Image Scene Classification based on Latent Semantic Analysis

Chu-Hui Lee*, Meng-Feng Lin, Kun-Cheng Chiang

*Department of Information Management, Chaoyang University of Technology, Taichung County 41349, Taiwan, ROC

Abstract

Digital image classification is becoming increasingly important. A digital image can be represented by its low-level features. Semantic analysis is a common technique for scene image classification. How to reduce the semantic gap between the high-level semantic and low-level features is a significant problem. This paper proposes a novel scene image classification method. Latent Semantic Analysis (LSA) is applied to scale feature dimensions, delete noise, and select the important latent semantic features of each scene. To increase classification accuracy, low-level features should contain diversity. The proposed mechanism is applied to classify images by measure the scene similarity of each image. Experimental results demonstrate that the proposed mechanism can classify image scenes successfully and has a better correct classification
ratio than other analytical methods. The proposed image classification method reduces the semantic gap and close to human semantics.

*Keywords:* Image Classification, Latent Semantic Analysis, Low-level Features, Scene Similarity Measurement.
1. Introduction

Via the Internet, transmission of digital images has become easy and image databases now require large storage capacities. As image databases are now massive, an efficient method for searching images is needed. Content-Based Image Retrieval (CBIR) is an important image retrieval technology. An image typically contains many objects, and is represented by low-level features that have color, texture, shape, and spatial relationships. However, a gap exists between high-level semantic content and low-level image features. This is referred to as the semantic gap problem [1-4]. The semantic gap problem is primarily due to the fact that low-level image features cannot be mapped to high-level semantic content smoothly. Thus, identifying and classifying high-level semantic content in an image is very difficult. Therefore, how to reduce the gap between high-level semantic content and low-level features in an image is a significant problem.

Good image classification can improve the effectiveness of CBIR systems and eliminate this semantic gap [5-8]. Many schemes to reduce the size of the semantic gap have been developed in many relevant researches [1,9-11]. Tsai et al. [12] utilized low-level features combined with Support Vector Machine (SVM) for image classification. Chow and Rahman applied a tree structure combined with global and regional features of an image for image classification via Self-Organizing Map (SOM) [13]. Yang et al. captured the semantic content of images in local and global regions via
SVM to reduce the size of the semantic gap [14]. Features in an image contain both
global and regional features. In current studies, regional features are represented by
detailed information of a scene image, which assists in global features classification.

The proposed mechanism uses a statistical histogram of local regions in an image
and utilizes Latent Semantic Analysis (LSA) to identify the latent semantic features in
images. Generally, LSA generates good classification results for text data. The proposed
approach uses LSA to locate correlations between low-level features and high-level
semantics and reduce the size of the semantic gap. Via LSA, the dimensions of features
are scaled and noise is deleted. To increase classification accuracy, low-level features
must contain both color and texture. Therefore, this paper can classify various scene
images through multiple latent semantic features. This paper proposes a novel scene
image classification method based on the semantics of image features. Each image is
classified into suitable scene classes, semantic gap is eliminated, and classification
performance is enhanced.

The remainder of this paper is organized as follows. Section 2 describes LSA and
low-level features. Section 3 demonstrates the use of LSA to obtain the important
features and calculate scene similarity. Experiments using the proposed mechanism are
presented in Section 4. Finally, Section 5 gives conclusions.
2. Related works

This section discusses LSA and low-level features in images. LSA is adopted to delete noise and selects the important features of the scene, and the low-level features here are color and texture in image.

2.1. Latent Semantic Analysis

LSA is a vector space model for indexing and retrieving information technology, primarily uses Singular Value Decomposition (SVD) and reduces the dimensions based on the theory of modules to find implicit concept in a document [5,15 -18]. Some LSA studies are used in text or web pages. Notably, SVD is a decomposition technology that reduces the size of a highly dimensional matrix[19]. Through dimensional reduction can extract important information from the semantic space. After decomposition, the new matrix and original matrix of features are similar and the new matrix can more accurately describe the matrix of the hidden semantic concept than can the original matrix.

