Graph Based Image Search Using Clustering Approach

Anuradha Karlekar

Siddhivinayak College of Higher Education & Research, RTU.

karlekaranuradha28@gmail.com

Abstract-- From the many approaches for image classification, graph based approach is gaining popularity due to its ability in reflecting global image properties. In this report, VEAM (Vertex and Edge Approximate graph Miner) algorithm is used for mining frequent connected sub-graphs over undirected and labeled graph collections. Slight variations of the data, keeping the topology of the graphs, are allowed. In this report we have proposed graph-based image representation by using Dynamic Region Merging (DRM) technique. DRM is used with watershed segmentation. It can tolerate some variations for grouping meaningful regions.

Keywords: Graph-based image representation, Dynamic region merging, Frequent approximate sub-graphs,Graph Clusturing,segmentation,Watershed,VEAM(Vertex and Edge Approximate graphMiner).

I. INTRODUCTION

In many innovative fields, hierarchical data structures, specifically graphs, are mainly used to store and process realistic data due to their ability to model complex relationships. Also, a vast amount of graph theories has been developed in order to work with graphs and process the information they represent. We intend to explore and combine two research fields where graphs are involved, in order to exploit both their advantages.

The first, image classification field is related to Computer Visualization. Our intension is to categorize images using a graph-based representation. The important factor for image classification is to extract low-level features, but it has been found that low-level information by itself cannot provide the high-level insight clues that exist in human minds to describe objects or images in general (semantic gap problem). Within the scope of low-level features developed so far, graphs are among the representations that can provide some kind of high-level information implicitly, making them advantageous choice of representation for researchers to find new solutions. One major concern related to this representation is that although graphs are dominant representational tools, they are inflexible or complex to handle, and usually includes algorithms with high computational costs. One of the solutions is to simplify the data structure, but it will lose some of the embedded information.

Now we will consider the other research field that we are trying to discover: Data Mining. Several graph-based approaches and procedures are developed to satisfy the necessity of using huge bulks of data and extract useful information from these heavy chunks of data [4, 22,].One such techniques include Frequent sub graph discovery [15]. An important obstacle in graph mining tasks is organizing information, such as image, text, and molecular datasets. Although using graph mining for classification purposes has been broadly explored, these approaches may not always produce the optimal results in all applications and there is a necessity to use approximate graph matching for frequent sub graph mining. Frequent approximate sub graph (FAS) mining is an important problem in graph mining, where the mined patterns are detected taking into account such semantic misrepresentations. Thus, such approximate solutions achieve classification results which are different from the other graph mining methods.

As mentioned before, we aim at combining and exploiting both research fields (image classification and frequent approximate sub graph mining) by proposing an image representation that can be used in a classification framework. The new contributions and changes with respect to the previous work are as following:
1. The images are segmented using Watershed algorithm.
2. The over segmented regions are automatically merged by DRM technique.
3. From these regions graphs and patterns are generated.
4. Using VEAM algorithm, these patterns are mined
5. By using KNN classifier these patterns are classified to generate final classified images.
We performed new experiments to show how these improvements largely and positively influence the results, while comparing it also with other state-of-the-art methods in image classification.

II. LITERATURE SURVEY

In this section, we start by providing a brief on a review of previous works related to graph based image representation and approximate graph mining.

A. Graph-based image representation
B.

Several ways are established to denote images in graph forms [10, 5, and 6]. This type of representation is advantageous for image processing since graphs are capable of describing the structural and topological information of images. The important idea of graph-based image representation is that the regions of the image, that contain similar properties, are denoted by graph vertices, and therefore the relations between different regions are denoted by graph edges. The vertex and edge attributes typically describe the characteristics of that region and therefore the relation between regions respectively. An easy approach to retain the structural and topological information of an image is to use digital image representation techniques; for example, quad-trees [10], etc. By modelling images as graphs, the task of image classification becomes the task of classifying graphs.

