more. In line with this research topic, this paper proposes an alternative semantic-image retrieval that utilizes ontology and KD-Tree data structures to improve the accuracy of the image searching. This works is extended from previous works published in (Nhi, 2022), (N. T. Dinh, & Le, T. M. , 2022).

There are two main components to consider in multiple-object image retrieval searches: the objects present in the image and their visual relationships. Currently, the visual relationships between objects are stored for later determination using scene graphs (Yoon, 2021) or knowledge graphs (Roopak, 2021). In this paper, the visual features of these objects and encode their visual relationships are stored in KD-Tree data structures.

In the previous work, the KD-Tree can be used to improve image retrieval performance (N. T. Dinh, & Le, T. M. , 2022). Furthermore, with the recent advanced machine learning techniques, the accuracy of image retrieval by applying machine learning can be further improved. Therefore, the KD-Tree data structure was enhanced by assigning trained weights, thus improving the accuracy of image searching in comparison to previous work (N. T. Dinh, & Le, T. M. , 2022). For instance, in the indexing phase, each object of an image is segmented by using Graph-cut technique. The weighted feature vector of each object is extracted and stored in a KD-Tree data structure. After that, the visual relationship between each pair of two objects is also stored in a weighted KD-Tree named Relationship KD-Tree. Hence, the trained weights will help us to cluster the images by the feature vectors of the objects and visual relationships, each leaf of KD-Tree stores a set of images with the same label or the same type of relationship.

To enhance the retrieval accuracy, the RDF data model is used to semantically annotate the images and create an ontology of object symbolic label and their visual relationships. For instance, for an input image, first the objects are segmented by using Graph-cut. Then, the features vector of each object is extracted. These feature vectors are used to predict the name (label) of the object and the type of their relationships. After that, a SPARQL query is constructed from these object labels and relationship labels in the form of triple statements. The precise description of this process is presented in **Section 4**.

The paper’s main contributions are as follows: (1) a design and algorithms for encrypting object feature vectors and visual relationships in weighted KD-Tree data structures that can be used for predicting later; (2) a method for facilitating semantic retrieval by constructing a SPARQL from the triples that describe the relationships and objects in the image; (3) a design of an ontology-based framework for semantic-based image retrieval using SPARQL query; (4) a precise demonstration of the framework and evaluation of the framework using image datasets such as MS-COCO (COCO, 2017), and Flickr (Flickr, 2017).

The rest of the paper is organized as follows: **Section 2** presents the related works including the works on image retrieval using KD-Tree, image retrieval using visual relationships and semantic-based image retrieval using ontology. The construction of the KD-Tree and the prediction algorithm are presented in **Section 3**. The utilization of semantic query and ontology is described in **Section 4**. **Section 5** presents the details of the experiment and evaluation results. The conclusion of the paper with future works is presented in Section 6.

# Related work

The performance of image retrieval is directly affected by the methods of implementation. So, in this section, some related works are investigated such as semantic-based image retrieval using ontology; image retrieval using a KD-Tree structure; classification the relationship of objects in the image including:

***2.1*. *Semanitc-based image retrieval using ontology***

M.N. Asim et al. (Asim, 2019) implemented an information retrieval method using ontology applied to multimedia data (images, videos, audio). In this work, the authors used the RDF triple language to retrieve similar images based on ontology and simultaneously compares its performance with the others of previous approaches. This article is evaluated as feasible, and effective; at the same time compared with other works (N. T. Dinh, V. T. Thanh, Thanh, L. M., 2022).

Nhi N T U et al. (Nhi, 2022) built an ontology framework and SPARQL query to retrieve images by semantics. In this work, an ontology framework is built capable of enriching many sets of images and extending the semantic relationship of images. On this basis, a SPARQL query is extracted to retrieve a set of similar images by semantics based on ontology. This work is evaluated as feasible and effective for semantic image retrieval based on ontology.

Besides, some works (N. T. Dinh, and Le, T. M., 2022), (N. T. Dinh, V. T. Thanh, Thanh, L. M., 2022) have performed semantic-based image retrieval based on ontology using SPARQL query. However, the query process is performed based on the classifications of the input images, so the performance is not optimized by the proposed method of combining the classification and relationships of objects on the proposed image in this article. Therefore, the design and query process using extended ontology is an improvement proposal in this paper to improve the performance of the semantic-based image retrieval.

Based on the ontology-based semantic image retrieval method; at the same time, inheriting ontology from works (Nhi, 2022), (N. T. Dinh, and Le, T. M., 2022), this paper continues to build an extended Ontology to perform semantic image search problems based on triples extracted from input images.

***2.2. KD-Tree structure for image retrieval***

The works using the KD-Tree structure for the image retrieval problem have been published many times with positive results in the past decades and are evaluated as feasible and effective, such as:

Y.H. Sharath Kumar et al. (Kumar, 2015) proposed a model of content-based image retrieval based on the KD-Tree structure that was evaluated as effective, consisting of two phases: The training phase includes: (1) Data set segmented images; (2) feature extraction; (3) building KD-Tree structure; (4) Building knowledge system. The query phase based on the sketch image consists of: (1) The input is an outline image; (2) feature extraction; (3) execute the query from the KD-Tree structure. After building the KD-Tree structure applied to the image query problem, the search and matching process is performed one more step to filter out the most optimal results. This is an advantage over direct searches on the KD-Tree structure that improves performance.

Fengquan Zhang, et al (Zhang, 2019) built the Vocabulary-KDTree structure to improve the image matching problem. In this work, the authors have carried out two processes: (1) clustering image data according to similarity; (2) online data matching with an input image. The Vocabulary-KDTree structure is based on the SIFT feature by adjusting the weights at the nodes on the tree. The image query model is performed in two phases: Offline phase, each image after feature extraction is matched and clustered with the KD-Tree structure; then build Vocabulary KD-Tree and perform clustering on this structure. In the online phase, an input image after feature extraction is compared to this feature with the Vocabulary KD-Tree structure, finding keywords as a basis for comparison with the extracted feature. This work is evaluated with good image query performance.

Y Narasimhulu et al. (Narasimhulua, 2021) proposed an image classification model based on KD-Tree. From an input image, perform a search on the KD-Tree structure using the search algorithm according to the maximum number of neighbors as the basis for determining the classification for the image. Finally, the author uses a distance scale to perform image classification of training image datasets. In this work, the KD-Tree is used directly to store data and to classify an input image with good results without much intermediate cost. This is a proposed model for the problem of classification and image retrieval based on the KD-Tree structure, which is considered to be quite good.

