Image Classification Using Neural Networks and Ontologies*

Casey Breen
Dep. of Computer Science,
University of Texas at Dallas
casey@utdallas.edu

Latifur Khan
Dep. of Computer Science,
University of Texas at Dallas
lkhan@utdallas.edu

Arunkumar Ponnusamy
Dep. of Computer Science,
University of Texas at Dallas
axp014210@utdallas.edu

Abstract

The advent of extremely powerful home PCs and the growth of the Internet have made the appearance of multimedia documents a common sight in the computer world. In the world of unstructured data composed of images and other media types, classification often comes at the price of countless hours of manual labor. This research aims to present a scalable system capable of examining images and accurately classifying the image based on its visual content. When retrieving images based on a user's query, the system will yields a minimal amount of irrelevant information (high precision) and insure a maximum amount of relevant information (high recall).

1. Introduction

The unstructured format of digital images tends to resist standard categorization and classification techniques. Traditional systems used to store and process multimedia images provide little to no means of automatic classification. Existing image storing systems such as QBIC [13] and VisualSEEK [14] limit classification mechanism to describing an image based on metadata such as color histograms [16], texture, or shape features. 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 description of an image based on the image's content. To provide automatic classification of images, our system uses neural networks together with domain-dependant ontologies [2, 7, 8].

We begin by using a neural network to classify objects from an image. Neural networks prove to be useful in applications ranging from medical imaging [15], to computer learning algorithms used in artificial intelligence [4]. The network takes an image as input and gives it a classification as output. The ontology then processes the classified output, discovering relationships among objects that can be used to provide semantic meaning to the entire image. Historically ontologies have been employed to achieve better precision and recall in text retrieval systems [6]. Ontologies may be described as collections of concepts and their interrelationships regarding a specific domain [3, 6]. By determining the relationship among a set of concepts, additional information can be deduced allowing retrieval systems to find relevant documents containing none of the words used in the original query [10]. Though ontologies poses the ability to extract additional relevant information from textual documents, no attempts have been made to transfer this method to the realm of image classification.

The proposed system joins neural networks and ontologies in an effort to provide automatic classification of images in the sports domain. Images are examined by processing individual objects according to the rules held in specific concepts present in the domain-dependant ontology. The system implements an automatic concept selection mechanism for images including a scalable disambiguation algorithm. This algorithm prunes irrelevant concepts while allowing relevant concepts to associate with images. These concepts create a domain-specific ontology that facilitates high precision and high recall in the area of image classification and retrieval.

Section 2 of this paper describes ontologies and how they are used to specify interrelationships among concepts that helps extract semantic meaning from images. Section 3 outlines the steps carried out by the neural network as it process an image, as well as the interaction between the neural networks and the domain-dependent ontology. Finally, section 4 details the neural network's ability to correctly classify objects, and presents the results of the completed system's ability to accurately discover semantic meaning based on relationships between objects in an image.

2. Structure of ontologies

An ontology is a specification of an abstract, simplified view of the world that we wish to represent for some purpose [3]. Therefore, an ontology defines a set of representational terms that we call concepts. Inter-relationships among these concepts describe a target

* This research was supported by NSF grant NGS-0103709

world. An ontology can be constructed in two ways, domain dependent and generic. CYC [9] and WordNet [11, 12], 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.

