PICTURE-GRAPHICS COLOR IMAGE CLASSIFICATION

Salil Prabhakar1, Hui Cheng2, John C. Handley3, Zhigang Fan3, and Ying-wei Lin3

1DigitalPersona Inc., 2Sarnoff Corp., 3Xerox Corp.
Redwood City, CA, 94063 Princeton, NJ, 08543 Webster, NY, 14580
salilp@digitalpersona.com, hcheng@sarnoff.com, {jhandley, zfan, ylin}@crt.xerox.com

ABSTRACT

High-level (semantic) image classification can be achieved by analysis of low-level image attributes geared for the particular classes. In this paper, we have proposed a novel application of the known image processing and classification techniques to achieve such a high-level classification of color images. Our image classification algorithm uses three low-level image features: texture, color, and edge characteristics to classify a color image into two classes: business graphics or natural picture. We have achieved an accuracy of 96.6% on our database of 209 images using a combination of tree and neural network classifiers.

1. INTRODUCTION

During the past several decades, monitors, printers, and copiers have evolved from monochrome to color. With the increasing use of color products and services, there is a growing demand for high quality reproduction, rendering, and transmission of color images.

Depending on the source, different images, such as “natural pictures” and “business graphics” (including text) have different characteristics and generally require to be treated differently in a given imaging pipeline. For example, in an image reproduction application, the best gamut mapping algorithms [1, 2] for business graphics and natural pictures are different. While Inverse-Power-Inverse followed by Centroid Clipping is most effective for natural pictures, Rotated Device Mapping achieves the best results for business graphics [1, 2]. In addition to gamut mapping, enhancement, rendering, and compression [3] algorithms are also different for graphics and pictures. However, the origin or “type” information about a scanned image is usually unavailable. Therefore, a picture-graphics classifier is required to automatically extract the type information from the scanned image.

Image classification is an important problem in pattern recognition and has been used in several different applications. For efficient image retrieval, Vailaya et al. [4] proposed methods for indoor-outdoor, city-landscape, and sunset-mountain classification. For document compression and reproduction, Revankar and Fan [5] and Cheng and Bouman [3] proposed text, graphics, picture, and background classification as part of an image segmentation system.

In this paper, we propose an algorithm that classifies an image as either picture or graphics. This high-level classification is achieved by the analysis of three low-level image attributes that represent color, texture, and edge information in an image. In addition, the classification is performed by a combination of a rule-based tree classifier and a neural network classifier. While the tree classifier is more effective in establishing a piecewise linear boundary and interpreting it, neural network classifier is good at modeling a complex non-linear boundary and is used only when the tree classifier rejects an image.

Fig. 1: Representative examples of picture and graphics.

2. FEATURES

As shown in Fig. 1, natural pictures and business graphics are different in several ways. A typical natural picture has many details, smooth color changes from one part of the
image to another, and a variety of rich textures from the presence of walls, ceilings, sky, vegetation, water, terrain, etc. On the other hand, a typical synthetic graphics is smoother, has a uniform background, many lines and drawings, text labels, homogeneous patches of colors, and abrupt changes from region to region. In past, many different types of features have been used for this particular classification problem. Revankar and Fan [5] proposed the ratio of edges to image region and ratio of strong edges to weak edges. Jain and Yu [6] and Fan et al. [7], both proposed the proportion of black pixels in binarized image to identify picture regions. Schettini et al. [8] used a large number of low-level color, texture, and shape features and their saliency derived from sub-images to train a tree classifier. However, these previous approaches either do not provide satisfactory accuracy or are computationally expensive for our applications. Therefore, we propose the following three texture, color, and edge features that are very fast to compute and improve the accuracy.

