THE IMPROVEMENTS OF SEMANTIC-BASED IMAGE RETRIEVAL USING HIERARCHICAL CLUSTERING TREE

Nguyen Minh H๐¹, Van The Thanh², Tran Van Lang³

¹Faculty of Physics, HCMC University of Education, Vietnam
²Office of Scientific Research Management and Post-Graduated Affairs, HCMC University of Food Industry, Vietnam
³Institute of Applied Mechanics and Informatics, Vietnam Academy of Science and Technology, Vietnam

haimn@hcmue.edu.vn, thanhvt@hufi.edu.vn, langtv@vast.vn

ABSTRACT: Finding similar images and extracting semantics of them is an important problem in many multimedia systems. In this paper, we present a semantic-based image retrieval on the H-Tree, which is a structure of a hierarchical clustering tree. The H-Tree is created based on the method of hierarchical clustering on the visual feature vectors of the images. In the H-Tree structure, the method of element partitioning and neighbor connection among clusters is proposed to build a multi-branch tree structure, which is a hierarchy on each node of the tree, for storing large image dataset. The features of image are extracted rely on convolutional and sub-sampling operations to describe the low-level visual features of images. The image retrieval model is proposed including the method of k-nearest neighbor (k-NN) and searching graph to increase efficiency for finding clusters in the H-Tree, which have the closest similarity to the query image. On the base of the proposed theory, this retrieval system has experimented on the ImageCLEF image dataset which has 20,000 images and 276 categories. The retrieval results of the paper are compared with some of the results of the recently published works on the same dataset. From there, the proposed image search method is highly feasible and effective.

Keywords: SIBR, hierarchical clustering, similar images, image retrieval.

I. INTRODUCTION

In recent years, there are many works about image retrieval which has been focused on reducing the semantic gap or the linguistics gap between low-level features and high-level semantics [20–24]. The state-of-the-art techniques have been used to decrease semantic gap and enhance the efficiency of image retrieval such as using ontology to define high-level concepts, using machine learning to associate low-level features with the high-level semantic concepts [17], creating the retrieval model based on tree or graph structures to improve the efficiency of the image retrieval systems [9,13], etc. Although there is a lot of research in this area, there is still no general approach for retrieving images at high-level semantics, due to the diversity and complexity of the visual semantics. Incorporating semantics into the search process based on an ontology that we have built to describe the semantic relationships among image subjects to expand of image search results according to linguistic concepts, that the human has accepted compared with the meaning of the information from image features; however, we still need the method to reduce the semantic distance from the content of the objects.

There are two main problems in content-based image retrieval (CBIR) that describe image visual content and construct data structure for creating index to image dataset. With respect to the semantic-based image retrieval (SIBIR), two above issues also have to be resolved because the input data of the retrieval process is an image; two more issues need to be added, however, generating the query image to the semantics of image contents as a visual word vector, and queries by image semantics to find out similar images as well as the semantics of them rely on the ontology was built by RDF triple language.

According to the above analysis, we present an improvement of the H-Tree [9] by incorporating the method of k-nearest neighbor (k-NN) and ontology to increase the accuracy of searching clusters on the H-Tree. The contributions of the paper include (1) constructing convolution operations to extract low-level visual features of images; (2) building a multi-branched H-Tree structure to classify image dataset and help image retrieval process achieves high efficiency; (3) proposing model and algorithms of SIBIR-HT to retrieve semantics of similar images; (4) building ontology for image dataset on the basis of triple language RDF and creating a SPARQL command to retrieve similar images and their semantics based on visual word vector; (5) constructing the SBIR-HT system based on proposed model and algorithms; evaluate the experiment on the imageCLEF dataset.

The rest of the paper is as follows. In section 2 analysis of related works and solutions, as well as feasibility analysis of the approach of the paper presented; components and the SBIR-HT model combining H-Tree and ontology are presented in section 3. In section 4 presents the experiment application and evaluates results base on the proposed theory, conclusions and future works are presented in section 5.

II. RELATED WORK

In recent years, there were many studies for improving the efficiency of content-based and semantic-based natural image retrieval such as image retrieval and semantic parsing using the image regional weighting factors and applying unsupervised learning technology [1], image retrieval based on the S-Tree multi-branch tree data structure [4], semantic-based image retrieval relied on hierarchical classification model using deep learning technique [5], query semantic based on the image visual features classification techniques and indexing to classify images into different
semantic classes using dictionary data technique [8]; Analysis the basic techniques for the CBIR and SBIR problems and propose the features of an image to be used for image retrieval processing [3].