***2.3. Image retrieval using relationships between objects***

There are many works to determine the relationship between objects on the image according to different proposed methods; some of the typical works such as:

**Alina Kuznetsova**et al.(Kuznetsova, 2020) proposed a method of visual analysis semantics of images. The paper has presented a collection of 9.2 million images (MS-COCO, PASCAL VOC) annotated with unified ground truth for image classification, object detection, and relationship detection. The authors explained how the data was collected and annotated, presented comprehensive data set statistics, evaluated its quality, and reported the performance of several modern models for image classification and object detection. In this work, firstly, objects are detected and classified by an R-CNN model, then visual relationships are built by this set of images to train the process of building visual relations between objects of the input image. The proposed method is evaluated the feasibility and effectiveness, and applied to different image data sets.

**Sangwoong Yoon** et al. (Yoon, 2021) performed a multi-object image retrieval method that was tested on the Flickr30k image set with an average query performance of 0.5670 if the TopK is equal to 5. In this work, the IRSGS-GCN combination method and Scene Graph construction are used to find the relationship between the objects on the image and at the same time retrieve the similar image set with the first image. This is an evaluated method with high performance for multi-object image retrieval problem.

**Brigit Schroeder** et al. (Schroeder, 2020) performed an image retrieval based on determining the relationship of objects on the image using Scene Graphs. In which, the objects with visual relationships are identified according to each MS-COCO and Flickr image set. Experimental results show that the proposed and implemented method is feasible and effective for multi-object image data sets.

**Sijin Wang** et al. (S. Wang, Wang, R., Yao, Z., Shan, S., & Chen, X, 2020) performed an image retrieval problem based on the relationships between objects on the image using the Scene graph context graph. In this work, the author used visual scene graph (VSG) and textual scene graph (TSG) to build and extract the visual relationship between any two objects on the image, thereby extracting captioning of the image. input and perform image queries based on captioning content. The experimental results are evaluated on two sets of MS-COCO and Flickr images with query performance of 0.7650 and 0.8650.

Based on analyzing the advantages and disadvantages of related works about semantic-based image retrieval using ontology; image classification using Random Forest; classify the relationship of objects in the image. In this paper, a model of semantic-based image retrieval is proposed including the following steps: First, a method of image classification using the KD-Tree Random Forest is performed with a majority vote. Secondly, a balanced multi-branch KD-Tree structure is built to classify the relationships between objects in the image. Thirdly, a SPARQL query is built by a triple of describing class names and the relationships between the objects on the input image. Finally, an extende ontology được xây dựng để truy vấn tập ảnh theo ngữ nghĩa với ảnh đầu vào. **Finally, extended ontology is proposed to retrieve a set of similar images based on the SPARQL query.**

# A model of image retrieval using KD-Tree and Ontology

## A proposed model

A model of image retrieval using KD-Tree and Ontology is implemented with the following steps:

(1) Extract feature vector from input images to form an object KD-Tree;

(2) Extract visual relationships of input images to form a relationship KD-Tree;

(3) Assign a label for each leaf of the object KD-Tree. When constructing the object  
KD-Tree, the frequency of the labels found on each leaf is recorded. The label of a leaf is  
the label that has the highest frequency;

(4) For each input image generating object images, each object finds the nearest objects on the KD-Tree.

(5) From the triple <Obj\_1> < Relationship> <Obj\_2>, construct the SPARQL query;

(6) For a triple <Obj\_1> <Relationship> <Obj\_2>, an image is retrieved.

(7) Retrieving all images of the same cluster on the original clustered Ontology;

(8) Combine all pairs of objects and select by TopK;

(9) Extract a set of images similar by semantics to the input image.

In this paper, a model of image retrieval using KD-Tree structure and Ontology is illustrated in **Figure 1**.

Diagram

Description automatically generated

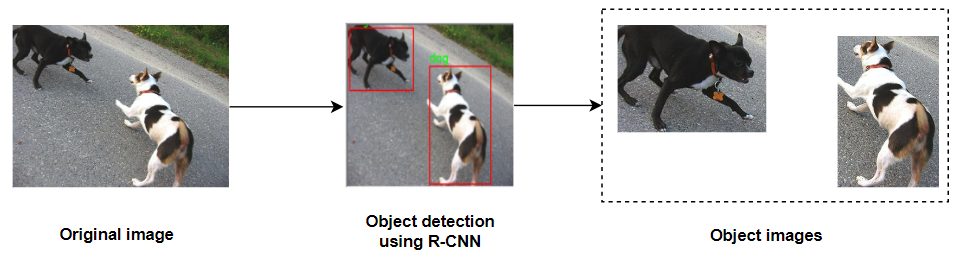
**Figure 1.** A model of image retrieval using Ontology and KD-Tree structure

The preprocessing phase performs segmentation into object images and feature vectors extraction (1) before building the KD-Tree structure to store the data (2). From KD-Tree, data stored at each leaf proceeds to build a Relationship KD-Tree structure to extract relationships between objects on the image (2). The Ontology development process is supplemented with relationships between objects and object image sets (3), (4), (5), (10).

At the retrieval phase, each input image is segmented (6), and feature vectors are extracted (7); then retrieving the same set of object images on the KD-Tree (9); classifies the relationship between objects on the input image by Relationship KD-Tree (8). Then, the SPARQL query is built based on the input image triple (8), (11). Finally, retrieving the Ontology in the preprocessing phase to extract the set of similar images to the input (12).

* 1. ***KD-Tree structure for image retrieval***

In this paper, a KD-Tree structure is built and trained to inherit the work (N. T. Dinh, and Le, T. M., 2022) (N. T. Dinh, Thanh The Van, and Thanh Manh Le, 2022). However, each original image consisting of many objects is cut into many opposite images, and each image region contains a single object. After image segmentation, each feature image region is extracted as a feature vector for the process of building a KD-Tree structure. The original image and subject images of the image 1001773457.jpg (Flickr) are illustrated as shown in **figure 2**.



**Figure 2.** Illustrate the object images are extracted from the original image

**Algorithm 1.** Build KD-Tree structure from object images

|  |
| --- |
| 1. **Input**: *F = {: = (, , …,)*; *i = 1 ... k*}; height of KD-Tree *h*; number of branches *b;* initial set of weights *= {: = (,, …,)*; *i = 0 … h-1*}; |
| 1. **Output**: KD-Tree |
| 1. **Function** **CKDT** (*F,, h, b*) |
| 1. **Begin** |
| 1. **Int** h, b; KD-Tree = ∅; |
| 1. **For** (int i = 0; i <= h-1; i++) **do** |
| 1. initialize the set of random weights ; |
| 1. = Epsilon; |
| 1. = Epsilon; |
| 1. **EndFor** |
| 1. **Foreach** ( *in F*) **do** |
| 1. **If** (KD-Tree = Null) **then** create KD-Tree with *h*, number of branch is 1; |
| 1. **Else** |
| 1. **For** (int j = 0; j <= h-1; j++) **do** |
| 1. **If** (*Sigmoid*(, ) < **then** |
| 1. Create left branch and update left threshold; |
| 1. **EndIF;** |
| 1. **If** (*Sigmoid*(, ) > ) **then** |
| 1. Create right branch and update right threshold; |
| 1. **EndIf;** |
| 1. **IF** ( <= *Sigmoid*(, ) and *Sigmoid*(, ) <= ) **then** |
| 1. Select best branch; |
| 1. **EndIf**; |
| 1. **EndFor**; |
| 1. **If** ( = h-1) **then** |
| 1. **Foreach** () **do** |
| 1. **Foreach** ( *in* ) **do** |
| 1. **IF** ( *=* ) **then** insert) |
| 1. **Else** |
| 1. Create left branch ); |
| 1. **EndIf**; |
| 1. **EndForeach;** |
| 1. **EndForeach**; |
| 1. **EndIf;** |
| 1. **Return** KD-Tree; |
| 1. **End.** |

