*Journal of Computer Science and Cybernetics, V.38, N.2 (2022), 193–212*
DOI no 10.15625/1813-9663/38/2/16786

# A METHOD OF SEMANTIC-BASED IMAGE RETRIEVAL USING GRAPH CUT

NGUYEN MINH HAI[1], VAN THE THANH[1], TRAN VAN LANG[2,*]

[1]*HCMC University of Education, 280 An Duong Vuong Street, Ward 4, District 5, Ho Chi Minh City, Viet Nam*
[2]*HCMC University of Foreign Languages - Information Technology (HUFLIT), 828 Su Van Hanh Street, Ward 13, District 10, Ho Chi Minh City, Viet Nam*

**Abstract.** Semantic extraction from images is a topical problem and is applied in many different semantic retrieval systems. In this paper, a method of image semantic retrieval is proposed based on a set of images similar to the input image. Since then, the semantics of the image are queried on the ontology by the visual word vector. The objects in each image are classified and features extracted based on Mask R-CNN, then stored on a graph cut to extract semantics from the image. For each image query, a similar set of images is retrieved on the graph cut and then a set of the visual words is extracted based on the classes obtained from Mask R-CNN, as the basis for querying semantics of an input image on ontology by SPARQL query. On the basis of the proposed method, an experiment was built and evaluated on the image datasets MIRFLICKR-25K and MS COCO. Experimental results are compared with recently published works on the same data set to demonstrate the effectiveness of the proposed method. According to the experimental results, the semantic image retrieval method in the paper has improved the accuracy to 0.897 for MIRFLICKR-25K, 0.873 for MS COCO.

**Keywords.** Image retrieval, ontology, clustering, data mining.

## 1. INTRODUCTION

With the development of the Internet and the proliferation of imaging devices such as digital cameras and photo scanners, the size of digital photo datasets is increasing rapidly. This storage and retrieval of images in response to user semantic expectations from big data has attracted widespread interest from scientists and in industrial fields. Therefore, the semantic query approach is an urgent need for various image retrieval applications.

The semantic-based image retrieval (SBIR) problem is posed as follows: For an input image dataset, with each received query image, a similar set of images should be given, the objects in the image and the annotations of these objects.

Semantic Image Query (SIQ) focuses on studying techniques to reduce the “semantic distance” between low-level features and high-level semantics of images [1]. Low-level image features can be extracted to identify objects of interest in the image, then these objects are

*Corresponding author.
*E-mail addresses*: hainm@hcmue.edu.vn (N.M.Hai); thanhvt@hcmue.edu.vn (V.T. Thanh); langtv@huflit.edu.vn (T.V. Lang).

semantically extracted with semantic descriptions stored in the database [2]. The image semantic retrieval can be queried based on the ontology to determine the concept, high-level semantics of the image [3]. Semantic mapping is used to find the best concept for an image object by supervised or unsupervised machine learning tools to associate low-level features with high-level semantics [4]. Despite a great deal of effort in research on semantic image retrieval, it is still not enough to provide satisfactory performance and satisfy users' desires. Therefore, image query by semantic approach is still a problem with many challenges. The first challenge is to associate low-level features with high-level semantics. The second challenge is bridging the "semantic gap" to query images from content to semantic concept. Therefore, an ontology framework and ontology enrichment are needed so that the extracted semantic features can be applied to any collection of images. In order to improve the efficiency of image query by semantic approach, in this paper, two problems are focused to solve: (1) improve the efficiency of mapping from low-level features to semantic concepts of images through the hierarchical clustering tree GP-Tree which was built in [5]; (2) improve the efficiency of ontology-based image semantic querying.

In addition to the works [1, 2, 3, 4] mentioned above, in recent years, many research groups to improve the efficiency of the semantic image retrieval problem based on ontology have been built [6, 7, 8, 9]; image retrieval based on related feedback techniques [10]; Ontology-based image querying is applied to querying text, multimedia data or identifying relationships between images by using annotations and image features [11, 12]. However, the similar set of obtained images has not really met the expectations of users because of the difference between the computational representation in machines and the natural language of humans. With the goal of reducing the semantic gap to improve the performance of image retrieval, many related research works have been published as follows.

Vijayarajan et al., (2016) [13] performed image retrieval based on natural language analysis to generate SPARQL query to search image set based on RDF-image description (Image Description RDF). The image search process depends on analyzing the grammar of the language to form keywords that describe the image content. This method has not yet implemented classification of image content from color features and spatial features to create keywords to perform search. Therefore, the search from a given query image has not been performed. Filali et al. (2016) [6] proposed an image query system based on visual vocabulary and ontology. For each query image, construct a visual vocabulary and ontology based on the annotation of the image. The ontology is enriched by concepts and relationships extracted from the BabelNet lexical resource. Experiments of the proposed method prove that the query performance is feasible. However, the proposed experimental method has not built a structure to store image data and has not combined image query by content with query by semantics.

Ritika Hirwane (2017) [10] introduced the article on image query by semantic approach. The author introduced techniques of related feedback, classification and evaluation of semantic metrics to build a semantic query model for images. In this work, the author only applies data mining techniques, not search models to improve the efficiency of the image search problem by semantic approach. Spanier et al (2017) [14] built a multi-modal ontology MMO (Multi-Modality ontology) to reduce image semantic distance by using object properties filter (OPF). However, the authors have only built the ontology on a small sample data set belonging to a specific image data domain and have not yet built a structure to store image data. Allani Olfa et al., (2017) [7] proposed the SemVisIR image retrieval system, which combines low-level features of images and high-level semantics. The image dataset is stored in a sample histogram automatically generated using clustering algorithms. SemVisIR modeled the visual aspects of the images through area of histograms and assigned them to automatically built Ontology modules.

Ouiem Bchir et al. (2018) [15] performed an image query based on extracting feature vectors of region objects to perform the partitioning process to speed up image search. In this method, the authors build a semantic mapping between visual features and high-level semantics. Safia Jabeen et al., (2018) [8] built an image search model by clustering visual features combined with semantics of image classifiers. However, clustering of low-level visual features can create clusters of images with different semantics, leading to erroneous search of the query image's semantics. Therefore, the method of semantic classification from low-level feature needs to be applied and at the same time convert this feature word into semantic for the image.

Binbin Yu (2019) [9], proposed an ontology model for semantically processing and retrieving text documents. Building an ontology for semantic information retrieval system includes the following steps: Enter the information to be queried; The system sends information to the ontology to find the corresponding semantic concept; Returns query results to the user. The group of authors experimented by extracting word concepts based on 1000 scientific articles to put them into ontology, creating concepts and literals including 10 groups, each group has 100 articles containing query words. At the same time, the author proposed a genetic algorithm that combined calculations with the frequency of words appearing in the text to return search results. From the experiment, the performance of information querying on the proposed model is feasible. However, in this work, it has not been applied to the image search problem, and has not yet proposed an automatic or semi-automatic ontology building model to enrich data for the ontology. In addition, flexible queries have not been implemented to meet user needs. MN Asim et al (2019) [11] reviewed ontology-based information retrieval methods applied to text queries, multimedia data (images, video, audio) and multimedia data. language. The authors compared the performance and previous approaches for text, multimedia, and multilingual data queries. In this work, the author uses the triad language RDF to perform storage and query on ontology.

Botao Zhong et al (2020) [12] proposed a method to determine the relationship between images by through annotation and image features. The authors built an ontology framework to retrieve the relationship of the image by performing on the protégé to classify the image objects, classify the attributes and determine the relationship between the images, image layers, and object layers. In this work, the authors introduce the HowNet structure and extend it by combining taxonomies to build relationships between image objects. Based on the semantic model, an ontology framework is developed in dealing with image semantic relationships. However, this is only the beginning of building ontology application for images and integrating HowNet into semantics based on ontology automatically.