Constructing binary hierarchical tree and SVM technique to classify visual object classes of large size images [2]; however, this binary tree structure is only used to classify images but has not yet created a storage structure for images. Content-based image retrieval and extract semantic of the image using SVM technique, and applying dictionary data technique for mapping from low-level visual features to the semantics of images [6, 7]; however, these articles only apply data mining techniques that have not built a search model nor created a SPARQL query to performing searching and semantic query of image.

The method of semantic-based image retrieval relied on the basis of data mining techniques in the form of partitioning of low-level visual features vectors on hierarchical clustering tree H-Tree [9] or the graph clustering structure C-Graph that achieve results quite well [13]; however, these articles have only given clustering model and similar image retrieval that have not yet processed the extraction features of input image dataset so the accuracy of this method will not be high if the extraction of features is not good.

Vijayarajan et al., (2016) implemented image retrieval based on analyzing natural language to find similar images by created a SPARQL query relied on the RDF image description [11]. The process of image retrieval depends on analyzing the grammar of the language to create keywords that describe the image content. This method has not yet classified image content from color and spatial features to create keywords to perform retrieval; therefore, it can not perform the image retrieval process from a given query image.

M. Jiu and H. Sahbi (2017) used a multilayered neural network based on different nonlinear activation functions on each layer. The SVM technique was used to determine semantics for similar images based on Bag-of-Words [15]. With this design, the classification according to deep learning technique is limited because the neural network is fixed in terms of the number of layers.

M. Tzelepi and A. Tefas (2018) proposed a convolutional neural network to content-based image retrieval using the Caffe Deep Learning framework. The image retrieval process based on classifying images from low-level features relied on relevant feedback [16].

Safia Jabeen et al. (2018), proposed a method clustering visual features combined with the semantics of the categories of images for building an image retrieval model based on bag-of-visual-words [19]; however, finding semantic of a query image can be inaccurate because of clustering low-level features can create clusters of images with different semantics. Therefore, classifying the semantic of low-level features needs to be applied to map these features into the semantics of images.

Xiao Xie et al., (2018) proposed a convolutional neural network to classify the visual features of images and rendering word vector for finding similar images. In this paper, the proposed method has not yet performed querying on Ontology to determine semantics for images [18]. Applying deep learning neural network to classify images and evaluate the similarity among two images based on semantic measure [5, 16]; however, the proposed methods in these papers have not built a finding model in order to speed up for semantic-based image retrieval problem.

Bjorn Barz et al. (2019) [14] applying a deep neuron network to classify images and evaluate the similarity between two bases on semantic measure images. However, this method does not build a retrieval model to speed up the semantic-based image retrieval problem.

Based on our initial results on this problem presented in [9], we continue to improve the H-Tree hierarchical clustering tree by combining with convolution and subsampling operations aim to extract the low-level visual features of the image dataset to improve the accuracy of the partitioning of the image sets on the H-Tree tree; the k-nearest neighbor algorithm we incorporate in querying similar image sets to classify the objects for query image in order to increase the performance of similar image retrieval on the H-Tree.

III. SBIR-HT: THE SEMANTIC-BASED IMAGE RETRIEVAL SYSTEM

A. The architecture of SBIR-HT system

Based on the constructed ontology and the data stored on the H-Tree tree, the semantic-based image retrieval model based on the H-Tree hierarchical clustering tree is described in Figure 1, called SBIR-HT. The retrieval system consists of two phases: the pre-processing phase to extract the features of the image in the image dataset and store it on the H-Tree; also create an ontology for the image dataset; In image retrieval phase, there are 2 processes: retrieving on the H-Tree and query on the ontology was built. Retrieval on the H-Tree for retrieving similar images and visual word vectors in the image dataset. Query on the ontology to find an extended similar set of images and corresponding semantics.

In the pre-processing phase of the model, the image data set is extracted directly on the image objects in order to extract feature vectors for the image (1) extract image features based on low-level visual features of an image as color, area, position, texture, boundaries of image objects,... help increase accuracy and reduce training time on the H-Tree data structure (2).
In addition, in order for the image data set to be more diverse, the system has collected more image datasets from WWW; from this, the system will extract the URI identifier and image-related descriptions (3); after that, these images will be created with sample data of classes, properties, instances, semantic description, ... (4). The final step is to create and add these image sets to the ontology for enriching the ontology model (5).