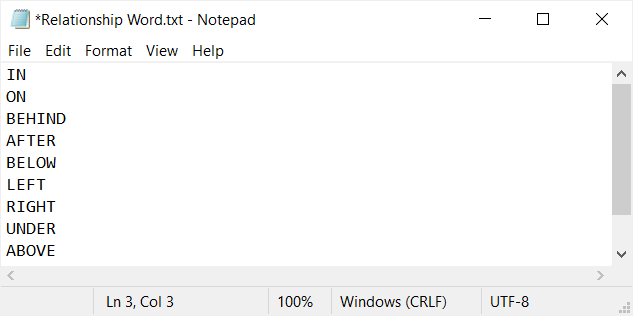
Let be the number of elements performing KD-Tree construction, is the height of the KD-Tree. When building a KD-Tree, **algorithm 1** needs to add elements to a tree of height . KD-Tree is a balanced tree, so when adding elements to the tree, every element must be traversed from the root node to the leaf node. is a constant, so the complexity of **algorithm 1** is )*.*

After building the KD-Tree structure and process of training the weights inherited from the work (N. T. Dinh, and Le, T. M., 2022) (N. T. Dinh, Thanh The Van, and Thanh Manh Le, 2022). The KD-Tree structure is an image classification tree, in which each leaf contains a set of images with the same label. On a KD-Tree, each leaf is assigned a name; From each of these leaves, each object is decomposed, combining the Relationship KD-Tree to create an Ontology for each leaf cluster. The purpose of clustering before inclusion in Ontology is to increase retrieval time and retrieval accuracy based on clustering*.*

***3.3. A KD-Tree structure for relationship classification***

The process of inserting object image vectors into KD-Tree is done in pairs of object images on the original image.

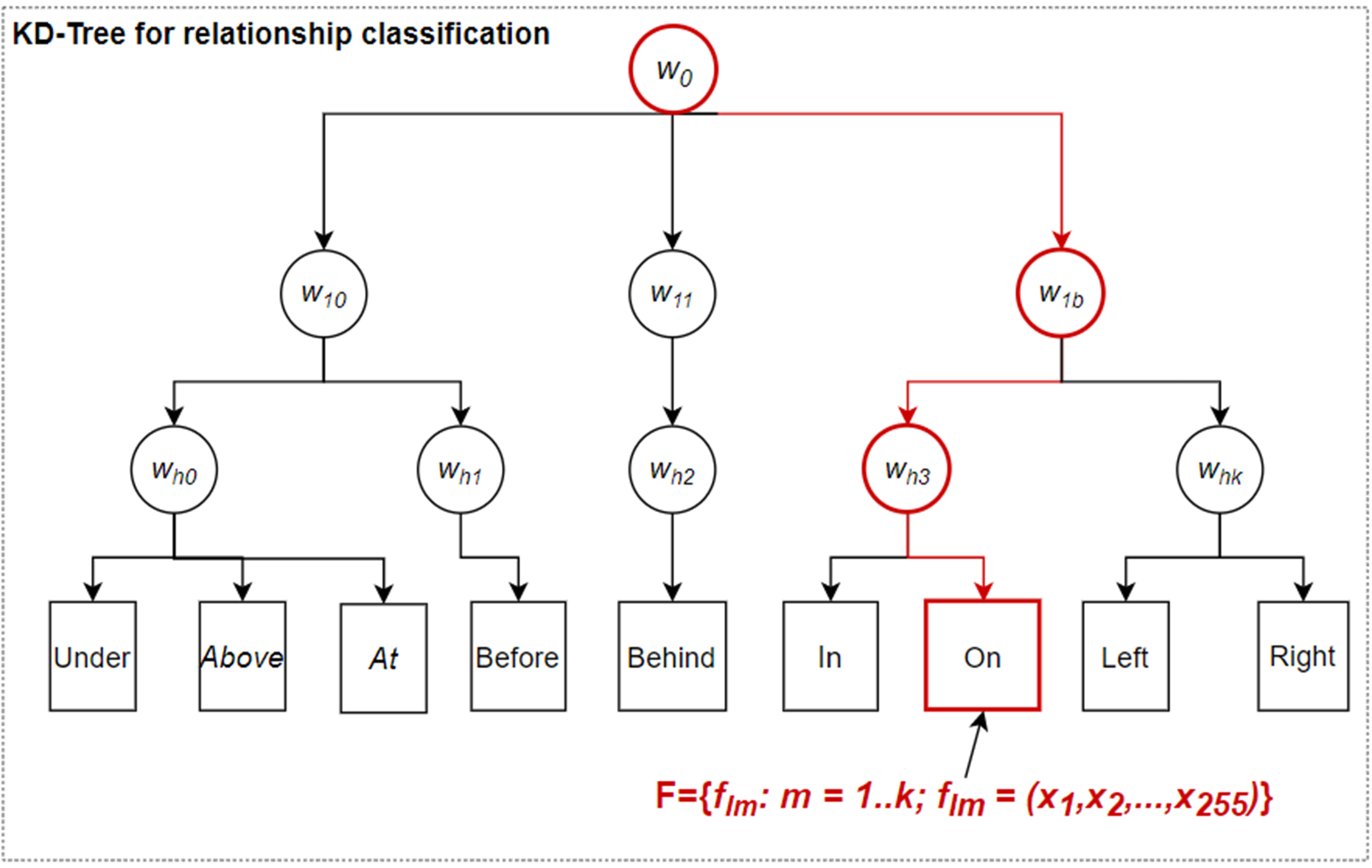
Based on the original KD-Tree structure (Bentley, 1975), a balanced multi-branch KD-Tree structure is built to apply to relationship classification. In this paper, the relationships between objects on the MS-COCO and Flickr data sets are determined for building and training a balanced multi-branch KD-Tree structure. A relationship between objects is determined to rely on relationship words including On, In, At, Before, Behind, Under, Above, Left; Right, etc (Schroeder, 2020). These relationship words are stored at the leaves on a balanced multi-branch KD-Tree structure. **Figure 3**illustrates the set of related words of objects on images are extracted from the MS-COCO image dataset



**Figure 3.** Illustration of relationship words on MS-COCO image dataset

In which, a balanced multi-branch KD-Tree structure is built and trained based on the development of the KD-Tree structure from the published work (N. T. Dinh, and Le, T. M., 2022) (N. T. Dinh, Thanh The Van, and Thanh Manh Le, 2022). Initially, a balanced multi-branch KD-Tree structure is built with a set of random weight vectors stored at the nodes, and the relationship words are randomly distributed to leaves. Each image is segmented into image regions Each image region is extracted into a *225*-dimensional vector ,…, ).