Bowen Liu et al (2021) [16] introduced an iterative min-cut clustering algorithm for the proposed non-linearly separable data sets. The proposed method bases on graph slicing theory and it does not require calculation of Laplacian matrix, eigenvalues, and eigenvectors. The proposed iterative minimal slicing clustering uses only one formula to map a non-linearly separable dataset into a linear separable one-dimensional representation. However, the weights of the graph edges base only on the distance measure between the data points. This does not lead to very efficient determination of the intersection points on the graph.

In general, recent approaches focus on methods of mapping low-level features with semantic concepts using supervised or unsupervised machine learning techniques; build data models such as graphs, trees, or deep learning networks to store low-level content of images; build ontology to define high-level concepts, etc. However, the SBIR problem relies heavily on reliable external resources such as automatically annotated images, ontology, and training datasets. Group N.M. Hai, V.T. Thanh, and T.V. Lang (2020) [5] also built on semi-supervised learning technique to store images automatically indexed from low-level features of the image. In this work, the GP-Tree tree will be built with each node on the GP-Tree being clustered based on similarity measures by the hierarchical clustering method to efficiently retrieve a set of similar images, and classify input query images, thereby query semantics of images based on ontology. GP-Tree is a multi-branch tree, clustering feature vectors, storing large image datasets, and retrieving images on GP-Tree fast. However, each time a node is split, the GP-Tree can cause similar elements to split into separate branches, so the most similar branch search will not find similar elements that have been branched. Therefore, the retrieval performance is not really high, so it is necessary to improve the retrieval efficiency on the GP-Tree. Graph search overcomes the disadvantage of missing similar data in the tree since all related node clusters can be found. However, this same advantage causes the graph to consume a lot of memory because it has to traverse all the node clusters.

The main purpose of this paper is to improve the GP-Tree to enhance the efficiency of semantic-based image retrieval with the combined model of Mask R-CNN and cluster graph. To solve this problem, the following three specific objectives are addressed. The first is to use the Mask R-CNN model to classify objects in the image, thereby extract the features of the objects to create a general feature descriptor for the input image. The second is to build a graph cut algorithm to cluster graphs based on GP-Tree into sub-clusters with high similarity between elements, thereby improve image data retrieval performance. The third is to build and enrich an ontology framework for querying the semantic of input images. To evaluate the effectiveness and correctness of the proposed method, experiments were performed on the MIRFLICKR-25K and MS COCO image datasets.

The rest of the paper presents the necessary steps on the image retrieval method according to the semantic approach as the main contribution of the paper (Section 2); The experimental results on the datasets, as well as the evaluations, are presented in Section 3; The conclusions are presented in the last section.

## 2. THE METHOD OF SEMANTIC-BASED IMAGE RETRIEVAL

The proposed semantic-based image retrieval method for image datasets consists of two main functions: retrieving similar image sets of a given image and mapping image features such as color, texture, and shape with the semantics of images based on ontology frameworks.

The main processing steps in the proposed model are as follows: perform image segmentation to identify objects and corresponding classes in the image; build cluster graph based on leaves of GP-Tree; Retrieve image on graph cut and semantic query image on the ontology.

## 2.1. Image segmentation and classification

In this paper, a pre-trained Mask R-CNN model is used to detect objects in the image, from there, determine the classifier for the input image. Figure 1 depicts the results of object recognition and classification on MS COCO dataset using Mask R-CNN based on ResNet-101-FPN.

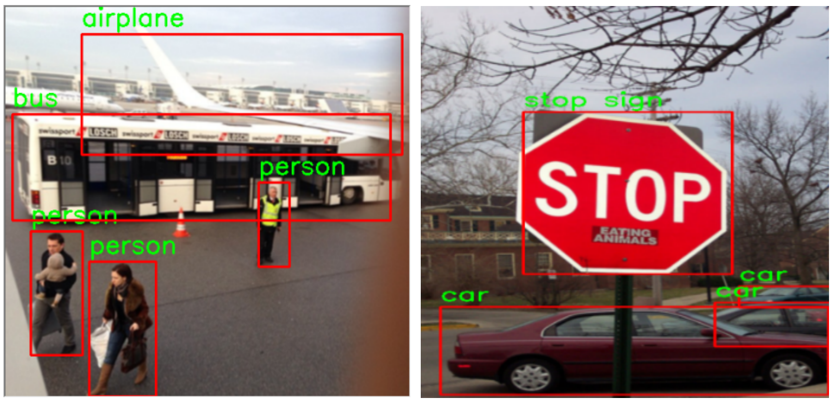
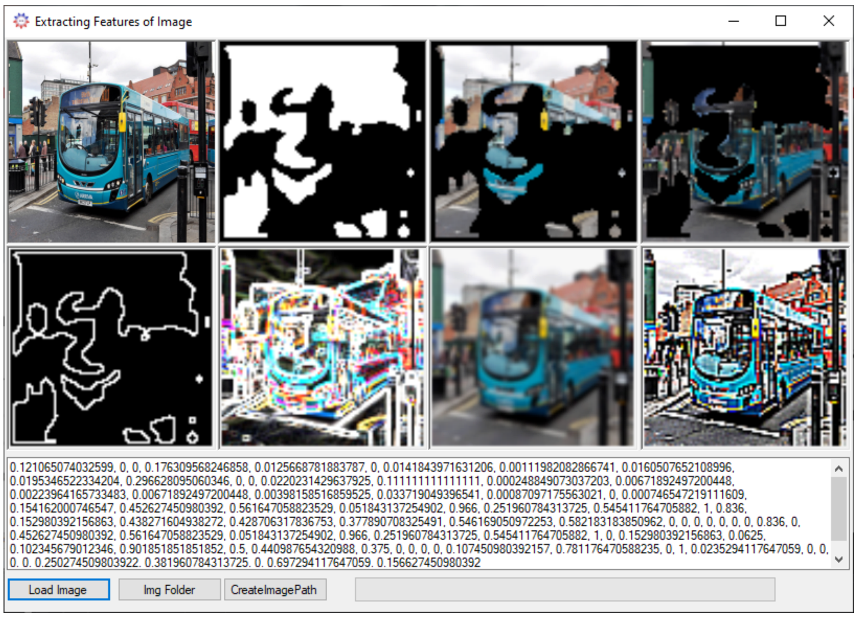


Figure 1: Results of Mask R-CNN using ResNet-101-FPN on images in the COCO dataset

The comparison results between Mask R-CNN and other modern image segmentation methods on the test-dev COCO dataset are described in Table 1 [16]. In which, MNC and FCIS are the winning models in the image segmentation challenges on the COCO dataset in 2015 and 2016, respectively. Mask R-CNN outperforms FCIS +++ in testing/training on various image sizes. Based on this comparison, the Mask R-CNN model using Feature Pyramid Network (FPN) and ResNet-101 deep learning neural network architecture is proposed to recognize and classify objects in the input image.



```
0.121065074032599, 0, 0, 0.176309568246858, 0.0125668781883787, 0, 0.0141843971631206, 0.00111982082866741, 0.0160507652108996,
0.0195346522334204, 0.296628095060346, 0, 0, 0.0220231429637925, 0.111111111111111, 0.000248849073037203, 0.00671892497200448,
0.00223964165733483, 0.00671892497200448, 0.00398158516859525, 0.033719049396541, 0.00087097175563021, 0, 0.000746547219111609,
0.154162000746547, 0.452627450980392, 0.561647058823529, 0.051843137254902, 0.966, 0.251960784313725, 0.545411764705882, 1, 0.836,
0.152980392156863, 0.438271604938272, 0.428706317836753, 0.377890708325491, 0.546169050972253, 0.582183183850962, 0, 0, 0, 0, 0, 0, 0, 0.836, 0,
0.452627450980392, 0.561647058823529, 0.051843137254902, 0.966, 0.251960784313725, 0.545411764705882, 1, 0, 0.152980392156863, 0.0625,
0.102345679012346, 0.901851851851852, 0.5, 0.440987654320988, 0.375, 0, 0, 0, 0, 0.107450980392157, 0.781176470588235, 0, 1, 0.0235294117647059, 0, 0,
0. 0. 0.250274509803922. 0.381960784313725. 0. 0.697294117647059. 0.156627450980392
```

