A New Proposal for Graph-Based Image Classification using Frequent Approximate Subgraphs

Annette Morales-González[a,*], Niusvel Acosta-Mendoza[a,*], Andrés Gago-Alonso[a],
Edel B. García-Reyes[a], José E. Medina-Pagola[a]

[a] *Advanced Technologies Application Center, 7a ♯ 21812, Siboney, Playa, CP: 12200, Havana, Cuba*
E-mail: {amorales, nacosta, agago, egarcia, jmedina}@cenatav.co.cu

Abstract

Graph-based data representations are an important research topic due to the suitability of this kind of data structure to model entities and the complex relations among them. In computer vision, graphs have been used to model images in order to add some high level information (relations) to the low-level representation of individual parts. How to deal with these representations for specific tasks is not easy due to the complexity of the data structure itself. In this paper we propose to use a graph mining technique for image classification, introducing approximate patterns discovery in the mining process in order to allow certain distortions in the data being modeled. We are proposing to combine a powerful graph-based image representation adapted to this specific task and Frequent Approximate Subgraph (FAS) mining algorithms in order to classify images. In the case of image representation we are proposing to use more robust descriptors than our previous approach in this topic, and we also suggest a criterion to select the isomorphism threshold for the graph mining step. This proposal is tested in two well-known collections to show the improvement with respect to previous related works.

Keywords: Approximate graph mining, frequent approximate subgraphs, graph-based image representation, image classification.

1. Introduction

In many research fields, graphs have been largely used to model data due to their representation expressiveness and their suitability for applications where some kind of entities and their relationships must be encoded within some data structure. Also, a vast graph theory has been developed in order to work with graphs and process the information they represent. In this paper, we intend to explore and combine two research fields where graphs are involved, in order to exploit both their advantages.

The first field is related to Computer Vision. Our intention is to classify images using a graph-based representation. The first step for image classification is to extract low-level

[*] Corresponding author

features that will encode relevant information for the task, but it has been shown that low-level information by itself cannot provide the high-level perception cues that exist in human minds to describe objects or images in general (this is the well-known semantic gap problem) [16]. Within the range of low-level features developed so far, graphs are one of the representations that can provide some kind of high-level information implicitly, making them a desirable representation choice for researchers to find new solutions. Many works have represented images as graphs (see Section 2) with this purpose, and have developed methods for classification using this type of data structure. One major concern in this area is that although graphs are powerful representational tools, they are hard to work with, leading usually to algorithms with high computational costs, or simplifying the data structure, thus losing some of the embedded information. Having a collection of images represented as graphs rises the question of whether graph mining techniques can be used to discover beneficial information and to perform certain tasks such as image classification.

Now we will approach the other research field that we are aiming to explore: Data Mining. Several authors have developed graph-based techniques and methods for satisfying the need to convert large volumes of data into useful information [24, 46]. Frequent subgraph discovery is an example of such techniques [15]. An important problem in graph mining tasks is classifying information, such as image [2, 13, 25, 35, 37], text [26], and molecular [11, 23] datasets. Although using graph mining for classification purposes has been widely explored, these approaches may not always produce the optimal results in all applications and several authors [22, 27, 7] have expressed the necessity to use approximate graph matching for frequent subgraph mining. These authors defend the idea that, by using approximation, more interesting subgraphs can be found for many applications, for instance, when processing graph databases that have distortions (in terms of different geometric, topological or semantic variations) of similar structures in several objects [2, 23]. Distortion in data is one of the challenges for developing classifiers based on frequent subgraphs in several domains of science [2, 9, 19, 23, 29, 41, 43, 44, 47, 48]. Frequent Approximate Subgraph (FAS) mining is an important problem in graph mining, where the mined patterns are detected taking into account such semantic distortions. Thus, such approximate solutions achieve classification results which are different from the other graph mining methods.

As mentioned before, in this paper we aim at combining and exploiting both research fields (Image Classification and Frequent Approximate Subgraph Mining) by proposing an image representation that can be used in a classification framework. Although graph mining techniques for image classification have been explored before (see Subsection 2.3), our main contribution resides precisely in the approximation part of the subject. This work is, in fact, an extension of a previous work [3] where we make use of a powerful graph-based representation adapted to the conditions of the problem. The new contributions and changes with respect to that work are the following:

- We use a different visual description of the regions in order to add context information to it. We employ a visual descriptor already reported in the literature, making some changes to take advantage of the structure of irregular pyramids.
- We propose a criterion to obtain the isomorphism threshold needed in the FAS mining

step, which is the parameter employed to determine whether two graphs are similar enough to be considered in the frecuency count.

We performed new experiments to show how these improvements largely and positively influence the classification results, while comparing it also with other state-of-the-art methods in image classification.

The remainder of the paper is distributed as follows. Section 2 is a summary of related works in the fields of graph-based image classification, graph mining and the combination of both. Section 3 presents some basic notions regarding graph mining techniques and some specific details of the FAS mining process. The graph-based image representation used in our proposal is described in Section 4. Section 5 depicts the classification framework, where both topics discussed in Section 3 and 4 are combined. In order to validate our approach, we present experimental results in Section 6 and conclusions are given finally in Section 7.

2. Related work

In this section, we start by providing a brief on classification methods using graph-based representations. Next, we present a review of previous works related to approximate graph mining and finally, we present a brief on classification using frequent patterns, which is the subject of this paper.

2.1. Brief overview of image classification using graph-based representations

Representing images as graphs has become popular because they are powerful tools to encode different types of information, and may provide a robust and rich representation for many applications. How to exploit this information is the main issue in the graph-based classification scope. A popular graph-based image representation is the Region Adjacency Graph (RAG) [6] where each vertex represents a region in the image and an edge exists between two vertices if the underlying regions are adjacent.

Different methods have been developed to use graphs for classification tasks, for instance, graph matching algorithms [21, 12, 32, 18, 17], which use distances (ex. graph edit distance), greedy matching techniques or matching kernels in order to compare graphs. Another way to perform classification is by using graph embedding methods [8, 20], which, in general terms, map graphs to a vector space and then perform regular classification operations with the resulting vectors.