Quad-trees [10] were used for representing images in the form of trees. A quad-tree consists of splitting an image in four equal-sized quadrants. Each of those quadrants can be additional split into four sub-quadrants (Northwest, Northeast, Southwest and Southeast), and so on. The global process of the quad-tree consists of dividing the sub-images recursively till the obligatory limit for the amount of divisions is met. An example of quad-tree with four as depth limit of divisions over a picture is showed in figure below. Once the depth limit of divisions is achieved and any sub-quadrant contains many properties to obtain, then the major property during this sub-quadrant is taken as attribute of a tree node. The property taken as an attribute of the nodes in the example of figure below is the color.
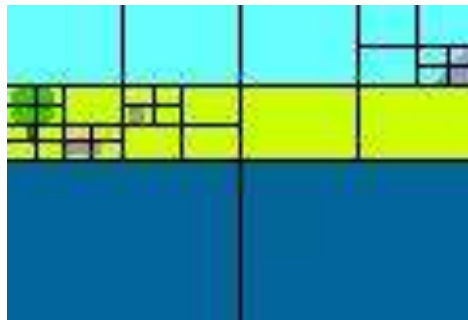

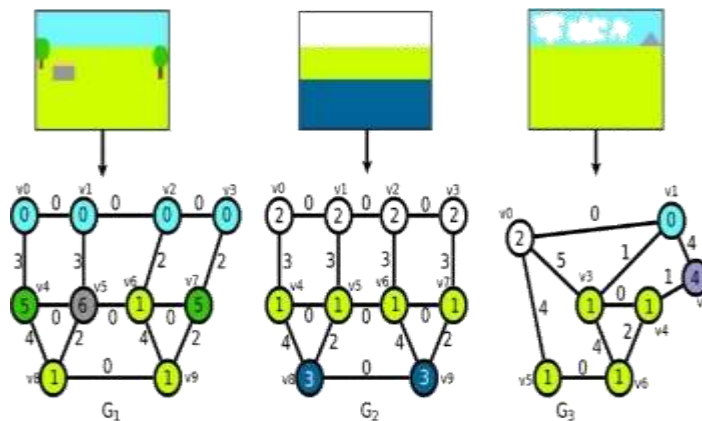Figure 1: Example image with the quadrants of quad-tree identified.

Figure 2: Collection of graph $D = fG_1; G_2; G_3g$.

After obtaining the quad-tree of an image, we produce a graph to characterize the image with the aim of observing the structural information such as the relations between entities. So, the graph (image) mining can be performed to discover hidden patterns among graph collection. The process of graph generation consists of three steps:

1. The quad-tree leaves (sub-quadrants) with its attributes become the vertices of the graph. The midpoint of the quadrant is the attributes of the leaf nodes of the quad-tree and the major attributes of the image is their labels (i.e. texture, color, etc.).

2. Iteratively for each level from top to bottom of the quad-tree each leaf is visited.

Example 1. Let a quadrant divisions showed in figure 2.1.1 be the quad-tree of an image. We say that the vertex denoted by the number eight and each vertex of the set $\{5; 7; 9; 13; 14; 17\}$ are connected by an edge.
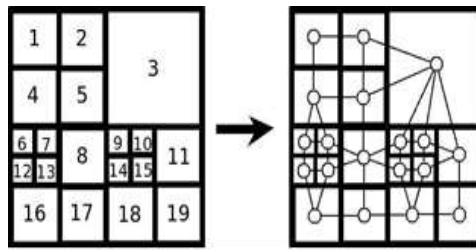
Figure 3: Example of quadrant divisions of an image

3. The label of each edge *e*, denoted by *label* (*e*), is an index obtained through the following function over the angle between this edge and the horizontal line. This function depends on a value of the number of classes *n* that categorize the possible angles. The possible angles cover 180 degrees supposing that these edges are undirected.

Though graphs provide powerful representation, they have a major drawback of lack of suitable classification methods where they can be used. Graph matching techniques have high computational complexity and graph embedding methods suppress part of the information encoded by the graphs. Graph mining techniques used for classification purposes could be consider a midway between graph matching and graph embedding methods, thus combining their advantages and reducing their individual disadvantages.

C. Background on FAS mining

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

1) Basic concepts:

There are several algorithms for frequent approximate sub graph mining in graph collections which use different similarity functions for graph matching. The approximate sub graph mining can be divided into five kinds according to the matching approach:

1. Based on graph edit distance: The algorithms SUBDUE [14] and RNGV [29] exploring possible edit paths of a graph as required by the candidate. These algorithms do not claim completeness.
2. Based on edge sub-isomorphism: The Monkey algorithm [35, 36] handles only missing edge and edge label mismatch, where the maximum number of edge differences allowed between sub graphs.
3. Based on node/edge disjoint sub-homeomorphism: The CS Miner algorithm [31, 32] discoveries approximate structures that share the similar topology.
4. Based on sub-isomorphism on uncertain graphs: The MUSE algorithm [37, 38] calculates the estimated support of each candidate through a given interval. This algorithm process uncertain graphs as they contain a non-occurrence probability.
5. Based on substitution probabilities: The algorithms g Approx [5] and APGM [17, 18] specifies which vertices, edges or labels can replace others. Thus, the idea that always a vertex label or an edge label can be replaced by any other vertex label or an edge label is defended.

In this paper, we propose a solution based on the approach of the last group, as we are looking for an algorithm which permits some variations in the data through the substitution probability, at the same time retaining the topology of the graphs.

2) Approximate FAS methods considered:

In APGM (Approximate graph miner) [22] and VEAM (Vertex and Edge Approximate graph Miner) [2] algorithms, the concept that always a vertex label or an edge label can be replaced by the other, is supported. Therefore, these algorithms specify which vertices, edges or labels can replace others using substitution matrices to perform the frequent sub graph mining task. APGM (Approximate graph miner) solely deals with the variations among vertex labels, whereas VEAM (Vertex and Edge Approximate graph Miner) performs the mining method using the vertex and edge label sets. These approaches use substitution matrices where each cell represents the suitability of approximating one vertex by another vertex (or an edge by another edge) and they provide frameworks for each frequent sub graph mining task.

Here, we propose a replacement algorithm for frequent approximate sub graphs mining referred to as VEAM (Vertex and Edge Approximate graph Miner). This algorithm incorporates in the approximate graph matching all labels that seem in the substitution matrices. Notice that if there are labels that do not seem in any graph of the collection and appear in the matrices, then these labels will be utilized in the mining method, and frequent approximate sub graphs with those labels can be obtained.

VEAM (Vertex and Edge Approximate graph Miner) computes the frequency of the vertices, and starts mining from the frequent ones. At a subsequent step, it adds an edge to an existing pattern, in all possible ways, to make new candidate sub graphs and determine their support value. In this algorithm depth-first search (DFS) is employed to search

out candidates by extending each current sub graphs pattern employing a new edge. VEAM (Vertex and Edge Approximate graph Miner) stops when no additional patterns can be extended.

III. **METHODOLOGY**

After briefing 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 as follows:

1. The images are over segmented using Watershed algorithm.
2. Automatically group the initially over segmented many small regions by DRM technique.
3. From these regions graphs and patterns are generated.
4. Using VEAM algorithm, these patterns are mined5. By using KNN classifier these patterns are classified to generate final classified images.

In the above framework following tools are combined.
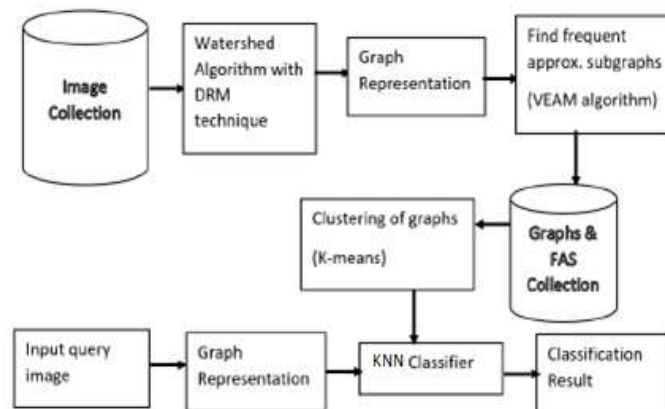
A. Overall System Design/Architectural

Figure 4. Framework of graph-based image classification

First of all, we obtain the grahased image representation of a given set of real imagesusing dynamic image merging, which gives us a graph collection to work with.After that,the algorithms for frequent subgraph mining are implemented with the goal of obtaining allrequent subgraphs over a graph collection? Later, these subgraphs are used as features andthe feature vectors of the original images are made.These feature vectors are compared withthe feature vector of input query image. Finally, we employ a classier generator i.e.KNNusing these vectors as data to produce an image classication result. The completearchitectureof our experiment procedure is shown in Fig.4.

B. Combine Approach:

Several techniques have been developed to represent images in graphical forms. This sort of representation has been a great help for image processing since graphs can describe the structural and topological information of images. The main plan of graph based image representation focuses on the regions of the image, which contain similar properties, are denoted by graph vertices, and the relations between different regions are denoted by graph edges. The vertex and edge attributes typically describe the characteristics of that region and the relation between regions severally. An easy approach to maintain the structural and topological information of an image is to use image representation techniques; for instance, dynamic image merging, etc. By modelling images as graphs, the task of image classification becomes one of classifying graphs. Starting from an over segmented image, neighbouring regions merged if there is an evidence for merging according to predicate. Given an over segmented image; there are several regions to be integrated for a significant segmentation. By taking the region merging as a labelling problem, the goal is to assign every region a label such that regions belong to the same object will have the same label. The proposed DRM algorithm starts from an initially over segmented image. For simplicity, we use the watershed algorithm to get the initial segmentation. It is clear that neighbouring regions with coherent colours are integrated into one, whereas the boundaries are well located on the reasonable places as shown in figure 5. A number of the massive regions have substantial variations inside, however, with comparatively slow changes of colours along the boundaries. This means that the DRM algorithm can tolerate some variations for grouping significant regions in an image. In Fig.5 from left to right, the first column shows the original image. The second column shows the over segmentation produced by watershed algorithm. The third column shows segmentation results exploitation DRM algorithm.

Figure 5: Segmentation results by the proposed algorithm

C. Watershed Based Image Segmentation:

Watershed transformation also known as, as watershed technique is a powerful mathematical morphological tool for the image segmentation.. In geography, watershed means the ridge that divides areas drained by different watercourse systems. If image is observed as geological landscape, the watershed lines decide boundaries which separate image regions. The watershed transform computes structure basins and ridgelines (also called watershed lines), where structure basins corresponding to image regions and ridgelines relating to region boundaries. Segmentation by watershed embodies several of the concepts of the three techniques such as threshold based, edge and region based segmentation. It gives excellent segmentation results, and meets the standards of less computational complexity for hardware implementation. Block diagram of Watershed based Segmentation is shown below. There are mainly three stages as indicated by Figure 6 for watershed based mostly image segmentation approach. Initial stage is defined as pre-processing, second stage as watershed based image segmentation and last stage as post-processing. Input image is initially processed by the pre-processing stage, and so given to watershed based mostly segmentation stage. The resulting image is post processed by the final stage to get a segmented image. Pre-processing and post-processing are necessary to beat the problem of over-segmentation in watershed based image segmentation.
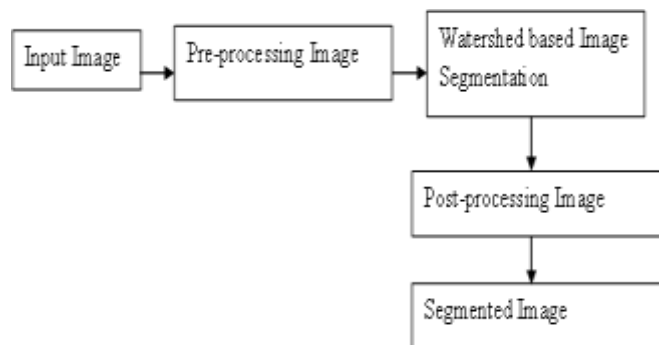

Figure 6: Block diagram of watershed based image segmentation
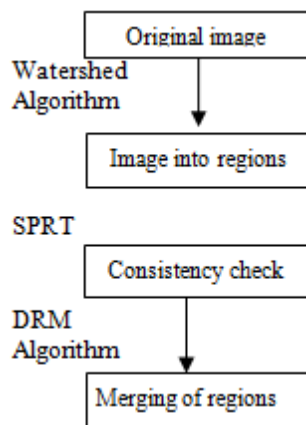
D. Dynamic Region Merging (DRM):

Figure 7: Block diagram of DRM.

In this section, we explain the proposed region merging algorithm as a dynamic region merging (DRM) process, as mentioned above, the proposed DRM algorithm is started from a set of over-segmented regions. This is because a small region can provide more stable statistical information than a single pixel, and using regions for merging can improve a lot the computational efficiency. For simplicity and in order to validate the effectiveness of the proposed DRM algorithm, we use the watershed algorithm, to obtain the initially over-segmented regions.

In Fig. 8, an example process of region merging is shown by embedding it into a 3D graph
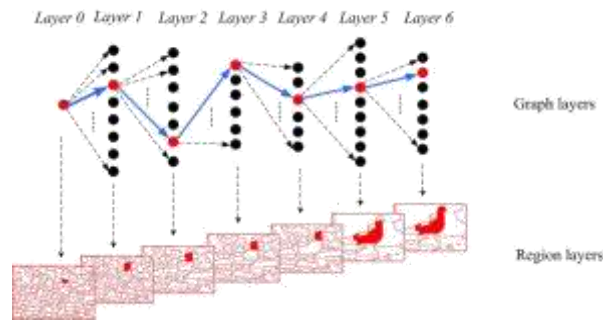
Figure 8: the dynamic region merging process as a shortest path in a layered graph. The upper row shows the label transitions of a graph node. The lower row shows the corresponding image regions of each label layer. Starting from layer 0, the highlighted region (in red) obtains a new label from its closest neighbor (in red). If the region is merged with its neighbor, they will be assigned to the same label. The shortest path is shown as the group of the directed edges (in blue).

Dynamic Region Merging Algorithm

DRM algorithm begins from a set of segmented regions. This is because a small region can provide more stable statistical information than a single pixel, and using regions for merging can improve a lot the computational efficiency.

The algorithm follows as:

1. Input (I/P) = initially over segmented image So.
2. Output (O/P) = region merging result
3. Steps:
4. For given over segmented image assign each region a label as L0 to Ln.
5. Consider n regions.
6. Assign label to regions as L0………. Ln.
7. The label of each region is sequentially transited from initial to final.
8. To find the optimal sequence of merges to produce optimal merging of all regions,
9. The problem is broken down into sub problems using dynamic programming.
11. Obtain the merging result by shortest path algorithm

In the DRM algorithm, there is at least one pair of regions to be merged in each iteration before the stopping criteria is satisfied.

E. Frequent Approximate Sub graphs:

Discussed earlier the use of approximate graph matching for frequent sub graph mining has been used in different applications. To meet this need, many algorithms have been developed, but there are applications where it has not been used yet, for example image classification. A new algorithm for mining frequent connected sub graphs over undirected and labelled graph collections, VEAM (Vertex and Edge Approximate graph Miner) is presented. Slight variation in the data, retaining the topology of the graph, is allowed. In VEAM, the approximate matching between edge labels set in frequent sub graph mining is included in the mining process. The approximate method of VEAM was tested on an artificial image collection using a graph-based image representation. The experimentation on this collection shows that our proposal gets better results than graph-based image classification using some algorithms reported in related work.

Algorithm 1: *VEAM*

Input: *D:* A graph collection, *MV* : Substitution matrix indexed by *LV* , *ME* : Substitution matrix indexed by *LE*, _ : Isomorphism threshold, _ : Support threshold.

Output: *F* : Frequent approximate sub graph set.

$F \leftarrow C \leftarrow 1$ { the single vertices labeled in LV , which are frequent approximate sub graphs in D};

    2 for each $T \in C$ **do**

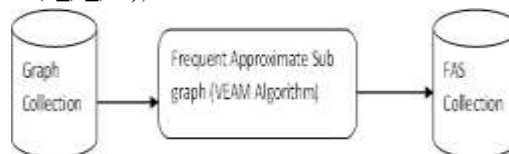   **3** VEAMSearch(*T;D;MV;ME; _; _; F*);

Figure 9: VEAM process

F. Clustering of graphs (Grouping):

Any no uniform data contains underlying structure due to the heterogeneity of the data. The process of identifying this structure in terms of grouping the data elements is called clustering, also called data classification. The resulting groups are called clusters. The grouping is usually based on some similarity measure defined for the data elements. Clustering is thoroughly associated to unsupervised learning in pattern recognition systems. A basic task in

unsupervised learning is to classify data set into two or more classes based on a similarity measure over the data, without resorting to any prior information on how the classification should be done. A graph is a mathematical structure made up of set of vertices or nodes and is connected by a set of edges or lines. Graph clustering is the task of grouping the vertices of the graph into clusters considering the edge structure of the graph such that there should be several edges within each cluster and relatively few between the clusters. Graph clustering means grouping the vertices of a given input graph into clusters.

Here we have used K-means Clustering Algorithm. It is one of the simplest algorithms for clustering of data (graph). The procedure that is followed is a simple and easy way to classify data (graph) through a certain number of clusters (say k clusters) fixed a priori. The main idea is to define k centres, one for each cluster.

Algorithm (k-means clustering):
Consider A = {$a_1,a_2,a_3,........,a_n$} be the set of data points and Z = {$z_1,z_2,.......,z_c$} be the set of centers.
1) Randomly select *'c'* cluster centers.
2) Calculate the distance between each data point and cluster centers.
3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers.
4) Recalculate the new cluster center.
5) Recalculate the distance between each data point and new obtained cluster centers.
6) If no data point was reassigned then stop, otherwise repeat from step 3.

G. Query Processing:

This block is the important section of the project. In this section, a query image is taken from the User. The image is represented as Graph as discussed above using DRM technique. Then Patterns are generated. Then stored Patterns and Generated Patterns are matched for relevant results and finally Results are displayed along with Accuracy and Precision.
Outcomes of this system are:
1. When a query image is given to the system, images in database are classified as per their relevance with query image.
2. Also for a given query, accuracy and precision is calculated by considering entire database and is displayed.

H. KNN Classifier:

Classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. K nearest neighbours is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions) KNN is a *non parametric lazy learning* algorithm. That is a pretty concise statement. When you say a technique is non-parametric, it means that it does not make any assumptions on the underlying data distribution. This is pretty useful, as in the real world, most of the practical data does not obey the typical theoretical assumptions made (e.g. Gaussian mixtures, linearly separable etc.)Non parametric algorithms like KNN come to the rescue here. It is also a lazy algorithm. What this means is that it does not use the training data points to do any *generalization.* In other words, there is *no explicit training phase* or it is very minimal. This means the training phase is pretty fast. Lack of generalization means that KNN keeps track of all the training data. More exactly, all the training data is needed during the testing phase. (Well this is an exaggeration, but not far from truth). More memory is needed as we need to store all training data .k-Nearest neighbour is an example of instance-based learning, in which the training data set is stored, so that a classification for a new unclassified record may be found simply by comparing it to the most similar records in the training set.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

Actually, in the graph mining community; the standards to perform tests are synthetic datasets or molecular datasets. when we try to use these techniques in real images, they have to deal with larger graphs, with sizes starting from two hundred to three hundred edges per graph, and also the current progressive ways for graph mining are not well ready to address such huge graph sizes, in massive collections, in terms of memory requirements and computational time. If this is considered a computational problem for graph mining methods, the procedure of approximation in graphs increases the complexity.

In order to certify our proposal, we have chosen well-known databases containing colourful images of simple objects captured from different viewpoints. The first one is that the COIL-100 [33] dataset, that possess a hundred objects with seventy two poses per object. We tend to took twenty five objects arbitrarily selected from this dataset to check our classification framework. The second dataset is ETH-80 Image Set [30], that contains eighty objects from eight classes and every object is described by forty one totally different views, yielding a complete of 3280 images. Example images from each datasets can be seen in Fig 10.

Figure 10: Sample images

The results shown in this work present that our current proposal is able to provide efficient result for real time image classification, and also underlines the importance of using approximate methods in graph mining when dealing with real time images. The analysis and selection of pivotal parameters in the process also play a significant role, as well as an appropriate choice of the underlying image representation.

Content based image retrieval (CBIR) using color histogram technique is used for result analysis. CBIR is the process of retrieving images from a database (library of digital images) based upon the visual content of the images. In other words, it is the retrieving of images that have similar content of colors, textures or shapes transparent. In CBIR system, usually the image features are grouped into three main classes: color, texture and shape. Color is the most common visual feature (parameter) used in CBIR, mostly because of the simplicity of extracting color information from images. CBIR system decides which images in the image database is the most similar one with the query image. Obtain the nearest neighbors of the query image, exact images are retrieved. In content-based image retrieval (CBIR) searching process for k most-similar images to a query image (required image) involves comparing the feature vectors of all the images in the database with that of the query image (required image) using some initially selected (pre-selected) similarity measure, and then sorting the results. However, this requires linear time (time complexity) with respect to the size of the database and becomes impractical for large databases.

CBIR is used to generate idle sequence of images. This sequence is compared with the system generated sequence of images and precision and accuracy is calculated and displayed. Accuracy and Precision are calculated as:

$$accuracy = \frac{number\ of\ true\ positives + number\ of\ true\ negatives}{number\ of\ true\ positives + false\ positives + false\ negatives + true\ negatives}$$

$$precision = \frac{number\ of\ true\ positives}{number\ of\ true\ positives + false\ positives}$$

The CBIR is applied to an image database and the result is shown below.
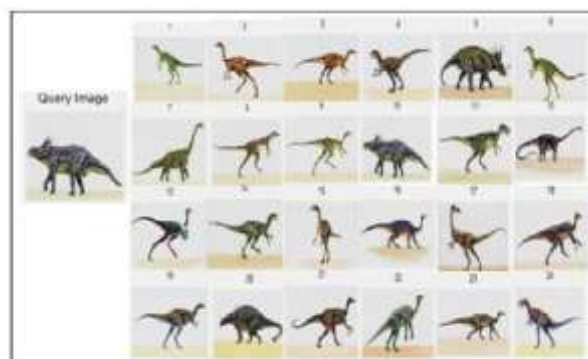

Figure 11: Output of CBIR

Figure 12: Output of Proposed system

CONCLUSIONS AND FUTURE SCOPE

In this work we proposed and presented an image representation scheme using FASs (Frequent Approximate Sub graph) as efficient and effective features that can be used in a classification framework. Here we have 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. Also, the experimental results show that our proposal is comparable with other state-of-the-art methods for image classification.

The goal of this project is graph based image classification in which DRM techniqueis used for graph generation which will improve performance of system.Graph generatedby DRM technique is very small in size as compared to qu adtree in existing system.Solots of space get saved and also time get reduced because of less number of patterns generate d from graphs.By using approximation in VEAM algorithm, more interestingpatterns can be found for many applications, f or instance, when processing graph databasesthat have distortions (in terms of different geometric, topological or semantic va riations) ofsimilar structures in several objects. Distortion in data is one of the challenges for developing classifiers based on frequent patterns in several domains of science.