Figure 2: Image feature extraction 000000133819 in the MS-COCO image dataset

For image segmentation, low-level features are used to form a combined feature descriptor. Colors are extracted based on the MPEG-7 dominant color descriptor. The texture features are extracted based on contrast, high-frequency filter, Sobel filter, and Gaussian filter. The shape features are extracted using the Gaussian Laplacian method. A 144-dimensional low-level feature vector is extracted for the image retrieval system in this paper. Table 1 describes the number of components of features in a feature vector representing an image. Figure 2 depicts the extraction of features of image 000000133819 in the MS-COCO image dataset.

Table 1: Description of the number of components of features in a feature vector representing an image

| Feature | Number of elements |
|---|---|
| The color features of the object are according to the Newton color range. | 25 |
| The intensity features of the image are based on the dominant neighbor intensity. | 25 |
| The area features of the object and the background. | 25 |
| The relative position features of the object. | 9 |
| The relative position features of the background image. | 9 |
| The boundary features of the object using the Laplace filter. | 16 |
| The perimeter features of the object and the background. | 16 |
| The surface features of the object using the Sobel filter. | 16 |

### 2.2. The hierarchical clustering tree GP-Tree

On the basis of the analyzed images, the storage locations of the images (URLs) allocated into the leaves of the GP-Tree are organized as follows.

**Definition 1.** The data element $\rho$ at a leaf node is a pair $(\tau, f)$, denoted by $\rho = (f, \tau)$, where $f = (f_1, f_2, \ldots, f_n), f_i \in [0, 1]$, $\forall i = \overline{1, n}$ is the feature vector of the image; $\tau$ is the path to the file stored on disk (URL).

**Definition 2.** The representative element $\sigma$ at an internal node (including the root) is a pair $(c, l)$, denoted by $\sigma = (c, l)$, where,

- $l = (l_1, l_2, \ldots, l_k)$ consists of $k$ links to $k$ nodes to which this internal node links to them.
- $c = (c_1, c_2, \ldots, c_n)$ is the center value corresponding to $n$ features, where each $c_i$ is the average of the $k$ centers of $k$ associated nodes.

**Definition 3.** The GP-Tree is a tree consisting of:

- A root node is a set $C_0$ having $n_0$ representavtive elements $C_0 = \left\{\sigma_i^0 = \left(c_i^0, l_i^0\right) / \forall i = \overline{1, n_0}\right\}$.
- A set $T$ consists of $N_T$ internal nodes, each of which is a set of $C_k$ having $n_k$ representative elements, denoted by $T = \left\{C_k = \left\{\sigma_i^k = \left(c_i^k, l_i^k\right) / \forall i = \overline{1, n_k}\right\} / \forall k = \overline{1, N_T}\right\}$.
- A set $L$ has $N_L$ leaf nodes, each of which is a set of $L_l$ having $m_l$ data elements, denoted by $L = \left\{L_l = \left\{\rho_i^l = \left(f_i^l, \tau_i^l\right) / \forall i = \overline{1, m_l}\right\} / \forall l = \overline{1, N_L}\right\}$.

*Comment:* The number of images stored in the GP-Tree is $\sum_{l=1}^{N_L} n_l$.

**Theorem 1.** *There exists only one branch from root to leaf to insert an data element $\rho$ into a leaf node whose internal nodes in $C_k$ are determined by*

$$\min\left\{\left\|\rho, \sigma_i^k\right\|_2 \mid \forall i = \overline{1, n_k}\right\}.$$

*Proof.* Let $\eta$ be any node on the GP-Tree, $\rho$ be an element to be added to the GP-Tree. If $\eta \in T$, according to Theorem 1, the element $\rho$ can always choose a sub-branch to create a path to the leaf node. If the sub-branch is an internal node, then continue to find the next sub-branch. If $\eta \in L$, then finds the leaf node $L_l$ containing the element $\rho$. Thus, there is always a leaf node containing the element $\rho$. ∎

According to Definition 3 for every leaf node $L_l \in L$, there always exists $C_k$ for this leaf node to link to $l^k$. Suppose, there are two different leaves $L_u$, $L_v \in L$ such that $\rho \in L_u \wedge \rho \in L_v$. That is, there are two paths from $C_k$ to find the leaf node that stores the element $\rho$. According to Theorem 1, only one branch can go to the next child node. So the above assumption is inconsequential. Deduce $L_u \equiv L_v$. Therefore, there is only one leaf node in the tree that stores the element $\rho$.

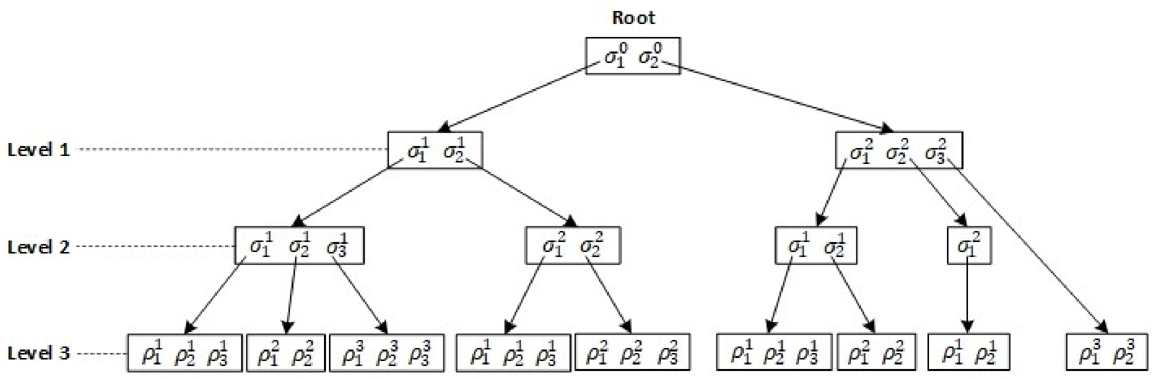


Figure 3: The hierarchical clustering tree GP-Tree with 3 levels

Creating the GP-Tree need to have Algorithm 1 (adding elements to a leaf node) and Algorithm 2 (separating a leaf node). Figure 3 depicts the hierarchical clustering tree GP-Tree with three levels. On the basis of Definition 3, the GP-Tree is a tree that grows by adding an element to the leaf node based on a given threshold $\theta$ and the distance measure $d$ between the element to be added and a representative, consisting of the following cases:

(1) If $d \leq \theta$ then $\rho$ belongs to the current leaf node.

(2) If $d > \theta$ then create a new leaf node at the current parent node and add $\rho$ to the newly created leaf node.