The segmented image vector set is also stored at leaves on a balanced multi-branch KD-Tree structure. So, a process training of weight vector on a balanced multi-branch KD-Tree structure is performed to improve the performance of relationship classification on an image. The weight vector training algorithm is inherited from published work (N. T. Dinh, and Le, T. M., 2022) (N. T. Dinh, Thanh The Van, and Thanh Manh Le, 2022). The structure of a balanced multi-branch KD-Tree structure is illustrated as shown in **figure 4,** and **algorithm 1** for constructing a balanced multi-branch KD-Tree structure is illustrated as follows:



**Figure 4.** A KD-Tree structure for relationship classification

A process of building a balanced multi-branch KD-Tree structure on the basis of inheriting the functions **kMiKDT** and **kNiKDT** from the work (N. T. Dinh, and Le, T. M., 2022) to be used for **algorithm 2** is as follows:

**Algorithm 2**. Building a KD-Tree structure for relationship classification

|  |
| --- |
| 1. **Input**: *F = {: = (, , …,)*; *i = 1 ... k*}; height of KD-Tree *h*; number of branches *b;* initial set of weights *= {: = (,, …,)*; *i = 0 … h-1*}; |
| 1. **Output**: Relationship KD-Tree |
| 1. **Function** **BRKDT** (*F,, h, b*) |
| 1. **Begin** |
| 1. **; ;** |
|  |
| 1. initialize the set of random weights ; |
| 1. = Epsilon; |
| 1. = Epsilon; |
| 1. **EndFor** |
|  |
| 1. ; |
| 1. **Else** |
|  |
| 1. **If** (*Sigmoid*(, ) < **then** |
| 1. Create left branch and update left threshold; |
| 1. **EndIF;** |
| 1. **If** () **then** |
| 1. Create right branch and update right threshold; |
| 1. **EndIf;** |
| 1. **IF** () **then** |
| 1. Select best branch; |
| 1. **EndIf**; |
| 1. **EndFor**; |
| 1. **If** ( = h-1) **then** |
| 1. **Foreach** () **do** |
| 1. **Foreach** ( *in* ) **do** |
| 1. **IF** ( *=* ) **then** insert) |
| 1. **Else** |
| 1. Create left branch ); |
| 1. **EndIf**; |
| 1. **EndForeach;** |
| 1. **EndForeach**; |
| 1. **EndIf;** |
| 1. **Return** Relationship KD-Tree; |
| 1. **End.** |

A process of building a KD-Tree structure has a height needs to be added elements. The KD-Tree is a balanced tree, so when adding elements to the tree, every element must be traversed from the root to the leaf. The complexity of the **algorithm 2** is ; is a constant. So, the complexity of the **algorithm 2** algorithm is .

Each experimental image data set after segmentation, feature extraction, and storage on a balanced multi-branch KD-Tree structure. Each input image after segmenting the objects, and extracting features into many vectors, the objects on the input image are determined by the relationship by **algorithm 2**. If  and belong to a leaf, the relationship between these two objects is determined by the relationship word at that leaf. If and belong to two different leaves, then the relationship between objects is determined by the word indicating the relationship between the two corresponding leaves.

**Algorithm 3***.* Relationship classification between objects using KD-Tree structure

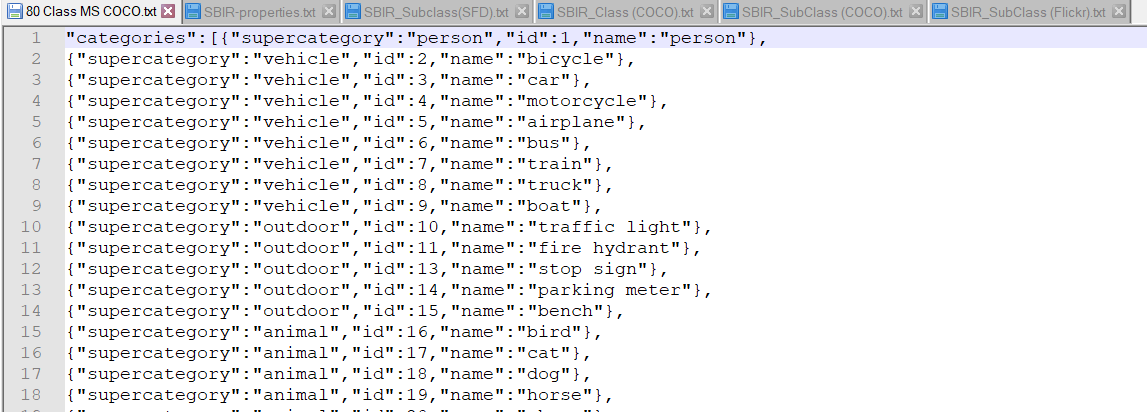
|  |
| --- |
| 1. **Input**: |
| 1. **Output**: |
| 1. **Function** |
| 1. **Begin** |
|  |
|  |
| 1. ; |
|  |
|  |
|  |
|  |
| 1. ; |
| 1. **;** |
| 1. **;** |
|  |
|  |
| 1. *;;* |
| 1. *;* |
|  |
|  |
| 1. **End.** |

Let be the height of the KD-Tree structure, is the maximum number of branches at any , and the input data is the feature vector with-dimensions. When passing vector to the KD-Tree, the **algorithm 3** traverses the levels of the tree. Let be a constant value and *,* so . So the complexity of algorithm **algorithm 3** is