Let \( T \) be a matrix with sized \( m \times n \). After SVD, matrix \( T \) is transferred to three matrices, where matrix \( S \) represents the semantic space for a diagonal matrix, and matrix \( A \) and matrix \( B^T \) are the semantic spaces of keywords and a document for an orthogonal matrix respectively; that is \( T = ASB^T \) [15]. After the SVD, matrix \( S \) of singular value will be sorted from large to small; thus, re-ordering of the matrix \( S \)
requires re-ordering of columns of $A$ and the rows of $B^T$. Larger singular values corresponding to the feature are important. This paper uses dimension reduction to popup the important scene features; that is $T = A'B'B'$ [15]. Matrix $T$ is expressed by Equation 1[15].

$$T = \sum_{i=1}^{c} a_i s_i b_i^T$$ (1)

### 2.2. Color

Color, one of the most dominant features of images for CBIR, is also one of the most widely used low-level features. Many color presentation schemes exist, such as RGB, LAB, HSV and CMYK. The HSV color attribute model, which presents color features, is closest to human vision. In the HSV model, $H$ (hue) is color type, $S$ (saturation) is color intensity, and $V$ (value) is color brightness. These three elements comprise the color space and are independent. Figure 1 shows the HSV color space. The value of $H$ is $0–360^\circ$; that of $S$ is $0–1$; and that of $V$ (brightness) is $0–1$. The color histogram proposed by Swain and Ballard is the most popular [20].

Figure 1 Here

### 2.3. Texture

Texture is important in both pattern recognition and computer vision. Obtaining useful information is an important task in texture analysis. The Grey-level
Co-occurrence Matrix (GLCM), which describes image texture, is a two-dimensional matrix of joint probabilities \( P(i, j : d, \theta) \), in which two neighboring pixels with grey levels \( i \) and \( j \) occur for a given direction \( \theta \) and distance \( d \). Haralick (1979) used 14 features derived from the co-occurrence matrix for texture characterization [21]. This paper uses three common gray-level statistical features along with four different angles (0°, 45°, 90° and 135°), which are presented as Equations 2, 3, and 4 [21].

\[
\text{Contrast} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (i - j)^2 P(i, j)  
\]

(2)

\[
\text{Inverse Difference Moment (IDM)} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \frac{P(i, j)}{1 + (i - j)^2} 
\]

(3)

\[
\text{Correlation} = \frac{\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (i - \mu_i)(j - \mu_j)P(i, j)}{\sigma_i \sigma_j} 
\]

(4)

3. Scene Image Classification

The scene images contain a large amount of semantic information. This semantic information is applied to classify each image into the appropriate scene class. Figure 2 presents the process of the proposed image classification mechanism based on the scene semantics. The size of each image is normalized in pre-processing. The low-level features of each image are extracted in image database. These features are applied to
The system then calculates scene similarity among images based on the important semantic features of each scene. Finally, the image is classified into the appropriate scene class and accurate semantic classification of scene image is achieved. The following subsections introduce the details.

3.1. Pre-processing

The classification result of the scene image is influenced by the size of each image. In pre-processing, the size of each image is normalized to \( D \times D \) cells in the image database, where \( D \) is a constant. Moreover, the proposed mechanism would be obtained more information from regional features. Each image is segmented equally into \( Dr \times Dr \) blocks, where \( Dr \) is a constant.

3.2. Image Feature Extraction

Images contain many low-level features such as color, texture, and shape. Previous studies that used color and texture to classify images achieved the good performance [3,6,12]. Therefore, this paper uses HSV and GLCM to extract low-level features from each image. Each segmented block of an image is extracted low-level features individual. Each low-level feature has \( Dr \times Dr \) information. Color and texture are quantified into \( c_i \) and \( t_i \) bins, respectively. Each pixel in an image is then distributed
into the corresponding bin according each low-level feature. Values of each bin are recorded to generate an image histogram. The color feature vector of each image is \( f^c_1, f^c_2, \cdots, f^c_i, \cdots, f^c_n \) and the texture feature vector is \( f^t_1, f^t_2, \cdots, f^t_i, \cdots, f^t_n \). The image feature is defined as concatenating all feature vectors.

