Image Semantic Classification by Using SVM

WAN Hua-Lin1*, Morshed U. Chowdhury2

1(Laboratory of Digital Technology, Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100080, China)
2(School of Information Technology, Deakin University- Melbourne Campus, Melbourne 3125, Australia)

+Corresponding author: E-mail: wanhl@ict.ac.cn
http://www.ict.ac.cn
Received 2002-09-28; Accepted 2002-12-04

http://www.jos.org.cn/1000-9825/14/1891.htm

Abstract: There exists an enormous gap between low-level visual feature and high-level semantic information, and the accuracy of content-based image classification and retrieval depends greatly on the description of low-level visual features. Taking this into consideration, a novel texture and edge descriptor is proposed in this paper, which can be represented with a histogram. Furthermore, with the incorporation of the color, texture and edge histograms seamlessly, the images are grouped into semantic classes using a support vector machine (SVM). Experiment results show that the combination descriptor is more discriminative than other feature descriptors such as Gabor texture.

Key words: content-based; image feature descriptor; color; texture; edge; classification; SVM

* WAN Hua-Lin was born in 1974. He received his Ph.D. degree in Computer Science from Institute of Computing Technology, the Chinese Academy of Sciences, Beijing China in 2002. He is currently an academic staff in Institute of Computing Technology. His research interests include multimedia information retrieval, image processing, computer vision, and pattern recognition. He is an author of over 10 research publications in these areas. Morshed U. Chowdhury received his Ph.D. degree in Computing from Monash University Australia in 1999. He is currently an academic staff in School of Information Technology Deakin University, Melbourne, Australia. His research interests are in the areas of image segmentation, classification and retrieval, suppression of video coding distortion, multimedia synchronisation, e-commerce, online distance learning, etc.
1 Introduction and Related Work

The rapid development of multimedia computing and communicating technology has led to increased demands for multimedia information, and larger and larger collections of images and videos are available to the public. Consequently, how to help users find their needs is becoming a challenge. In order to solve this problem, a content-based image retrieval method is proposed, which represents image content with low-level feature descriptors, such as color, texture, shape, etc. Here, “content” is some kind of objective statistic character of images, which couldn’t be understood by human beings directly.

During image retrieval, human beings are used to the human-computer interaction at a semantic level, so image semantic feature extraction and representation are necessary; but because of the enormous gap between low-level visual feature and high-level semantic information, image semantic representation is still an unresolved problem.

We believe that the nature of CBIR (content-based image retrieval) is to search the relevant or similar images based on low-level visual features, which implies that relevant images have similar visual features. This is to say, similar or relevant images are adjacent in some ideal feature space, so it is possible to cluster or classify images according to low-level visual features. Image classification is limited to image understanding, and the purpose is to group images into some semantic class, so that semantic features of images could be extracted automatically, which will not only help organize image databases, but also help label images automatically. This will drive CBIR from the laboratory into industry.

Image classification and clustering include supervised and unsupervised classification of images\cite{1}. In supervised classification, we are given a collection of labeled images (or priori knowledge), and the problem is to label a newly encountered, yet unlabeled image. On the other hand, for unsupervised classification (or image clustering), the problem is to group a given collection of unlabeled images into meaningful clusters according to the image visual feature without a priori knowledge\cite{2}.

We have tried to cluster images into semantic categories using SOFM (Self-organism Feature Mapping) and C-Means. The experiment results show that the error rates are very high. The reason perhaps is that the low-level visual features are not related to the human perception about image content. The clustering algorithm couldn’t automatically bridge the enormous gap between low-level visual feature and high-level semantic content without priori knowledge.

