Binarization of Color Characters in Scene Images
Using $k$-means Clustering and Support Vector Machines

Kohei Kita  Toru Wakahara
Faculty of Computer and Information Sciences
Hosei University
3-7-2 Kajino-cho, Koganei-shi, Tokyo, 184-8584 Japan
E-mail: wakahara@hosei.ac.jp

Abstract—This paper proposes a new technique for binarizing multicolored characters subject to heavy degradations. The key ideas are threefold. The first is generation of tentatively binarized images via every dichotomization of $k$ clusters obtained by $k$-means clustering in the HSI color space. The total number of tentatively binarized images equals $2^k - 2$. The second is use of support vector machines (SVM) to determine whether and to what degree each tentatively binarized image represents a character or non-character. We feed the SVM with mesh and weighted direction code histogram features to output the degree of “character-likeness.” The third is selection of a single binarized image with the maximum degree of “character-likeness” as an optimal binarization result. Experiments using a total of 1000 single-character color images extracted from the ICDAR 2003 robust OCR dataset show that the proposed method achieves a correct binarization rate of 93.7%.

Keywords—figure-ground discrimination; binarization of multicolored characters; $k$-means clustering; support vector machines;

I. INTRODUCTION

Recently, recognition of web documents and characters in natural scenes has emerged as a hot, demanding research field [1]. In particular, recognition of characters in scene images with a wide variety of image degradations, multiple colors, and complex backgrounds poses the following two key problems: figure-ground discrimination or correct binarization and distortion-tolerant character recognition. This paper addresses the first problem of figure-ground discrimination or binalization of color characters.

Most of binarization methods are based on global, local/adaptive or multi-stage selection of threshold [2], [3]. Wang et al. [4] applied color-based clustering to scene images for locating and binarizing characters assuming that characters have a uniform, single color. Yokobayashi et al. [5] proposed the technique for selecting a maximum separability axis in the color space and an appropriate threshold for binalization. However, these techniques could not deal with multicolored characters and/or heavy image degradations.

To resolve the above-mentioned problem, this paper proposes a new, promising technique composed of three parts: generation of tentatively binarized images via every dichotomization of $k$ clusters obtained by $k$-means clustering in the HSI color space, use of support vector machines (SVM) to determine the degree of “character-likeness” of each tentatively binarized image, and selection of a single binarized image with the maximum degree of “character-likeness” as an optimal binalization result.

The proposed method in this paper is a preparatory and steady step to our main goal of binarizing multicolored degraded character strings in scene images.

Experiments made on single-character color images extracted from the ICDAR 2003 robust OCR dataset show that the proposed method can successfully binarize multicolored characters subject to heavy image degradations.

II. GENERATION OF TENTATIVELY BINARIZED IMAGES USING $k$-MEANS CLUSTERING

We apply $k$-means clustering to points in the HSI color space for a given input image, and generate a set of tentatively binarized images by every dichotomization of a total of $k$ clusters or subimages.

First, values of $R$, $G$, and $B$ in the RGB color space are converted to values of $H$, $S$, and $I$ in the HSI color space, where $H$, $S$, and $I$ represent hue, saturation, and intensity, respectively [6]. In particular, we scale each value of $H$, $S$, and $I$ to range from 0 to 255. When an input image has $M \times N$ pixels, a total of $M \times N$ points corresponding to those pixels are scattered in the HSI color space.

Preliminary experiments showed that the conversion from the RGB color space to the HSI color space helped the clustering of colors used for contrasting characters against backgrounds.

Second, $k$-means clustering is applied to a total of $M \times N$ points in the HSI color space to generate $k$ clusters, where a number of clusters, $k$, is determined in advance. Of course, the parameter $k$ should be determined so that a number of tentatively binarized images is necessary and sufficient to include a correctly binalized image.

The $k$-means clustering algorithm or nearest mean reclassification algorithm [7] is as follows.

Step 1: Select $k$ points at random from a total of $M \times N$ points scattered in the HSI color space as initial
cluster centers, \( \{ \mu_i^{(\tau=0)} \}_{i=1}^k \), \( \tau \) specifies an iteration number. Then, assign each of \( M \times N \) points to its nearest cluster center among \( \{ \mu_i^{(\tau=0)} \}_{i=1}^k \), and a set of points assigned to the same cluster center forms one cluster.