![Image Retrieval Phase](image.png)

Figure 1. The model of SBIR-HT

In the image retrieval phase, the image query is also extracted features automatically to feature vector (6). Then, execute a retrieval algorithm with this feature vector (7) and retrieving directly on the H-Tree hierarchical clustering tree was built (8); The result of this retrieval process is similar images with image query (9). From this, the system will proceed to determine the visual word vectors for the query image using the k-NN method (10). After that, the system will automatically create a SPARQL query to query the ontology was built. The query on the ontology aims to expand the similar images and extract the corresponding semantics for each image. The results of the query are the resource identifiers, the meta-data of the image data set; from there, the system can extract the similar images and the semantic descriptions of the images.

- **Pre-processing**
  - Step 1: Extract the low-level visual features and classify the images in the image dataset by convolutional and sub-sampling operations, with each image being a vector;
  - Step 2: Store image feature vectors on the H-Tree hierarchical clustering tree base on Euclidean similarity;
  - Step 3: Select image datasets from WWW, extract the URI identifier and image-related descriptions (3); from this, these images will be created with sample data of classes, properties, instances, semantic description, ...; after that create and add image ontology.

- **Image retrieval**

  The image retrieval phase includes two main processes: retrieving on the H-Tree and query on the ontology were built.

  - Step 1: Image retrieval based on the H-Tree
    - For each input query image, the system performs image feature vector extraction;
    - Retrieve on the H-Tree to find the index set of similar images;
    - Retrieve similar images and visual word vectors.
  
  - Step 2: Semantic query based on ontology
    - Query semantic on ontology is based on categories of visual word vectors;
    - Apply k-NN on the set of similar images to define visual word vectors of the query image;
    - Create a SPARQL query from visual word vectors;
    - Query on the ontology has been built;
    - The result of the query is the identifier and meta-data of the image.

**B. The components in SBIR-HT**

1. **Extracting image features**

  In this section, we present the proposed method of extracting features from images based on convolutional and sub-sampling operations; then, normalize these feature vectors in the range [0, 1]. Each image is extracted as a feature vector that includes area features such as area, the circuit of the object, image color features, texture features and object...
position based on convolution operation; the original image is divided into 9 mini images and continues to extract the same features as the original image. Figure 2 shows the image feature extraction model and Figure 3 describes the extracted features of the image according to our proposed method.

![Figure 2. The model of extracting image’s features](image)

The image features are extracted by the following steps:

- Step 1: normalize the original image size.
- Step 2: determine the contrast based on the maximum and minimum brightness of each image and divide the input image into 2 regions of the background and object according to the contrast.
- Step 3: Extract the background image of the original image.
- Step 4: implement convolutional and sub-sampling operations to extract features of the image, including:
  - The color features follow the Newton color range.
  - The bright intensity features of the image are based on the dominant neighbor intensity.
  - The area features of the object and the background image.
  - The relative position features of the object image.
  - The relative position features of the background image.
  - The boundary features of the object image using the Laplacian filter.
  - The circuit features of the object and the background image.
  - The texture features the surface of the object by filtering the Sobel filter.

![Figure 3. The example of extracting features of image 35570.jpg of the ImageCLEF dataset.](image)
2. The hierarchical clustering tree

a. H-Tree structure

The H-Tree is a multi-branches tree consisting of a root node, a set of internal and leaf nodes. Each node in the H-Tree contains a set of image feature vectors. Creating H-Tree based on adding, deleting, and modifying nodes. The image retrieval process is performed from the root node and selects an appropriate branch on the tree, then sums the feature vectors from the root node to the leaf node. Based on the rendering the uniform resource identifier (URI) of the image, as well as the meta-data of that image, the set of similar images and the semantics respectively are extracted [9].