2.1. Spatial Gray Level Dependence Texture Features

Synthetic graphics are usually very smooth and contain very little texture. Natural pictures often contain water, sky, vegetation, buildings, walls, and objects in them and thus are texture rich. A variety of texture features can be extracted for image classification. We use a modified spatial gray level dependence (SGLD) approach, which is relatively simple in computation. SGLD was originally proposed for texture analysis in multi-level images [9]. It establishes a set of two-dimensional histograms that record the joint probability occurrence of gray levels i and j for two pixels with a defined spatial relationship in an image. The spatial relationship is represented by a distance and orientation. We use only one histogram that combines the relationships in the horizontal and the vertical directions at a fixed distance d (d is typically 1 or 2). We use only the luminance component in the image to generate the modified SGLD histogram because it contains most of the texture energy. The histogram is first initialized to zero and then the entry $SGLD[Y(m,n),X(m,n)]$ is incremented by one for each pixel with a value of $Y(m,n)$ and a neighbor with a value of $X(m,n)$. $X(m,n)$ is determined as:

$$
X(m,n) = \begin{cases} 
Y(m,n + d), & \text{if } |Y(m,n + d) - Y(m,n)| > |Y(m + d, n) - Y(m,n)| \\
Y(m + d, n), & \text{otherwise}
\end{cases}
$$

Applying this to one dimension, if pixel $(m,n)$ is in a flat area where $Y(m,n)$ is equal to $X(m,n)$, the entry $[Y(m,n),X(m,n)]$ is on the diagonal. On the other hand, if $(m,n)$ is on an edge, the difference between $Y(m,n)$ and $X(m,n)$ will be significant, and the entry will be far away from the diagonal. As illustrated in the examples given in Fig. 2, the SGLD histograms differ for graphics and pictures. The graphics images have narrower diagonal peaks. In addition, there exist many entries that are far away from the diagonal. This is more evident when the 2-D histograms are sliced at horizontal axis into 1-D histograms. The 1-D histograms for the graphics are composed of a peaky Gaussian contributed by the pixels in the flat regions, and many spikes that represent contributions from the edges. These spikes make the overall distribution non-Gaussian. Furthermore, they are usually non-symmetrical about the main peak. On the other hand, the 1-D histograms for the pictures can be well modeled as a Gaussian. These differences are captured by skewness ($S$) and variance ratio ($V$) of the SGLD histogram. $S$ measures the average skewness of the 1-D histograms. It is smaller for pictures than for graphics. $V$ compares the variances estimated using two methods. It is calculated as $V = \sum_i \alpha_i \sigma_{1i}^2 / \sigma_{2i}^2$ where $\alpha_i$ is a weighting factor proportional to $C_i$, the population of the i-th 1-D histogram, $\sigma_{1i}^2$ is the sample variance, and $\sigma_{2i}^2$ is defined such that

$$
\int_{d=0}^{d_0} SGLD(m,m + d) + SGLD(m,m - d) = 0.6 \times C_i.
$$

A graphic image has a smaller $\sigma_{2i}^2$ due to the narrow main peak and a larger $\sigma_{1i}^2$, due to the contributions from the additional spikes. As a result, $V$ is large. In comparison, for the pictures, $\sigma_{1i}^2$ and $\sigma_{2i}^2$ are similar and the ratio is close to 1.

![Fig. 2: Corresponding modified SGLD histograms and value of the feature V for images shown in Fig. 1.](image)

2.2. Color discreteness features

We extract color features in the CIELUV space and therefore, we first apply a transform to convert the input color space to CIELUV color space. The CIELUV color space provides perceptual uniformity, that is, the Euclidean distance between a pair of colors specified in CIELUV roughly correspond to perceived difference in color, and therefore is preferred. Other color spaces that contain a luminance component and two chrominance channels can also be used here. We first smooth the image with a 4x4 averaging filter to remove the noise due to half-tone. We then compute the color histograms ($H_L$, $H_U$, $H_V$) for the three channels, $L$, $U$, and $V$, respectively, from the image and normalize them by the number of pixels in the image. The color histograms are invariant under rotation and translation of the input image and the normalization provides scale invariance. If $I(i)$ is a bin in the histogram $I$ of an image, then the normalized histogram $H$ is defined as: $H(i) = I(i) + \sum_{i=0}^{C_i} I(i)$. Since
graphics are generated using limited number of colors, graphics images usually contain areas of uniform colors and hence the color histograms contain many sharp peaks. On the other hand, natural pictures contain smoothly varying colors due to lighting and hence the color histograms contain a few smooth peaks. This difference in the histograms is captured by the color discreteness statistics: 

\[ R = \sum_{i=1}^{GL} |H(i+1) - H(i)| \]

where \( GL \) is the number of bins in the H color histogram (typically, 256). We use three color discreteness features (\( R_L, R_U, \) and \( R_V \)) for the three color channels \( L, U, \) and \( V \), respectively. See Fig. 3 for the \( H_L \) histograms and the corresponding values of the feature \( R_L \) for the images shown in Fig. 1.