**Step 2:** Compute a mean vector of each cluster and set the mean vector as an update on its cluster center. Then, \( \tau = \tau + 1 \), and cluster centers thus updated are denoted by \( \{ \mu_i^{(\tau)} \}_{i=1}^k \).

**Step 3:** Each point is re-assigned to a new set according to which is the nearest cluster center among \( \{ \mu_i^{(\tau)} \}_{i=1}^k \), and each new set of points corresponds to a cluster. If there is no further change in the grouping of the data points, output the present \( k \) clusters as the clustering result and stop. Otherwise, go to **Step 2**.

Also, it is well known that the \( k \)-means clustering results depend on the initial selection of \( k \) cluster centers. Therefore, we adopt the multi-start \( k \)-means clustering technique. That is, we choose the clustering result with the minimum within-cluster variance among multiple trials with different initializations.

Then, by inverse mapping of a set of points forming each cluster in the HSI color space onto a 2D image plane, respectively, we obtain a total of \( k \) subimages the sum of which is equivalent to the input image.

Finally, we dichotomize \( k \) subimages into two groups, and set values of pixels belonging to the one group at 0 (black) and the other group at 255 (white). As a result, we obtain one binarized image, where black pixels represent figure and white pixels represent background. By considering every possible dichotomization of \( k \) subimages we can generate multiple tentatively binarized images the total number of which, \( N_{\text{binary}} \), is given by

\[
N_{\text{binary}} = \sum_{i=1}^{k-1} \binom{k}{i} = 2^k - 2, \tag{1}
\]

where \( \binom{k}{i} \) denotes a binomial coefficient.

Fig. 1 shows one example of generation of tentatively binarized images from an input image.

From Fig. 1, it is seen that a correctly binarized image is included in a set of tentatively binarized images even when the input character “M” is in two colors.

**III. SELECTION OF A CORRECTLY BINARIZED IMAGE VIA CHARACTER-LIKENESS USING SVM**

In this section we calculate the degree of “character-likeness” of each tentatively binarized image using SVM in an appropriately chosen feature space. In advance, SVM is trained to determine whether and to what degree each binarized image represents a character or non-character. Then, we output the binarized image with the maximum degree of “character-likeness” as an optimal binarization result.

First, we extract a feature vector from a binary image so that a feature vector should represent a kind of “character-likeness” as much as possible. Selection of a good feature vector is a clue in achieving the high ability of SVM that determines whether and to what degree each binarized image represents a character or non-character in the feature space.

As preprocessing, position and size normalization is conducted by using 1st and 2nd moments. Namely, the center of gravity of black pixels is shifted to the center of the image, and the second moment around the center of gravity is set at the predetermined value. Then, we set a size of a preprocessed binary image at \( 80 \times 120 \) pixels.

Next, we extract two kinds of feature vectors well-known in the field of character recognition: mesh feature and weighted direction code histogram feature [8].

**Mesh feature:**

We divide the input binary image into a total number of \( 8 \times 12(=96) \) square blocks and, then, calculate the percentage of black pixels in each block. Finally, those measurements together form the 96-dimensional mesh feature vector.

**Weighted direction code histogram feature:**

One of 4-directional codes, i.e., \( H \) (horizontal), \( R \) (right-diagonal), \( V \) (vertical), and \( L \) (left-diagonal), is assigned to every contour pixel of black regions. Then, we divide the input binary image into a total number of \( 8 \times 12 \) square blocks, and count the number of contour pixels assigned to \( H \), \( R \), \( V \), and \( L \), respectively, in each block. Their measurements together form the 384-dimensional feature vector. Then, a locally weighted sum using a \( 5 \times 5 \) Gaussian mask around each block taken at intervals of two blocks, horizontally and vertically, reduces the dimension of the feature vector from 384 to 96.

The support vector machines (SVM) map the input feature
vectors, \( x \), into a high-dimensional feature space through nonlinear mapping, \( \Phi(x) \), to construct an optimal separating hyperplane that maximizes the margin between two classes.

Then, SVM assigns a new data point, \( x \), to one class or the other, according to the sign of \( f(x) \) given by

\[
f(x) = \sum_{i} \alpha_i y_i (\Phi(x_i) \cdot \Phi(x)) - b
\]

where \( \{x_i\} \) are training data with corresponding target values \( \{y_i\} \) where \( y_i \in \{-1, +1\} \); non negative coefficients \( \{\alpha_i\} \) and a scalar \( b \) are trained to maximize the margin in advance.