We believe supervised classification is a promising method, and there have been many achievements in this field. Smith\cite{3} proposed a multi-stage image classification algorithm based on visual features and related text. Bruzzone and Prieto\cite{4} developed a variety of classifiers to label the pixels in a landset multi-spectral scanner image. MM-classifier developed by Zaiane et al.\cite{5} classifies multimedia images based on some provided class labels. Vailaya et al.\cite{6} used a binary Bayesian classifier for the hierarchical classification of vacation images into indoor/outdoor classes. Outdoor images are further classified into city/landscape classes. Li and Wang\cite{7} classified textured and non-textured images using region segmentation. Experiments performed by Chapelle\cite{8} have shown that SVM could generalize well to image classification, however the only features are high-dimension color histogram, which simply quantizes RGB color space into 4096 color bins. In fact, image content cannot be represented effectively by only color feature. For example, lawn and trees may have the same color histogram, while lawns in spring and autumn have different color histograms whilst they have the same shape or semantic feature.
Therefore, we have to find an efficient method to describe the image content and bridge the gap between the low-level visual feature and high-level semantic information. Taking these into consideration, we propose more effective texture and edge descriptors in this paper. Based on this, by combining color, texture and edge features seamlessly, images are grouped into semantic categories using SVM.

The paper is organized as follows. In Section 2, a novel texture and edge descriptor is proposed; Sections 3 and 4 briefly discuss the SVM framework and its application to image classification; Classification experiments and results are reported in Section 5, and finally Section 6 states the conclusions.

2 Feature Selection and Extraction

Currently, most researchers have dedicated the finding of an efficient method to represent the image low-level visual content and to measure the similarity of the image pairs, e.g. Refs.[9,10], because the accuracy of CBIR depends greatly on the representation of the low-level visual features. The more discriminative the low-level features, the more accurate the content-based classification and retrieval. We know that only one type of low-level features does not work well. It is difficult to incorporate color, texture and shape feature seamlessly, because they belong to different metric systems. They are not comparable.

In content-based retrieval, color descriptors originating from the histogram or spectrum analysis have played a central role in image content description. Can we develop a texture and shape descriptor just like color histogram? If we can, things will get easy, because histogram descriptor is no metric. In this paper, we propose novel texture and edge descriptors, from which we can conveniently describe image texture and edge feature. Moreover, color, texture and edge histograms are combined into image feature, which can be respectively described as follows.

2.1 Texture descriptor

Image texture means a kind of change of pixels’ intensity (or gray) in some neighborhood, which is spatially a statistical relative. It comprises two elements—texture unit and its arrangement. Combining this idea with the concept of texture spectrum proposed by He[11], we present a new method to describe the image texture histogram.

In order to get the local texture information of an image pixel, we take its 3*3 neighbor into consideration, as shown in Fig.1. Where \( I_i \) are the intensities of the corresponding pixels and can be calculated according to the following formula:

\[
I = 0.299R + 0.587G + 0.114B
\]

(a) (b) (c)

Fig.1 3*3 neighbor of a pixel, the location of \( I_4 \) is the central pixel

Fig.2  Texture models

Intensity differences calculated along different directions will be different, such as (a), (b), (c) in Fig.2. After the image retrieval experiments and the analysis of the many texture images, we found that the texture distribution in a neighborhood is ordinarily like a) or b) in Fig.2. So we record the salient intensity changes of the neighboring pixels using a binary sequence \( V_i (i = 0,1,...,7) \), where \( V_i \) is denoted as:

\[
V_i = \begin{cases} 
0 & \text{if } |I_i - I_{i+1}| \leq T_i \\
1 & \text{if } |I_i - I_{i+1}| > T_i 
\end{cases} \quad i = 0,1,...,7
\]
where \( T_j \) is a predefined constant. From Eq.(2), we can see it focuses on the salient changes of a pixel’s intensity, which is consistent with human vision perception for texture, because human vision perception is more sensitive to the changes of intensities than to the intensity itself. Furthermore, Eq.(2) indicates that the smaller the \( \sum V_i \) the smoother the neighborhood. Therefore, from the binary sequence \( V_0, V_1, ..., V_7 \), we can easily extract many statistical features to describe the image texture.