\[ \text{Fig. 3: Corresponding } H_L \text{ histograms and values of the feature } R_L \text{ for the images shown in Fig. 1.} \]

### 2.3. Edge Feature

Graphics images contain several areas of uniform color, line drawings, text, and have very sharp, prominent, and long edges. On the other hand, natural pictures are very noisy and contain short broken edges. Therefore, we propose to use statistics based on edges in an image to distinguish between picture and graphics. First, we use a standard Canny edge detector (sigma = 1, mask size = 9, lower threshold = 0.5, higher threshold = 0.9) to extract edges from an image. Then we use a standard 8-connected connected component algorithm to find all the connected edges in the edge-map. The average number of pixels per connected edge in the image is used as a feature (\( E = \text{No. of Edge Pixels} / \text{No. of Connected Edges} \)). Typically, graphics have fewer connected edges but each connected edge consists of large number of pixels while pictures have lots of connected edges with very few pixels in each connected edge (see Fig. 4).

\[ \text{Fig. 4: Corresponding edge maps and values of the feature } E \text{ for the images shown in Fig. 1.} \]

### 3. CLASSIFIER

Classifiers that can be used to solve a certain classification problem include statistical, structural, neural networks, fuzzy logic, and machine learning classifiers. Several of these classifiers are available in public domain and commercial packages. However, no single classifier seems to be a clear winner in solving all complex real-world problems. Each classifier has its own strengths and weaknesses. A combination of classifiers can perform better that any single classifier alone. Our experiments confirm this claim as our best classifier uses a combination of a tree classifier and a neural network classifier. A neural network, a two-stage neural network, CART, C4.5, and Logistic Regression classifiers gave about 13-18 (6.2%-8.6%) errors on our database of 209 images when used alone. During our tests, we observed that the CART classifier gave significant importance to the first color discreteness feature (\( R_L \)). We also observed that the edge feature (\( E \)) was very accurate only at large values, i.e., if the feature value is large, it is almost certain that the image is a graphics. However, when the edge feature value is small, nothing can be said about whether the image is picture or graphics. This is because, in certain cases, the value of \( E \) is low for scanned graphics that have low frequency half-tone or complex background. We combined all these observations into a rule-based tree classifier that uses a standard back-propagation neural network (with one neuron in its hidden layer) at one of its nodes (see Fig. 5). The tree classifier uses several thresholds. We determined these thresholds...
empirically and set them as: $TE=120$, $TH = 0.15$, and $TL = 0.10$. The rule-based portion of the classifier does not need any training. The neural network was trained with the samples that were correctly classified by the rule-based classifier.

![Diagram of the classification algorithm](image)

**Fig. 5:** Picture-graphics classification algorithm.

### 4. RESULTS

This combined classifier made only 7 (3.4%) errors (see Fig. 6) on our database, thus giving us far better performance than any single classifier alone. The analysis of the errors made by this classifier indicates that the misclassifications of graphics into picture were made for those graphics images that have a lot of details in them. Even though these graphics images are synthetic, they look like natural pictures. The picture images that are misclassified as graphics have very little detail in them. Also, from a reproduction point of view, pictures misclassified as graphics incur more penalty than graphics misclassified as picture. So, essentially, our classifier made six mild errors on graphics images while only one serious error on the picture images, and thus is suitable for the application in consideration. Although our current database is small, these results are very encouraging. A future version of our classifier will classify the input image into picture, graphics, text (text is currently classified as graphics), and mixture and will be tested on a much larger database.

![Example of errors made by our classifier](image)

**Fig. 6:** Example of errors made by our classifier. (a)-(c) are of the mild type, i.e., graphics mis-classified as pictures; (c) is the only case where a picture is mis-classified as a graphics.

### 5. REFERENCES