We implemented SVM via SVM\textsuperscript{light} [9], and used the RBF kernel function given by

\[
K(x, y) = \exp \left(-||x - y||^2\right)
\]

Also, training data fed into SVM are as follows.

Training data for \textit{character} class:

We selected correctly binarized images from a total of \((2^k - 2)\) tentatively binarized images obtained for each training sample. Furthermore, we added a total of 136 available font sets to the training data.

Training data for \textit{non-character} class:

We selected incorrectly binarized images from a total of \((2^k - 2)\) tentatively binarized images obtained for each training sample.

Fig. 2 shows examples of training data for character and non-character classes.

From Fig. 2, it is seen that we chose not only perfectly but also nearly correctly binarized images as training data belonging to a character class.

Finally, we select a single tentatively binarized image with the maximum value of \( f(x) \) among those obtained for \((2^k - 2)\) tentatively binarized images and output the selected one as an optimal binarization result.

IV. EXPERIMENTAL RESULTS

In experiments, we used a total of 1,000 single-character color images extracted from “TrialTrain” subset of ICDAR 2003 robust OCR dataset [10].

Fig. 3 shows examples of images with multiple colors, degradations, and complex backgrounds.

Fig. 2. Examples of training data. (a) Character class (correctly binarized images). (b) Character class (available fonts). (c) Non-character class (incorrectly binarized images).

Here, we regard the value of \( f(x) \) of (2) as estimating the degree of “character-likeness”, and also assume that the larger the value of \( f(x) \) is the more its character-likeness is.

In preliminary experiments, we examined the values of \( k \) ranging from 3 to 6 in the \( k \)-means clustering to generate tentatively binarized images. Too small a \( k \) value fails to generate a correctly binarized image while too large a \( k \) value generates surplus binarized images and increases the processing time. Actually, the number of clusters, \( k \), in the \( k \)-means clustering was set at 5, and, hence, a total number of tentatively binarized images was 30 \((=2^5 - 2)\).

First, we evaluated the ability of discrimination between character and non-character via SVM using three kinds of feature vectors: mesh feature, weighted direction code histogram feature, and concatenation of these two features. Also, we applied the technique of 10-fold cross-validation [7] to a total of 1,000 samples for training and validation.

Fig. 4 shows ROC (Receiver Operating Characteristics) curves obtained by moving the threshold on the value of \( f(x) \) of (2) for discrimination between character and non-character, where FRR and FAR denote false reject rate and false acceptance rate, respectively.

From Fig. 4, it is found that SVM fed with the concatenation of mesh and weighted direction code histogram features achieved the minimum EER (Equal Error Rate) of 5.7%.

Next, we investigated the ability of selecting a correctly binarized image from a total of 30 candidate binary images based on the outputs of SVM fed with the concatenation of mesh and weighted direction code histogram features. Namely, a total of 30 candidate binary images were arranged in the decreasing order of SVM outputs, and the top one was selected as an optimally binarized image.

Fig. 5 shows cumulative binarization rates. The \( p \)th cumulative binarization rate is an average rate at which the top \( p \) candidate binary images arranged in the decreasing order of SVM outputs contain a correctly binarized image.

From Fig. 5, it is found that the correct binarization rate or the 1st cumulative binarization rate is 93.7%, and the 7th cumulative binarization rate exceeds 99.9%.

Figure 3. Examples of images used in our experiments.
From these results, we can say that the proposed method provides a very promising tool for binarizing multicolored characters with a variety of image degradations and complex backgrounds.

V. CONCLUSION

Binarization of color, low-quality characters in scene images is most challenging as a crucial step to the success of subsequent recognition.

This paper proposed a very promising solution composed of three steps; generation of tentatively binarized images via \(k\)-means clustering in the HSI color space, evaluation of the degree of “character-likeness” of binarized images using SVM, and selection of a single binarized image with the maximum degree of “character-likeness” as an optimal binarization result.

Experiments using the ICDAR 2003 robust OCR dataset containing single-character images in natural scenes showed that the proposed method achieved a correct binarization rate of 93.7%.

Future work is to reinforce this technique so as to tackle the problem of binarization of multicolored character strings or words in scene images.

REFERENCES