The H-Tree is organized according to a hierarchical clustering structure based on Euclidean distance to cluster the feature vectors set of images to serve the similar image retrieval problem. Therefore, H-Tree stores the set of image description elements \( T = \{ ED_1 = \langle v_i, l_p, id \rangle | i = 1, \ldots, N, j = 1, \ldots, M \} \), where \( N \) is the number of features of the image, \( M \) is the number of visual words in the image dataset. The result of tree creation is a set of nodes such that each node contains the element set \( \mu \) representing the image of the original image set \( \mathcal{S} = \{ \mu_k = \langle ED_i, cent_k, links_k > | k = 1..K, t = 1..T \} \), where \( K,T, cent_k, links_k \) are the number of nodes, number of images in the image set, center, and link at the k node in the H-Tree [9].

b. Creating H-Tree

In the beginning, the H-Tree was empty, it has only one root node containing the empty links. Each vector \( v_i \) then is added into the tree to create branches corresponding to the nodes in the tree. Based on the Euclidean measure, vector \( v_i \) is distributed into nodes in the tree. Let \( d(v_i, cent_k) \) be the Euclidean distance between the feature vector \( v_i \) and center \( cent_k \), this distance is normalized in the value domain \([0,1]\) because each component of the feature vector is also normalized in the value domain \([0,1]\). Let the two threshold distances among two feature vectors respectively are \( \varepsilon, \sigma \), with \( 0 < \varepsilon < \sigma < 1 \). The distribution rule is defined as follows:

i. Select the direction from the current node to the nodes of the neighboring branch and select the branch with the shortest distance \( d(v_i, v_{cj}) \).

ii. If \( d(v_i, v_{cj}) < \varepsilon \) then \( v_i \) is added to the current node, the node centered on \( v_{cj} \).

iii. If \( \varepsilon \leq d(v_i, v_{cj}) \leq \sigma \) then \( v_i \) is added to the child node of the node centered on \( v_{cj} \). Finding a suitable child node is executed again from the rule (i).

iv. If \( d(v_i, v_{cj}) > \sigma \) then create a new node at the same level as the node has center \( v_{cj} \).

Each time an element is added, this element is added to the current node, or the child node, or at the new node at the same level as the current node. Therefore, the H-Tree is a growing tree suitable for rapidly increasing image data storage.
neighboring branches. If it does not belong to an existing branch, a new branch is created, which makes the partition at the same nodes smoother, meaning that the correctness of the partition is higher. The algorithm for adding an element on the H-Tree is done as follows:

**Algorithm 1: Insert an element data into H-Tree**

**Input:** $E$, $D$ element, $Root$, threshold $\epsilon$, $\sigma$

**Output:** H-Tree

**Function INE($E, Root, \epsilon, \sigma$)**

1. **Begin**
   
   $NODE = Root$;

   **If** $NODE = null$ **then**
   
   Initialize root = \{links$_k$| links$_k$ = null; $k = 1..n_k$\};

   Create new node $=< E, cent, links >$, links = null;

   root. links$_0$ = node;

   **Else**
   
   $i = \arg\min (\text{euclidean}(NODE.\text{links}.\text{cent}, E. v), k = 1..|NODE.\text{links}|)$;

   $d = \text{Euclidean}(NODE.\text{links}_i.\text{cent}, E. v)$;

   **If** $d \leq \epsilon$ **then**
   
   $NODE.\text{links}_i.\text{e} = NODE.\text{links}_i.\text{e} \cup E$;

   **ElseIf** $d > \epsilon$ **and** $d \leq \sigma$ **then**
   
   $INE(E, NODE.\text{links}_i, \epsilon, \sigma)$;

   **ElseIf** $d > \sigma$ **then**
   
   Create new node $=< E, cent, links >$, links = null;

   $NODE.\text{links}_i[\text{NODE.\text{links}_i}+1] = \text{node}$;

   **EndIf**

2. Return H-Tree;

**End**

The complexity of the INE algorithm is $O(NxK)$, where $N$, $K$ are the number of nodes and the maximum number of elements of a node in the H-Tree.

c. **The meaning of H-Tree**

   In the H-Tree, a node can be both a leaf and a middle node, and a leaf node can become a middle node if a new branch is added. The H-Tree creates a multi-level partition, so when finding a cluster with a radius of $\epsilon$, the highest accuracy. If within radius $\sigma$, the tree will divide the feature vector into a branch, so the partitions will tend to be regular because the elements that are too different will not belong to one branch of the tree.

   In addition, the H-Tree hierarchical clustering tree aims to hierarchical clustering of data samples by clustering or dividing data samples according to the Euclidean similarity measure. Clusters in the H-Tree store the index of elements from a root node to a leaf node; this makes the image retrieval faster, more accurate, and H-Tree capable of storing large datasets.