Based on Theorem 1, the algorithm for adding an element to the GP-Tree - insertED is described as Algorithm 1.

```
Algorithm 1: insertED ($\rho, \eta, \theta$, GP-Tree )
Input: $\rho, \eta, \theta$, GP-Tree
Output: GP-Tree
Begin
        If $\eta$ is leaf Then
            $\eta = \{\eta\} \cup \rho$
            If $|\eta| > M$ Then GP-Tree = splitLeaf ($\eta$, GP-Tree )
            EndIf
            Return GP-Tree
        EndIf
        $C_k = \text{argmin}\left(\rho, \sigma_i^k\right)$
        If ($\|\rho, C_k\|_2 \leq \theta$) insertED $(\rho, C_k, \theta)\, q$
        Else
            $L_l \leftarrow$ initialize a leaf node
            $L_l = L_l \cup \rho$
            $C_k = C_k \cup (\rho, L_l)$
            GP-Tree = GP-Tree $\cup L_l$
        EndIf
End
```

On the basis of Definition 3, if the number of elements at any leaf node $L_l$ is greater than the maximum number of elements $M$ in the leaf node, then this leaf node is split into two leaves, creates a parent node linking to these two leaves, and then the parent node become the children of the current parent node. After that, the elements $\rho_i^l$ are distributed to about the two newly created leaves. Splitting a leaf node into two new leaves goes through the following steps:

- Let $L_l$ and $L_r$ be the two new leaves after splitting the leaf $L_s$. Determine the center of leaf $L_s$ the average value of the feature vectors in $L_s$.
- Select an element $\rho_i$ furthest from the center of leaf $L_s$ as the center of leaf $L_l$. Next, select the element $\rho_j$, which is the element furthest from $\rho_i$, is the center of the leaf $L_r$. Then create a new parent node (internal node) of $L_l, L_r$, and add two center elements $\rho_i$ and $\rho_j$ to this new parent node.
- Elements in leaf $L_s$ are allocated to two new leaves according to the nearest node selection rule based on Euclidean distance. Update the central element at the parent node cluster and perform recursion to the root.
- The algorithm for splitting a leaf node on the GP-Tree is described by Algorithm 2 - splitLeaf.

```
Algorithm 2: splitLeaf ($L_s$, GP-Tree )
Input: $L_s$, GP-Tree
Output: GP-Tree Begin
        $C_k \leftarrow$ get the parent node of the leaf $L_s$
        $\rho_i^l = \text{argmax}\left(\sigma_i^k\right)$
        $\rho_j^r = \text{argmax}\left(\rho_i^l\right)$
        $L_l, L_r \leftarrow$ initialize two leaf nodes
        $L_l = \{L_l\} \cup \rho_i^l$
        $L_r = \{L_r\} \cup \rho_j^r$
        **Foreach** $\rho_i^s \in L_s$ **do**
            If $\left\|\rho_i^s, \rho_i^l\right\|_2 < \left\|\rho_i^s, \rho_j^r\right\|_2$ **then**
                $L_l = \{L_l\} \cup \rho_i^S$
            **Else**
                $L_r = \{L_r\} \cup \rho_i^S$
        **EndForeach**
        GP-Tree = GP-Tree $\cup L_l \cup L_r$
        $C_h \leftarrow$ initialize an internal node
        $\sigma_h^k \cdot l^k = C_h$
        $\sigma_l^h \cdot l^h = L_l; \sigma_l^h \cdot c^h = \frac{1}{m_l} \sum_{i=1}^{m_l} \rho_i^l \cdot f_i^l$
        $\sigma_r^h \cdot l^h = L_r; \sigma_r^h \cdot c^h = \frac{1}{m_r} \sum_{i=1}^{m_r} \rho_i^r \cdot f_i^r$
        GP-Tree = GP-Tree $\cup C_h$
        **Return** GP-Tree
**End**
```

On the basis of Definition 3 and Theorem 1, GP-Tree is a multi-branch structure and grows in the direction of leaves. The GP-Tree is created by adding each data element to the structure of the tree. The added element selects only one direction in the tree to determine the leaf node to store; Therefore, if going from the root node to the leaf node, only the most suitable leaf node can be selected for storage. This adding process will perform node splitting and the tree will grow to accommodate the original dataset. The process of creating a tree is described by the following Algorithm 3.

```
**Algorithm 3: createGPT** $(\Gamma, \theta)$
**Input:** Image dataset $\Gamma$, threshold $\theta$
**Output:** GP-Tree
**Begin**
        GP-Tree = null
        $L_0 \leftarrow$ initialize a leaf node
        **Foreach** $(\rho \in \Gamma)$ **do**
            GP-Tree = insertED $(\rho, L_0, \theta$, GP-Tree )
        **EndForeach**
        **Return** GP-Tree
**End**
```

**Theorem 2.** *The complexity of Algorithm 1 is $O(h \times n_k)$, and Algorithm 2 is $O(M)$.*

*Proof.*

For Algorithm 1 - insertED, the main time complexity of the algorithm is used in lines 9 to 10 , which is a recursive step to find the leaf node that best matches the element to be added. We have the number of elements at the internal nodes is $n_k$. Let $h$ be the height of the GP-Tree. The loop $n_k$ times to find the branch closest to element $\rho$ at an internal node has $n_k$ execution time. Therefore, the time complexity of insertED algorithm is $O(h \times n_k)$.■

For Algorithm 2 - splitLeaf, the main time complexity of the algorithm is from lines 7 to 12, this is the step that allocates elements in leaf node $L_s$ to two new leaves. The maximum number of elements in the leaf node is $M$, so the loop $M$ times, which adds data elements to the new leaf node, will take the execution time of $M$. Hence, the time complexity of splitLeaf algorithm is $O(M)$.

### 2.2.1. The clustering graph

The hierarchical clustering graph is generated based on the set of leaf nodes obtained from the construction of the GP-Tree. The tree clustering problem will become a graph clustering problem whose main task is to group related clusters together into a large cluster to avoid missing similar image data in the retrieval process.

Let $v_l = \left(v_1^l, v_2^l, \ldots, v_n^l\right)$ be the center vectors of the leaf node $L_l$, with $v_j^l = \sum_{i=1}^{m_l} f_{ij}^l, \forall j = \overline{1,n}$. Based on Definition 3, the clustering graph Graph-GPTree is defined as follows.

**Definition 4.** The clustering graph Graph-GPTree $G = (V, E)$ is an undirected graph, consisting of:

- Set of vertices $V = \left\{v_l \ \forall l = \overline{1, N_L}\right\}$.
- Set of edges $E = \{e_{lk} = \|v_l, v_k\|_2 / v_l, \ v_k \in V, \ l \neq k\}$.

The Graph-GPTree $G = (V, E)$ is clustered according to the following steps:

(1) Find $e_{ij} = \min(E)$, then two vertices $v_i, v_j$ are grouped into a cluster

(2) Let $v_k$ be the vertex representing the cluster of two vertices $v_i, v_j$, then the edge between $v_k$ and the remaining vertices is determined as follows

$$e_{kt} = \max\left(\|v_t, v_i\|_2, \|v_t, v_j\|_2\right), \forall v_t \in V \backslash \{v_i, v_j\}.$$

(3) Repeat process (1),(2) until $N_L - 1$ clusters are created.

Let $(m_{ij})_{N_L \times N_L}$ be the matrix representing the graph $G$, where $m_{ij} = (e_{ij}, \tau)$, with $\forall e_{ij} \in E, \tau$ is the index of the element $m_{ij}$ in the matrix $(m_{ij})_{N_L \times N_L}$. The GP-Tree-based clustering graph generation algorithm - createGraph is described as Algorithm 4 - createGraph.

**Theorem 3.** *The complexity of Algorithm 4 - createGraph is $O(N_L^2)$.*

*Proof.*

For Algorithm 4 - createGraph, the main time complexity of the algorithm is from lines 10 to 23, which is the step of grouping clusters with similar measures into a single cluster. The maximum associative clustering algorithm using array $\wp$ records the best merged cluster for each cluster. After merging two clusters $i_1$ and $i_2$, the first cluster ($i_1$) represents the merged cluster. If $I[i] = i$, then $i$ is representative of its current cluster. If $I[i] \neq i$, then $i$ has been merged into the cluster represented by $I[i]$ and will therefore be ignored when updating $\wp[i_1]$. In algorithm 4, the two top-level for loops are $O(N_L^2)$, so the overall complexity of the createGraph algorithm is $O(N_L^2)$. ■

```
Algorithm 4: createGraph $(G, \alpha)$
Input: $(v_1, \ldots, v_{N_L})$
Output: list of clusters $\delta$
Begin
        For $i = 1$ to $N_L$ do
                For $j = 1$ to $N_L$ do
                    $m_{ij} \cdot e_{ij} = \|v_i, v_j\|_2$
                    $m_{ij} \cdot \tau = j$
                Endfor
                $I[i] = i$
                $\wp[i] = \text{argmin}\{m_{ik} \cdot e_{ik} \mid i \neq j\}, \forall k = 1 \ldots N_L$
        Endfor
        $\delta = [\ ]$
        For $k = 1$ to $N_L - 1$ do
                $i_1 = \text{argmin}\{\delta\rho[i].e_i \mid I[i] = i\}, \forall i = 1..N_L$
                $i_2 = I[\wp[i_1].\tau]$
                $\delta.\text{push}((i_1, i_2))$
              For $i = 1$ to $N_L$ do
                If $I[i] = i \wedge i \neq i_1 \wedge i \neq i_2$ then
                $m_{i_1 i} \cdot e_{i_1 i} = m_{i i_1} \cdot e_{i i_1} \leftarrow \max(m_{i_1 i} \cdot e_{i_1 i}, m_{i_2 i} \cdot e_{i_2 i})$
              Endif
                If $I[i] = i_2$ then
               $I[i] = i_1$
                Endif
                $\wp[i_1] = \text{argmin}\{m_{i_1 i} \cdot e_{i_1 i} \mid I[i] = i \wedge i \neq i_1\}, \forall i = 1..N_L$
              EndFor
        EndFor
        Return $\delta$
End
```

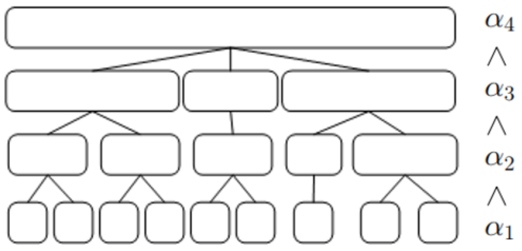


Figure 4: Hierarchical tree with corresponding $\alpha$ values

## Graph cut

The cut-clustering algorithm generates clusters of nodes of $G$ with a certain value of $\alpha$. Once such a clustering is produced, the clusters are grouped into single nodes and the same algorithm is applied on the resulting graph. When collecting a set of nodes, these nodes are replaced by a new single node. By repeating the same process a number of times with smaller $\alpha$, a hierarchy of clusters is created. Figure 4 depicts a hierarchical tree of clusters with different $\alpha$ values.

The clustering algorithm cuts iteratively until the desired number of clusters or the desired size is satisfied, or when only a single cluster containing all nodes is identified. The hierarchical cut clustering algorithm is described in Algorithm 5 - cutGraph as follows.

```
Algorithm 5: cutGraph ($G, \alpha$)
Input: $G, \alpha$
Output: $\gamma$ # Set of all clusters at values $\alpha$
Begin
    $G_0 = G$
    For ($i = 1; ; i++$) do
        Assign a value of $\alpha_i$ less than
        $\delta_i \leftarrow$ cut cluster $G_i$ with $\alpha_i$
        $\Upsilon = \{\Upsilon\} \cup \delta_i$
        If (clusters returned are of desired number and size) or
            (clustering failed to create nontrivial clusters) then break
        EndIf
        $G_{i+1} = G_i \backslash \delta_i$
    EndFor
    Return $\Upsilon$
End
```

### 2.3. Image retrieval on the clustering graph

The process of retrieving images on the clustering graph is a combination of two retrieval phases: retrieve on the tree and retrieve on the graph. First, the query image gets feature extracted to create the data element $\rho$. The element $\rho$ is then compared with the elements representing $\sigma$ at the internal nodes of the tree from root to leaf. At each internal node, the algorithm will navigate to the branch closest to $\rho$ in Euclidean measure until it finds a suitable leaf node. This leaf node is retrieved on the clustering graph $G$ to find the set of its neighbor leaf node $\Omega$, thereby extract the set of elements $\Psi$ similar to $\rho$. The algorithm for retrieving the image on the cluster graph is described by Algorithm 6 - retrieveGraph.

**Theorem 4.** *The complexity of Algorithm 6 - retrieveGraph is* $O(h \times \log(h) \times k)$.

*Proof.*

For Algorithm 6 - createGraph, the main time complexity of the algorithm is from lines 5 to 13, which is the step of extracting data elements in the set of leaf nodes. The image retrieval algorithm on the clustering graph performs a recursive call on a branch of the tree to find a suitable leaf node and extract the elements in that leaf node, each time a node traverses the representative element of the tree. The obtained leaf node will continue the query on the graph with $k$ clusters. Therefore, the Image Query Algorithm has a complexity of $O(h \times \log(h) \times k)$, where $h$ and $k$ are the height of the GP-Tree and the number of clusters of the graph, respectively. Therefore, this algorithm is feasible and has finite steps to execute.

■

```
Algorithm 6: retrieveGraph $(r, \rho, G)$
Input: $r, \rho, G$
Output: $\Psi$
Begin
        If $r$ is a leaf then
            $\Psi \leftarrow \{\rho_i^r \mid i = 1..m_r\}$ d
        Else
            $\varphi \leftarrow \text{argmin}\left\{\|\rho, \sigma_i^r\|_2 \mid i = 1..n_r\right\}$
            If $\varphi$ is a leaf then
                $\Omega \leftarrow$ get the neighbor leaf nodes in the cluster containing the leaf
                node $\varphi$
                $\Psi \leftarrow \rho_i^\varphi$
                For $L_k \in \Omega$ do
                    $\Psi \leftarrow \Psi \cup \left\{\rho_i^k \mid i = 1..m_k\right\}$
                End For
            Else
                retrieveGraph$(\varphi, \rho)$
            EndIf
        EndIf
        Return $\Psi$
End
```

## 2.4. Querying image semantics on ontology

### 2.4.1. Ontology

In order to reduce the semantic gap between low-level visual features and semantics of images, an ontology framework is built for image datasets based on the RDF/XML and OWL triple languages. With the input query image, the visual word set is extracted based on the classes obtained from Mask R-CNN; From there, a SPARQL query is created and queried on the ontology to determine the concept and semantics of the image. The semantic mapping process is used to analyze and find the best annotation for the objects in the image.

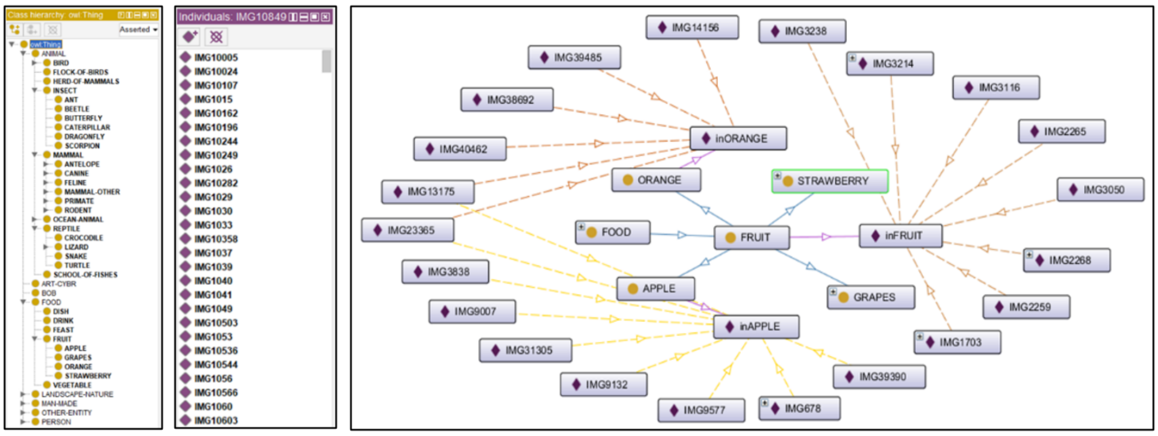


Figure 5: An example of an ontology applied to the image dataset ImageCLEF

The classes of the image datasets are built in a hierarchical form. A semantic dictionary to define classes of images extracted from WordNet. Each image is an individual/instance of one or more classes in ontology. Figure 5 is an example of an ontology built on Protégé for ImageCLEF image datasets.

### 2.4.2. Image semantic query

SPARQL is a query language on data sources described as RDF or OWL triples. With the input query image that can contain one object or many objects, Mask R-CNN is used to extract the visual word vector; This vector contains one or more semantic classes of the image, and automatically generates a SPARQL statement (AND or OR), from which to query the ontology to find the annotations of the query image. The algorithm for automatically generating SPARQL queries is described through Algorithm 7 - createSPARQL as follows

**Algorithm 7: createSPARQL(W)**

```
Input: Visual word vector $W$
Output: SPARQL
Begin
        SPARQL = $\emptyset$;
        $n = W.\ count$;
        Select DISTINCT ?Img
        WHERE{";
        For ($i = 0.n$) do
              SPARQL+= " ⟨subject⟩ : $W(i)$+ "rdf:type" +
              ⟨object⟩ : + ? Img" + "UNION/AND";
        End
        SPARQL+=}";
        Return SPARQL;
End
```

The query result on the ontology is a set of URIs and the metadata of the similar image dataset and its semantics. For each image class, the system performs semantic analysis of the ontology-based class and queries the annotation for this class from the cognitive synonym of the ontology.

## 3. EXPERIMENTAL RESULTS

### 3.1. Image datasets

To demonstrate the effectiveness of the proposed method, two popular image datasets are used for testing MIRFLICKR-25K and MS COCO. The MIRFLICKR-25K dataset consisting of 20,015 images and 24 class labels was collected from the Flickr website. Each image represents a single object and is annotated with a number of semantically relevant text tags. The MS COCO dataset consists of 85,000 and 80 class labels. Each image consists of many objects and there is a caption for each object in the image. The details of these two image datasets are described in Table 2.

Table 2: Information on experimental image datasets

| Image dataset | Number of images | Number of class labels |
|---|---|---|
| MIRFLICKR-25K | 20,015 | 24 |
| MS COCO | 85,000 | 80 |

## 3.2. The proposed semantic-based image retrieval approach

The semantic-based image retrieval system based on cluster graph and ontology model is called GP-SIR. The retrieval system consists of two stages: (1) the preprocessing stage which generates the graph cut of the leaf node clusters and builds a semi-automatic ontology framework based on the RDF triple language and stores it in N3 format; (2) the retrieval stages searches for a set of similar images based on the GP-SIR model, image semantic segmentation by Mask R-CNN to find visual words and generates SPARQL queries, to query on the ontology to extract the semantics of the image. Figure 6 shows the architecture of GP-SIR based on the graph cut model with two specific phases as follows.

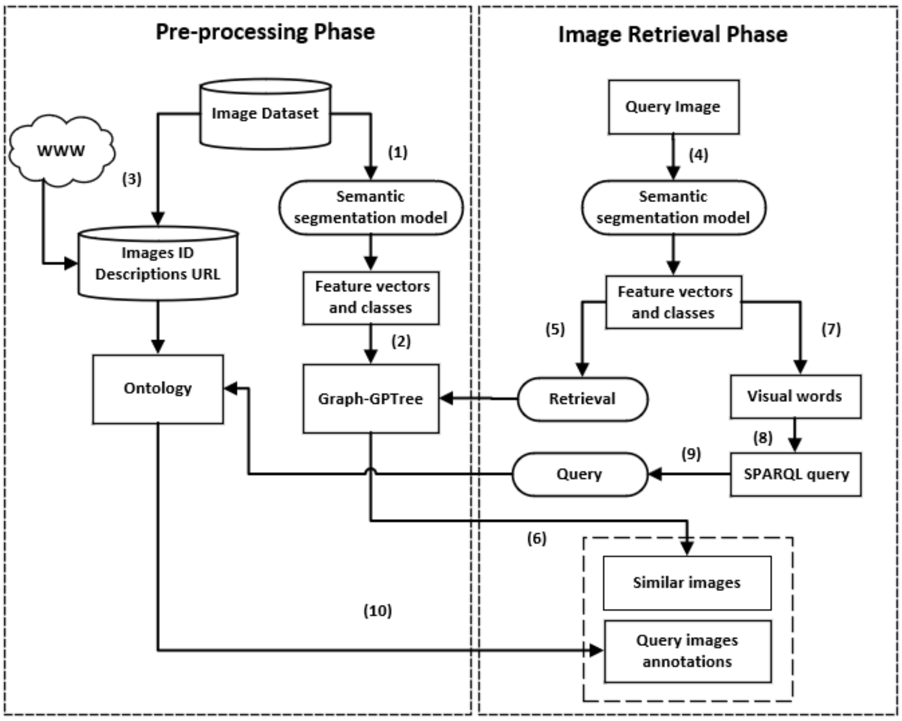


Figure 6: The semantic-based image retrieval system model GP-SIR

### Pre-processing phase

Step 1: The input image is segmented to determine the classes of the objects in the image, and at the same time, the low-level feature vectors and corresponding class labels(1).

Step 2: Create a combined model of GP-Tree and clustering graph (Graph-GPTree) from the data sample set (2).

Step 3: Build a semi-automatic ontology framework from the WWW dataset and image datasets (3).

**Image retrieval phase**

Step 1: Create a representative data sample for the query image consisting of feature vectors and corresponding class labels (4);

Step 2: First, perform a retrieve on GP-Tree, based on the Euclidean measure to find the most suitable leaf node on the tree (5), then, get the set of neighboring nodes belonging to the cluster with the leaf node on the clustering graph Graph-GPTree - the result is a set of similar images sorted by measure (6);

Step 3: The visual word set is created based on the class labels of the retrieval image obtained in Step 1(7), from which the SPARQL query is generated (8) and executed on a built ontology framework (9). The result of the query is the annotations of the query image (10);

Step 4: Combine the results obtained from Step 2 and Step 3 (11), a set of similar images and the annotations of the query image is obtained.

## 3.3. Application

Each image in image datasets is segmented and determined classes for objects in the image by Mask R-CNN. These objects are extracted features and described as feature vectors 144 dimensions. The acquired feature vectors are stored on GP-Tree based on the Euclidean distance. Then, the clustering graph Graph-GPTree was built according to Definition 4.

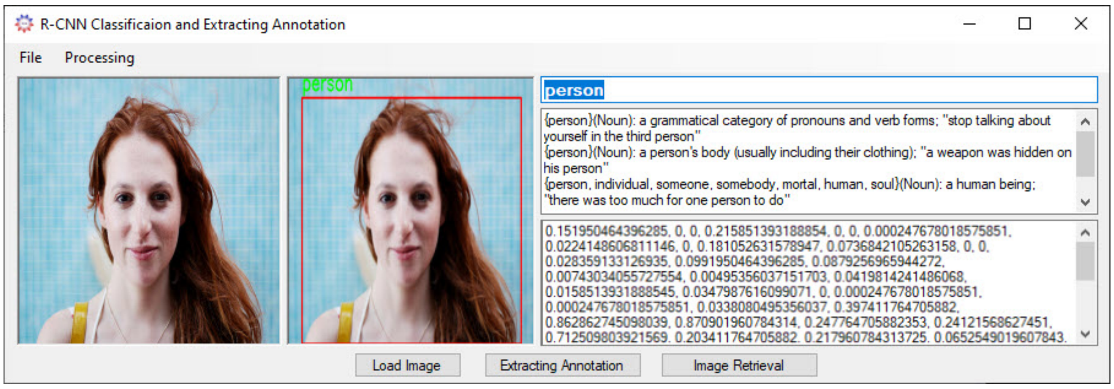
With a query image, the GP-SIR system determines the classes of objects in the image and extracts the features of the image, then retrieves images by content on GP-Tree and Graph-GPTree to find a set of similar images by content. A visual word set is extracted from the obtained classes of the query image. At the same time, the SPARQL query is also automatically generated to query the semantics of the input image on the ontology. Figure 7 depicts an image search result of the MIRFLICKR-25K dataset according to the semantics of the GP-SIR system.

## 3.4. Experimental evaluation

To evaluate the effectiveness of image retrieval, the factors used to evaluate include: precision, recall, and F-measure. On the basis of the existing performance values, Table 3 compares the image retrieval performance values on the GP-Tree tree and on the Graph-GPTree of the image datasets MIRFLICKR-25K and MS COCO.

Based on the experimental data, Figures 8 and 9 describe the Precision-Recall curves and ROC curves made to evaluate the accuracy of the proposed image retrieval system. Each curve describes the retrieval result of a set of image data in the image dataset. The area under these curves shows the accuracy of the image query. The image retrieval performance of GP-SIR on the image datasets MIRFLICKR-25K and MS COCO shows that the proposed method is effective.

To evaluate the accuracy and efficiency of the proposed image query system, the experimental results are compared with other studies on the same image data set. The average accuracies of the proposed query system compared with the ones of other semantic image query systems on each dataset in Tables 4 and 5 show that the query results of the proposed method are relatively accurate.

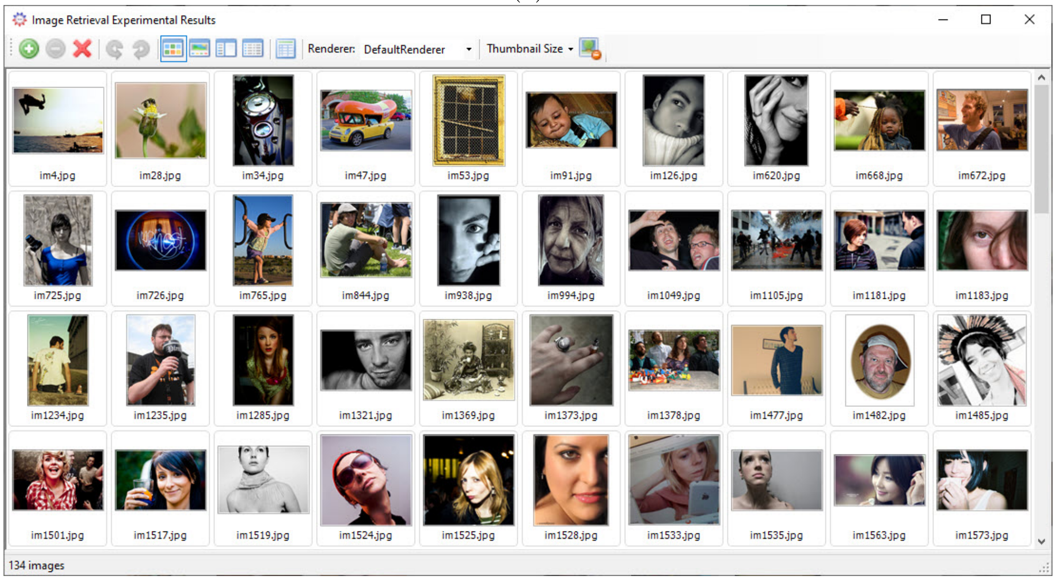


person

```
{person}(Noun): a grammatical category of pronouns and verb forms; "stop talking about
yourself in the third person"
{person}(Noun): a person's body (usually including their clothing); "a weapon was hidden on
his person"
{person, individual, someone, somebody, mortal, human, soul}(Noun): a human being;
"there was too much for one person to do"
```

```
0.151950464396285, 0, 0, 0.215851393188854, 0, 0, 0.000247678018575851,
0.0224148606811146, 0, 0.181052631578947, 0.0736842105263158, 0, 0,
0.028359133126935, 0.0991950464396285, 0.0879256965944272,
0.00743034055727554, 0.00495356037151703, 0.0419814241486068,
0.0158513931888545, 0.0347987616099071, 0, 0.000247678018575851,
0.000247678018575851, 0.0338080495356037, 0.397411764705882,
0.862862745098039, 0.870901960784314, 0.247764705882353, 0.24121568627451,
0.712509803921569, 0.203411764705882, 0.217960784313725, 0.0652549019607843,
```

(a)



(b)

Figure 7: (a) The results of classification, feature extraction, and annotations query for the img1553.jpg image of the MIRFLICKR-25K dataset; (b) the result of retrieving the image set is similar to the query image of the GP-SIR system.

Table 3. Image retrieval performance on GP-Tree and on Graph-GPTree of the image datasets MIRFLICKR25 K and MS COCO.

| Method | MIRFLICKR-25K | | | MS COCO | | |
|---|---|---|---|---|---|---|
| | Avg. precision | Avg. recall | Avg. Fmeasure | Avg. precision | Avg. recall | Avg. F-measure |
| GP-Tree | 0.717 | 0.683 | 0.700 | 0.647 | 0.564 | 0.603 |
| Graph-GPTree | 0.897 | 0.793 | 0.842 | 0.873 | 0.764 | 0.815 |

The data from the above tables show that our proposed method has higher accuracy when compared with other retrieval methods on the same set of images. It shows that our proposed method is effective.

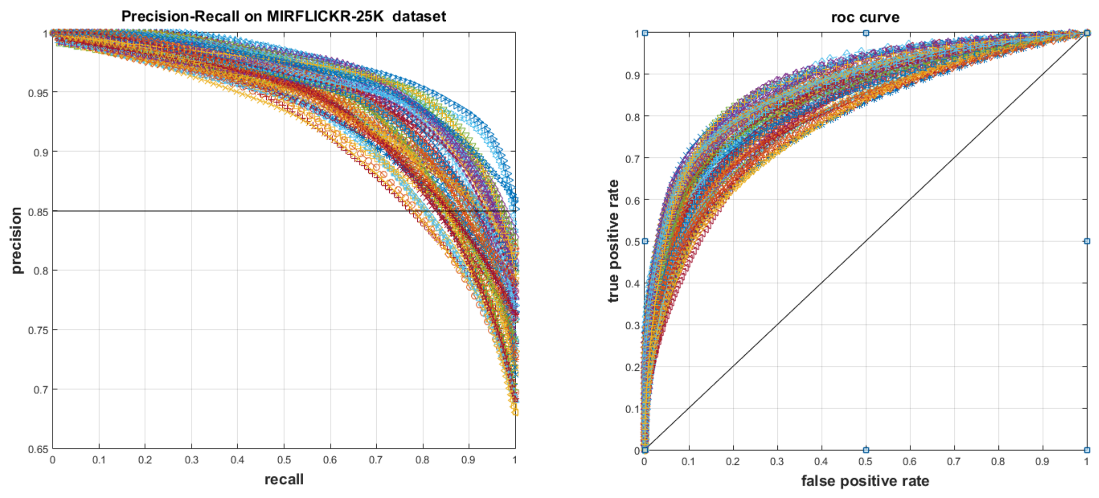


Figure 8. Image retrieval performance on MIRFLICKR-25K image dataset of GP-SIR

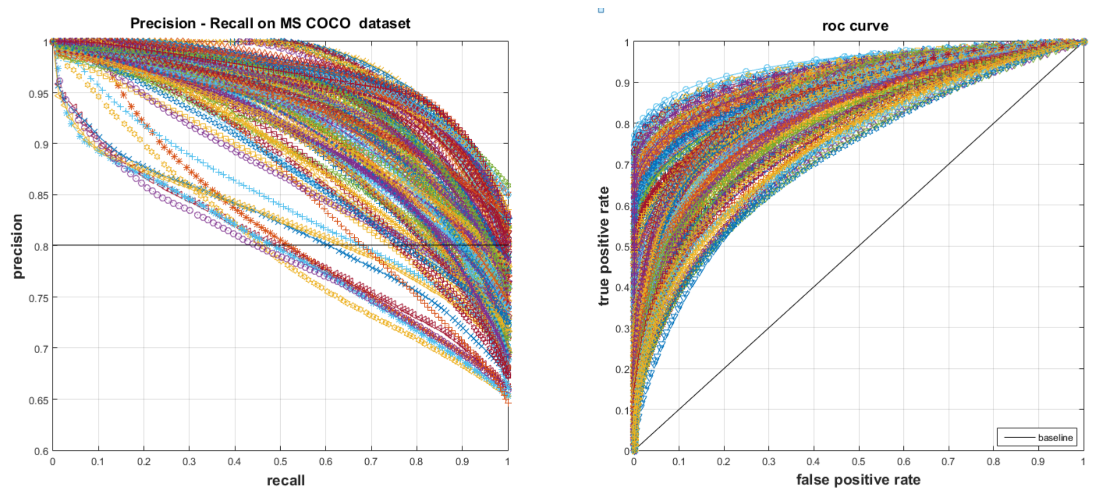


Figure 9: Image retrieval performance on MS COCO image dataset of GP-SIR

Table 4: Accuracy comparison between methods on dataset MIRFLICKR-25K

| Method | Mean Average Precision (MAP) |
|---|---|
| Xuelong Li, 2017 [17] | 0.621 |
| Di Wang, 2017 [18] | 0.652 |
| GP-SIR | **0.897** |

Table 5: Accuracy comparison between methods on dataset MS COCO

| Method | Mean Average Precision (MAP) |
|---|---|
| Yue Cao, 2018 [19] | 0.843 |
| Wen Gu, 2019 [20] | 0.835 |
| GP-SIR | **0.873** |

## 4. CONCLUSION

In this paper, a semantic image retrieval method is proposed with the combination of the Mask R-CNN network and graph-cut cluster graph. For each input image, features and image classes are extracted by Mask R-CNN to form a visual word. From there, the SPARQL query is automatically generated from the visual word and executed on the ontology to retrieve the similar and semantic image sets. An image query model based on Mash R-CNN, clustering graph, and ontology (GP-SIR) is proposed and tested on the popular image datasets, MIRFLICKR-25K and MS COCO with the accuracy of 0.897 and 0.873, respectively. The experimental results are compared with other studies on the same set of images, showing that our proposed method has higher accuracy. In the future research direction, we continue to improve the feature extraction methods, supplement and enrich the ontology, and build an ontology based image search system on WWW.

## REFERENCES

[1] C. Bai, J.-n. Chen, L. Huang, K. Kpalma, and S. Chen, "Saliency-based multi-feature modeling for semantic image retrieval," *Journal of Visual Communication and Image Representation*, vol. 50, pp. 199–204, 2018.

[2] M. Jiu and H. Sahbi, "Nonlinear deep kernel learning for image annotation," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1820–1832, 2017.

[3] M. Fachrurrozi, A. Fiqih, B. R. Saputra, R. Algani, A. Primanita *et al.*, "Content based image retrieval for multi-objects fruits recognition using k-means and k-nearest neighbor," in *2017 International Conference on Data and Software Engineering (ICoDSE).* IEEE, 2017, pp. 1–6.

[4] A. S. M. Arif, A. Y. Ali, and S. R. Hasan, "A modularized approach for similarity based object retrieval," *JCIT*, vol. 1, no. 1, 2010.

[5] N. M. Hai, T. Van Lang *et al.*, "Semantic-based image retrieval using hierarchical clustering and neighbor graph," in *World Conference on Information Systems and Technologies.* Springer, 2022, pp. 34–44.

[6] J. Filali, H. Zghal, and J. Martinet, "Towards visual vocabulary and ontology-based image retrieval system," in *International Conference on Agents and Artificial Intelligence*, vol. 2, 2016, pp. 560–565.

[7] O. Allani, H. B. Zghal, N. Mellouli, and H. Akdag, "Pattern graph-based image retrieval system combining semantic and visual features," *Multimedia Tools and Applications*, vol. 76, no. 19, pp. 20 287–20 316, 2017.

[8] S. Jabeen, Z. Mehmood, T. Mahmood, T. Saba, A. Rehman, and M. T. Mahmood, "An effective content-based image retrieval technique for image visuals representation based on the bag-of-visual-words model," *PloS One*, vol. 13, no. 4, p. e0194526, 2018.

[9] B. Yu, "Research on information retrieval model based on ontology," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 1–8, 2019.

[10] R. Hirwane, "Semantic based image retrieval," *Int. J. Adv. Res. Comput. Commun. Eng*, vol. 6, no. 4, pp. 120–122, 2017.

[11] M. N. Asim, M. Wasim, M. U. G. Khan, N. Mahmood, and W. Mahmood, "The use of ontology in retrieval: a study on textual, multilingual, and multimedia retrieval," *IEEE Access*, vol. 7, pp. 21 662–21 686, 2019.

[12] B. Zhong, H. Li, H. Luo, J. Zhou, W. Fang, and X. Xing, "Ontology-based semantic modeling of knowledge in construction: classification and identification of hazards implied in images," *Journal of Construction Engineering and Management*, vol. 146, no. 4, p. 04020013, 2020.

[13] V. Vijayarajan, M. Dinakaran, P. Tejaswin, and M. Lohani, "A generic framework for ontology-based information retrieval and image retrieval in web data," *Human-centric Computing and Information Sciences*, vol. 6, no. 1, pp. 1–30, 2016.

[14] A. B. Spanier, D. Cohen, and L. Joskowicz, "A new method for the automatic retrieval of medical cases based on the radlex ontology," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, no. 3, pp. 471–484, 2017.

[15] O. Bchir, M. M. B. Ismail, and H. Aljam, "Region-based image retrieval using relevance feature weights," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 18, no. 1, pp. 65–77, 2018.

[16] B. Liu, Z. Liu, Y. Li, T. Zhang, and Z. Zhang, "Iterative min cut clustering based on graph cuts," *Sensors*, vol. 21, no. 2, p. 474, 2021.

[17] D. Hu, F. Nie, and X. Li, "Deep binary reconstruction for cross-modal hashing," *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 973–985, 2018.

[18] D. Wang, Q. Wang, and X. Gao, "Robust and flexible discrete hashing for cross-modal similarity search," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2703–2715, 2017.

[19] Y. Cao, M. Long, B. Liu, and J. Wang, "Deep cauchy hashing for hamming space retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1229–1237.

[20] W. Gu, X. Gu, J. Gu, B. Li, Z. Xiong, and W. Wang, "Adversary guided asymmetric hashing for cross-modal retrieval," in *Proceedings of The 2019 on International Conference on Multimedia Retrieval*, 2019, pp. 159–167.

*Received on December 09, 2021*
*Accepted on July 04, 2022*