# Developing an Ontology for image retrieval

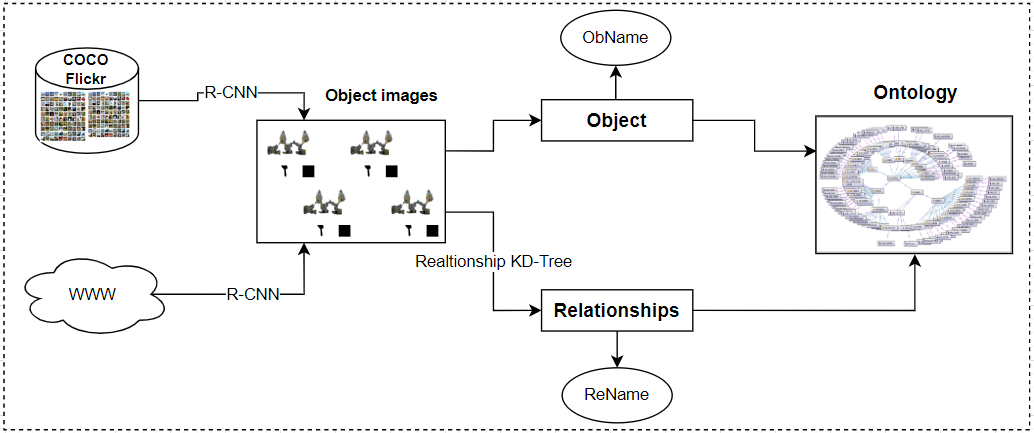
**4.1. Hierarchy of experimental image set and developing Ontology**

In this paper, Ontology is developed on the basis of inheriting and supplementing data from the ontology framework of the work (Nhi, 2022), (N. T. Dinh, and Le, T. M., 2022). This part is presenting the classification hierarchy of multi-object image sets MS-COCO, Flickr. The process of enriching and adding data to sets of images. The data hierarchy on the MS-COCO and Flickr image files is illustrated in **Figure 5**.



**Figure 5.** Structure of Class, SubClass on MS-COCO image dataset

The process of developing an ontology for experimental image sets is illustrated in **figure 6**. For each multi-object image, after segmenting the object to form object images; each object is stored by a name (*ObName*) and between objects, there is a relationship (*Relationship*) classified by the Relationship KD-Tree structure. On this basis, the process of adding objects and relationships between objects on the image to build an ontology.



**Figure 6.** Model for adding data into the ontology for multi-object images

## 4.2. Structure of Ontology

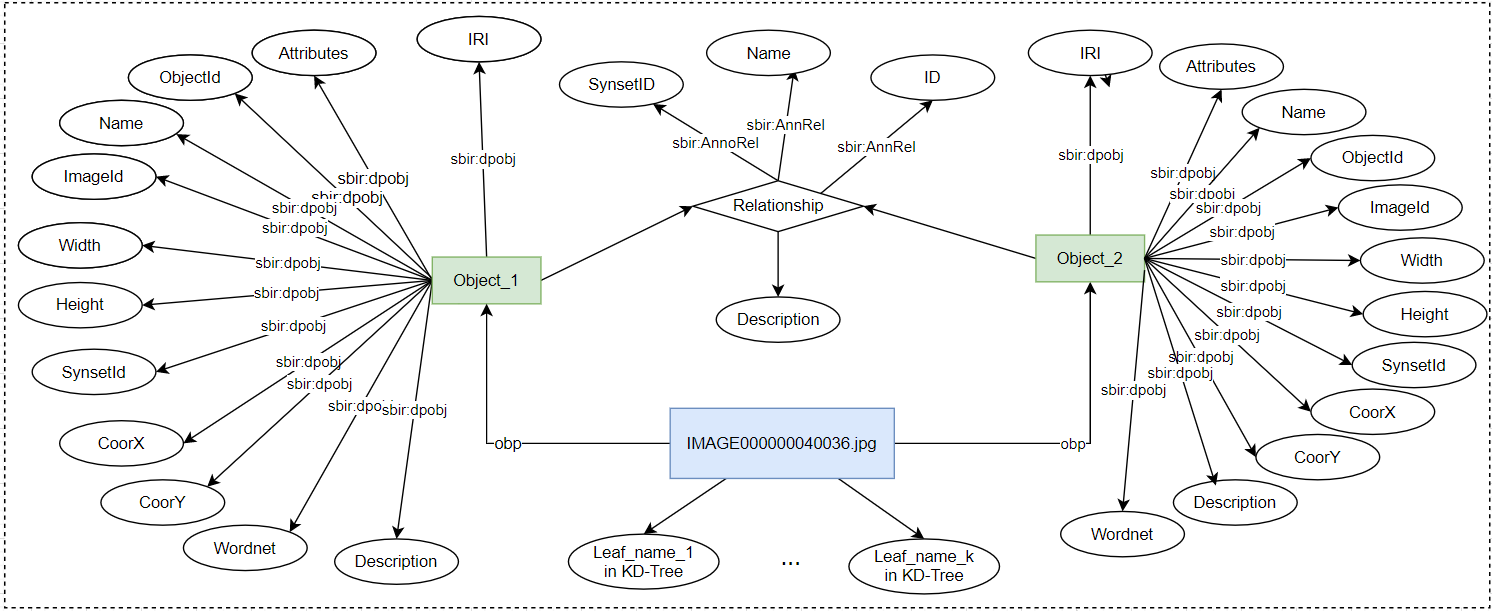
An ontology is designed and enriched for many sets of experimental images to improve the performance of the image search problem according to the semantic approach. Based on inheriting ontology built from the works (Nhi, 2022), (N. T. Dinh, and Le, T. M., 2022), in this topic, the ontology structure is proposed and built to apply to the problem of semantic image search on sets of tuples. MS-COCO and Flickr multi-object images. Based on the ontology framework (Nhi, 2022), an ontology structure is built in **algorithm 4**.

**Algorithm 4**. Build an Ontology from similar images and relationships

|  |
| --- |
| 1. **Input**: |
| 1. **Output**: |
| 1. **Function** |
| 1. **Begin** |
|  |
|  |
| 1. *;* |
|  |
|  |
| 1. **End.** |

Let be the number of classes and subclasses of the experimental set. When adding classifiers and subclasses of each set of images to the ontology framework, the complexity of the **algorithm 4** is . Since , and are constants, set to be constant as well. The complexity of the **algorithm 4** is .

The ontology structure is built with the components described on each object area image as shown in **Figure 7**, each stored object includes the following components: storage address (*IRI*), object identifier (*ObjectId*), attributes (*Attributes*), object image name (*Name*), object image identifier (*ImageId*), width (*Width*), height (*Height*), object image wordnet identifier (*SynsetId*), horizontal axis coordinates (*CoorX),* vertical axis coordinates (*CoorY*), object wordnet description (*Wordnet*), object description (*Description*).