For example: If we transform the sequence \( V_0, V_1, ..., V_7 \) from binary to decimal, obviously the value range of the sequence is \([0, 1, 2, ..., 255]\). This is to say, the total number of the texture model for the image pixels is 256. The texture unit (TU) of the image pixel can be obtained according to the following formula:

\[
TU = \sum_{i=0}^{7} 2^i V_i
\]

(3)

The value range of \( TU \) is \([0, 1, 2, ..., 255]\). Thus, calculating the distribution of the image texture unit in the value range, we can get the image texture spectrum. Suppose we denote the texture unit \( TU \) of Pixel\((i, j)\) as \( T(i, j) \), and denote the image texture spectrum as \( \{ h[i]\} [0, 1, 2, ..., 255] \); then

\[
h[k] = \frac{\sum_{j=0}^{n-1} \sum_{i=0}^{m-1} f(i, j, k)}{m \times n}
\]

(4)

where \( f(i, j, k) \equiv \begin{cases} 1, & \text{if } T(i, j) = k \\ 0, & \text{otherwise} \end{cases} \), \( m, n \) are image height and width respectively.

In fact, we can extract other statistical features from the binary sequence \( V_0, V_1, ..., V_7 \), e.g. smoothness, moment feature, etc.

It is believed that the image texture is an interwoven distribution of the intensity change of the pixels, therefore, compared with structural approaches\(^{[12]}\), a statistical one for a texture description is more reasonable. Compared with texture descriptors\(^{[12]}\), the texture description method in this paper has the following characteristics: (1) it describes the intensity changes of the neighboring pixels, not the absolute intensity, which indicates that texture is some kind of changes of pixel intensity, (2) the texture unit obtained using our method is local, which makes it possible to get texture spectrum. Compared with the method proposed in Ref.[11], (1) our method decreases the dimension of texture spectrum from 6561 to 256. This greatly reduces the time- and spatial-complexity. Also, (2) our method pays more attention to salient changes of the pixels’ intensities. This is more consistent with human vision perception for image texture, so it can describe the smoothness degree of images efficiently.

### 2.2 Edge descriptor

Edge is a basic feature of images. It contains contour information of a valuable object in the image and can be used to represent image contents, recognize the objects, etc. Although Prewitt, Sobel and Canny edge detectors can successfully filter an image background, and extract an object contour, the extracted edges may be composed of a large number of short lines or curves. It is very difficult to describe and represent them. Rosin\(^{[13]}\) proposed an algorithm to simulate edge points into a combination of representations, such as lines, circular, elliptical and superelliptical arcs, and polynomials. But it is so complex and time-consuming that it isn’t feasible for real-time CBIR. Recently, Park\(^{[14]}\) proposed a local edge histogram descriptor, but the number of image-blocks is a predefined constant. This means, for different images, the sizes of image blocks may be different, which affects the edge type of the image block.
A simple and effective image edge spectrum descriptor for the spatial distribution of six types of edges is presented and implemented as shown in Fig.3. First, after detecting the image edge by canny detector, partition the edge image into $3 \times 3$ non-overlapping sub-images, and then partition each sub-image into $8 \times 8$ image blocks (Fig.4). It is consistent with JPEG format of image compression so that the width and height of images can be divided exactly by the width and height of image blocks, respectively, which ensures the information of image border will not lose. Since there exist six edge types, we can define a global edge type histogram with 6 global bins. In the same way, according to the sub-images, 54 local histogram bins are defined.

Suppose image width and height are denoted by $\text{image\_width}$ and $\text{image\_height}$, respectively, then the number of $8 \times 8$ image blocks in the horizontal direction is $N = \frac{\text{image\_width}}{8}$ and the number of $8 \times 8$ image blocks in the vertical direction is $M = \frac{\text{image\_height}}{8}$, so $MN$ is the total number of image blocks.

If we denote an edge type array with $ET(i, j)$, where $i = 0, 1, ..., M - 1$, $j = 0, 1, ..., N - 1$, then a global edge histogram $\{H_{EG}(k)\}$ can be computed by the following formula:

$$H_{EG}(k) = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} E(i, j)}{M \times N}$$

(5)

where $E(i, j) = \begin{cases} 1, & \text{if } ET(i, j) = k \\ 0, & \text{otherwise} \end{cases}$, $k = 0, 1, ..., 5$.