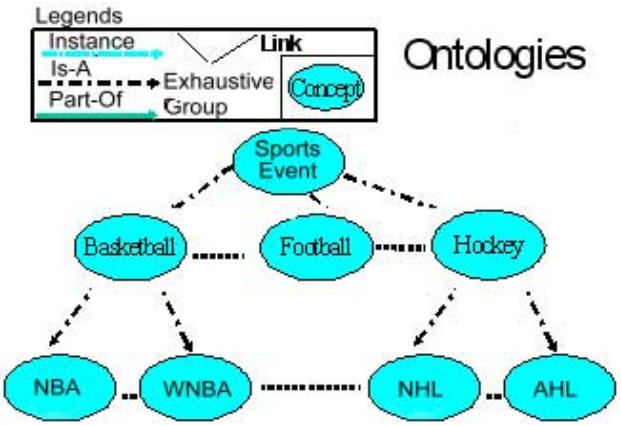

Figure 1: Sample ontology from the sports domian

Figure 1 illustrates an example ontology for the sports domain. The ontology is described by a directed acyclic graph (DAG). 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 represents an object of an image, such as a basketball or baseball. Note also that the label name connected to a feature is unique in the ontology, serving 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. Furthermore, the weight of each feature of a concept may vary. A particular feature may have more importance in one concept than another, thus deserving more weight in that concept.

2.1 Ontology 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].

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 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.

3. 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. We now outline the steps taken to successfully process and classify an input image presented to our system.

3.2 Neural network specifications

Our system employs a neural network that classifies objects into pre-defined output categories. This type of system is known as a supervised classifier [5]. These networks take an image (or in our case an object from an

image) as input and place it in a certain category as output. The system uses an image segmentation algorithm to find the individual objects present in an image [2]. The network developed for our system uses the hue value of each pixel in the segmented object as input. We chose the hue value because this value represents the most information about a pixel's color. Giving the network information on an object's color distribution allows it to find patterns relating to areas of similar color, and patterns relating to areas of similar shape based on the object's edges. The neural network itself learns the shape and color of each object by adjusting its weights as it processes the training data.

The hue values are given as input to the network in vector form. A XxY pixel object would results in an input layer with X nodes. Each node would receive an input vector of Y values. The height of the image represents the number of input nodes, while the length represents the size of the vector each input node receives. For testing purposes, an input image size of 50x50 pixels was selected. This size reduced the dimensionality of the input object to a reasonable quantity, while still keeping the visual content of the object in tact. Each node then maps the given input vector to an output vector of size x. The number of feature detectors and the number of iterations used to train the network were determined from experimental results, and are covered in section 4. Two output nodes were chosen for this network, with each one having a vector size of Y elements (based on the width of the training data). In the case of a network used to identify a basketball, a basketball based input image is mapped to an output vector of all 1's. Training data representing a non-basketball image is mapped to a value of 0.

Two output nodes were chosen to allow the network to have a greater ability to adjust its weights after processing each test image. After successfully training the network, the system may be used to classify test images of segmented objects representing (for example) basketballs and non-basketballs. By having the two classification categories map to the two extremes of output values allowable by the network's sigmoid function (0 and 1), a threshold value of .5 can be used to determine if an image is a basketball based its output vector values. When the network processes a test image, it takes the average sum of the vector values. If this average is greater than .5, the image may be classified as a basketball. If the value is less than .5, it may be classified as a non-basketball.

3.3 Concept selection

After the network identifies a set of objects from an input image, these objects may be used to select concept(s) from ontologies. Recall that each concept in an ontology contains a set of features (objects) and weights.

$$\sum_{i=0}^{x} Ci = \left(\left(\sum_{j=0}^{y} Wi, j \right) \geq Ti \right) \quad (1)$$

The objects from the sample image are processed using equation 1 to determine if the image may be classified by a concept from the ontology. The variables of the equation are defined as follows:

x = number of concepts present in the ontologies

C_i = for concept C_i in the ontology this value will be 1 (input image is classified as concept C_i) or 0 (input image is not classified as concept C_i)

y = number of objects from input image identified by the neural networks

Wx,y = weight given to object j when associated with concept C_i. If object j does not exist in C_i, the weight is 0

Ti = threshold used to determine if the input image may be classified as concept C_i. (Set through experimentation, see section 4)

After applying Equation 1, it is possible the sample image has been classified as belonging to several different concepts in the ontology. For example, an input image may be classified as both a basketball game and an NBA basketball game. It is also possible that a set of concepts may be selected where some of them may be wrongly associated. For instance it is possible an image may be classified as both an NBA basketball game and a college basketball game at the same time. However, we can employ the following heuristic-based pruning techniques to narrow down the selection of concepts [17]:

- The parent and children concepts are selected so that the parent concept has the higher rank. This allows the child concept to be discarded. The rationale behind this is that the most generic concept will share a subset of features with specific concepts.
- In a case in which parent and children concepts are selected so that parent concepts have a lower rank than children, the children concept with the highest rank will be kept and the parent concept can simply be discarded. The rationale behind this is that the parent concept might share some common feature with children.

