2011 International Conference on Document Analysis and Recognition

Binarization of Color Character Strings in Scene Images Using K-means Clustering and Support Vector Machines

Toru Wakahara Kohei Kita
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 addresses the problem of binalizing multicolored character strings in scene images subject to heavy image degradations and complex backgrounds. The proposed method consists of four steps. The first step generates tentatively binarized images via every dichotomization of K clusters obtained by K-means clustering of constituent pixels of a given image in the HSI color space. The total number of tentatively binarized images equals $2^K - 2$. The second step divides each binarized image into a sequence of "single-character-like" images using an average aspect ratio of a character. The third step is use of support vector machines (SVM) to determine whether each "single-character-like" image represents a character or non-character. We feed the SVM with the mesh feature to output the degree of "character-likeness." The fourth step selects a single binarized image with the maximum average of "character-likeness" as an optimal binarization result. Experiments using a total of 1000 character strings extracted from the ICDAR 2003 robust word recognition dataset show that the proposed method achieves a correct binarization rate of 80.8%.**

***Keywords*-binarization of multicolored character strings; 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 three key problems: detection and localization of characters, figure-ground discrimination or correct binarization, and distortion-tolerant recognition of binarized characters. This paper addresses the second 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. Therefore, these techniques could not deal with multicolored characters and/or heavy image degradations.

In our previous paper [5] we proposed the technique for optimally binalizing a multicolor single-character image using K-means clustering and support vector machines (SVM). The key idea was use of SVM to calculate the degree of "character-likeness" as a two-class classification problem between character and non-character. However, binalization of not a single-character image but a character string image remained unsolved as a future work.

This paper proposes an extended version of our previous method [5] in order to be able to binarize multicolored, degraded character strings in scene images.

Our new method consists of four steps: (1) generation of tentatively binarized images via every dichotomization of K clusters obtained by K-means clustering of constituent pixels of a given image in the HSI color space, (2) segmentation of every binarized image into a sequence of "single-character-like" images using an average aspect ratio of a character, (3) use of support vector machines (SVM) to calculate the degree of "character-likeness" of each single-character-like image, and, (4) selection of a single binarized image with the maximum average of "character-likeness" as an optimal binarization result. All steps but the first one are newly introduced.

Experiments made on character strings extracted from the ICDAR 2003 robust word recognition dataset show that the proposed method can successfully binarize multicolored character strings subject to heavy image degradations and complex backgrounds.

II. ICDAR 2003 robust word recognition dataset

Several datasets used in ICDAR 2003 robust reading competitions are available for download from the website [6]. We use the robust word recognition dataset containing JPEG character string images in natural scenes. In particular, we select a total of 1000 images from "TrialTrain" subset.

Fig. 1 shows examples of character string images used in our experiments.

From Fig. 1, it can be seen that those examples include multicolored character strings and also are subject to a wide variety of image degradations and complex backgrounds.

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.63

Figure 1. Examples of color character strings used in our experiments.

III. Generation of Tentatively Binarized Images Using K-means Clustering

We apply K-means clustering to constituent pixels of a given image in the HSI color space, and generate 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. 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 was useful 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 set of generated binarized images never fails 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 $\{\boldsymbol{\mu}_i^{(\tau=0)}\}_{i=1}^{K}$. τ specifies an iteration number. Then, assign each of $M \times N$ points to its nearest cluster center among $\{\boldsymbol{\mu}_i^{(\tau=0)}\}_{i=1}^{K}$, 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 $\{\boldsymbol{\mu}_i^{(\tau)}\}_{i=1}^{K}$.

Step 3 : Each point is re-assigned to a new set according to which is the nearest cluster center among $\{\boldsymbol{\mu}_i^{(\tau)}\}_{i=1}^{K}$, and each new set of points corresponds to a cluster. If there is no further change in the grouping of the points, output the present K clusters as the clustering result and stop. Otherwise, got 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_{binary}, is given by

$$N_{binary} = \sum_{i=1}^{K-1} \binom{K}{i} = 2^K - 2, \tag{1}$$

where $\binom{K}{i}$ denotes a binomial coefficient.

Fig. 2 shows one example of generation of tentatively binarized images by K-means clustering.