### 3.3. LSA Model Construction

To enhance image classification accuracy, the important semantic low-level features are selected by LSA model. Images of the same scene in the image database are extracted feature to constitute a matrix \( H_{ij} \) (Figure 3), where \( I_1, I_2, \cdots, I_j, \cdots, I_n \) represents \( m \) training images. \( F_1, F_2, \cdots, F_i, \cdots, F_n \) represents features of an image scene. \( W_{ij} \) is the frequency of the \( j \)-th feature in the image \( i \). Therefore, each column represents the feature vector of each image and each row represents the frequency of each feature in all images.

The goal of this paper is to find the distinguishing features of a specific scene; the matrix for a specific scene can then be constructed. The SVD of LSA is used to analyze the matrix (Figure 4). Re-ordering the matrix of \( S \) requires re-ordering of columns of \( A \) and rows of \( B' \). The features corresponding with the larger singular values are more important. After matrix reorganization, the singular values are sorted from largest
to smallest based on their singular values.

The LSA discards the least significant values and retains the rest of the matrix. A threshold $\varepsilon$ is used to enhance the matrix reduction process, where $s_{k,k} \geq \varepsilon$ and $k \leq i$. Via the dimension reduction process, matrix noise is eliminated (Figure 5). Matrix $H_{ij}'$ is re-structured by $A'$, $S'$ and $B'$ (Figure 6). The frequency is summed according to each row of $H_{ij}'$ (Figure 7). Frequency corresponds to the $i$-th important feature, where $w_{i,total} = w_{i,1} + w_{i,2} + \cdots + w_{i,m}'$. Each image is classified based on the selected pre-$k$ important features of a specific scene.
3.4. Scene Similarity Measurement

The features with the largest singular values are more important in the LSA. To increase classification accuracy, this paper uses this characteristic to locate the pre-\( k \) suitable features of each scene. If there are \( r \) scenes exist, then \( r \) times LSA analysis are applied to locate the suitable features for \( r \) kinds of scene types. Equation 5 is used to derive scene similarity for an image with scene \( s \).

\[
C_s = \frac{(\log S_s^{i,i})^n}{\log S_{total}} h_s^{i} + \frac{(\log S_s^{i,i})^n}{\log S_{total}} h_s^{i} + \cdots + \frac{(\log S_k^{i,i})^n}{\log S_{total}} h_k^{i} + \cdots + \frac{(\log S_k^{i,i})^n}{\log S_{total}} h_k^{i} 
\]  

(5)

where \( S_s^{i,i} \) is the singular value corresponding to the \( i \)-th important feature in the \( s \)-th scene and \( S_{total} = S_s^{i,i} + S_s^{i,i} + \cdots + S_k^{i,i} \). If an image has the \( i \)-th important feature for the \( s \)-th scene, then \( h_s^{i} \) is the ratio of important feature \( i \) in an image; otherwise \( h_s^{i} = 0 \). Let \( n \) be the magnification factor. Different scenes have their own magnification factor \( n \) to enhance the importance ratio \( h_s^{i} \).

3.5. Image Classification

After LSA analysis, different scenes have different important features and corresponding singular values. Each image can calculated \( r \) degrees for each scene similarities, that is \( C_1, C_2, \cdots, C_r \). Therefore, this paper can classify an image as scene \( t \) when \( C_t \) is the largest scene similarity. Equation 6 is the majority function.
$C_i = \text{Max}(C_1, C_2, \cdots, C_r)$  \hspace{1cm} (6)

4. Experimental Results

This section has two subsections. The first subsection presents the feature selection for each scene. The second subsection shows classification performance results. The image database in this experiment was downloaded from http://wang.ist.psu.edu/docs/related.shtml (2010) [23]. This database has 10 different scenes in 788 images. Table 1 shows one example of each scene and the number of scene images. All images are stored in the JPEG format and are $384 \times 256$ pixels in size. The experimental environment was as follows. The hardware was an Inter Core2 Quad Q8200 CPU with 2.00GB RAM running Microsoft Windows XP. All programs were implemented using MATLAB 2007b.

Table 1 HERE

4.1. Feature Selection of Each Scene

In this paper, each image is segmented into 64 blocks; each block is $48 \times 32$ pixels. The HSV model is used and quantified into 40 bins, with 10 bins for the H value, 2 bins for the S value, and 2 bins for the V value. The texture attribute model is quantified into 40 bins, composed of 10 bins for Contrast, 2 bins for Correlation, and 2
bins for IDM. This experiment addresses two feature types. This paper randomly selects 30 training images of each scene from the image database. Via LSA, the important bins of each scene are selected. The first experiment identified the distribution of important features in the two feature types (Table 2). Before applying the proposed method, the number of selected bins is set at 5, 10, 15 and 20 in the experiment. For instance, for Building scene are 5.5 bins of the color feature and 7 bins of the texture feature are selected in average case at the sixth column. For the Flower scene, 10 bins of the color feature and 2.5 bins of the texture feature are selected in average case at fifth column. This shows that different scenes have different numbers of important bins. The overall average number of bins selected for color and texture is 6.675 and 5.825, respectively.

Table 2 HERE

### 4.2. Classification Performance

This subsection discusses image classification performance, which is improved by the proposed mechanism. The performance of the proposed mechanism is measured by the correct classification ratio (CCR) using Equation 7.

\[
CCR = \frac{\text{Correct classification images}}{\text{Test images}}
\]  

(7)

The second experiment shows the CCR for different numbers of selected bins (Table 3). The total number of correctly classified images is 472 and the average CCR is 59.90% for ten scenes with 5 bins. The average CCR of the 15 bins is 71.95%, which is better performance
than that for the other bins. Therefore, the moderation scale of selected bins can help classify scene images effectively.

Table 3 HERE

Table 4 shows the experimental result for the third experiment, which compares the performance of the proposed mechanism with that of other classification methods such as SVM, K-NN, Bayesian, and Frequency Analysis (FA). In the initial stage, this paper selected important 15 bins in this experiment. The total number of correctly classified images is 567 and the average CCR is 71.95% for the ten scenes. Experimental results demonstrate that the proposed mechanism has better retrieval performance than the other methods, indicating that the proposed image classification scheme is efficient. The proposed mechanism can be used to filter out noise, stand out the latent semantic information, and choose the important bins for each scene using SVD. Therefore, the important bins selected for low-level features can represent all bins in each scene.

Table 4 HERE

5. Conclusions

To retrieve relevant images from a large image database rapidly, images are classified into suitable semantic classes based on the low-level features in each image.
Therefore, reducing the gap between low-level features and the high-level semantic is challenging. This paper applies LSA to select the important semantic features in scenes and classifies scene image. Generally, LSA has good classification results for text data. This paper modifies the LSA model and applies it to image data to locate latent semantic features. This paper also develops an effective scene similarity measurement. Experimental results indicate that the proposed method has excellent performance in scene image classification. Future work will explore how to integrate low-level features other than color and texture features to increase classification accuracy further and apply the proposed mechanism to image retrieval to extract the important features of multi-query.

**Reference**


Figure 1 HSV color space model\textsuperscript{21}.

\begin{center}
\includegraphics[width=0.5\textwidth]{hsv_color_space.png}
\end{center}

H : specific angles around the vertical axis  
S : the radial distance from the vertical axis  
V : along the central axis

Figure 2 The process of the proposed image classification mechanism.
Figure 3 A scene matrix $H_y$.

