Learning Semantic Concepts from Visual Data Using Neural Networks

Xiaohang Ma and Dianhui Wang

Department of Computer Science and Computer Engineering,
La Trobe University, Melbourne, VIC 3083, Australia
{x4ma,dhwang}@cs.latrobe.edu.au
http://homepage.cs.latrobe.edu.au/dhwang/

Abstract. For content-based image retrieval techniques, query image is used to pick up and rank some relevant images from a database using some certain similarity metric. If semantic features are not involved in the modeling of visual data, the resulting system may demonstrate a disability of retrieving images likely associated with interesting semantic concepts of objects in the images. Therefore, issues on semantics representation, automatic extraction of semantic concepts from visual data, and effects of window size on the concepts recognition are needed to study. This paper describes an approach towards these problems. We first define a set of semantic concepts characterizing the outdoor images. Then, a neural network is employed to memory the semantic concepts through pattern learning techniques. Lastly, the well-trained neural networks will perform as a classifier to identify the predefined semantics within an image. Empirical studies and comparison with decision tree techniques are carried out.

1 Introduction

In the past decades, content-based multimedia information retrieval techniques have been paid a great attention and developed rapidly due to their extensive applications [1]. Various attempts on building effective content-based information retrieval (CBIR) systems have been made, and some of them have been applied in the commercial domain [1] such as IBM QBIC [3], MIT Photobook [4], Columbia VisualSEEK and WebSEEK [5] and UCSB NeTra [6] etc. Although features associated with the visual data have been widely investigated for better representing the content information, most of the existing CBIR systems have a rudimentary understanding on image content [2]. However, CBIR is the set of techniques for retrieving semantically relevant images rather than feature relevant images from an image database based on automatically derived image features [9]. Those CBIR systems cannot guarantee a meaningful query completion with semantic context concern [7]. On the other hand, the performance of those CBIR systems depends upon the choice of the visual attributes [4], such as color, texture or shape et al [10-16]. No matter how sophisticated they are, color, texture or shape features do not adequately model abstract semantic concepts within images, which is a limit to apply the CBIR systems to broad databases. Therefore we need to design a set of semantic features to bridge the gap between image semantics and pixel representations.

Image semantics classification is a limited form of image understanding, the goal of image classification is not to understand images the way human being do, but merely to assign the image to a semantic class. Minka and Picard [8] introduced a method, which generate many segmentations or regions based on different combinations of features. Although region-based systems aim at decomposing images into constituent objects, it is well known that the image segmentation is almost as difficult as image understanding. In this work, we formulate our semantics extraction task as a local image classification problem using heterogeneous features extracted from block windows and neural networks. One significant and interesting issue associated with this task is the window size effect on the classification system. Like our human being's visual system, an appropriate distance is necessary to well identify details within objects. Neural networks also perform their classification jobs differently as the window size changes. The rest of this paper is organized as follows. Section 2 presents the system description including approach overview, visual data representation, feature selection and the determination of neural network architecture. Section 3 reports and discusses the performance evaluation of the system with a comparison. Conclusion of the work is presented in the last section.

2 System Description

This section describes our methodology for learning semantics using neural networks.

2.1 Overview

The automatic derivation of semantically meaningful information from the content of an image is the focus of interest for most research on image databases. The image "semantics", i.e., the meanings of an image, has several levels. From the lowest to the highest, these levels can be roughly categorized as follows [9]:

1. Semantic types (e.g., landscape photograph, clip art)
2. Object composition (e.g., a bike and a car parked on a beach, a sunset scene)
3. Abstract semantics (e.g., people fighting, happy person, objectionable photograph)
4. Detailed semantics (e.g., a detailed description of a given picture)

To extract semantics, boundary or shape information of objects is important. However it is almost as difficult as semantic learning to get that information. In order to avoid the complex and difficult segmentation job, alternative solution is to use visual features within block windows. It is obvious that a semantic concept can only be drawn or inferred from the feature information carried by the given windows. If the window size is too small, the concept will not be identified at all. On the contrary, if the window size is too big, multiple or mixed concepts may be include. Therefore, it is meaningful and important to select a proper window size for concepts detection from local visual feature vectors.

Firstly, we define the window size as follows:

$$r_l = \frac{l}{L}, \quad r_w = \frac{w}{W} , \tag{1}$$

where r_l is the ratio between the actual block length $\boldsymbol{l}$ and the image length $\boldsymbol{L}$, r_w is the ratio between the actual block width $\boldsymbol{w}$ and the image width $\boldsymbol{W}$. Fig. 1 depicts four windows with the preset window size 20%*20%. From these windows, we can recognize some semantics, for example, mountain in W1, sky in W2 and tree in W3. For window W4, it contains three different semantics, i.e., mountain, sky and tree. In such a case we label the class or semantics as “unknown”.

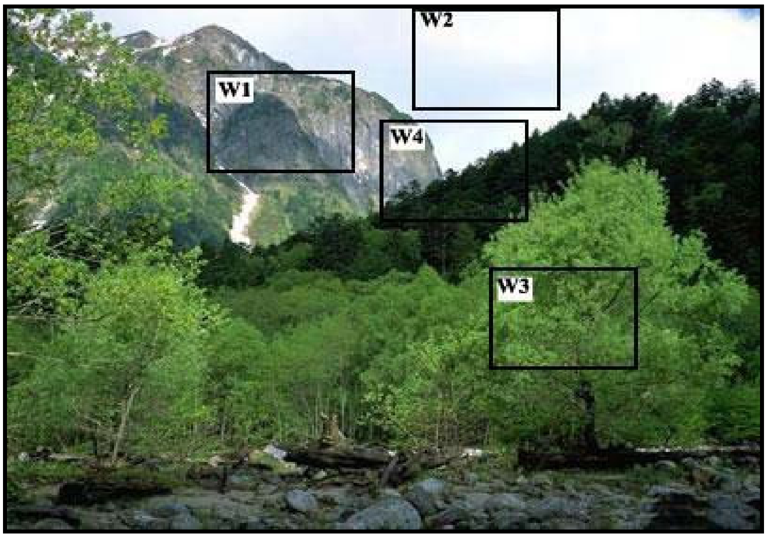

Fig. 1 Semantic concepts within the image

Secondly, a selected set of visual features is extracted from the window candidatures, which may be located in some ways. Heterogeneous features are employed to characterize the semantics from different perspectives. The underlying assumption is that objects concepts are usually related to visual characteristics, therefore, high-level scene properties may be inferred from low-level image features suck as color and texture information. Because of the multiple feature representation, the dimension of feature space is very high. So a reduction of the use of features is needed for neural classifier design. Lastly, neural networks, acting as a concept detector, map the features into semantics labels. In this study, neural networks are trained using standard supervising learning techniques, and they, with the generalization power, are used to recognize the similar semantics, which the neural networks met before.

2.2 Visual Features

Feature selection is a critical issue in retrieval system design. Although there are various criteria and techniques available in literature [10-16], it is still hard to tell which feature is necessary and sufficient to result in a high performance retrieval system. This is because the system performance relies on not only the features but the types of retrieval system as well. In this work, a set of multiple features is adopted to build the semantics learning system. The feature vector $\vec{f}$ consists of three components $< \vec{c}, \vec{m}, \vec{v} >$. $\vec{c}$ stands for Color Histogram and is 8 dimensions , i.e., $\vec{c} \in \Re^8$. $\vec{m} \in \Re^9$ is three channels color moments and $\vec{v} \in \Re^{42}$ is the mean and variance of coefficients of DCT for three R, G and B channels. The following briefly gives the calculation formulas.

Color Histogram. Color Histogram (CH) is the most commonly used color feature representation [10]. Statistically, it expresses the joint probability of the intensities of the three prime color channels, that is, R G and B.

$$h_{R,G,B}(r,g,b) = N * prob\{R = r, G = g, B = b\} \ . \tag{2}$$

where *N* is the number of pixels in the image. The color histogram is computed by discretizing the colors within the image and counting the number of pixels of each color. For convenience, CH is usually rewritten as $m = r + N_r * g + N_r * N_g * b$, where N_r, N_g and N_b are the numbers of bins for *R, G* and *B* respectively. However, the dimension of Color Histogram vector is usually high. To reduce it, a sequence similarity measure with spatial considerations is adopted [20]. The formula is given by

$$c_i = \sum_{1 \le j \le n} \frac{h((i-1) \times n + j)}{2} \times \sqrt{(i-r)^2 + h((i-1) \times n) + j)^2}$$
$$i = 1, 2, ..., (N_r \times N_g \times N_b) / n \,, \tag{3}$$

where *r* is the gravity center of the consider segments, which is calculated by

$$r = \arg \min_{1 \le j \le n} \left| \sum_{l=1}^{j} h((i-1) \times n + l) - \sum_{k=j}^{n} h((i-1) \times n + k) \right| \ . \tag{4}$$

In this work, $N_r = 8, N_g = 8, N_b = 4$ and $n = 32$, therefore, there are 256 bins totally which are set to 8 groups.

Color Moments (CM) [11]. Since most of the information is concentrated on the low-order moments, only the first moment (mean), the second and third central moments (variance and skewness) will be used as the color features. Let the value of the i-th color channel at the j-th image pixel is p_{ij}. The index entries related to this color channel are calculated by:

$$E_i = \frac{1}{N}\sum_{j=1}^{N} p_{ij} \ . \tag{5}$$

$$\sigma_i = \left(\frac{1}{N}\sum_{j=1}^{N}\left(p_{ij} - E_i\right)^2\right)^{\frac{1}{2}} \ . \tag{6}$$

$$s_i = \left(\frac{1}{N}\sum_{j=1}^{N}\left(p_{ij} - E_i\right)^3\right)^{\frac{1}{3}} \ . \tag{7}$$

where N is the number of pixels in the image.

The detailed formulas for the Color Histogram (CH) and the Color Moments (CM) can be found in [10] and [11], we here omit them.

Discrete Cosine Transform (DCT) [13]. It helps separate the image into spectral sub-bands of differing importance with respect to the image's visual quality. A signal or an image in spatial domain can be re-represented by a set of coefficients in frequency domain. Lower frequencies contribute more to an image than higher frequencies. Therefore, some coefficients related to higher frequencies might be discarded without significant loss in terms of visual quality. Tradeoff between the dimensionality and fidelity of information is made by a partition technique, which groups the coefficients into 7 categories as shown in Fig.2. The DCT features $\vec{v}$ are constituted by the mean and variances of the DCT coefficients in the 7 groups for three color channels.

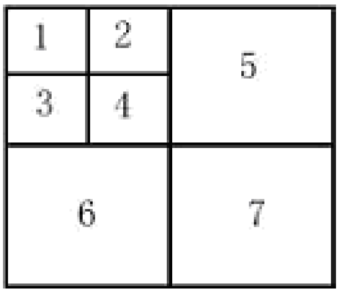

Fig. 2 DCT transformation

2.3 Determination of Neural Networks Architecture

Neural networks have been successfully applied in a variety of applications due to its merits. The Back-Propagation Neural Network (BPNN) [18] is adopted to perform the task of learning image semantics. The following describes a method for determining the BPNN architecture.

Usually, the topology of a BPNN is concerned primarily with the number of hidden layers and the numbers of hidden nodes on each hidden layers. The generalization ability of a BPNN is closely associated with its topology. It has been shown that a three-layered network (one hidden layer) provides enough representational power for most nonlinear discriminating decision problems [19]. An inappropriate number of nodes in hidden layer would make the BPNN under-fit or over-fit the training data. The training error rate usually decreases as the number of hidden nodes grows, which only implies that the network becomes more flexible in fitting the data. As the number of hidden nodes further increases, the average error rate eventually levels off and then increases, suggesting a movement toward over-fitting. Unfortunately, up to date, there is no really feasible method for determining the number of hidden nodes so that the resulting neural network model owns an optimal property. In this study, we determine this important number through experiments. Fig. 3 shows the correlation between the number of neurons in hidden layer and average training errors. It can be seen that the average training errors decrease from 10 to 50 (hidden nodes), and they vary slightly after 50 hidden nodes. Since larger hidden nodes will weaken the generalization power of the neural networks, we eventually decide to employ a BPNN with architecture 59-50-6 in our study.

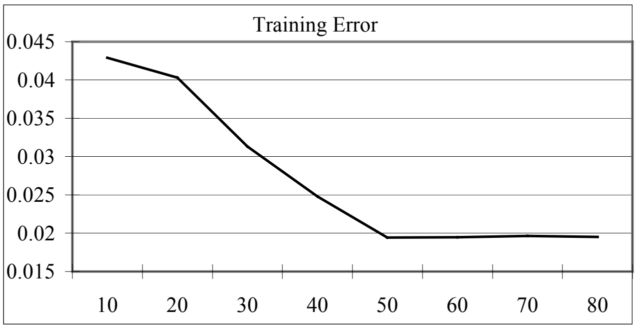

Fig. 3 Number of hidden nodes vs. average training errors

The output of the BPNN can be represented with a local or distributed scheme. The local representation demands a one-to-one matching between an output class and an output node, i.e., each output node represents a unique class. Differently, the distributed representation uses a cluster of output nodes that jointly represent all possible classes. For example, consider a system with 5 classes. The local representation scheme would require 5 distinct output nodes, one for each value. For class1, the output of the BPNN would be (1,0,0,0,0) and for class 3, the output would be

(0,0,1,0,0). The distributed representation way requires only 3 output nodes (e.g., 000 for class1, 001 for class2, and 101 for class5). The distributed representation is more efficient than local representation because it requires fewer nodes to represent the same set of classes. However, it is more complex than its counterpart and is not suitable to manage multiple outputs. Thus, we adopt the local representation form in this work.

3 Performance Evaluation

Eighty outdoor scene images stored in JPEG format with twenty-four bits color are used in this study. Five classes - tree, sky, mountain, cloud and water are selected which are the most common components in natural outdoor scene images. Another class "Unknown" is set if the image block cannot be recognized or include more than one semantics. Totally, there are six classes.

According to the preset windows size, we define the semantics in the block. For each image, 10 windows are sampled. The locations, the class and the window size are stored in a text format as following:

```
$Pic_10420.jpg
@X=0.335    Y=0.168     Width=0.2   Height=0.2  Class=1
@X=0.322    Y=0.261     Width=0.2   Height=0.2  Class=2
@X=0.672    Y=0.507     Width=0.2   Height=0.2  Class=3
@X=0.513    Y=0.385     Width=0.2   Height=0.2  Class=4
# (End of the image file)
```

where X and Y are the top-left position of the window. Width and Height are the size of the window defined by (1). Class attribute specifies which class this image block belongs to. There are total 800 examples generated from the image database, 500 of them are assigned as training data and the rest 300 examples are used as the test data.

In training program, the BPNN runs 2000 epochs with the momentum factor as 0.0 and the learning rate as 0.1, respectively. The effectiveness of the proposed method can be measured by Recall and Precision, which are often referred to together since they measure the different aspects of the system performance. Recall measures the system's ability to retrieve relevant items from the database. It is defined as the ratio between the number of retrieved relevant items and the total number of relevant items in the database. Thus, the recall rate demonstrates the power of a learning system by revealing its level of false negatives. Precision measures the retrieval accuracy and is defined as the ratio between the number of retrieved relevant items and the number of total retrieved items. This measure demonstrates the efficiency of a learning system and is closely related to its level of false positives. Table 1 summarizes the recall and precision of the performance of the BPNN. At 25% windows size, the network achieves the highest recall rate (95.42%) and highest precision rate (96.21%) for the training data set. For the test data set, the recall rate (56.31%) and precision rate (55.56%) are better than that obtained by using other window sizes, too.

Table 1 Performance measures for different window sizes

	Training Data Set		Test Data Set	
	Recall	Precision	Recall	Precision
10%	0.87	0.90	0.47	0.49
15%	0.90	0.91	0.50	0.52
20%	0.93	0.93	0.53	0.51
25%	0.95	0.96	0.56	0.56
30%	0.94	0.93	0.54	0.52

Table 2 gives a comparison with decision tree techniques (C4.5). The decision tree classifier (DTC) is unable to achieved better performance. For example, the recall rate for BPNN is 95.42 %. It is much higher than that of DTC, only 62.68%. Neural nets can extract nonlinear combinations of features, and the resulting discriminating surfaces can be very complex. Those characteristics of neural networks can be very attractive than decision tree classifiers where one has to determine the appropriate feature subsets and the decision rules at each internal node.

Table 2 Performance comparison for 25% window size

	Training Data		Test Data	
	Recall	Precision	Recall	Precision
BPNN	0.95	0.96	0.56	0.56
DTC	0.63	0.89	0.39	0.43

Table 3 and 4 summarizes the detailed recall and precision performance of the BPNN classifier with 25% window size for the test data and the training data, respectively.

Table 3 Performance of 25% window size for the test data

ClassID	Miss	Cover	OAll	RAll	Recall	Precision
0	12	16	24	30	0.67	0.53
1	19	50	68	69	0.74	0.72
2	17	21	56	31	0.38	0.68
3	18	35	72	57	0.49	0.61
4	25	14	28	42	0.50	0.33
5	41	32	52	71	0.62	0.45
Average					0.56	0.56

Table 4 Performance of 25% window size for the training data

ClassID	Miss	Cover	OAll	RAll	Recall	Precision
0	5	102	104	105	0.98	0.97
1	0	50	53	51	0.94	0.98
2	5	74	76	80	0.97	0.93
3	2	52	54	58	0.96	0.90
4	2	55	60	52	0.92	1.06
5	8	145	153	154	0.95	0.94
Average					0.95	0.96

4 Conclusion

This paper presents an empirical study on machine learning approach for automatic semantics extraction from visual data. Simulation results demonstrated that about 25% block size seems to be an appropriate option for achieving a better result for this image database. Comparative studies showed that neural networks outperform the decision tree techniques in terms of the recall and precision performance. Although the obtained results are not so satisfactory, which may be caused by the use of improper visual features, the presented methodology in this paper has its value to help us in getting a better understanding about the potentials and limitations of the proposed methodology. Further efforts on automatic semantics identification using machine learning techniques will focus on feature selection, use of fuzzy classifiers or neuro-fuzzy classification techniques.

Lastly, we point out that an appropriate formulation of the domain problem plays a key role in performance evaluation.

References

1. Y. Rui, T. S. Huang, S. F. Chang. Image Retrieval: Current Techniques, Promising Directions and Open Issues. Journal of Visual Communication and Image Representation, Vol. 10 (1999) 39-62
2. A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain: Content Based Image Retrieval at the End of the Early Years. IEEE Trans. on PAMI, Vol. 22 (2000) 1349-1380
3. M. Flickner, H. Sawhney, etc: Query by Image and Video Content: The QBIC System. IEEE Computer, vol. 28, (1995)
4. A. Pentland, R. W. Picard, and S. Sclaroff: Photobook: Tools for Content-Based Manipulation of Image Databases. Proc. SPIE, vol. 2185, (1994) 34-47

5. J. R. Smith and S. F. Chang. VisualSEEK: A Fully Automated Content-Based Image Query System. Proc. ACM Multimedia, (1996) 87-98
6. W. Y. Ma and B. Manjunath. NaTra: A Toolbox for Navigating Large Image Databases. Proc. IEEE Int'l Conf. Image Processing, (1997) 568-571
7. M. Nappi, G. Polese, and G. Tortora: FIRST: Fractal Indexing and Retrieval System for Image Databases. Image Vis. Comp., Vol. 6 (1998) 1019-1031
8. T. P. Minka and R. W. Picard. Interactive Learning Using a Society of Models. Pattern Recognition, vol. 30, (1997) 565
9. R. Schettini, G. Ciocca, S.Zuffi: A Survey of Methods for Colour Image Indexing and Retrieval in Image Databases. http://www.itim.mi.cnr.it/Linee/Linea1/Sottolinea3/methodsCIIR-9-1-01a.PDF
10. M. Swain, D. Ballard: Color indexing. Int. J. Comput. Vis., Vol 7 (1991) 11-32
11. Markus Stricker, Markus Orengo: Similarity of color images. In Proc. SPIE Storage and Retrieval for Image and Video Databases (1995)
12. R. M. Haralick etc: Texture features for image classification. IEEE Trans. on Sys, Man, and Cyb, SMC-3(6) (1973) 610-621,
13. K. Rao and P. Yip: Discrete Cosine Transform, Algorithms, Advantages, Applications. Academic Press (1990)
14. J. Z. Wang, J. Li and G. Wiederhold: IMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries. IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 23, no. 9 (2001) 947-963
15. W.Y. Ma and B. Manjunath: NaTra: A Toolbox for Navigating Large Image Databases. Proc. IEEE Int. Conf. Image Processing, (1997) 568-571
16. C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein, and J. Malik: Blobworld: A System for Region-Based Image Indexing and Retrieval. Proc. Visual Information Systems, (1999) 509-516
17. Lefteri H. Tsoukalas and Robert E. Uhrig et al.: Fuzzy and Neural Approaches in Engineering. John Wiley & Sons, Inc., (1996)
18. D. Rumelhart, G. Hinton, and R. Williams: Learning Internal Representations by Error Propagation. Parallel Distributed Processing: Explorations in the Microstructures of Cognition, Vol. 1, D. E. Rumelhart and J. L. McClelland, eds., MIT Press, Cambridge, Mass. (1986) 318-362
19. J.de Villiers and E. Barnard: Backpropagation Neural Nets with One and Two Hidden Layers. IEEE Trans. on Neural Networks, Vol. 4, No. 1 (1992) 136-141
20. K. J. Cios and I. Shin: Image recognition neural networks: IRNN, Neurocompting, Vol. 7, No. 2 (1995) 159-185