Although graphs are considered to be a powerful representation, they have a major drawback, which is the lack of suitable classification methods where they can be used. Graph matching techniques usually have the problem of the computational complexity involved in the process and graph embedding methods usually tend to suppress part of the information encoded by the graphs. Graph mining techniques used for classification purposes could be considered a halfway between graph matching and graph embedding methods, thus using the strength of both and reducing their individual disadvantages.

2.2. Summary of approximate graph mining

Traditional exact graph mining has become an important topic of research in recent years [7, 15, 34, 45]. However, there are concrete problems where these solutions could not be applicable with positive outcomes [19]. Sometimes, the interesting subgraphs show slight differences throughout the data. An example of these differences can be seen on image processing, where these differences may be due to noise and distortion, or may just illustrate slight spatial differences between instances of the same objects. This means that we should tolerate slight semantic variations or some mismatches of vertices (and edges) in frequent subgraph pattern search.

In the last years, some approximate graph mining algorithms have been published where several similarity functions are used. For example, the algorithms SUBDUE [19] and RNGV [43] are based on graph edit distance; Monkey [47] is based on β-edge sub--isomorphism; UGRAP [41] and MUSE [48] are based on sub-isomorphism on uncertain graph collections; GREW [29] is based on sub-isomorphism employing ideas of edge contraction and graph rewriting; CSMiner [44] uses node/edge disjoint sub-homeomorfismo; gApprox [9], APGM [23] and VEAM [2] are based on substitution probabilities.

gApprox, *APGM* and *VEAM* algorithms defend the idea that a vertex label or an edge label can not always be replaced by any other. Therefore, these algorithms specify which vertices, edges or labels can replace others using substitution matrices to perform frequent subgraph mining. However, only APGM and VEAM perform frequent approximate subgraph mining on graph collections and we are interested in this kind of mining. APGM only deals with the variations between the vertex labels, while VEAM performs the mining process using both the vertex and edge label sets.

In this paper, the last two algorithms are applied in order to create an image representation that will be used for classification purposes. This is due to the need of an algorithm that allows some variations in the data using substitution probabilities and keeping at the same time the topology of the graphs.

2.3. Brief review of classification using frequent patterns

As mentioned before, frequent subgraph patterns has been successfully used for classification tasks in different domains of science [2, 11, 13, 23, 26, 35, 37]. However, among these solutions, there are only two that use frequent approximate subgraphs in classification tasks [2, 23]. These methods perform feature (subgraph) selection taking into account semantic distortions. APGM algorithm [23] allows these distortions only between vertices of graphs, while VEAM algorithm [2] allows distortions between vertices and between edges. Feature vectors are built from the identified FAS , which are later used for classification. However, only the scheme presented by Acosta-Mendoza *et al.* [1, 2, 3] reflects the degree of semantic distortions (in case they exist) and not only the occurrence or not of each subgraph in the graphs.

Our present work is an extension of [3], and as new enhancements with respect to the later, we implement (and adjust to our specific conditions) a new low-level representation for images and we introduce a new criterion to select the isomorphism threshold, which is basically the keystone in the FAS mining process.

3. Background on FAS mining

In this section we intend to provide the reader with the basic tools for understanding the principles of graph mining techniques and, specifically in this context, how approximation methods work in order to take into account possible data distortions.

3.1. Basic concepts

In this work we use simple undirected labeled graphs as basis for subgraph mining. From now on, when we refer to graph we assume this type of graph. Before presenting their formal definition, we will define the domain of labels.

Let L_V and L_E be label sets, where L_V is a set of vertex labels and L_E is a set of edge labels. The domain of all possible labels is denoted by $L = L_V \cup L_E$.

A *labeled graph* in L is a 4-tuple, $G = (V, E, I, J)$, where V is a set whose elements are called *vertices*, $E \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}$ is a set whose elements are called *edges* (the edge $\{u, v\}$ connecting the vertex u with the vertex v), $I : V \to L_V$ is a *labeling function* for assigning labels to vertices and $J : E \to L_E$ is a *labeling function* for assigning labels to edges.

Let $G_1 = (V_1, E_1, I_1, J_1)$ and $G_2 = (V_2, E_2, I_2, J_2)$ be two graphs, we say that G_1 is a *subgraph* of G_2 if $V_1 \subseteq V_2$, $E_1 \subseteq E_2$, $\forall u \in V_1, I_1(u) = I_2(u)$, and $\forall e \in E_1, J_1(e) = J_2(e)$. In this case, we use the notation $G_1 \subseteq G_2$.

Given G_1 and G_2, we say that f is an *isomorphism* between these graphs if $f : V_1 \to V_2$ is a bijective function, where $\forall u \in V_1, f(u) \in V_2 \wedge I_1(u) = I_2(f(u))$ and $\forall \{u, v\} \in E_1, \{f(u), f(v)\} \in E_2 \wedge J_1(\{u, v\}) = J_2(\{f(u), f(v)\})$. When there is an isomorphism between G_1 and G_2, we say that G_1 and G_2 are *isomorphic*.

Let Ω be the set of all possible labeled graphs in L, the *similarity* between two elements $G_1, G_2 \in \Omega$ is defined as a function $sim : \Omega \times \Omega \to [0, 1]$. We say that the elements are very different if $sim(G_1, G_2) = 0$, the higher the value of $sim(G_1, G_2)$ the more similar the elements are and if $sim(G_1, G_2) = 1$ then there is an isomorphism between these elements.

Let $D = \{G_1, \ldots, G_{|D|}\}$ be a graph collection and let G be a labeled graph in L, the *support* value of G in D is obtained through the following equation:

$$supp(G, D) = \sum_{G_i \in D} sim(G, G_i)/|D| \qquad (1)$$

If $supp(G, D) \geq \delta$, then the graph G is approximately frequent in the collection D, saying that G is a *FAS* in D. Notice that when we refer to a graph collection we assume that it is the representation built from a real graph collection. The value of the support threshold δ is in $[0, 1]$ assuming that the similarity is normalized to 1. *FAS mining* consists in finding all the FASs in a collection of graphs D, using a similarity function sim and a support threshold δ.

There are several similarity functions used by different algorithms in the graph matching process [10]. In Section 2.2, we presented the most relevant algorithms which use approximate graph matching techniques in frequent subgraph mining. All of these algorithms use the definitions presented above, implementing a specific similarity function.