Figure 4 The matrices of SVD.

Figure 5 The matrices of reduced dimension.

Figure 6 The re-structured scene matrix $H_{ij}'$. 
Figure 7 The feature matrix.
### Table 1
The example of each scene

<table>
<thead>
<tr>
<th>Scene</th>
<th>Images</th>
<th>Scene</th>
<th>Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>62</td>
<td>Snow</td>
<td>64</td>
</tr>
<tr>
<td>Flower</td>
<td>86</td>
<td>Food</td>
<td>61</td>
</tr>
<tr>
<td>Coast</td>
<td>83</td>
<td>Elephant</td>
<td>85</td>
</tr>
<tr>
<td>Bus</td>
<td>98</td>
<td>Building</td>
<td>69</td>
</tr>
<tr>
<td>Dinosaur</td>
<td>100</td>
<td>Horse</td>
<td>80</td>
</tr>
</tbody>
</table>

### Table 2
The number of bins of the selected feature in each scene

<table>
<thead>
<tr>
<th>Scene</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Color</td>
<td>Texture</td>
<td>Color</td>
<td>Texture</td>
<td>Color</td>
</tr>
<tr>
<td>Human</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>Snow</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>Elephant</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Bus</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>Flower</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>Building</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Food</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Dinosaur</td>
<td>4</td>
<td>1</td>
<td>7</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>Coast</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Horse</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Average</td>
<td>2.3</td>
<td>2.7</td>
<td>4.7</td>
<td>5.3</td>
<td>8</td>
</tr>
</tbody>
</table>
Table 3
The C.C.R. of each scene using LSA

<table>
<thead>
<tr>
<th>Scene</th>
<th>Images</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>62</td>
<td>34</td>
<td>44</td>
<td>45</td>
<td>27</td>
</tr>
<tr>
<td>Snow</td>
<td>64</td>
<td>40</td>
<td>38</td>
<td>43</td>
<td>47</td>
</tr>
<tr>
<td>Elephant</td>
<td>85</td>
<td>35</td>
<td>48</td>
<td>53</td>
<td>52</td>
</tr>
<tr>
<td>Bus</td>
<td>98</td>
<td>55</td>
<td>53</td>
<td>71</td>
<td>62</td>
</tr>
<tr>
<td>Flower</td>
<td>86</td>
<td>54</td>
<td>69</td>
<td>70</td>
<td>77</td>
</tr>
<tr>
<td>Building</td>
<td>69</td>
<td>37</td>
<td>42</td>
<td>41</td>
<td>35</td>
</tr>
<tr>
<td>Food</td>
<td>61</td>
<td>39</td>
<td>43</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>Dinosaur</td>
<td>100</td>
<td>91</td>
<td>98</td>
<td>99</td>
<td>100</td>
</tr>
<tr>
<td>Coast</td>
<td>83</td>
<td>27</td>
<td>29</td>
<td>32</td>
<td>30</td>
</tr>
<tr>
<td>Horse</td>
<td>80</td>
<td>60</td>
<td>67</td>
<td>72</td>
<td>72</td>
</tr>
<tr>
<td>Total</td>
<td>788</td>
<td>472</td>
<td>531</td>
<td>567</td>
<td>544</td>
</tr>
</tbody>
</table>

C.C.R. 59.90% 67.39% 71.95% 69.04%

Table 4
The performance of the different methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Proposed Mechanism</th>
<th>SVM</th>
<th>K-NN</th>
<th>Bayesian</th>
<th>FA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td>567</td>
<td>431</td>
<td>564</td>
<td>261</td>
<td>530</td>
</tr>
<tr>
<td>C.C.R.</td>
<td>71.95%</td>
<td>54.70%</td>
<td>71.57%</td>
<td>33.12%</td>
<td>64.97%</td>
</tr>
</tbody>
</table>