When the pruning algorithm completes the selected concepts will be sorted based on their ranking in descending order. Thus, in the example of an image being classified as both a NBA game and a college basketball game, the image would ultimately be assigned the more generic label of a basketball game.

4. Experimental Results

The following section presents the results of training the system to recognize and apply relevant concepts to sample images from the sports domain. This section

begins with experimental results created by the basketball neural network. We then conclude the paper with detailed results of the completed system's ability to apply relevant concepts to sample images from the sports domain.

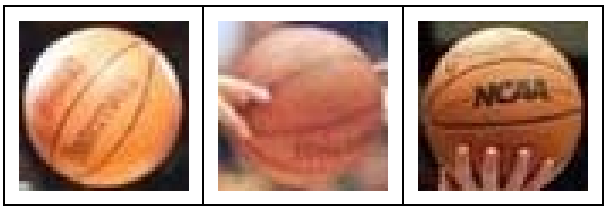

Figure 2: Images used to train basketball network

The basketball neural network was trained using sample images found on the Internet. Of the sample images, 65% were used to train the network while 35% were used to test the trained network. To successfully train a neural network to identify an object, we include 3 types of training data. Figure 2 illustrates several of the images used in the first category of training data, basketball images (for clarity and brevity we only mention one object/concept). Images were chosen based on varying shades of orange, and varying background colors present in the minimum bounding rectangle surrounding each ball. By including a large amount of sample basketball images, each with varying shades and backgrounds, we create a more robust system capable of successfully identifying a wide range of basketball images.

Figure 3: Various objects specified in the ontology

Figure 3 illustrates several of the images used in the second category of training data, objects present in the domain-dependant ontology used by the network. These objects train the basketball networks to recognize non-basketball images. Other objects present in the ontologies include baseballs, soccer balls, and basketball courts. Each of these objects has their own neural network used for its identification by the system. Since these objects will appear often in the sports domain, the basketball-based network must recognize that they are not basketballs.

The final category of data includes noise images used to train the network. Noise images consist of various segmented regions that may exist in a sports image, but do not contain a concept in the ontology. Including these regions helps the system converge on a set of weights that successfully identifies the input image [3]. The presence of these objects in the training data allows the system to be more robust in terms of various types of input.

We show results for the basketball network based on 4 different network configurations. Figure 4 displays setup information and results for each configuration. Each network reached a set of weights capable of correctly identifying the training images after 1,000 iterations. Additional training of the network had no effect on its ability to correctly identify input objects. The network using only 5 feature detectors is excluded. Using 5 feature detectors resulted in a network not capable of storing enough features to correctly identify the training or test images. All networks using more than 10 feature detectors yield roughly the same results, but at the cost of increased training times. By adding more feature detectors, the systems only adds more hidden layer nodes that detect the same pattern.

Num. Feature Detectors	Num. of Iterations	Training Time (ms)	Correct Results	Incorrect Results
10	10000	9691505	92.5%	7.5%
25	10000	26181796	90.0%	10.0%
37	10000	42308868	92.5%	7.5%
50	10000	85052104	92.5%	7.5%

Figure 4: Results of training the basketball network

4.1 Results of combined system

We begin the study of the performance of our concept selection algorithm by considering the percentage of images it can successfully identify. Furthermore, we would like to study the impact of threshold values on pruning irrelevant concepts associated with images, while retaining those which are relevant. We ran the disambiguation algorithms over 15 sample images. Among these images 4 belonged to the basketball, baseball, and soccer concepts, while the final 3 belonged to a category of noise images. The purpose of the noise images is to test the network's ability to classify an image that does not belong to any concept in the domain-dependant ontology. In this implementation, we have assumed that the weight of each feature of a concept is equal. In Figure 5 the X axis represents the value of threshold t, and the Y axis represents the percentage of images associated with correct concepts (category I) and incorrect concepts (category II). It is important to note that category II may contain images that were classified with the correct concept, but were also classified with an additional concept that proved to be incorrect.