3.2. Approximate FAS methods considered

In APGM [23] and VEAM [2] algorithms, the idea that not always a vertex label or an edge label can be replaced by any other is upheld. Therefore, these algorithms specify which vertices, edges or labels can replace others using substitution matrices to perform the frequent subgraph mining task. APGM only deals with the variations among vertex labels, while VEAM performs the mining process using the vertex and edge label sets. These methods use substitution matrices where each cell represents the suitability of approximating one vertex by another vertex (or an edge by another edge) and they offer frameworks for each frequent subgraph mining task.

A *substitution matrix* $M = (m_{i,j})$ is an $|L| \times |L|$ matrix indexed by a label set L. An entry $m_{i,j}$ $(0 \leq m_{i,j} \leq 1, \sum_j m_{i,j} = 1)$ in M is the probability that the label i is replaced by the label j. When M is diagonally dominant (i.e. $M_{i,i} > M_{i,j}$, $\forall j \neq i$) then M is known as *stable matrix.*

Let $G_1 = (V_1, E_1, I_1, J_1)$ and $G_2 = (V_2, E_2, I_2, J_2)$ be two labeled graphs in L, MV be a substitution matrix indexed by L_V, ME be a substitution matrix indexed by L_E, and τ be the isomorphism threshold. We say that G_1 is *approximate isomorphic* to G_2, denoted by $G_1 =_A G_2$, if there exists a bijection $f : V_1 \to V_2$ such that:

- $\forall \{u, v\} \in E_1, \{f(u), f(v)\} \in E_2$,
- $S_f(G_1, G_2) = \prod_{u \in V_1} \frac{MV_{I_1(u), I_2(f(u))}}{MV_{I_1(u), I_1(u)}} * \prod_{e=\{u,v\} \in E_1} \frac{ME_{J_1(e), J_2(\{f(u), f(v)\})}}{ME_{J_1(e), J_1(e)}} \geq \tau.$

The bijection f is an approximate isomorphism between G_1 and G_2, and $S_f(G_1, G_2)$ is the product of normalized probabilities called *approximate isomorphism score* of f. When G_1 is approximately isomorphic to a subgraph of G_2, we say that G_1 is *approximately sub-isomorphic* to G_2. Notice that this is a generalization of the APGM approach [23].

The *approximate matching score* between two graphs, denoted by $S_{max}(G_1, G_2)$, is the largest approximate isomorphism score.

$$S_{max}(G_1, G_2) = max_f\{S_f(G_1, G_2)\} \tag{2}$$

Given a graph collection D and an isomorphism threshold τ, the *approximate support* of a graph G, denoted by *supp*(G, D), is the average score of the graph in the collection, where G is approximately isomorphic to a subgraph of graphs in the collection:

$$supp(G, D) = \sum_{G_i \in D} S_{max}(G, G_i)/|D| \tag{3}$$

If *supp*$(G, D) \geq \delta$, then graph G is approximately frequent in the collection D, saying that G is a *frequent approximate subgraph* in D, with δ as support threshold. Notice that the values of the products of normalized probabilities $S_f(G_1, G_2)$ is in the interval $(0, 1]$. The value of the support threshold δ is in $[0, 1]$ assuming that $S_{max}(G, G_i)$ is normalized. The *frequent subgraph mining* task used in this paper consists in finding all the connected frequent subgraphs in a collection of graphs D, using (3), δ as support threshold, and τ as isomorphism threshold.

4. Graph-based image representation

In the present work we choose to explore the approach proposed in [31, 32] to obtain a graph-based image representation that can serve as input to a graph mining algorithm. In this case we construct an irregular pyramid for each image [5], which provides a hierarchy of partitions at different levels of resolution. Each level is a RAG and the whole pyramid is built from bottom to top, being the base level (level 0) the entire image (i.e. each vertex of the base level represents one pixel in the image, and the edges are the 4-connections of each pixel). Each level l is built from its previous level $l-1$ by contracting several edges in the process. In a given level, a set of surviving vertices is chosen to be part of the new level, and the set of vertices to be merged into a surviving one is called its Contraction Kernel (CK). In the new level l, each surviving vertex will represent all the vertices from level $l-1$ in its contraction kernel, and will keep a connection to them. Further information regarding the construction of the pyramid can be obtained in [5, 28].

Since the pyramid provides several graphs at different levels of resolution for a single image, we use the B measure proposed in [32] to evaluate the segmentation "quality" of each level. This evaluation will serve as a guide to choose one graph per image. The measure evaluates each level of the pyramid against a border map of the image, in terms of how much each partition preserves the borders present in the map. The best level evaluated by B is selected to represent the image.

4.1. Low-level features to describe image regions

In our previous work [3], we employed a very basic low-level representation for the image regions, which consisted in a 48-dimensional color histogram (quantized from the RGB color space) and a 256-dimensional Local Binary Pattern (LBP) histogram as texture descriptor. Nevertheless, in order to gain more robustness in our visual representation, we decided to use a low-level description that allows us to include contextual information in the description.

As stated in [39], context plays a crucial role in many cases, especially when using irregular segmentations of an image. Usually, segmentation methods tend to create mostly homogeneous regions that provide poor information when low-level features are extracted from them. Therefore, including some area outside of each region may provide the necessary clues for the region representation to be somewhat more discriminative. To illustrate this, please refer to Fig 1. We can see the original image first, and in the second image we can see an example of an irregular region (with green border) that may result from applying a segmentation method. As we can notice here, this region is homogeneous, white, and it does not have a discriminative texture. Therefore, trying to use this single region for classification may introduce several errors, since this type of region may be part of a very large number of different objects. Including more area to the region representation (as the one depicted in blue in the third image), may introduce some clues about the context of this single white blob, and may supply more useful information to it.

Based on this, we decided to use the Region-based Context Features (RCF) proposed in [39], which combines irregular regions and regular patches to obtain a more accurate representation. We chose the representation used in [40], which involves a 100-dimensional

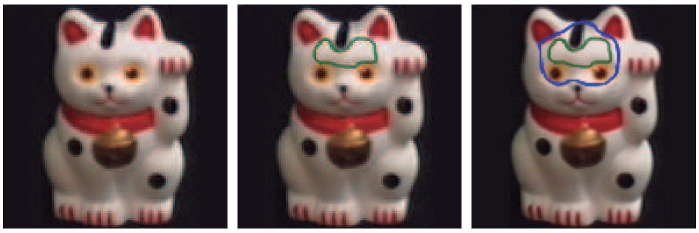

Figure 1: Example of low-level representation problems when using irregular segmentation methods. First we can see the original image, next, an arbitrary homogeneous region that we may obtain by means of a segmentation method, and last, an example of using information outside the region to add some context information.

histogram of quantized hue features for describing color within the regions and a 300-dimensional RCF histogram. In order to construct the RCF features, we computed dense SIFT features through the entire image for two different scales and we created a vocabulary by clustering this set of features. Each dense SIFT patch is then represented by a visual word in the vocabulary. The first 150 bins of the resulting histogram correspond to the occurrence of each visual word within the region being analyzed, only for the first scale. In the original approach, the next 150 bins correspond to the occurrence of each visual word in the second scale, that appears close enough (by setting a distance) to the region in question, adding context information in this way.

For this last step we introduced a modification. Instead of using a distance to count the features of the second scale, we counted the occurrences of the features that appear within the father region in the pyramid. This means that if we have a region r_l in level l of the pyramid, the first 150 bins of the RCF histogram will count the occurrences of the visual words (corresponding to the first scale) that are inside r_l, and the second 150 bins will count the occurrences of the visual words (corresponding to the second scale) that are inside r_{l+1} in level $l+1$, where r_l belongs to the Contraction Kernel of r_{l+1}. In this way, we take advantage of the spatial relations of the pyramid (by means of the graphs at each level) and the hierarchical relations, introducing context to the representation.

After this step, we have a 400-dimensional feature vector (by concatenating color and RCF histograms) for each region. We also add another 400 dimensions to this descriptor, by including a 100-dimensional color histogram and a 300-dimensional RCF histogram of the entire image, in order to add overall image context, yielding a final 800-dimensional feature vector for each region.

4.2. Spatial relations to describe edges

In a RAG, the edges implicitly encode adjacency relations between image regions, but we can add an explicit and more detailed information about the relations shared by two regions, in order to exploit it in the FAS mining algorithm. For the edge representation, we chose to use the spatial descriptor proposed in [31, 32], which is a binary vector encoding several topological and orientation relationships between pairs of regions.

4.3. Automatically building substitution matrices

FAS mining algorithms work with substitution matrices, which intuitively represent the probability of substituting one label with another, using some meaningful criteria for the matter (see the formal definition in Subsection 3.2). For this case, we have a graph-based image representation and we are interested in knowing which vertices can be substituted by others in terms of visual similarity of the underlying regions. We also want to know which edges can be considered equivalent (in the approximation framework), in terms of the similarity of the spatial relations that they encode.

To create the substitution matrix for the vertices, it is necessary to reduce the set of vertices labels. According to the pyramid representation explained before, there will be as much labels as possible different feature vectors. In order to reduce the set of vertices labels, we use a clustering algorithm to group similar features. The centroid of each cluster will be the new label of all the vertices with features belonging to this cluster. Then, the substitution matrix will be a $n \times n$ matrix, where n is the number of labels (clusters). Each element of this matrix will store the similarity between two labels, given by the similarity between the centroids of the clusters they belong to. In this case, we decided to use the Euclidean distance between the feature vectors for each node. This means that an element of this matrix can be interpreted as the confidence of substituting a node with label x with a node with label y in a matching scheme.

The substitution matrix for edges is easier to construct, since using the spatial descriptor representation we can only have 27 possible configurations of spatial relations. The similarity between edges, i.e. the value that will be stored in the elements of the matrix, can be obtained by the Sokal-Michener measure proposed for this purpose in [31] for computing the similarity between spatial descriptors.

5. Classification Framework

After presenting the basic concepts and details regarding the graph-based image representation and the FAS mining methods used, we can finally describe the overall classification framework where we combine these tools.

First of all, we obtain the graph-based image representation of a given set of pre-labeled real images, which gives us a graph collection to work with. Next, we label all the vertices and edges, and create the corresponding substitution matrices as presented in Subsection 4.3. Once we have these substitution matrices we proceed to select the appropriate parameters for the graph mining algorithm (see Subsection 5.1 for further details in this step). Next, we apply the FAS miners to the entire collection in order to obtain all the FASs present in it, according to some given support threshold δ. Similar to the vocabulary creation step in the Bag of Words (BoW) model [38], the FASs obtained by the miners can be regarded as the words in a vocabulary, thus giving us some sort of bag of subgraphs approach. After this, each image will be represented by a histogram where each bin corresponds to each FAS of the vocabulary. Therefore, the dimension of the new feature will be the number of FASs found in the collection. For each FAS in the image, the similarity of the subgraph found with respect to the one represented in the vocabulary is stored in the corresponding bin,

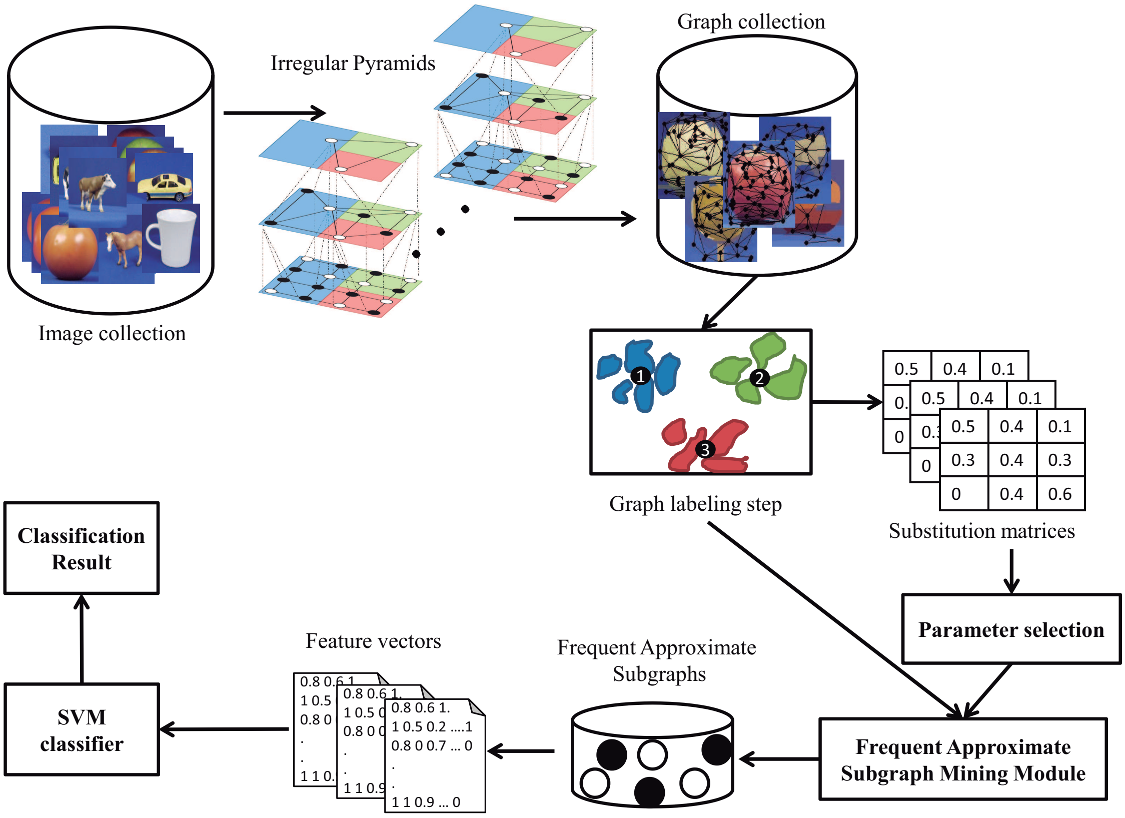

Figure 2: Framework of graph-based image classification.

and the final vector for the image is created after the pooling step, where the maximum similarity of each subgraph found is taken into account.

When all the new features are built, a classifier generator (SVM using 10 cross-validation) is used having such vectors as data to produce an image classifier. The complete flowchart of our classification framework is shown in Fig. 2.

5.1. Parameter selection

In our framework, two parameters play a decisive role in the classification performance. Both are used in the FAS mining step and they are presented in Section 3. They are the support threshold δ and the isomorphism threshold τ.

For the case of the support threshold, it determines the frequency of the graphs considered to be frequent by the miner. If we choose a high value for δ (for instance 80%), we might obtain graphs that are very frequent, therefore, less discriminative, since they appear in most objects in the collection. On the contrary, if we choose a small value (for instance, less than 10%), we are considering too many graphs to be frequent, and more noise can be introduced in this step. In order to illustrate the behavior of this parameter, we ran our experiments using different support thresholds, starting from 80% to 20%, with a 10% step.

The isomorphism threshold is a very delicate parameter, because it will determine what is considered to be similar or not. Therefore, a bad selection of this parameter may lead to bad classification results, very long processing times and very high memory requirements, if it finds too many graphs as approximate patterns. This threshold is highly dependent on the collection and the representation used for low-level features, but fortunately, we can count on the substitution matrices to make an analysis of how the similarities behave according to the labels found for each collection. The first step is to normalize each row of the substitution matrices (MV and ME) with respect to their main diagonal and then we find the mean and standard deviation per row (without the value of the main diagonal). We analyze the maximum value of these means and standard deviations (denoted by $\mu(M)$ and $sd(M)$ respectively), and, if the standard deviation is not too high (i.e. lower than 0.3), this means that the similarity values are well distributed and we can take the mean value as our isomorphism threshold. In case the standard deviation is high, we suggest the criteria of an expert in the current dataset for selecting this parameter. This process is done for both MV and ME and we select the minimum mean value from these two (see Equation 4).

$$\tau = \begin{cases} min(\mu(MV), \mu(ME)) & \text{if } sd(MV) \leq 0.3 \geq sd(ME) \\ expert\ decision & \text{otherwise} \end{cases} \tag{4}$$

In Section 6 we perform an experimental validation of the entire framework, which shows the advantages of the new image representation employed and the effects of selecting the isomorphism threshold in this way.

6. Experimental Results

Development in the field of approximate graph mining is still incipient when it comes to applying it to problems with real images. Tests to validate this approach have been performed so far in simple collections [25, 2]. Actually, in the graph mining community, the standards to perform tests are synthetic datasets or molecular datasets [42]. When we try to use these techniques in real images, they have to deal with bigger graphs, with sizes ranging from 200 to 300 edges per graph, and the current state-of-the-art methods for graph mining are not well prepared to cope with such big graph sizes, in large collections, in terms of memory requirements and computational time. Even more, if this is considered a computational problem for graph mining techniques, the use of approximation among graphs increases the complexity.

In order to validate our proposal, we chose two well known databases containing color images of simple objects taken from different viewpoints. The first one is the COIL-100 [33] dataset, which possess 100 objects with 72 poses per object. We took 25 objects randomly selected from this dataset to test our classification framework. The second dataset is ETH-80 Image Set [30], which contains 80 objects from 8 categories and each object is represented by 41 different views, yielding a total of 3280 images. This database is more challenging than the COIL-100 database in the sense of the viewpoint diversity. For the experiment in this database we took the same 6 categories employed by [31] and [3]: *apples*, *cars*, *cows*, *cups*, *horses* and *tomatoes*, for the sake of comparison. Example images from both datasets can be seen in Figure 3. We represented all images by a single graph, which corresponds to the "best" segmented level of each pyramid (see Section 4).

Figure 3: Example images from the COIL-100 Image Set database (first 6 images), and from the ETH-80 Image Set database (last 6 images).

In Table 1 and 2, we can see the results of the experiments using the proposed framework. These results aim at showing a comparison among three different graph miners, i.e, gdFill [15] (representing the exact methods), APGM and VEAM (both representing FAS methods). The different support thresholds (δ) used in our experiments are shown in columns and the algorithms used are presented in rows. Each cell shows the accuracy achieved by each method and the highest accuracy per column is represented in bold style. In Table 2, the results for δ lower than 40% are not shown, since for these cases they were not relevant enough.

Table 1: Accuracies achieved by gdFil, APGM and VEAM algorithms in COIL database using 25 random classes.

				Support (δ)			
Algorithm	**80%**	**70%**	**60%**	**50%**	**40%**	**30%**	**20%**
gdFil	-	-	-	-	-	21.61%	32.61%
APGM	**72.78%**	**84.06%**	**86.78%**	89.72%	94.06%	98.22%	99.11%
VEAM	**72.78%**	**84.06%**	**86.78%**	**90.39%**	**94.89%**	**98.33%**	**99.44%**

Table 2: Accuracies achieved by gdFil, APGM and VEAM algorithms in ETH-80 database using 6 classes.

			Support (δ)		
Algorithm	**80%**	**70%**	**60%**	**50%**	**40%**
gdFil	-	-	-	25.70	47.59%
APGM	77.28%	**81.02%**	**82.11%**	**84.39%**	**76.22%**
VEAM	**76.06%**	80.16%	78.46%	82.68%	75%

For the APGM and VEAM algorithms, the isomorphism threshold used was automatically set to $\tau = 0.66$ according to the criteria described in Subsection 5.1, specifically using (4), in both databases (COIL-100 and ETH-80). As we can see, the approximate graph miners achieve a higher accuracy in most cases than the exact ones. These results show the relevance of treating distortions in real data. In this direction, we can see with VEAM in COIL-100, that the values of the accuracies obtained are better in most cases than using APGM. This indicates that taking into account edge distortions in FAS mining can benefit the classification task in this dataset. On the other hand, the aforementioned distortions affected negatively the features identified by VEAM in the ETH-80 dataset.

In addition, we present in Figures 4 and 5, a comparison between the results of this work and the previous work [3] (which is extended in this paper) to show the improvement achieved using our current classification framework. These results are from the COIL-100 dataset.

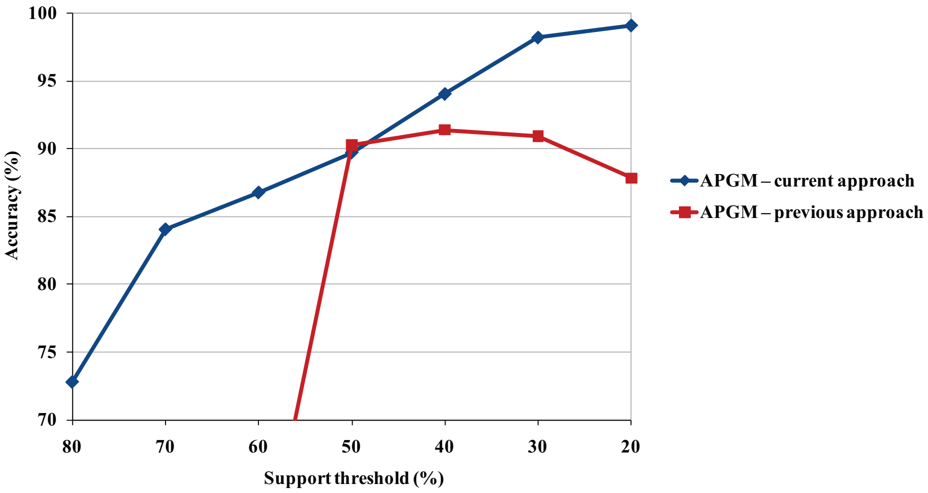

Figure 4: Comparison between the results obtained by APGM in both classification approaches, using the COIL-100 dataset. ”Previous approach” refers to the work presented in [3]

In figure 4, the results obtained by using the APGM algorithm in both works are shown. As we can see, the accuracies achieved by APGM using our current approach are much better than using the previous classification approach. This fact proves that the automatic parameter selection combined to the proposed image representation provides important information that is translated in the identification of better features in the mining process.

In figure 5, the results achieved using VEAM in both works over the same dataset are presented. As we can see, the performance of the classification using both VEAM and APGM in the current and previous approaches is very similar; however, in this case (using VEAM) is where we achieved the highest classification accuracy (99.44%).

Although the same charts for the ETH-80 dataset are not shown, the comparison with our previous work also reveals the improvement achieved. In [3], the best result for this dataset was 82.03% and with the new proposal the result is 84.39%.

Aside from comparing with our previous approach, we also compared our current approach with other state-of-the-art classification methods that do not use FAS mining techniques. In [32] the best accuracy obtained in COIL-100 dataset is 91.6% while our proposal scored 99.44%. In the case of the ETH-80 dataset, the state-of-the-art results range between 76% and 88% according to the comparison performed by [32], and we obtained 84.39% with our approach, which is comparable to the reported results. Although a comparison with the traditional Bag of Words approach would have served as a baseline to validate our proposal, we consider it is not fair to compare segmentation-based methods with patch-based methods, because it has been shown before that the later usually displays better results [14].

Regarding the running times of these experiments, we can illustrate this aspect in general terms for the ETH-80 dataset. In this case, the step of building the graph-based image

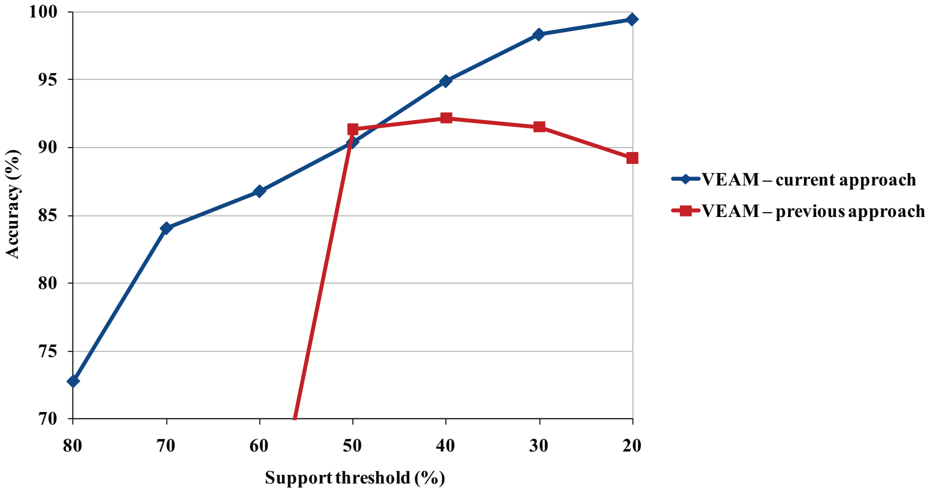

Figure 5: Comparison between the results obtained by VEAM in both classification approaches, using the COIL-100 dataset. "Previous approach" refers to the work presented in [3]

representation (building irregular pyramids, evaluating each level and extracting features for the regions) can take about 4 hours. In the next step, we need less than an hour to create labels for the graph's vertices and computing substitution matrices. The FAS mining step usually takes the longest time in the process, ranging from 4 hours to 4 days, from 80% to 20% supports. After the FAS are obtained, the construction of the FAS histograms and the classification step can be done in less than 5 minutes. This shows that the computational overhead of using a graph-based representation approach is restricted to the extraction of the subgraphs in the training stage, while the classification of new instances can be done very quickly. Nevertheless, a current shortcoming of our approach is related precisely to the computational cost of using these graph mining techniques. At the present moment it is not possible to use larger or more complex datasets to perform experiments, due to memory requirements (as mentioned at the beginning of this section). Currently, we are working on reducing the computational overhead in the mining process in order to test our approach in other scenarios. We believe that this aspect has presented new challenges for the graph mining community, in terms of making their approximate algorithms applicable to larger, real and more complex problems.

The results presented in this work show that our current proposal is able to provide good results for real image classification, and also reveals the importance of using approximate approaches in graph mining when dealing with real images. The analysis and selection of decisive parameters in the process also plays an important role, as well as a proper choice of the underlying image representation.

7. Conclusions

In this work we proposed an image representation using FASs as features that can be used in a classification framework. The FAS are obtained by means of FAS miners reported in the literature. The FAS miners are able to identify FAS patterns in graph collections allowing distortions in the data (in terms of edge and vertex label). We propose to automatically compute substitution matrices and the isomorphism threshold for the mining process, based on image features embedded in the framework, which proves that it can also produce good outcomes for the classification task when expert knowledge is not used. A new visual description of regions is employed, showing the relevance of making accurate choices in this direction, since the underlying representation is the basis for obtaining useful features in the mining process. The classification accuracy results obtained with our current approach using FAS miners outperform the results of traditional miners in most cases. Also, the experimental results show that our proposal is comparable with other state-of-the-art methods for image classification.

In the future, we plan to take advantage of FAS selection strategies for improving graph classification (such as, using discriminative FASs, representative FASs, contrast FASs, etc.). These strategies in combination with FAS miners can improve the efficiency of graph classifiers and may be a way of reducing dimensionality as well. We also plan to keep working on reducing the computational cost of the FAS miners, in order to test this approach in larger and more varied image datasets.

References

[1] Acosta-Mendoza, N., Gago-Alonso, A., Medina-Pagola, J.E.: Clasificación de imágenes utilizando minería de subgrafos frecuentes aproximados. Revista Cubana de Ciencias Informáticas (RCCI), Vol 5, No. 4, 2012.

[2] Acosta-Mendoza, N., Gago-Alonso, A., Medina-Pagola, J.E.: Frequent Approximate Subgraphs as Features for Graph-Based Image Classification. Knowledge-Based Systems 27:381–392, March, 2012.

[3] Acosta-Mendoza, N., Morales-González, A., Gago-Alonso, A., García-Reyes, E.B., Medina-Pagola, J.E.: Classification using Frequent Approximate Subgraphs. In: Proceedings of the 17th Iberoamerican Congress on Pattern Recognition (CIARP'12), Buenos Aires, Argentina, Springer-Verlag Berlin Heidelberg, 2012.

[4] Aggarwal, C.C., Wang, H.: Graph Data Management and Mining: A Survey of Algorithms and Applications. Managing and Mining Graph Data, Springer-Verlag New York, 2010.

[5] Brun, L., Kropatsch, W.: Introduction to combinatorial pyramids. Digital and image geometry: advanced lectures, 108–128, 2001.

[6] Brun, L., Kropatsch, W.: Contains and inside relationships within combinatorial pyramids. Pattern Recognition, 39(4), 108–128, 2006.

[7] Borgelt, C., Berthold, M.R.: Mining Molecular Fragments: Finding Relevant Substructures of Molecules. In *Proceedings of the 2002 International Conference on Data Mining*, pages 211–218, Maebashi, Japan, 2002.

[8] Bunke, H., Riesen, K.: Graph Classification Based on Dissimilarity Space Embedding. Proceedings of the 2008 Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, SSPR & SPR '08, pages 996–1007, 2008

[9] Chen, C., Yan, X., Zhu, F., Han, J.: gApprox: Mining Frequent Approximate Patterns from a Massive Network. IEEE International Conference on Data Mining, 445–450, 2007.

[10] Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. International Journal of Pattern Recognition and Artificial Intelligence, 2004.

[11] Deshpande, M., Kuramochi, M., Wale, N., Karypis, G.: Frequent Substructure-Based Approaches for Classifying Chemical Compounds. IEEE Trans. Knowledge Data Engineering, 17:1036–1050, 2005.

[12] Duchenne, O., Joulin, A., Ponce, J.: A graph-matching kernel for object categorization. IEEE International Conference on Computer Vision (ICCV), pages 1792 – 1799, 2011.

[13] Elsayed, A., Coenen, F., Jiang, C., García-Fiñana, F., Sluming, V.: Corpus Callosum MR Image Classification. Knowledge-Based Systems, 23:330–336, 2010.

[14] Everingham, M. and Van Gool, L. and Williams, C. K. I. and Winn, J. and Zisserman, A.: The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html, 2011.

[15] Gago-Alonso, A., Carrasco-Ochoa, J.A., Medina-Pagola, J.E., Martínez-Trinidad, J.F.: Full Duplicate Candidate Pruning for Frequent Connected Subgraph Mining. In *Integrated Computer-Aided Engineering*, 17:211–225, August 2010.

[16] Galleguillos, C., Belongie, S.: Context Based Object Categorization: A Critical Survey. Computer Vision and Image Understanding (CVIU), 114:712-722, 2010.

[17] Glantz, R., Pelillo, M., Kropatsch, W.G.: Feature selection for graph-based image classifiers. Proceedings of the Second Iberian conference on Pattern Recognition and Image Analysis - Volume Part II, IbPRIA'05, pages 147–154, 2005

[18] Glantz, R., Pelillo, M., Kropatsch, W.G.: Matching Segmentation Hierarchies. International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), vol 18(3):397-424, 2004.