For each Sub-image, a local edge histogram can be calculated by a similar method, denoted as $\{H_{EL}(m, n, l) | m = 0, 1, 2, n = 0, 1, 2, l = 0, 1, ..., 5\}$. 
2.3 Color descriptor

As for the descriptor of an image content, a color spectrum or histogram is a simple and efficient low-level feature. However, if calculated directly in a triple-dimension color space (e.g., RGB), both the storage space and computing time will be very high. So color quantization is necessary before extraction of the image color features. In this paper, we adopt a quantization algorithm based on subjective vision perception proposed in Refs.[15,16]. First, transforming the color image from RGB to HSV, then, according to human vision perception for color, quantizing the triple-color components (H, S, V) into non-equal intervals. The transformation formula is described briefly as follows:

\[
H = \begin{cases} 
0 & \text{if } h \in [316,20] \\
1 & \text{if } h \in [21,40] \\
2 & \text{if } h \in [41,75] \\
3 & \text{if } h \in [76,155] \\
4 & \text{if } h \in [156,190] \\
5 & \text{if } h \in [191,270] \\
6 & \text{if } h \in [271,295] \\
7 & \text{if } h \in [296,315] 
\end{cases}, \quad S = \begin{cases} 
0 & \text{if } s \in [0,0.2] \\
1 & \text{if } s \in (0.2,0.7], \quad V = \begin{cases} 
0 & \text{if } v \in [0,0.2] \\
1 & \text{if } v \in (0.2,0.7] \\
2 & \text{if } v \in (0.7,1] 
\end{cases}
\end{cases}
\tag{6}
\]

Based on the quantization level aforementioned, the triple-color components are mapped into an one-dimension vector using (7),

\[
l = HQ_SQ_Vs + HQ_Vs + V
\tag{7}
\]

where \(Q_S\) and \(Q_V\) are the numbers of quantization levels for color components \(S\) and \(V\), respectively. From (6), it is known that \(Q_S=3, Q_V=3\), so (6) can be further transformed as:

\[
l = 9H + 3S + V
\tag{8}
\]

Therefore \(H, S, V\) can be represented by a vector, according to (8), with the value range of \(l=[0,1,2,…,71]\). Using a similar method as Eq.(4), we can get the color spectrum with 72 color bins.

Compared with the quantization algorithms proposed in Refs.[8,9], the advantages of this algorithm are: (1) the triple-color components are quantized into non-equal intervals, respectively, which is consistent with human vision perception for color; (2) combining similar colors together avoids the problem of color redundancy appearing in other color quantation algorithms.

3 Support Vector Machine (SVM)

Support vector machine is a well-known pattern classification method. It is an approximate implementation of the structural risk minimization (SRM) principle\(^{[17]}\) and creates a classifier with minimized Vapnik-Chervonenkis (VC) dimension. For pattern classification, SVM has a very good generalization performance without domain knowledge of the problems. This is one of the reasons why we select SVM as an image classifier.

Let the separable training set be \(\{(\tilde{x}_i, y_i)\}_{i=1}^N\), where \(\tilde{x}_i\) is the input pattern vector, \(y_i \in \{-1,1\}\) is the class label, \(+1\) denotes the positive example, and \(-1\) denotes the negative example. If the training set is linearly separable, and the discriminating function is \(g(x) = \tilde{w}^T \tilde{x} + b\), we can easily get the classifier hyper-plane by calculating: \(\tilde{w}^T \tilde{x} + b = 0\), where \(\tilde{w}\) is a weight vector, \(b\) is a bias. The goal of the SVM is to find the parameters \(\tilde{w}_0\) and \(b_0\) such that the distance between the hyper-plane and the nearest sample point is more than 1:

\[
\tilde{w}_0 \tilde{x} + b_0 \geq 1 \quad \text{for } y_i = +1,
\]

\[
\tilde{w}_0 \tilde{x} + b_0 \leq -1 \quad \text{for } y_i = -1.
\]

If it is not linearly separable, it is necessary to map the input training vector into a high-dimension feature
space using a kernel function \( K(\tilde{x}, \tilde{x}_i) \), then create the optimal hyper-plane and implement the classification in the high-dimension feature space. For more details, refer to Ref.[18].