For t=0.05, 100% of the images are associated with at least some concepts from the ontology (category I & II).

Among these, 33% of the images are associated with relevant concepts (category I). In addition, 67% of the images are associated with at least one irrelevant concept, and possibly several correct concepts (category II). In this case, precision will be hurt due to the addition of irrelevant concepts.

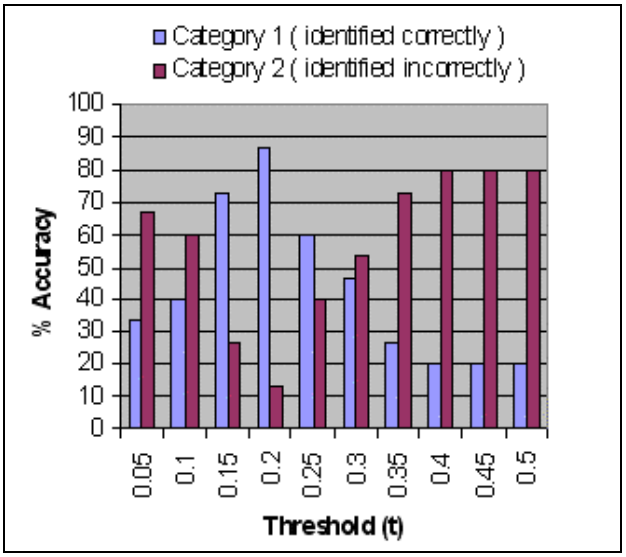

Figure 5: Effect of varying thresholds on concept selection

As the threshold value t is increased, the number of concepts associated with an image begins to decline as well as the number of incorrect concepts applied to the image. For example, with t=0.2, 87% of the images are associated with only relevant concepts (category I) and 13% of the images are associated with at least one irrelevant concept (category II). Unfortunately, as the value of t increases for a given image, we may loose a relevant concepts as we shed those which are irrelevant (t=0.25 and above in Figure 5). Thus, recall will be diminished at the expense of improving precision. The results presented in Figure 5 show that given the correct threshold, the proposed system can efficiently extract meaning from a sample image, and do so with an amazing level of precision and accuracy.

5. 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] C. Breen, L. Khan, A. Kumar and L.Wang, "Ontology-based Image Classification Using Neural Networks" to appear in Proc. of SPIE, Boston, MD, July 2002.

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

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

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

[6] 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.

[7] L. Khan and D. McLeod, "Disambiguation of Annotated Text of Audio Using Onologies," in Proc. of ACM SIGKDD Workshop on Text Mining, Boston, MA, August 2000.

[8] L. Khan and D. McLeod, "Audio Structuring and Personalized Retrieval Using Ontologies," in Proc. of IEEE Advances in Digital Libraries, Library of Congress, Washington, DC, May 2000.

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

[10] Y. S. Maarek, "An Information Retrieval Approach for Automatically Constructing Software Library", IEEE Transactions on Software Engineering, 17(8), 1991.

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

[12] G. Miller, "Nouns in WordNet: a Lexical Inheritance System", International Journal of Lexicography, Volume 3. no. 4, pp. 245-264, 1994.

[13] 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.

[14] 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.

[15] Y. Sun, R. Nekovei. Medical Imaging. From Image Processing and Pattern Recognition, Chapter 1, C. T. Leondes, Academic Press, 1998.

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

[17] E. M. Voorhees, "Implementing Agglomerative Hierarchic Clustering Algorithms for use in Document Retrieval", Information Processing & Management, Volume 22, No.6, pp 465-476 1986.