**Figure 7.** Illustrate elements on Ontology

Table

Description automatically generated

**Figure 8.** Illustration Ontology by N3

After building the ontology applied to the semantic image search problem, the input data is a SPARQL query (Icarte, 2022). To build the SPARQL query, it is necessary to have the classification results for the input image and the word describing the relationship between the objects on the image after determining based on the Relationship KD-Tree structure. After building the ontology, the process of performing an image query on the ontology is based on the SPARQL query. The results of the SPARQL query are illustrated as shown in **Figure 9.**



**Figure 9.** Illustrate the SPARQL query is built from triple

# Experiment experimental and evaluate results

## Experimental environment and data description

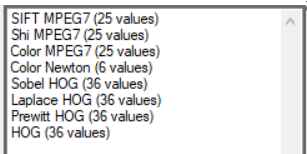
In this paper, a system of semantic-based image retrieval using Ontology and KD-Tree (**SO-KDT)** is built on the dot NET Framework 4.5 platform the C# programming language. The graphs are built on MathLab 2015. The **SO-KDT** system is performed on computers with Intel(R) Core i7-5200U processors, CPU 2.70GHz, RAM 16GB, and Windows 10 Professional operating systems. Server configuration for training image classification models using KD-Tree Random Forest: Xeon(R) Gold 6258R CPU 2.70Ghz CPU, 1024GB SSD, 16GB RAM, Server Datacenter 2019 operating system.

To prove the correctness of the proposed model, several experimental image data sets were performed to classify the relationship of objects on an image, including MS-COCO and Flickr. The MS-COCO data set is large-scale object detection, segmentation, key-point detection, and captioning data set. The data set consists of 163,957 images. The Flickr dataset contains 31,783 images collected from Flickr, together with five reference sentences provided by human annotators.

**Table I.** Describe experimental image data sets

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Data sets*** | ***Number of images*** | | ***Training images*** | ***Testing images*** | | ***Validation images*** | |
| MS-COCO | 163,957 | 118,287 | | | 40,670 | | 5,000 |
| Flickr | 31,783 | 29,000 | | | 1,783 | | 1,000 |

Feature vectors influence the performance of image retrieval. In this paper, after segmenting each image by the Graph-cut method, each image region is extracted a feature vector of 225 dimensions which is a combination of feature groups SIFT, Shi, Color MPEG7, Color Newton, Sobel HOG, Laplace HOG, Prewitt HOG, HOG as shown in **Figure 10.**



**Figure 10.** Illustrate a vector features 225-dimensional

## Experimental results and evaluation

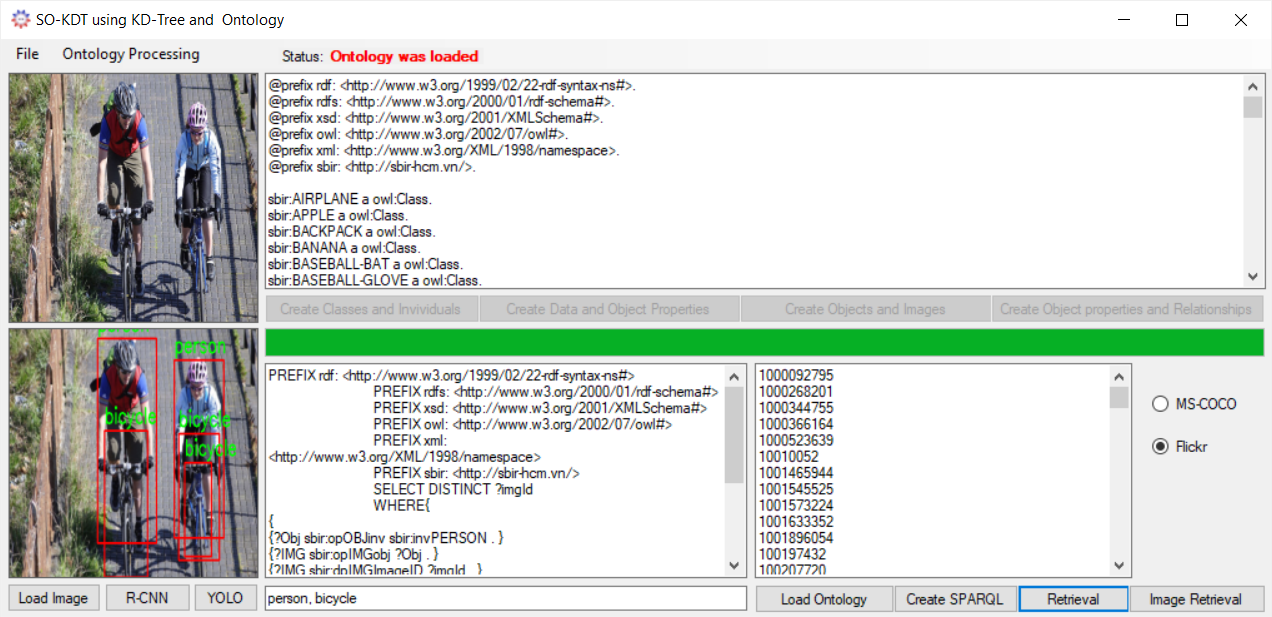
After building the Ontology and the SPARQL query is formed from the triple describing the input image content. The process of querying on the ontology to extract the image set similar to the input image is performed by **algorithm 5**.

**Algorithm 5:** Algorithm of semantic-based image retrieval using an Ontology

|  |
| --- |
| 1. **Input**: |
| 1. **Output**: |
| 1. **Function** |
| 1. **Begin** |
| 1. **;** |
|  |
|  |
| 1. **End.** |

Call is the number of classifiers in the input image; is the number of classifiers of the image data set used to build an ontology. So, and are constants, assign . In conclusion, the complexity of **algorithm 5** is .

**Figure 11** describes the process of building KD-Tree from object images (***Create KD-Tree***) from MS-COCO, and Flickr image data sets. After building the KD-Tree structure with a set of the training weight vectors. From the set of similar images by contents of an input image; the Relationship KD-Tree structure is built to classify the relationships between objects on the image. Then, the Ontology is built from the KD-Tree by a set of images of leaves and relationships. For each input image, after extracting objects, describing the relationship between objects will create a SPARQL query and perform retrieval on Ontology to extract a set of similar images by semantics to the input image. Therefore, the training weight (***Training weight***) is performed in this experiment. The process of weight training on KD-Tree is inherited **TKDT** algorithm from the work (N. T. Dinh, and Le, T. M., 2022).



**Figure 11.** Building a KD-Tree structure for classifying relationships between objects

Graphical user interface, website

Description automatically generated

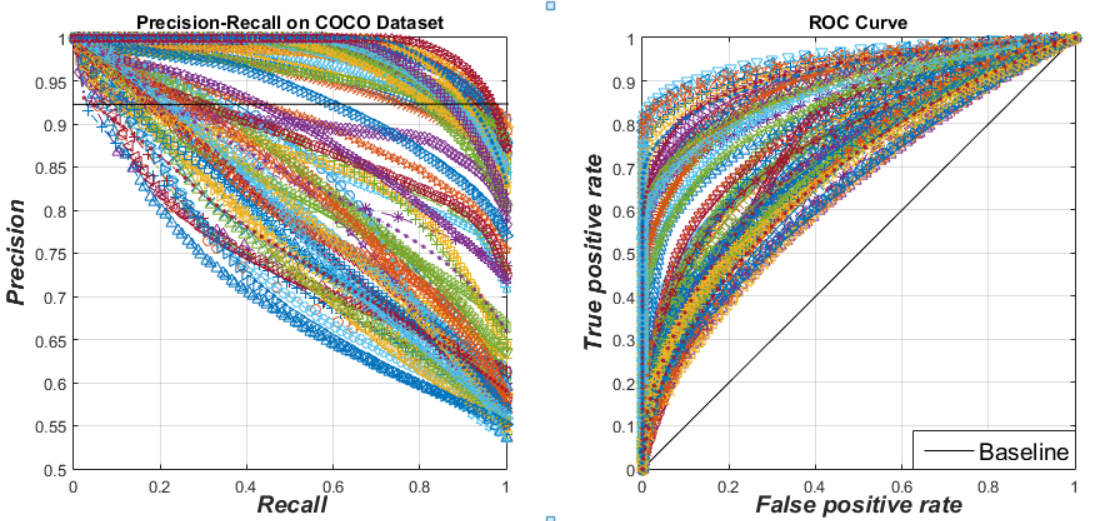
**Figure 12**. The similar images set by semantics of image 11205420.jpg (Flickr)

Experimental results of image retrieval using KD-Tree and Ontology on MS-COCO and Flickr image data sets with validation images and precision, recall, F-measure, and time retrieval are presented in **Table III**.

**Table III.** Experimental results of image retrieval using KD-Tree and Ontology

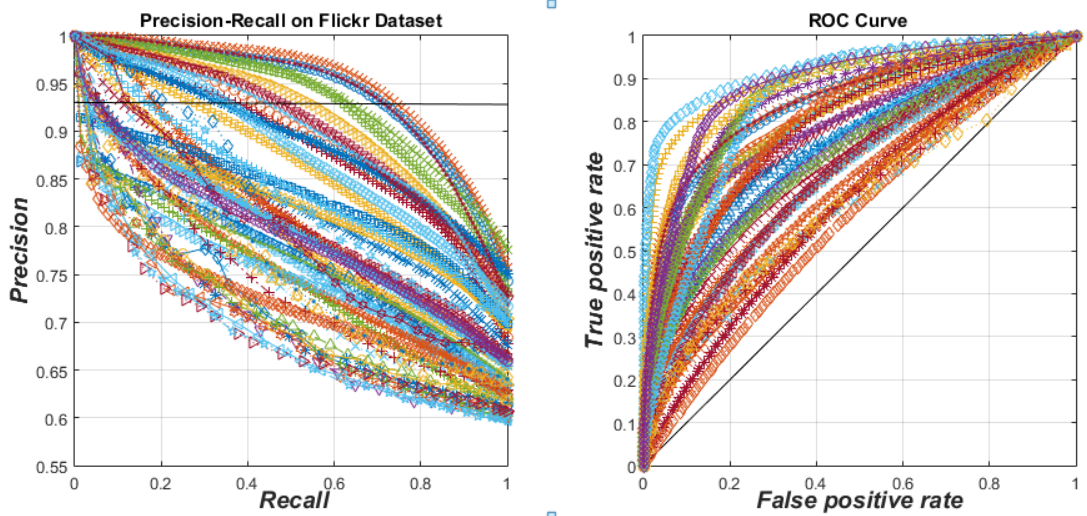
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Data sets*** | ***Validation***  ***images*** | ***Precision*** | ***Recall*** | ***F-measure*** | ***Time query***  ***(ms)*** |
| MS-COCO | 5,000 | **0.9218** | 0.8372 | 0.8774 | 105.90 |
| Flickr | 1,000 | **0.9370** | 0.8939 | 0.9149 | 89.20 |

To illustrate the accuracy of the **SO-KDT** system, the ROC graphs were performed on the MS-COCO and Flickr image sets performed on Matlab 2015 as shown in **figures 13 - 14**. Each curve on the map description of retrieval results with precision and coverage of an image subject in the MS-COCO, Flickr image data sets. At the same time, the corresponding curve in the ROC graph shows the ratio of true and false retrieval results, that is, the area under this curve evaluates the correctness of the retrieval results. The ROC curve graph shows true positive and false positive values according to Recall coverage, values are concentrated on the baseline, and more values are in the true positive region than in the false-positive region.





**Figure 13**. Precision, Recall and ROC curve on MS-COCO image data set





**Figure 14**. Precision, Recall and ROC curve on Flickr image data set

**Table V**. Comparison of mean average precision (MAP) of methods on MS-COCO dataset

|  |  |  |
| --- | --- | --- |
| **Methods (Authors)** | **TopK** | **MAP** |
| CN MAX, (Icarte, R. T.), TopK = 4, 2017 (Icarte, 2022) | 10 | 0.3910 |
| CAMP, (Wang, liu et al.), 2019 (Wang Z., 2019) | 10 | 0.6890 |
| CNN-RNN, (Wang, J., et al.,), 2016 (J. Wang, et al., 2016) | - | 0.6120 |
| Hamming Ranking using AlexNet, (Zhiwei Zhang, et al.), 2021 | 3 | 0.7640 |
| Resnet, (Wen, S., et al.), 2020 (Wen, 2020) | **-** | 0.8110 |
| D-MVE-Hash, (Chenggang Yan et al.), 2020 (Yan, 2020) | **-** | 0.8892 |
| Vision Transformer based Hashing, (Shiv Ram Dubey et al.), 2022 (Dubey, 2022) | **-** | 0.9110 |
| **SO-KDT (Re KD-Tree, RF KD-Tree, Ontology)** | **-** | **0.9218** |

The results comparing the experimental image retrieval performance on the MS-COCO image set with other methods show that the SO-KDT system has a higher image retrieval performance because of the following reasons: (1) the works in **Table V** have not integrated many learning machines to improve the performance of image classification and image retrieval; (2) the SO-KDT system improves the image classification performance by KD-Tree Random Forest before performing to retrieve a set of similar images on KD-Tree; (3) semantic-based image retrieval using ontology also contributes to improved image retrieval performance.

**Table VI**. Comparison of mean average precision (MAP) of methods on Flickr dataset

|  |  |  |
| --- | --- | --- |
| **Methods (Authors)** | **TopK** | **MAP** |
| IRSGS-GCN, (Yoon, 2021) | 10 | 0.5670 |
| CAMP, 2019 (Wang Z., 2019) | 10 | 0.7710 |
| BGAN- 48 bit, (Song, 2018) |  | 0.7030 |
| Unsupervised Deep Discriminative Hashing, (Hu, 2017) | - | 0.8530 |
| LocNar using text-only queries, (Changpinyo) | **-** | 0.8970 |
| **SO-KDT (Re KD-Tree, RF KD-Tree, Ontology)** | **-** | **0.9370** |

The results comparing the experimental image retrieval performance on the Flickr image set with other methods show that the works in table VI use single techniques such as CAM, VSRN, BGAN, and Unsupervised Deep Discriminative Hashing. The **SO-KDT** system has integrated many classification techniques, training the classification model before performing clustering for a semantic-based image retrieval problem.

Experimental results are higher than other works with the same image data set for the following reasons: (1) combining more 225-dimensional features, the results are better with 183-dimensional feature experiments; (2) implementing weight training on KD-Tree Random Forest to improve image classification performance; (3) performing image classification by KD-Tree Random Forest, so the classification efficiency is high; (4) semantic-based image retrieval using ontology also contributes to improving the performance of SO-KDT system. This shows that the combination of ontology and KD-Tree Random Forest is completely feasible and effective for the semantic-based image retrieval problem.

# Conclustions and development

In this paper, a model for image retrieval using weighted KD-Tree data structure and ontology is proposed. For instance, a KD-Tree structure is designed and implemented to store feature vectors of objects present in the images along with their labels. The other KD-Tree also is built to store the visual relationships of the objects in the images. Algorithms are presented to classify objects in an image and to predict their labels based on KD-Tree. Furthermore, a model is introduced how to facilitate semantic query to search for a set of similar images using SPARQL and ontology. Several extensive experiments have been conducted to evaluate model image retrieval. The results showed that MS-COCO and Flickr datasets achieved higher precision scores than previous work on this problem. The precision scores of the newer framework are 0.9128 and 0.9370 respectively. However, there is still room for improving and extending this work. For example, the weighted KD-Tree structure can be trained to work with content-based image retrieval. Other image datasets such as Visual Genome can be used to improve the quality of the relationship KD-Tree. The quality of the ontology which provides the semantics for the symbolic labels can be enriched using knowledge graphs such as Wordnet or Wikidata.

# Acknowledgment

The authors would like to thank the Faculty of Information Technology, University of Science, Hue University for their professional advice for this study. We would also like to thank HCMC University of Food Industry, HCMC University of Education which are sponsors of this research. We also thank anonymous reviewers for their helpful comments on this paper.

**Corresponding author**  
Van The Thanh can be contacted at: [thanhvt@hcmue.edu.vn](mailto:thanhvt@hcmue.edu.vn)

**References**

Asim, M. N., et al. (2019). The use of ontology in retrieval: a study on textual, multilingual, and multimedia retrieval. *IEEE Acces*, 21662-21686.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM, 18*(9), 509-517.

Changpinyo, S., Pont-Tuset, J., Ferrari, V., & Soricut, R. Telling the what while pointing to the where: Multimodal queries for image retrieval. *In Proceedings of the IEEE/CVF International conference on computer vision*, 12136-12146.

Dinh, N. T., & Le, T. M. . (2022). An Improvement Method of Kd-Tree Using k-Means and k-NN for Semantic-Based Image Retrieval System. *In World Conference on Information Systems and Technologies*, 177-187.

Dinh, N. T., Thanh The Van, and Thanh Manh Le. (2022). Semantic Relationship-Based Image Retrieval Using KD-Tree Structure. *Cham: Springer International Publishing, Intelligent Information and Database Systems: 14th Asian Conference, ACIIDS 2022, Ho Chi Minh City, Vietnam, November 28–30, 2022, Proceedings, Part I*.

Dinh, N. T., V. T. Thanh, Thanh, L. M. (2022). A semantic-based image retrieval system using KD-Tree Structure. *Annales Univ. Sci. Budapest. Sect. Comp, 53*.

Dubey, S. R., Singh, S. K., & Chu, W. T. (2022). Vision transformer hashing for image retrieval. *In 2022 IEEE International Conference on Multimedia and Expo (ICME)*, 1-6.

Flickr. (2017). https://[www.kaggle.com/datasets/hsankesara/flickr-image-datas](http://www.kaggle.com/datasets/hsankesara/flickr-image-datas).

COCO (2017). "https://cocodataset.org/download."

Hu, Q., et al. (2017). Pseudo label based unsupervised deep discriminative hashing for image retrieval. *Proceedings of the 25th ACM international conference on Multimedia*.

Icarte, R. T., Baier, J. A., Ruz, C., Soto..A. (2022). How a general-purpose commonsense ontology can improve performance of learning-based image retrieval. *arXiv preprint arXiv*.

Kumar, Y. S., and N. Pavithra (2015). (2015). KD-Tree approach in sketch based image retrieval. *International Conference on Mining Intelligence and Knowledge Exploration, Springer, Cham*.

Kuznetsova, A., et al. (2020). The open images dataset v4. *International Journal of Computer Vision, 128*(7), 1956-1981.

Mehmood Zahid, M. T., Javid, Muhammad Arshad. (2017). Content-based image retrieval and semantic automatic image annotation based on the weighted average of triangular histograms using support véc-tơ machine. *Applied Intelligence, 48*(1), 166-181.

Narasimhulua, Y., et al. (2021). Ckd-Tree: An Improved Kd-Tree Construction Algorithm.

Nhi, N. T. U., & Le, T. M. (2022). A Model of Semantic-Based Image Retrieval Using C-Tree and Neighbor Graph. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 1-23.

Roopak, N., & Deepak, G. (2021). OntoKnowNHS: ontology driven knowledge centric novel hybridised semantic scheme for image recommendation using knowledge graph. *In Iberoamerican Knowledge Graphs and Semantic Web Conference*, 138-152.

Schroeder, B., & Tripathi, S. . (2020). Structured query-based image retrieval using scene graphs. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 178-179.

Song, J., et al. (2018). Binary generative adversarial networks for image retrieval. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Wang, J., et al. (2016). Cnn-rnn: A unified framework for multi-label image classification. *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Wang, S., Wang, R., Yao, Z., Shan, S., & Chen, X. (2020). Cross-modal scene graph matching for relationship-aware image-text retrieval. *In Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 1508-1517.

Wang Z., L. X., Li H., Sheng L., Yan J., Wang X., Shao J. (2019). Camp Cross-modal adaptive message passing for text-image retrieval. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5764-5773.

Wen, S., Liu, W., Yang, Y., Zhou, P., Guo, Z., Yan, Z., ... & Huang, T. (2020). Multilabel image classification via feature/label co-projection. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51*.

Yan, C., Gong, B., Wei, Y., & Gao, Y. (2020). Deep multi-view enhancement hashing for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(4), 1445-1451.

Yoon, S., Kang, W. Y., Jeon, S., Lee, S., Han, C., Park, J., & Kim, E. S. . (2021). Image-to-image retrieval by learning similarity between scene graphs. *Proceedings of the AAAI Conference on Artificial Intelligence, 35*(12), 10718-10726.

Zhang, F., Yahui Gao, and Liuqing Xu (2019). (2019). An adaptive image feature matching method using mixed Vocabulary-KD tree. *Multimedia Tools and Applications*, 1-19.