4 Image Semantic Classification by Using SVM

In order to improve the precision of image classification, the recognition of each class is taken as an independent two-class classification. Suppose there are \( K \) classes in the image database, denoted as \( L = \{\alpha_1, \alpha_2, \ldots, \alpha_k\} \), \( N_i \) is the number of images in class \( \alpha_i \). We transform the \( K \)-class classification to two-class classification by the following scheme: For each image class \( \alpha_i \), the positive training samples are all the images in class \( \alpha_i \); while the negative training samples are all the other images, i.e. the image number of positive samples is \( N_i \), while the image number of negative samples is \( \sum_{j=1, j \neq i}^k N_j \).

Given an image class \( \alpha \in L \), the training sample set is \( E = \{(\tilde{x}_i, y_i)\mid i = 1,2,\ldots,l\} \), where \( \tilde{x}_i \in \mathbb{R}^N \) is an image feature vector composed of color, texture and edge histograms, \( y_i \in \{+1,-1\} \). If \( y_i = +1 \), \( x_i \in \alpha \), while \( y_i = -1 \) means \( x_i \not\in \alpha \).

Before utilizing SVM to classify a pattern, a feature vector should be extracted from the original space. In order to solve nonlinear problems, it is mapped into a higher dimension feature space. For different applications, the input spaces are different, e.g. for image classification, each point in input space is an image.

Through feature map \( \Phi \), the pattern classification can be transformed into an optimization problem in \( \mathbb{R}^N \). The image classification algorithm is briefly described as follows:

[Condition] Training sample set \( E = \{(z_i, y_i)\mid i = 1,2,\ldots,l\} \), where \( z_i \in \mathbb{R}^N \), \( y_i \in \{-1,+1\} \)

[Problem] \((w_0, b_0) = \arg \min_{(w,b)} R(w,b)\)

where \( R(w,b) = \int \frac{1}{2} |f_{w,b}(z) - y| \, d\rho(x,y) \), \( \rho(x,y) \) is the density of the joint probability distribution for feature vector \( x \) and its class \( y \), \( f_{w,b}(z) = \text{sgn}[w \cdot z + b] \).

In order to get \((w_0, b_0)\), or to get classifier \( f_{w,b}(z) \), we need to solve the following quadratic optimization problem: \( \alpha_0 = \arg \max_a W(a) \), where \( W(a) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j (z_i \cdot z_j) \), \( \alpha_i \geq 0 \) (\( i = 1,2,\ldots,l \)), \( \sum_{i=1}^l \alpha_i y_i = 0 \).

After getting \( \alpha_0 \), we can easily get \( w_0 \) using formula \( w = \sum_{i=1}^l y_i \alpha_i z_i \). At the same time, there must exist \( z_i \) such that \( f_{w,b}(z_i) = 1 \), thus we can get \( b_0 \).

Finally, in order to decide whether sample \( \tilde{x} \) belongs to class \( \alpha \), first compute \( z = \Phi(\tilde{x}) \), then compute the decision function \( f(z) = \text{sgn} \left[ \sum_{i=1}^l y_i \alpha_i (z \cdot z_i) + b \right] \). If \( f(z) = 1 \), then \( \tilde{x} \) belongs to class \( \alpha \); otherwise, \( \tilde{x} \) is not in class \( \alpha \).