3. **Creating ontology for a image dataset**

![Figure 5. An example about ontology apply to image dataset](image-url)
In order to reduce the "semantic gap" between low-level features and high-level semantics of the image, an ontology was built for image datasets based on the triple languages RDF/XML and OWL. The image querying process relies on clusters of data, resulting in a set of similar images sorted by precision. From this image set, the k-NN classification algorithm is implemented to take the closest neighbor classes and save it as a vector visual word containing the classes representing the query image. These classes are queried based on the ontology to identify the concept, high-level semantics of the image. Semantic mapping is used to analyze and find the best concept for objects in an image.

Based on the classification of the image datasets, a class hierarchy was built. A semantic dictionary that defines the subclasses of images extracted from WordNet. Each image is an individual/instance of one or more subclasses in the ontology. Figure 5 shows an example of the ontology built on Protégé for the ImageCLEF image datasets.

The output for the ontology based on Protégé is the data model in the RDF, OWL trilogy language described in Figure 6.

4. SPARQL query

SPARQL is a query language on data sources described in terms of the RDF and OWL triad. Based on the visual word vector, a SPARQL query is automatically generated to query on the ontology. The query result on the ontology is a set of URIs and metadata of similar images and their semantics. Figure 7 illustrates the SPARQL query created from a visual vector in two ways: "UNION Query" or "AND Query".

4. The operations on SBIR-HT system

The image retrieval and semantic analysis of images include the main operations: hierarchical clustering of visual word vectors of the image on a multi-branched tree H-Tree; image retrieval relies on clusters of data that have been allocated in a tree to retrieve a set of similar images; thereby determining classification for the query image to find the set of visual word vectors based on the k-NN method on the similar images being retrieved; high-level semantic query using SPARQL query on ontology was built to determine semantics for similar images and query images.

a. Extracting word vectors and similar images

From the H-Tree hierarchical clustering tree has been built, we propose a word vector extraction algorithm and similar images as the basis for extract semantic images. The process of extracting the vector set from the query image consists of 2 phases: (1) retrieve similar images with the query image; (2) extract the word vector set for the query image.

The similar image retrieval was done with input is a feature vector of the image; Then, the SBIR-HT retrieves system on the H-Tree to get the set of URIs, visual word vectors, and metadata data of similar images. The process of
performing similar image retrieval on the H-Tree tree will follow a branch going from the root node to the leaf node through the middle nodes. Figure 8 shows the path from the root node to leaf node along a branch of the H-Tree.

![Figure 8. The path from the root node to a leaf node on H-Tree](image)

The similar image retrieval algorithm on the H-Tree tree is implemented as follows:

**Algorithm 2: Image retrieval on H-Tree**

**Input:** feature vector $v$ of query image $I_q$, root node $root$, H-Tree, threshold $\varepsilon$, $\sigma$.

**Output:** Set of similar image $S_I$.

**Function:** IRHT($v$, $root$, $\varepsilon$, $\sigma$)

**Begin**

$NODE = root$;

**If** $NODE = null$ **then**

Return null;

**Else**

$i = \text{argmin}\{\text{euclidean}(NODE.links.cent}_k, E, v), k = 1..|NODE|\}$;

d = $\text{Euclidean}(NODE.links_i, cent, E, v)$;

**If** $d \leq \varepsilon$ **then**

$SI = NODE.links_i, E$;

**ElseIf** $d > \varepsilon$ **and** $d \leq \sigma$ **then**

IRHT($v$, NODE.links_i, $\varepsilon$, $\sigma$);

**EndIf**

End If

Return $SI$;

**End.**

The complexity of the IRHT algorithm is $O(hxK)$, where $h$, $K$ are respectively the height and a maximum number of elements of a node of H-Tree.

b. **Extracting image semantic**

To analyze the semantics of the image, the first, the system classifies an input image using the k-NN algorithm on a set of similar images to find the subclass of the query image and visual word vectors; These visual word vectors are then used to automatically generate a SPARQL query and query on the ontology to retrieve similar images and its semantics.

The image classification algorithm according to k-NN is performed as follows:

**Algorithm 3: Image classification**

**Input:** $I_q$, H-Tree

**Output:** The set of image’s categories

**Function:** $IC(I_q, HTree)$

**Begin**

$f_i = \text{ExtractVector}(I_q)$;

$NODE = root$;

Create new Stack;

nodeResult = null;

Stack.push($NODE$);
While Stack.count > 0 then
    Node = Stack.pop();
    If Node.links[i] = null then
        nodeResult = Node
    Else
        i = argmin \{eucled(Node.links.cent_k, f_i)\}, k = 1..|Node.links|
        d = Euclidean(Node.links_i , f_i)
        If d ≤ ε then
            nodeResult = Node.links_i
        Else
            Stack.push(Node.links_i)
    EndIf
EndIf

ListElementData = getListElementData(nodeResult);
ListClassRanking = null;
Foreach (var item in ListElementData)
    ListClassRanking.Add(item.Class)
End

TOP = 0;
If ListElementData.count > 100 then
    TOP = 3;
ElseIf ListElementData.count < 100 then
    TOP = 2;
ElseIf ListElementData.count < 10 then
    TOP = 1;
EndIf
ListClassRanking = Ranking(ListClassRanking).Take(TOP);
Return ListClassRanking;
End.

The complexity of the IC algorithm is \(O(nK)\), where \(n, K\) are respectively the number of clusters and a maximum number of elements of a node of H-Tree.

Thus, for each category of the image, the system performs a semantic analysis of the category concept based on the ontology and searches the definition for the image categories from the cognitive synonym of the ontology. Figure 9 is an example of a definition of a category which is executed of querying and classifying images in semantics based on ontology.

![Figure 9. An example about semantic of CASTLE category](image-url)
IV. EXPERIMENT

A. Experimental application

To evaluate the approach of the problem, based on the proposed algorithms, we build the SBIR-HT query system aimed at semantic-based image retrieval. The data set of 20,000 ImageCLEF images consisting of 276 layers and stored in 41 folders (from folder 0 to folder 40) was used for the experiment. There are many various types of images in the image dataset and describes various topics with annotations in English. Images are segmented into labeled and indexed areas with different features. The image data set has 99535 annotated regions with 276 subclasses. On average, each photo uses 5 words to annotate.

The experiment consisted of two phases: (1) The preprocessing stage performed training H-Tree tree to hierarchical clustering visual feature vectors of the image. (2) The image retrieval phase is done semantic-based image retrieval of the query image and similar images. SBIR-HT query system is built on the dotNET Framework 4.5 platform, C# programming language. Figure 10 shows the SBIR-HT query system for semantic-based image retrieval.

The preprocessing stage of SBIR-HT was tested on a server with Intel (R) Core (TM) i7-9700 CPU @ 3.00GHz (2 processors), Windows Server 2012 64-bit operating system, 32 GB RAM. The semantic-based image retrieval phase is executed on a computer with Intel (R) Core(TM) i7-8750H processor, 2.70GHz CPU, 8GB RAM, and Windows 10 Professional operating system.

B. Experimental results

In order to assess the effectiveness of proposed method, we used the following as evaluation metrics: precision, recall, F-measure. The formulas of these values are as follows:

\[
\text{precision} = \frac{|\text{relevant images} \cap \text{retrieved images}|}{|\text{retrieved images}|}
\]

\[
\text{recall} = \frac{|\text{relevant images} \cap \text{retrieved images}|}{|\text{relevant images}|}
\]

\[
F\text{-measure} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]

We obtained experimental results for image retrieval performance of the proposed method on the ImageCLEF dataset in Table 2.

<table>
<thead>
<tr>
<th>ID</th>
<th>Number of Images</th>
<th>Avg. precision</th>
<th>Avg. recall</th>
<th>Avg. F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>00-10</td>
<td>2250</td>
<td>0.696950</td>
<td>0.672100</td>
<td>0.661573</td>
</tr>
<tr>
<td>11-20</td>
<td>1700</td>
<td>0.711957</td>
<td>0.671962</td>
<td>0.670907</td>
</tr>
<tr>
<td>21-30</td>
<td>1260</td>
<td>0.730768</td>
<td>0.663194</td>
<td>0.669101</td>
</tr>
<tr>
<td>31-40</td>
<td>1780</td>
<td>0.689987</td>
<td>0.669142</td>
<td>0.659092</td>
</tr>
<tr>
<td>Average</td>
<td>6990</td>
<td>0.707416</td>
<td>0.676600</td>
<td>0.665168</td>
</tr>
</tbody>
</table>

Figure 11 shows the Precision-Recall and ROC curves, each representing a retrieved data set; the area under these curves indicates the accuracy of image retrieval. Figure 12 depicts the average value of Precision, Recall, and F-measure of 41 sets of images on the ImageCLEF image set.
Figure 11. The graph of Precision-Recall and ROC of SBIR-HT on the ImageCLEF dataset

Figure 12. The mean averages of precision, recall and F-measure on the ImageCLEF dataset

The average value of the MAP accuracy of the SBIR-HT system is compared with other methods on the same imageCLEF data set described in Table 3 and shows that the retrieval results of the proposed method are relatively accurate compared to semantic-based image retrieval systems.

Table 3. Comparison of mean average precision (MAP) of methods on ImageCLEF dataset

<table>
<thead>
<tr>
<th>Methods</th>
<th>Mean Average Precision (MAP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.A. Hernández-Gracidas, 2013 [10]</td>
<td>0.5826</td>
</tr>
<tr>
<td>Hakan Cevikalp, 2017 [2]</td>
<td>0.4678</td>
</tr>
<tr>
<td>Vijayarajan, 2016 [11]</td>
<td>0.4618</td>
</tr>
<tr>
<td>S. Fakhfakh, 2015 [12]</td>
<td>0.5400</td>
</tr>
<tr>
<td>SBIR-HT</td>
<td>0.7074</td>
</tr>
</tbody>
</table>

From the comparison in Table 3, it shows that the semantic-based image retrieval results of the SBIR-HT system have quite a good accuracy compared to recent studies in the same field and data set. This result proves that our proposed method is effective.

V. CONCLUSION AND FURTHER WORK

The paper has built a semantic-based image retrieval method based on the hierarchical clustering tree for visual features of images that are extracted from convolutional and sub-sampling operations. The H-Tree is a partition model of feature vectors to find similar images. Here’s an improved model of the hierarchical clustering method incorporating the semi-supervised learning technique. Based on the proposed theory, a semantic-based image retrieval SBIR-HT was constructed. The query process on the SBIR-HT system consists of three main stages: (1) Extracting the visual features of the query image and semantic classification from the image content; (2) Create a visual word vector from the frequency of occurrence of classification of image and similar images; (3) Execute SPARQL query from visual word vectors to query image from ontology describing the semantics of image data set. Experimental results on ImageCLEF image datasets evaluated and compared with other projects on the same image data set showed that the proposed method is effective. So our proposed method can be used for multimedia data lookup systems. Building a hierarchical cluster tree based on visual features of the image creates a quick image search structure. Our next development is to build a graph on a tree to increase the accuracy of the image search process.
REFERENCES


MỘT SỐ CẢI TIẾN CHO HỆ TRUY VẤN ẢNH THEO TIỆP CẤN NGỮ NGHĨA DỤA TRÊN CÂY PHÂN CỤM PHÂN CẤP

Nguyễn Minh Hải, Văn Thế Thành, Trần Văn Lăng

TÓM TẮT: Việc tìm kiếm hình ảnh tương tự và trích xuất ngữ nghĩa là một bài toán quan trọng trong các hệ thống đa phương tiện. Bài báo trình bày một hệ truy vấn ảnh theo ngữ nghĩa dựa trên cây H-Tree, một cấu trúc cây phân cụm phân cấp. Cấu trúc để liệu cây H-Tree được tạo ra dựa trên phương pháp phân cụm phân cấp các vector đặc trưng của hình ảnh. Trong cấu trúc H-Tree này, phương pháp phân hoạch phân tử và kết nối lang giang giữa các cụm được để xuất để từ đó xây dựng một cấu trúc cây đa nhân, lưu trữ được dữ liệu lớn và phân cấp trên từng nút của cây. Các đặc trưng của hình ảnh được trích xuất dựa trên các phép toán theo một tích chập nhằm mô tả các đặc trưng gián đoạn của ảnh, Mô hình truy vấn ảnh được để sử dụng kỹ thuật tìm phân tử.lang giang gần nhất kNN (k-Nearest neighbor) và tìm kiếm trên đó chỉ nhằm tăng hiệu suất cho việc tìm kiếm các cụm dữ liệu trong cây H-Tree có độ tương tự gần với ảnh truy vấn nhất. Trên cơ sở những đề xuất, hệ truy vấn này được thử nghiệm trên bộ dữ liệu ảnh ImageCLEF (20,000 ảnh, 276 phân lớp). Kết quả truy vấn của bài báo được so sánh với một số kết quả của các công trình đã được công bố gần đây trên cùng bộ dữ liệu. Từ đó cho thấy phương pháp tìm kiếm ảnh đã để xuất có tính khả thi và đạt được hiệu quả cao.