As future work, we will propose a new interest measures to reduce the number ofpatterns identified which k eep or increase the classification accuracy. We will developnovel prune strategies to obtain algorithms more efficient for larger.

REFERENCES

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

[2] N. Acosta-Mendoza, A. Gago-Alonso, J.E. Medina-Pagola, Frequent approx- imate subgraphs as features for graph-based image classification, Knowledge- Based Systems 27 (March) (2012) 381–392.

[3] N. Acosta-Mendoza, A. Morales-González, A. Gago-Alonso, E.B. García-Reyes, J.E. Medina-Pagola, 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] C.C. Aggarwal, H. Wang, Graph Data Management and Mining: A Survey of Algorithms and Applications, Managing and Mining Graph Data, Springer- Verlag, New York, 2010.

[5] L. Brun, W. Kropatsch, Introduction to combinatorial pyramids, Digital and Image Geometry: Advanced Lectures

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

[7] C. Borgelt, M.R. Berthold, Mining molecular fragments: finding relevant substructures of molecules, in: Proceedings of the 2002 International Con- ference on Data Mining, Maebashi, Japan, 2002, pp. 211–218.

[8] H. Bunke, K. Riesen, Graph classification based on dissimilarity space embed- ding, in: Proceedings of the 2008 Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, SSPR & SPR '08,.

[9] C. Chen, X. Yan, F. Zhu, J. Han, gApprox: mining frequent approximate patterns from a massive network, in: IEEE International Conference on Data Mining, 2007, pp. 445–450.

[10] D. Conte, P. Foggia, C. Sansone, M. Vento, Thirty years of graph matching in pattern recognition, International Journal of Pattern Recognition and Artificial Intelligence 18 (3) (2004) 265–298.

[11] M. Deshpande, M. Kuramochi, N. Wale, G. Karypis, Frequent substructure- based approaches for classifying chemical compounds, IEEE Transactions on Knowledge Data Engineering 17 (2005) 1036–1050.

[12] O. Duchenne, A. Joulin, J. Ponce, A graph-matching kernel for object categor- ization, in: IEEE International Conference on Computer Vision (ICCV), 2011,

[13] A. Elsayed, F. Coenen, C. Jiang, F. García-Fiñana, V. Sluming, Corpus Callosum MR image classification, Knowledge-Based Systems 23 (2010) 330–336.

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

[15] A. Gago-Alonso, J.A. Carrasco-Ochoa, J.E. Medina-Pagola, J.F. Martínez-Trinidad, Full duplicate candidate pruning for frequent connected subgraph mining, in: Integrated Computer-Aided Engineering, 17 August, 2010, pp.–

[16] C. Galleguillos, S. Belongie, Context based object categorization: a critical survey, Computer Vision and Image Understanding (CVIU) 114 (2010) 712–722.

[17] R. Glantz, M. Pelillo, W.G. Kropatsch, Feature selection for graph-based image classifiers, in: Proceedings of the Second Iberian Conference on Pattern Recogni- tion and Image Analysis – Volume Part II, IbPRIA'05, 2005, pp.–

[18] R. Glantz, M. Pelillo, W.G. Kropatsch, Matching segmentation hierarchies, International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI) 18 (3) (2004) 397–424.

[19] L.B. Holder, D.J. Cook, H. Bunke, Fuzzy substructure discovery, in: Proceedings of the Ninth International Workshop on Machine Learning, San Francisco, CA, USA, 1992, pp. 218–223.

[20] X. He, M. Ji, H. Bao, Graph embedding with constraints, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI 2009, 2009,

[20] A. Hlaoui, S. Wang, A new algorithm for graph matching with application to content-based image retrieval, in: Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, ,

[21] M. Hossain, R.A. Angryk, GDClust: a graph-based document clustering technique, in: Proceedings of the Seventh IEEE International Conference.