5 Experimental Results

We tried to learn each of the 8 concepts (Flower, Bird, Mountain, etc) using SVM. For training and testing, we use natural images mostly from “Corel Image Gallery”, where are about 67000 images. In this paper, we select 2516 images from image database, and manually organize them into 8 semantic categories, including flowers, birds,
mountains, tools, landscape, etc. Each class comprises training set and testing set. The image category used and its number are shown in Table 1.

Table 1 Training and testing set for Image classification experiments

<table>
<thead>
<tr>
<th>NO</th>
<th>Category</th>
<th>N_{\text{Training}}</th>
<th>N_{\text{Testing}}</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Flower</td>
<td>400</td>
<td>189</td>
</tr>
<tr>
<td>2</td>
<td>Bird</td>
<td>150</td>
<td>86</td>
</tr>
<tr>
<td>3</td>
<td>Mountain</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>Tool</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>Landscape</td>
<td>240</td>
<td>76</td>
</tr>
<tr>
<td>6</td>
<td>Building</td>
<td>400</td>
<td>103</td>
</tr>
<tr>
<td>7</td>
<td>Animal</td>
<td>400</td>
<td>160</td>
</tr>
<tr>
<td>8</td>
<td>Leave</td>
<td>80</td>
<td>32</td>
</tr>
</tbody>
</table>

We experimented with the combined histogram of images, which consist of 72 color features, 256 texture features and 6 global edge features, and selected radial-basis function as the SVM kernel function.

To demonstrate the practicability of image classifier and evaluate the discriminating performance of image features, classification accuracy and precision are computed as:

\[
\text{Accuracy} = \frac{\text{correctly classified samples}}{\text{total samples}},
\]

\[
\text{Precision} = \frac{\text{correctly classified positive samples}}{\text{total positive samples}}.
\]

We compared the accuracy and precision curves of 3 image features (combined histogram, color histogram+Gabor texture, and 4096 color bins used in Ref.[8]). The Gabor texture algorithm was proposed by Manjunath[12], the source code can be downloaded from web address http://vision.ece.ucsb.edu/texture/featuure.html. In the experiments, Gabor textures of images are not pre-processed. Under this condition, training time is much longer than that of using a combination histogram, and its classification accuracy and precision are lower. The possible reason is that Gabor texture and color histogram belong to different metric spaces, and they are not comparable, so combining them is not reasonable. While histogram features are statistical and normalized, they do not belong to any metric space, so the combination of color, texture and edge histograms is more discriminative than the combination of color and Gabor texture[12].

From Figs.5 and 6, we can see that the accuracy and precision of image classification combining color, texture and edge histograms are on average 7.37% and 7.51% higher than those combining color and Gabor feature, and on average 13.36% and 9.38% higher than those for only 4096 color bins[8]. Ordinarily speaking, (1) the combination of color, texture and edge histogram has more discriminative power than that of color and Gabor, or only color feature; however for image classification, it is not overwhelmingly better, e.g., for leaves, precision of the latter is better than the former; (2) for images whose background is clear and the objects are salient, the classification performances are satisfying, e.g., birds, tools, etc. It may be explained by the fact that the semantic objects in these images are clear and non-ambiguous. For example, human beings will never take birds as tools, but may take mountain as landscape, this is to say, an image may belong to more than one semantic category, which will inevitably disturb the classification results; (3) the bigger the number of positive samples, the better the classification performance, because there are enough training examples. These are all consistent with subject perception of human beings.

6 Conclusions

In this paper, we propose and present a new histogram method to describe image texture and edge types. Our texture method is based on the description of the intensity changes of the neighboring pixels in the local texture instead of absolute intensity. We also incorporated color in our texture and edge histogram, and grouped the images
into semantic categories using SVM. Experimental results show that the combined image histogram is more discriminative than the color histogram used in Ref.[8] and the Gabor texture.

Fig.5 Accuracy curves comparing classification performance of 8 natural image categories using combined feature histogram vs. color histogram+Gabor texture, and 4096 color bins

Fig.6 Precision curves comparing classification performance of 8 natural image categories using combined feature histogram vs. color histogram+Gabor texture, and 4096 color bins

References:

附中文参考文献: