Object Boundary Detection for Ontology-based Image Classification*

Lei Wang, Latifur Khan, and Casey Breen
Department of Computer Science
University of Texas at Dallas, TX 75083-0688
Email: [leiwang, lkhan, casey]@utdallas.edu

ABSTRACT

Technology in the field of digital media generates huge amounts of non-textual information, audio, video, and images, along with more familiar textual information. The potential for exchange and retrieval of information is vast and daunting. The key problem in achieving efficient and user-friendly retrieval in the domain of image is the development of a search mechanism to guarantee delivery of minimal irrelevant information (high precision) while insuring that relevant information is not overlooked (high recall). The traditional solution to the problem of image retrieval employs content-based search techniques based on color, histogram, texture or shape features. The traditional solution works well in performing searches in which the user specifies images containing a sample object, or a sample textural pattern, in which the object or pattern is indexed. One can overcome this restriction by indexing images according to meanings rather than objects that appear in images, although this will entail a way of converting objects to meanings. We have solved this problem of creating a meaning based index structure through the design and implementation of a concept-based model using domain dependent ontologies. An ontology is a collection of concepts and their interrelationships which provide an abstract view of an application domain. With regard to converting objects to meaning the key issue is to identify appropriate concepts that both describe and identify images. For this, first we need to identify all object boundaries accurately that appear in images. We propose an automatic scalable object boundary detection algorithm based on edge detection and region growing techniques. We also propose an efficient merging algorithm to join adjacent regions using an adjacency graph to avoid the over-segmentation of regions. To illustrate the effectiveness of our algorithm in automatic image classification we implement a very basic system aimed at the classification of images in the sports domain. By identifying objects in images, we show that our approach works well when objects in images have less complex organization.

1. INTRODUCTION

The development of technology in the field of digital media generates huge amounts of non-textual information, such as audio, video, and images, as well as more familiar textual information [17]. The potential for the exchange and retrieval of information is vast, and at times daunting. In general, users can be easily overwhelmed by the amount of information available via electronic means. The need for user-customized information selection is clear. The transfer of irrelevant information in the form of documents (e.g. text, audio, video) retrieved by an information retrieval system and which are of no use to the user wastes network bandwidth and frustrates users. This condition is a result of inaccuracies in the representation of the documents in the database, as well as confusion and imprecision in user queries, since users are frequently unable to express their needs efficiently and accurately. These factors contribute to the loss of information and to the provision of irrelevant information. Therefore, the key problem to be addressed in information selection in the domain of image is the development of a search mechanism which will guarantee the delivery of a minimum of irrelevant information (high precision), as well as insuring that relevant information is not overlooked (high recall).

Images consist of various objects, each of which may be used to effectively classify the image. The unstructured format of images tends to resist standard categorization and classification techniques. Traditional systems used to store and process multimedia images provide no means of automatic classification. The ability of these systems to retrieve relevant documents based on search criteria could be greatly increased if they were able to provide an accurate and semantic description of an image based on image content.

The traditional solution to the problem of image retrieval employs content-based search technique based on color, histogram, texture or shape features. The traditional solution works well in performing searches in which the user specifies images containing a sample object, or a sample textural pattern [9, 24, 28, 29, 30]. Should a user ask for an image depicting a basketball game, the results become less accurate. This is due to the fact that though an image may contain a basketball, it does not necessarily depict a basketball game. In order to overcome the shortcomings of traditional technique in responding to image classification we have designed and implemented a concept-based model using ontologies [3, 17, 18, 4, 19, 20]. This model, which employs a domain dependent ontology, is presented in this paper. An ontology is a collection of concepts and their interrelationships, which can collectively provide an abstract view of an application domain [5, 14, 15].

In our system we would like to address two distinct questions: the extraction of the semantic concepts from the images and the construction of an ontology. With regard to the first problem, the extraction of semantic concepts, the key issue is to identify appropriate concepts that describe and identify images. We would like to make sure that irrelevant concepts will not be associated and matched, and that relevant concepts will not be discarded. In other words, it is important to ensure that high precision and high recall will be preserved during concept selection. To the best of our knowledge there are no attempts to connect images and concepts through the use of ontologies in any traditional image retrieval systems. We propose an automatic mechanism for the selection of these concepts (for more details see [3, 4]). In ontologies each concept is described by a set of features (objects). To select concept(s) for each image, we need first to identify object boundaries. For this, an object detection algorithm is invoked. In this paper we only address the problem of the extraction of object boundary. Although we detect object boundaries of images, we will not identify or label these objects. For this, we use neural networks to identify objects that appeared in images. Neural networks prove to be an effective method used to automatically find a wide range of patterns in sample data. After the objects have been identified, their identifications are fed into a concept selection module using ontologies to select appropriate concepts.

We propose an automatic scalable object boundary detection algorithm. Our algorithm works in three stages. First, we detect all edge pixels in images and divide pixels into two sets, edge pixel and region pixel sets. Second, we grow a region from the region pixel set surrounded by edges taken from the edge pixel set. Finally, we may merge adjacent regions using an adjacency graph to avoid over segmentation of regions and to detect boundary of objects accurately. To illustrate the effectiveness of our algorithm in automatic image classification we implement a very basic system aimed at the classification of images in the sports domain. By identifying objects in images, we show that our approach works well when objects in images have less complex organization.

Section 2 of this paper discusses work related to image segmentation and ontologies for use in image retrieval, as well as the current systems used for image processing. Section 3 describes ontologies, and how they may be used to specify interrelationships among concepts that help draw meaningful conclusions about images. Section 4 describes outline of our approach. Section 5 presents elaborately our approach to detect object boundary. Section 6 presents preliminary result of our approach. Section 7 presents our conclusion and possible areas of future work.

2. RELATED WORK

Several systems exist today that attempt to classify images based on their content. Successful classification of an image and its contents relates directly to how well relevant images may be retrieved when a search is preformed. Most image storing systems such as QBIC [24] and VisualSEEK [28, 29] limit classification mechanism to describing an image based on metadata such as color histograms [30], texture, or shape features [2, 25]. These systems have high success in performing searches in which the user specifies images containing a sample object, or a sample texture pattern. Should a user ask for an image depicting a basketball game, the results become less accurate. This is due to the fact that though an image may contain a basketball, it does not depict a basketball game. Systems that only contain metadata regarding the objects contained in an image cannot provide an accurate classification of the entire image.

Other systems attempt to provide images with a more precise description by analyzing other elements surrounding the images, such as captions [26, 27], or HTML tags on web pages [37]. These systems use this information to help classify the image and give it a meaningful description. This approach, tied together with metadata on images such as histograms, texture, and color sampling has the potential to yield high precision results in image classification. Examining the textual descriptions associated with an image provides additional information that may be used to help better classify the image. Unfortunately, this approach does not take into account the connections among individual objects present in a sample image. Such connections provide useful information in the form of relationships among objects present in the image, which could be used to help classify the image's content.

To classify images we first need to segment images to detect objects. For this, simple color based segmentation techniques described in [13, 16, 31, 32, 34, 35, 36] may be used effectively to find regions rather than objects in a sample image. For example, Y. Deng et al. [36] propose a statistical method for segmenting color images based on a "J value." For region merge, agglomerative clustering technique is used. On the other hand, in our approach our main concern is to detect an object boundary in an image. For this, we detect edge pixels, and then use these pixels to locate regions. Furthermore, to avoid regions which are over-segmented, we propose a new method based on the use of an adjacency graph which is similar to [34]. However, to check the adjacency of two regions A. Trmeau et al. [34] use a minimum bounding rectangle that may identify some non adjacent regions as adjacent (false positive). We use a matrix method, which may substantially avoid false positives.

3. ONTOLOGIES

An ontology is a specification of an abstract, simplified view of the world that we wish to represent for some purpose [15]. Therefore, an ontology defines a set of representational terms that we call *concepts.* Inter-relationships among these concepts describe a target world. An ontology can be constructed in two ways, domain dependent and generic. CYC [22], WordNet [23], and Sensus are examples of generic ontologies. For our purposes, we choose a domain-dependent ontology. A domain-dependent ontology provides concepts in a fine grain, while generic ontologies provide concepts in coarser grain. The fine-grained concepts allow us to determine specific relationships among features in images that may be used to effectively classify those images.

Figure 1 illustrates an example ontology for the sports domain [11]. This ontology may be obtained from generic sports terminology and domain experts. The ontology is described by a directed acyclic graph (DAG). Here, each node in the DAG represents a concept. In general, each concept in the ontology contains a label name and feature vector. A feature vector is simply a set of features and their weights. Each feature may represent an object of an image, such as a basketball or baseball. Note also that this label name connected to the feature is unique in the ontology. Furthermore, this label name is used to serve as an association of concepts to images. The concept of football may be further expanded to objects present in a football game (i.e. the features of the concept). For instance, a green field, goalposts, and football players would indicate the image is a football game. Should only one or two of the features common to a football game (as specified in the ontology) be present, a less specific classification of the image would be given. In other words, a more generic concept will be assigned to the image. An image containing only a football would be classified as an image containing a football, not as a football game. Furthermore, the weight of each feature of a concept may not be equal. In other words, for a particular concept some feature may serve as more discriminating as compared to some other; it will be assigned higher weight. For example, in the concept of a game of football the weight of goalpost feature is higher than the weight of the feature, green field.

3.1 Inter-relationships

In Ontologies, concepts are interconnected by means of inter-relationships. If there is a inter-relationship R, between concepts C_i and C_j, then there is also a inter-relationship R' between concepts C_j and C_i. In Figure 1, inter-relationships are represented by labeled arcs/links. Three kinds of inter-relationships are used to create our ontology: IS-A, Instance- Of, and Part-Of. These correspond to key abstraction primitives in object-based and semantic data models [1].

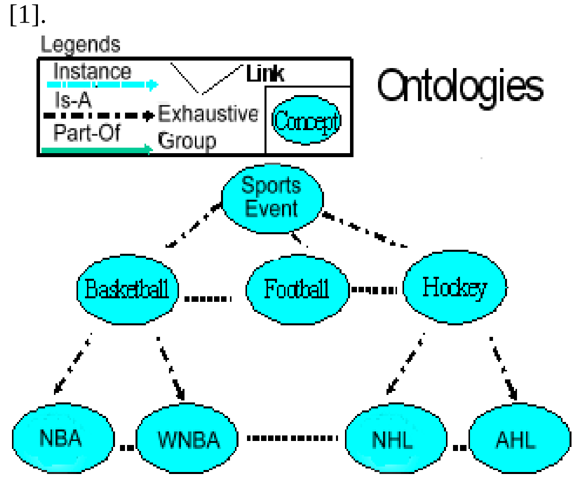


Figure 1. A Portion of an Ontology for the Sport Domain

IS-A: This inter-relationship is used to represent concept inclusion. A concept represented by C_j is said to be a IS-A inter-relationship between C_i and C_j if it goes from generic concept C_i to specific concept, C_j represented by a broken line. Specialized concepts inherit all the properties of the more generic concept and add at least one property distinguishes them from their generalizations. For example, "NBA" inherits the properties of its generalization, "Professional" but is distinguished from other leagues by the type of game, skill of participant, and so on.

Instance-Of: This is used to show membership. A C_j is a member of concept C_i. Then the inter-relationship between them corresponds to an Instance-Of denoted by a dotted line. Player, "Wayne Gretzky" is an instance of a concept, "Player." In general, all players and teams are instances of the concepts, "Player" and "Team" respectively.

Part-Of: A concept is represented by C_j is Part-Of a concept represented by C_i if C_i has a C_j (as a part) or C_j is a part of C_i. For example, the concept "NFL" is Part-Of "Football" concept and player, "Wayne Gretzky" is Part-Of "NY Rangers" concept. Once the concepts have been fully identified in an ontology they may be used to draw a meaningful conclusion about an image based on its content. Objects identified by the neural network are used to develop relationships. These relationships specify useful information that is used to accurately classify a sample image.

4. PROPOSED SYSTEM

Our system combines the use of ontologies and neural networks as object identifiers to provide a high level of precision in the automatic classification of an image based on its content. This system circumvents the low precision classification techniques of other systems by examining the actual objects within an image and using them to discover relationships that reveal information useful in classifying the entire image. The concepts behind these relationships are held in our knowledge base of domain-dependant ontologies as described in section 3. Before feeding to ontologies or neural network, object boundaries are required to be identified in images. We now outline the steps taken to successfully process and classify an input image presented to our system.

4.1 Our Approach

In our system we would like to address two distinct questions: the extraction of the semantic concepts from the images and the construction of an ontology. With regard to the first problem, the extraction of semantic concepts, the key issue is to identify appropriate concepts that describe and identify images. We propose an automatic mechanism for the selection of these concepts [3]. In ontologies each concept is described by a set of features (objects). To select concept(s) for each image, we need first to identify object boundaries. For this, an object detection algorithm (box 1 in Figure 2) is invoked. In this paper we only address the problem of the extraction of object boundary (see section 5). However, we will briefly touch upon some other issues.

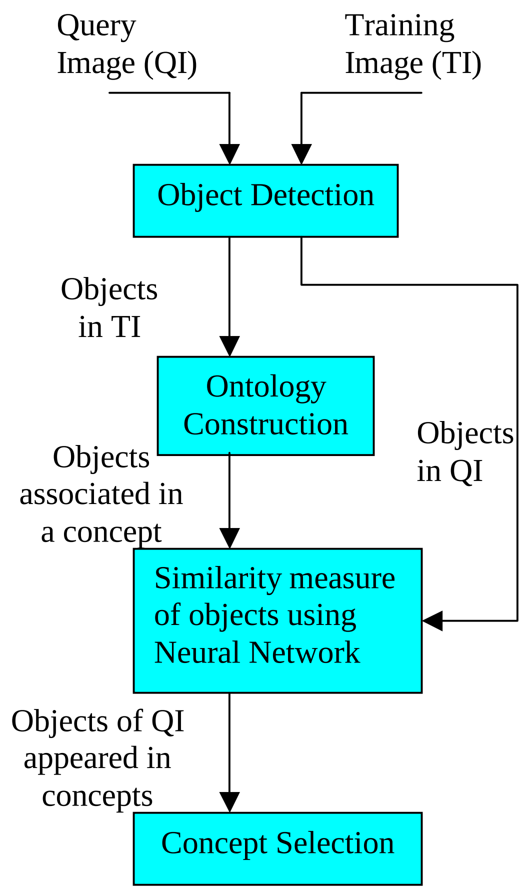


Figure 2. Flow of Our System

After identifying object boundaries in a query image to select concepts from ontologies, we identify objects that appear in the image using neural networks (box 3 in Figure 2) [4]. Neural networks prove to be an effective method used to automatically find a wide range of patterns in sample data [8]. Given a large amount of input data to work with, a neural network can automatically find the most dominant patterns of information. In most cases, a neural network takes an input vector and maps it onto an output pattern. The result is similar to a black box that takes an input and produces the desired output. In the case of a neural network, the inside of this black box is actually a set of adjustable weights, each of which is applied to the input data in an attempt to map this data to the correct output. The ability of a neural network to map an input image to a specified output category makes neural networks a popular method for object identification.

After the objects have been identified, their identifications are fed into a concept selection module (box 4 in Figure 2). The ontologies use this information to provide a meaningful description of the image by selecting concepts based on image content (i.e., individual objects within the image). Our concept selection mechanism includes a novel, scalable disambiguation algorithm using a domain specific ontology. This algorithm will prune irrelevant concepts while allowing relevant concepts to become associated with images [3].

With regard to the second problem, we would like to build ontologies automatically (box 2 in Figure 2). This will be part of future work. For this, we will rely on a self-organizing tree (SOTA) that constructs a hierarchy from top to bottom [21]. To construct the tree we need to measure similarity between images. We would like to propose similarity between images based on the objects appeared in images similar to vector space model. Furthermore, each object in an image will be treated as a keyword along with its weight.

5. IMAGE SEGMENTATION

First, several pre-processing steps must be carried out to prepare the individual objects as input into the neural network. One of them is image segmentation. In our approach image segmentation process has three steps. First, we need to extract color edges from areas of different color. Second, based on the color edges we discovered in step one, we divide the image into several sub-regions by using region-growing techniques. In the final step, adjacent regions having the similar colors are merged together.

5.1 Edge Detection

In our method, we use the I color space [33]. Edge pixels are discovered by values of intensity, hue and saturation. So, at first, we need to apply color conversion to transform all image pixels from the RGB color space to the I space. I, H and S stand for the value of intensity, hue and saturation correspondingly.

1	2	1
0	0	0
-1	-2	-1

HOE

1	0	-1
2	0	-2
1	0	-1

VOE

2	1	0
1	0	-1
0	-1	-2

NOE

0	1	2
-1	0	1
-2	-1	0

SOE

(x-1, y-1)	(x-1, y)	(x-1,y+1)
(x, y-1)	(x, y)	(x, y+1)
(x+1,y-1)	(x+1,y)	(x+1,y+1)

Figure 3: IHS Definitions.

In Figure 3, HOE, VOE, NOE, SOE stand for horizontal, vertical, northeast diagonal and northwest diagonal edge patterns respectively.

Using Figure 3 as a guide, we make the following definitions to carry out our calculations,

$$HOE(x, y)_I = | I(x-1,y-1) + 2I(x,y-1) + I(x+1,y-1) - I(x-1,y+1) - 2I(x,y+1) - I(x+1,y+1) |$$
$$VOE(x, y)_i = | I(x-1,y-1) + 2I(x-1,y) + I(x-1,y+1) - I(x+1,y-1) - 2I(x+1,y) - I(x+1,y+1) |$$
$$NOE(x, y)_i = | I(x,y-1) + 2I(x-1,y-1) + I(x-1,y) - I(x+1,y) - 2I(x+1,y+1) - I(x,y+1) |$$
$$SOE(x, y)_i = | I(x,y-1) + 2I(x+1,y-1) + I(x+1,y) - I(x-1,y) - 2I(x-1,y+1) - I(x,y+1) |$$
$$MOE(x, y)_i = \max \{HOE(x, y)_i , VOE(x, y)_i, NOE(x, y)_i, SOE(x, y)_i \}$$

If $MOE(x, y)_I$ is greater than a threshold T_I, the pixel (x, y) is an edge pixel [7]. Similarly, we use the same method to find values for H and S. If the value of MOE for H and S is more than threshold T_H and T_S correspondingly, the pixel (x, y) is also an edge pixel. The three thresholds discussed above are determined through experimentation. They may be adjusted to achieve better edge detection result. The pseudo code of edge detection is as follows.

```
Read image and save it in a two dimensional array
Pixel[imageWidth][imageHeight]
 for (int y = 0; y < imageHeight; y ++) {
     for (int x = 0; x < imageWidth; x ++) {
          if ( (MOE(x, y)_I > T_I) OR (MOE(x, y)_h > T_H)
OR (MOE(x, y)_s > T_S) )
```

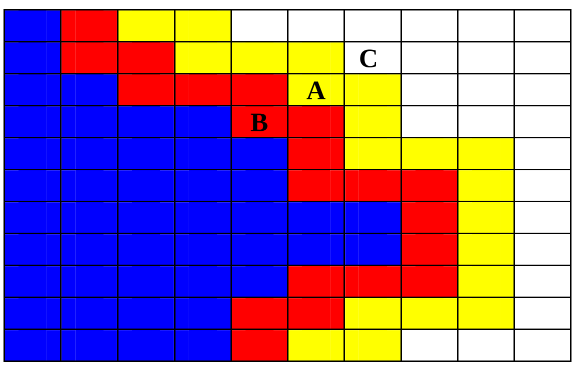
```
            Pixel[x][y] is an edge pixel
        else
            Pixel[x][y] is an region pixel
    }
}
```

Figure 4. Pseudo code for Edge detection

After edge detection, all image pixels are divided into two sets; the edge pixel set (EPS) and the region pixel set (RPS). We move on to the region growing calculations.

5.2 Region Growing

The detected edges cut the image into a set of regions. We pick a pixel from the RPS randomly as a seed for a new region, R_i. During region growing of R_i, all pixels in this region are moved out from the RPS and are assigned to this newborn region. After this region is fully grown, if the RPS is not empty, the algorithm simply picks a pixel randomly as a seed for another new region. This process continues until all pixels in the RPS are placed in a set of regions.



& Pixels in the growing region R

& Pixels not assigned yet

Boundary pixels of the region R

Outer neighbor pixels of the region R

Figure 5. Region growing

The growth of the regions must satisfy certain criteria. If the criteria cannot be satisfied, the growth in the given direction will be stopped. A. Trémeau et al. introduced three criteria for region growing, one local homogeneity criterion (LHC) and two average homogeneity criteria (AHC) [34]. We define p as the pixel to be processed, R is the set of pixels in the current region (possibly not fully-grown) and V is the subset of pixels from the current region which are neighbors to p. LHC states the color differences between p and its neighbors in R is sufficiently small. AHC1 states that the color difference between p and the mean of the colors in V is sufficiently small. AHC2 states that the color difference between p and the mean of the colors in R is sufficiently small. Each of the 3 criteria must be satisfied for p to be merged into R.

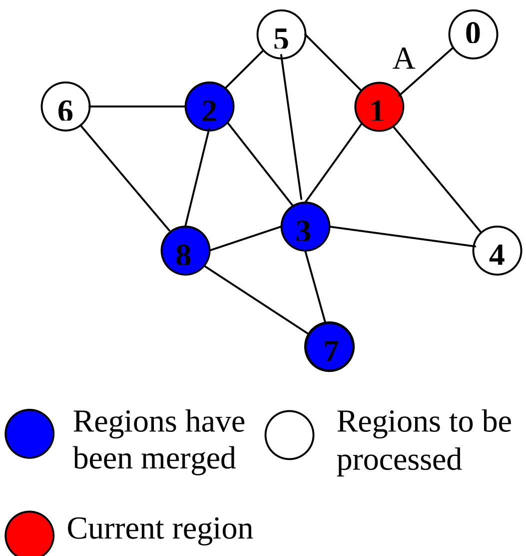
Growth of a region is as follows. First, the seed pixel is the only pixel that the region R has. Pixels of R are fallen into two categories such as boundary pixel (BP) and inner pixel (IP). A pixel is boundary pixel if at least one pixel among its 8 neighbor pixels is not in the region it belongs. On the other hand, a pixel is inner pixel if all its 8 neighbor pixels are in the region it belongs. At the beginning, the seed pixel is the only boundary pixel of the region. Next, we check the availability of 8 neighbor pixels of this boundary pixel. A pixel is available only when it is contained in RPS. This means the pixel is not an edge pixel and has not been assigned to some other region yet. If any of these pixels is available and satisfies the criteria, the pixel is qualified to be a member of R. After addition of a pixel into region R, it will be a new boundary pixel of the region. The inner pixels and boundary pixels of the region are also required to update. For example, in Figure 5, after adding pixel A into region R, A will be a new boundary (red) pixel. Pixel C will be a current neighbor (yellow) pixel of boundary pixel, A. Thus, pixel B is not a boundary pixel any more and will be an inner (blue) pixel. Based on these two characteristics, we keep checking and updating boundary pixels until the region stops to extend. Then, we can say the region is fully grown. The pseudo code is as follows.

```
int i = 0;
while (RPS is not empty ) {
        i ++;
        pick a pixel from RPS randomly as a seed
and assign it to new set R_i
         for each boundary pixel(r) of R_i {
            for each neighbor pixel(n) of r that is
not in BP and IP
                  if (LHC and AHC are satisfied for n)
                  { Move the pixel, n from RPS to R_i;
                    Update RPS and R_i; }
        }
}
```

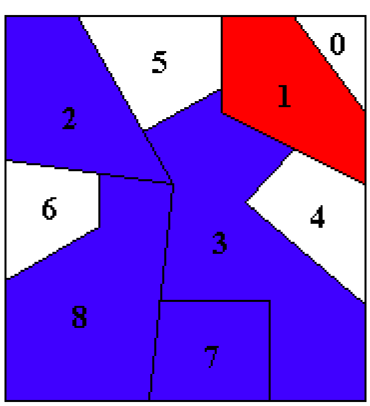
Figure 6. Pseudo Code for Region Growing

5.3 Merging Adjacent Regions

We still encounter several shortcomings. First, it is possible to achieve some noise regions which may not be the true region. Second, it is still possible to cut one object into several sub regions even if it has a unique color. For example, a basketball could be divided into several sub regions due to its black lines (see second image of Figure 10). Intuitively, these two problems can be solved by merging adjacent regions. At first, we need to construct a region adjacency graph (RAG) based on regions [34]. In a RAG each vertex represents a sub region. An edge will appear to connect the two vertices, which stand for two adjacent regions. (Shown in Figure 7) The edges are weighted by color difference between these two regions.



(a)



(b)

Figure 7. Region Adjacency Graph

To construct RAG, we have to know whether any two given regions are adjacent or not. Two following approaches can be used.

5.3.1 Minimum Bounding Rectangle Technique (MBRT)

In this approach, minimum bounding rectangle has been constructed [35]. Two regions are considered to be adjacent to each other if their minimum bounding rectangles overlap. Minimum bounding rectangle of a region not only encompasses the region but may also surround some regions which may contribute false positive (not true adjacent regions).

5.3.2 Matrix Oriented Technique (MOT)

Here we keep a two dimensional matrix where each cell corresponds to a pixel. Furthermore, content of the cell corresponds to a region index where the pixel belongs. Note that for edge pixel we have a special treatment: -1 will be used as a region index. To find adjacent regions, we simply scan matrix row-by-row and column-by-column. For example, in Figure 8, each gray pixel labeled by –1 is edge pixel, other pixels are region pixels and the number indicates the region index in which the pixel belongs to.

-1	-1	5	5	5	5	-1	3
2	-1	5	5	5	5	-1	3
2	-1	-1	5	5	-1	-1	3
2	2	-1	5	5	-1	3	3
2	2	-1	5	5	-1	3	3
2	2	2	-1	-1	3	3	3
2	2	2	2	2	-1	-1	3
2	2	2	2	2	2	-1	3
2	2	2	2	2	2	2	-1

(a)

2	-1	5	5	5	5	5	-1
2	-1	5	5	5	5	-1	-1
2	-1	5	5	5	5	-1	3
2	-1	5	5	5	5	-1	3
2	-1	-1	-1	-1	-1	-1	3
2	-1	4	4	4	4	-1	3
2	-1	4	4	4	4	-1	3
2	-1	4	4	4	4	4	-1
2	-1	4	4	4	4	4	-1

(b)

Figure 8. Examples of Adjacent Regions Detection

When we scan through the matrix row by row and column by column, and if the region index changes from a to b (say), we can say that the region a is adjacent to region b. For example, when we scan the first row in Figure 8(a), we know that region 5 and 3 are adjacent to each other. When we scan the seventh column in Figure 8(a), we know region 3 and 2 are adjacent. This method is easy to implement and the computation complexity is O(n). On the other hand, MOT has a shortcoming. In some special cases, it may detect regions adjacent wrongly. For example, in Figure 8(b), when we scan the fifth row in the matrix, region 2 and 3 are declared as adjacent. However, these two regions are separated by six edge pixels. Now, the issue will arise such as: What is the maximum number of edge pixels used as a separator to determine that two regions are adjacent? This threshold depends on the edge detection result and the region size scale.

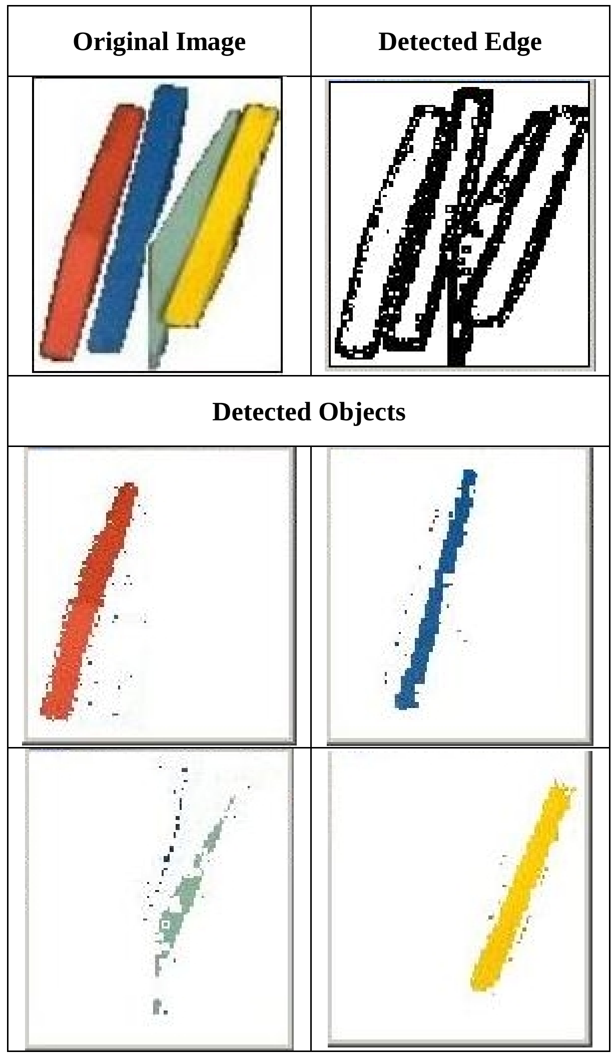
With regard to the first problem (i.e., noise region), based on the adjacency graph, first we identify noise regions. If a region only contains a small number of pixels, we declare this region is a noise region. For this, we merge the noise region to one of its neighbor regions that has smallest color difference. With regard to the second problem (i.e., over segmentation of sub regions), we merge adjacent regions by using a modified minimum spanning tree algorithm (MMSTA). In the MMSTA a threshold t_w is defined (see Figure 9). Furthermore, a tree will be constructed by adding an additional constraint: weight of each edge in the tree will fall below t_w.. All regions in the tree compose an object. This is because color difference between a region and all its neighbor regions in the tree falls below t_w.

Calculate average color value for each R_i;
Construct a RAG;
Define T_w;
Sort all edges;
while (still have edges and vertex not added in the tree) {
For each edge in order, test whether it creates a cycle in the tree we have thus far built or the weight is more than T_w –
if so
discard;
else
add to the tree.
}

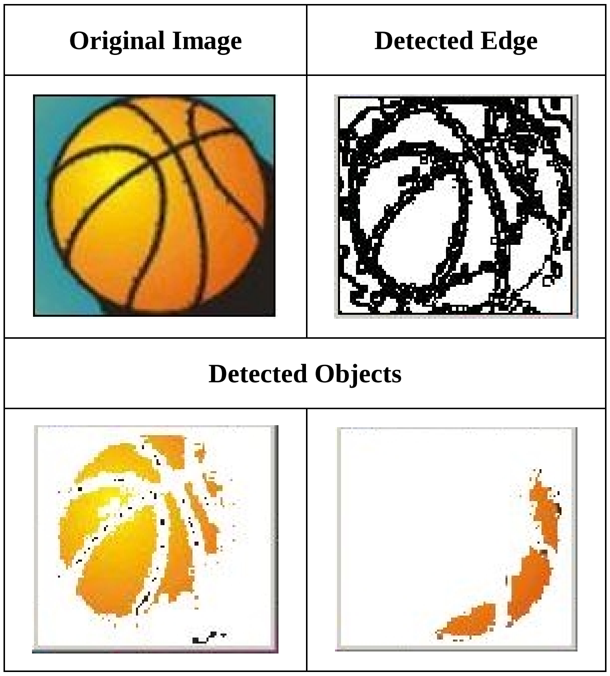
Figure 9. Pseudo Code for Merge Adjacent Regions

6. EXPERIMENTAL PRELIMINARY RESULTS

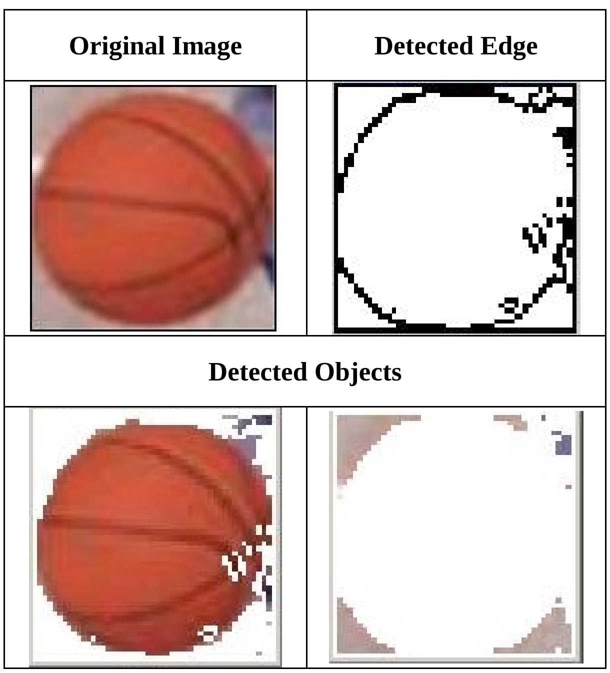
The object detection algorithm was tested using sample images found on the Internet. Here we reported results for only 4 images due to space limitations. These four images consist of varying degree of complex objects. The first image consists of 4 simple objects. The second and third images consist of basketball objects along with a set of lines. The fourth image consists of net, and player. Figure 10 shows these 4 images and displays detected objects. For each image, the original test images and edge detection results are shown first; and then all major detected objects are displayed.



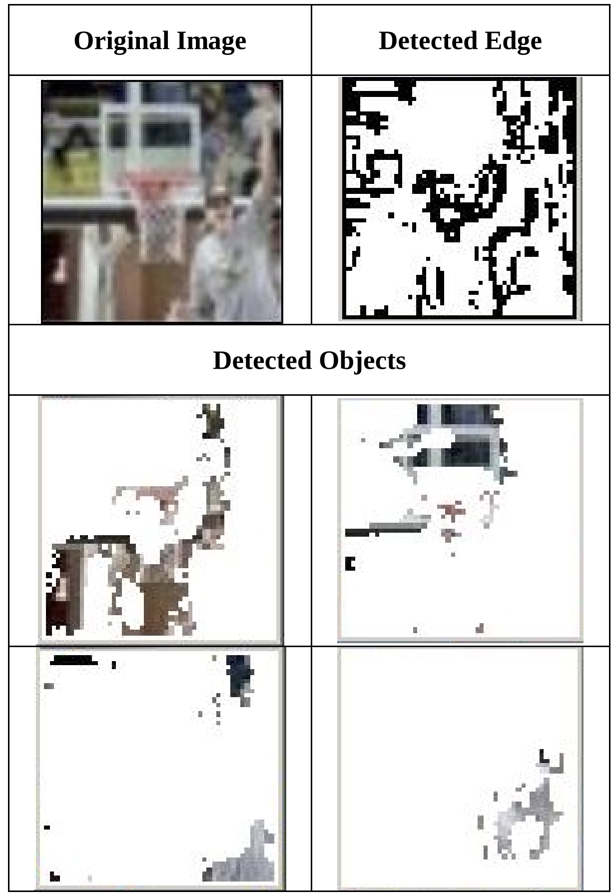
(a)



(b)



(c)



(d)

Figure 10. Image Segmentation Results

In the first image, each object has a unique color. We detected the four major objects correctly. The second and third images are more complicated, but the color distribution of the object is still simple, so the test results are also satisfactory. In the third image, objects are correctly classified. On the other hand, in the second image regions are correctly identified. However, merging adjacent regions algorithm fails to merge adjacent regions due to substantial change of hue property. Therefore, rather than unified one object two splitted objects are shown. Note that in the fourth image our algorithm fails to detect all objects correctly due to the presence of too many objects along with varying color.

7. CONCLUSIONS AND FUTURE WORKS

The success of ontology-based image classification model entirely depends on the detection of object boundaries. We have proposed an automatic scalable object boundary detection algorithm based on edge detection, and region growing techniques. We have also proposed an efficient merging algorithm to join adjacent regions using adjacency graph to avoid over segmentation of regions. To illustrate the effectiveness of our algorithm in automatic image classification, we implement a very basic system aimed at the classification of images in the sports domain. By identifying objects in images, we have shown that our approach works well when objects in images have less complex organization. We would like to extend the work in the following directions. First, we would like to build ontologies automatically based on object similarity. Next, we will update weight of objects automatically appeared in images.

REFERENCES

[1] G. Aslan and D. McLeod, "Semantic Heterogeneity Resolution in Federated Database by Metadata Implantation and Stepwise Evolution", *The International Journal on Very Large Databases*, Vol. 18, No. 2, October 1999.

[2] R. Barber, W. Equitz, C. Faloutsos, M. Fickner, W. Niblack, D. Petkovic, and P. Yanker, "Query by Content for Large On-Line Image Collections", *IEEE Journal*, 1995.

[3] C. Breen, L. Khan, Arun Kumar and Lei Wang, "Ontology-based Image Classification Using Neural Networks," to appear in *SPIE*, Boston, MA, July 2002.

[4] C. Breen, L. Khan and Arun Kumar, "Image Classification Using Neural Networks and Ontologies," to appear in *IEEE DEXA, International Workshop on Web Semantics*, France, Sept 2002.

[5] M. A. Bunge, "Treatise on Basic Philosophy: Ontology: The Furniture of the World", Reidel, Boston, 1977.

[6] S. F. Chang, J. R. Smith, "Extracting Multi-Dimensional Signal features for Content-Based Visual Query", in Proc. of *Visual Communications and Image Processing '95*, SPIE Volume 2501, pp. 995-1006, ed. T. Wu Lance, Bellingham, WA: The International society for Optical Engineering, 1995.

[7] L. H. Chen, S. Chang, "Learning Algorithms and Applications of Principal Component Analysis", Image Processing and Pattern Recognition, Chapter 1, C. T. Leondes, Academic Press, 1998.

[8] J. E. Dayhoff, "Neural Network Architectures An Introduction", VNR Press, 1990.

[9] C. Djeraba, "When Image Indexing Meets Knowledge Discovery", in Proc. of *The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, Boston, MA, August 2000.

[10] Joaquin Dopazo, Jose Maria Carazo, "Phylogenetic Reconstruction using an unsupervised growing Neural Network that adopts the Topology of a Phylogenetic Tree", *Journal of Molecular Evolution*, Volume 44, pp. 226-233 1997.

[11] ESPN CLASSIC, http://www.classicsports.com.

[12] Fritzke, Bernd, " Growing cell structures - a self-organizing network for unsupervised and supervised learning", Neural Networks, Volume 7, pp. 1141-1160 1994.

[13] Y. Gong and H. J. Zhang, "An Effective Method for Detecting Regions of Given Colors and the Features of the Region Surfaces", in Proc. of *Symposium on Electronic Imaging Science and Technology: Image and Video Processing II*, pp. 274-285, San Jose, CA, February 1994, IS&T/SPIE.

[14] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications Knowledge Acquisition," *An International Journal of Knowledge Acquisition for Knowledge-based Systems*, Volume 5, no. 2, June 1993.

[15] N. Guarino, C. Masolo, and G. Vetere, "OntoSeek: Content-based Access to the Web," *IEEE Intelligent Systems*, Volume 14, no. 3, pp. 70-80, 1999.

[16] N Ito, Y. Shimazu, T. Yokoyama, and Y. Matushita, "Fuzzy Logic Based Non-Parametric Color Image Segmentation with Optional Block Processing", in Proc. of *ACM*, 1995.

[17] A. K. Jain, "Fundamentals of Digital Image Processing", Prentice Hall, Englewood Cliffs, NJ, 1989.

[18] L. Khan, "Structuring and Querying Personalized Audio using Ontologies," in Proc. of *ACM Multimedia*, vol. 2, pp. 209-210, Orlando, FL, Nov 1999.

[19] L. Khan and D. McLeod, "Audio Structuring and Personalized Retrieval Using Ontologies," in Proc. of *IEEE Advances in Digital Libraries, Library of Congress*, pp. 116-126, Bethesda, MD, May 2000.

[20] L. Khan and D. McLeod, "Efficient Retrieval of Audio Information from Annotated Text Using Ontologies," in the Proc. of *ACM SIGKDD Workshop on Multimedia Data Mining*, Boston, MA, August 2000.

[21] T. Kohonen, "Self -Organizing Maps", Second Edition, Springer 1997.

[22] D. B. Lenat, "Cyc: A Large-scale investment in Knowledge Infrastructure", *Communications of the ACM*, pp. 33-38, Volume 38, no. 11, Nov 1995.

[23] G. Miller, "Wordnet: A Lexical Database for English", in Proc. of *Communications of CACM*, Nov 1995.

[24] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, G. Taubin, “The QBIC Project: Querying Images by Content Using Color, Texture, and Shape”, in Proc. of *Storage and Retrieval for Image and Video Databases*, Volume 1908, pp. 173-187, Bellingham, WA, 1993.

[25] A. Pentland, R.W. Picard, S. Sclaroff, “Photobook: Tools for Content-Based Manipulation of Image Databases”, in Proc. of *Storage and Retrieval for Image and Video Databases II*, Volume 2185, pp. 34-47, Bellingham, WA, 1994.

[26] N. Row, and B. Frew, “Automatic Classification of Objects in Captioned Depictive Photographs for Retrieval”, Intelligent Multimedia Information Retrieval, Chapter 7, M. Maybury, AAAI Press, 1997.

[27] A. F. Smeaton and A. Quigley, “Experiments on Using Semantic Distances between Words in Image Caption Retrieval,” in Proc. of *The Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1995.

[28] J. R. Smith, S. F. Chang, “Automated Binary Texture Feature Sets for Image Retrieval”, in Proc. of *The International Conference On Acoustic Speech and Signal Processing (ICASSP)*, pp. 2241-2244, Atlanta, GA, 1996.

[29] J. R. Smith, S. F. Chang, “Tools and Techniques for Color Image Retrieval”, in Proc. of *The Symposium on Electronic Imaging: Science and Technology Storage and Retrieval for Image and Video Databases IV*, pp. 426-437, San Jose, CA, 1996.

[30] M. J. Swain, D. H. Ballard, “Color Indexing”, *International Journal of Computer Vision*, 7(1), pp. 11-32, 1991.

[31] D. Tseng and C. Chang, “Color Segmentation Using Perceptual Attributes”, in Proc. of *11th International Conference on Pattern Recognition*, pp. 228-231, Amsterdam, Holland, September 1992, IAPR, IEEE.

[32] S. Wong and W. K. Leow, “Color Segmentation and Figure-Ground Segregation of Natural Images”, *IEEE Journal*, 2000.

[33] D. C. Tseng and C. H. Chang, “Color segmentation using perceptual attributes,” In Proc. of *11th International Conference on Pattern Recognition*, pages 228-231, Amsterdam, HOLLAND, September 1992. IAPR, IEEE.

[34] A. Trémeau and P. Colantoni, “Regions adjacency graph applied to color image segmentation,” *IEEE Transactions on Image Processing*, 1998.

[35] S. Wong and W. Leow, “Color segmentation and figure-ground segregation of natural images,” in *Proc. Int. Conf. on Image Processing (ICIP 2000)*, volume 2, pages 120--123, 2000.

[36] Y. Deng, B.S. Manjunath, and H. Shin, "Color image segmentation", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1999.

[37] C. Frankel, M.J. Swain and V. Athitsos, “WebSeer: An Image Search Engine for the World Wide Web,” *University of Chicago Technical Report TR-96-14*, July 31, 1996.

[38]. Chakrabarti, K., Ortega-Binderberger, M., Porkaew, K & Mehrotra, S. (2000) Similar shape retrieval in MARS. Proceeding of IEEE International Conference on Multimedia and Expo.

[39]. G. Lu and A. Sajjanhar, Region-based shape representation and similarity measure suitable for content-based image retrieval. Springer Verlag Multimedia Systems, 1999.

[40]. Ricardo Baeza-Yates, Berthier Ribeiro-Neto, Modern Information Retrieval, ISBN 0-201-39829-X, 1999.