[19] Holder, L.B., Cook, D.J., Bunke, H.: Fuzzy Substructure Discovery. In *Proceedings of the ninth international workshop on Machine learning*, pages 218–223, San Francisco, CA, USA, 1992.

[20] He, X., Ji, M., Bao, H.: Graph Embedding with Constraints. Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI 2009, pages 1065–1070, 2009

[21] Hlaoui, A., Wang, S.: A New Algorithm for Graph Matching with Application to Content-Based Image Retrieval. Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, pages 291–300, 2002.

[22] Hossain, M., Angryk, R.A.: GDClust: A Graph-Based Document Clustering Technique. In *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops*, pages 417–422, Washington, DC, USA, 2007.

[23] Jia, Y., Zhang, J., Huan, J.: An efficient graph-mining method for complicated and noisy data with real-world applications. Knowledge Information Systems, 28(2):423–447, 2011.

[24] Jiang, C., Coenen, F., Zito, M.: A Survey of Frequent Subgraph Mining Algorithm. To appear in Knowledge Engineering Review, 2012.

[25] Jiang, C., Coenen, F.: Graph-based Image Classification by Weighting Scheme. In *Proceedings of the Artificial Intelligence*, pages 63–76, Springer, Heidelberg, 2008.

[26] Jiang, C., Coenen, F., Sanderson, R., Zito, M.: Text Classification using Graph Mining-based Feature Extraction. Knowledge-Based Systems, 23:302–308, 2010.

[27] ketkar, N., Holder, L.B., Cook, D.J.: Mining in the Proximity of Subgraphs. Analysis and Group Detection KDD Workshop on Link Analysis: Dynamics and Statics of Large Networks, 2006.

[28] Kropatsch, W., Haxhimusa, Y., Pizlo, Z., Langs, G.: Vision pyramids that do not grow too high. Pattern Recognition Letters, vol 26(3), pages 319–337, 2005.

[29] Kuramochi, M., Karypis, G.: GREW-A Scalable Frequent Subgraph Discovery Algorithm. In *Proceeding of the 4th IEEE International Conference on Data Mining*, pages 439–442, 2004.

[30] Leibe, B., Schiele, B.: Analyzing Appearance and Contour Based Methods for Object Categorization. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03), pages 409–415, 2003.

[31] Morales-González, A., García-Reyes, E.B.: Assessing the Role of Spatial Relations for the Object Recognition Task. In *Proceeding of the 15th Iberoamerican Congress on Pattern Recognition*, pages 549–556, 2010.

[32] Morales-González, A., García-Reyes, E.B.: Simple object recognition based on spatial relations and

visual features represented using irregular pyramids. Multimedia Tools and Applications, Springer Netherlands, 1380-7501, pages 1–23,
DOI:http://dx.doi.org/10.1007/s11042-011-0938-3, 2011.

[33] Nene, S., Nayar, S., Murase, H.: Columbia Object Image Library (COIL-100). Technical Report, Department of Computer Science, Columbia University CUCS-006-96, 1996.

[34] Nijssen, S., Kok, J.: A Quickstart in Frequent Structure Mining can Make a Difference. In *Proceedings of the 2004 ACM SIGKDD International Conference on Kowledge Discovery in Databases*, pages 647–352, Seattle, WA, 2004.

[35] Nowozin, S., Tsuda, K., Uno, T., Kudo, T., BakIr, G.: Weighted Substructure Mining for Image Analysis. In *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1–8, IEEE Computer Society, 2007.

[36] Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distribution. Pattern Recognition, 29(1):51–59, 1996.

[37] Ozdemir, B., Aksoy, S.: Image Classification Using Subgraph Histogram Representation. In *Proceedings of the 20th International Conference on Pattern Recognition*, pages 1112–1115, IEEE Computer Society, 2010.

[38] O'Hara, S., Draper, B.A.: Introduction to the Bag of Features Paradigm for Image Classification and Retrieval. Computing Research Repository (CoRR), abs/1101.3354, 2011

[39] Pantofaru, C., Dork, G., Schmid, C., Hebert, M.: Combining Regions and Patches for Object Class Localization. The Beyond Patches Workshop in conjunction with the IEEE conference on Computer Vision and Pattern Recognition (CVPR), 2006.

[40] Pantofaru, C., Schmid, C., Hebert, M.: Object Recognition by Integrating Multiple Image Segmentations. Proceedings of the 10th European Conference on Computer Vision: Part III, ECCV '08, pages 481–494, 2008.

[41] Papapetrou, O., Ioanno, E., Skoutas, D.: Efficient Discovery of Frequent Subgraph Patterns in Uncertain Graph Databases. In *Proceedings of the 14th International Conference on Extending Database Technology*, 12:355–366, New York, USA, 2011.

[42] Riesen, K., Bunke, H.: IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning. pages 287–297. 2008.

[43] Song, Y., Chen, S.: Item Sets Based Graph Mining Algorithm and Application in Genetic Regulatory Networks. In *Proceedings of the IEEE International Conference on Granular Computing*, pages 337–340, Atlanta, GA, USA, 2006.

[44] Xiao, Y., Wu, W., Wang, W., He, Z.: Efficient Algorithms for Node Disjoint Subgraph Homeomorphism Determination. In *Proceedings of the 13th International Conference on Database Systems for Advanced Applications*, pages 452–460, New Delhi, India, 2008.

[45] Yan, X., Han, J.: gSpan: Graph-Based Substructure Pattern Mining. In *Proceedings of the 2002 International Conference on Data Mining (ICDM'02)*, pages 721–724, Maebashi, Japan, 2002.

[46] Yun, U., Ryu, K.H.: Approximate weighted frequent pattern mining with/without noisy environments. Knowledge-Based Systems, 24: 73–82, 2011.

[47] Zhang, S., Yang, J.: RAM: Randomized Approximate Graph Mining. In *Proceedings of the 20th International Conference on Scientific and Statistical Database Management*, pages 187–203, Hong Kong, China, 2008.

[48] Zou, Z., Li, J., Gao, H., Zhang, S.: Mining Frequent Subgraph Patterns from Uncertain Graph Data. IEEE Transactions on Knowledge and Data Engineering, vol 22(9):1203–1218, 2010.