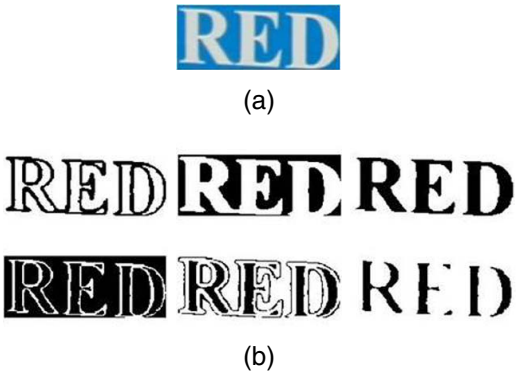

Figure 2. One example of generation of tentatively binarized images by K-means clustering (K = 5). (a) An input image. (b) A part of tentatively binarized images the total number of which equals 30 (= $2^5 - 2$).

From Fig. 2, it is seen that a correctly binarized image is included in those tentatively binarized images.

IV. Segmentaion of a Binarized Image into Single-character-like Images

Conventional techniques for extracting characters from scene images make full use of characteristics of words or character strings directly.

For example, Ashida et al. [9], who took the first prize in ICDAR 2003 Text Locating Competition [8], extracted several kinds of features from a candidate region of a word or a character string: distributions of runs of black pixels, black-white reversing frequencies in horizontal and vertical directions, local densities of black pixels, and an aspect ratio of the candidate region.

However, those features of a word or a character string have been investigated in a rather ad hoc manner.

On the other hand, in our previous paper [5], we calculated successfully the degree of "character-likeness" using well-known recognition features extracted from tentatively binarized images of a single-character image.

Therefore, in this section, we propose to segment every tentatively binarized image into a sequence of "single-character-like" images. It is to be noted that those segmented images are not single-character images but "single-character-like" ones.

In particular, we use an average aspect ratio of a character to estimate the number of characters contained in a given binarized image. We calculated the average aspect ratio of a character using a total of 1000 single-character color images extracted from the ICDAR 2003 robust OCR dataset [6], and got the value of 0.68. We denote this value by β.

First, for the given binarized image with M pixels in width and N pixels in height, we estimate the number of characters contained in the image according to

$$p = \frac{M}{N \times \beta}, \qquad n_1 = \lfloor p \rfloor, \qquad n_2 = \lceil p \rceil, \tag{2}$$

where $\lfloor\ \rfloor$ and $\lceil\ \rceil$ denote floor and ceiling functions, respectively.

Second, we generate one sequence of "single-character-like" images by dividing the given binarized image into a total number of n_1 images with $\frac{M}{n_1}$ pixels in width and N pixels in height. Similarly, another sequence of "single-character-like" images is obtained by dividing the given binarized image into a total number of n_2 images with $\frac{M}{n_2}$ pixels in width and N pixels in height.

Fig. 3 shows one example of segmentation of a binarized image into two sequences of single-character-like images. For the given binarized image Eq. (2) gives the values of $p = 3.79$, $n_1 = 3$, and $n_2 = 4$.

Figure 3. One example of segmentation of a binarized image into two sequences of single-character-like images. (a) A binarized image. (b) One sequence of $n_1(= 3)$ single-character-like images. (c) Another sequence of $n_2(= 4)$ single-character-like images.

V. Calculation of the degree of character-likeness using SVM

In this section we propose to calculate the degree of "character-likeness" of each single-character-like image using SVM in an appropriately chosen feature space.

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 single-character-like 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 80×120 pixels.

Next, we extract the well-known and simple feature in the field of character recognition: the mesh feature or local densities of black pixels.

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.

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

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

$$\begin{aligned} f(\boldsymbol{x}) &= \sum\nolimits_i \alpha_i y_i (\boldsymbol{\Phi}(\boldsymbol{x}_i) \cdot \boldsymbol{\Phi}(\boldsymbol{x})) - b \\ &= \sum\nolimits_i \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) - b, \end{aligned} \tag{3}$$

where $\{\boldsymbol{x_i}\}$ 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^{light} [10], and used the RBF kernel function given by

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\|\boldsymbol{x} - \boldsymbol{y}\|^2\right). \tag{4}$$

Following the procedure described in our previous paper [5] dealing with binarization of single-character color images, SVM was trained using a total of 1000 single-character color images extracted from the ICDAR 2003 robust OCR dataset [6] different from the 1000 character string color images selected in Section II.

Fig. 4 shows examples of SVM training data for character and non-character classes. In particular, training data for character class were reinforced with available fonts for English.

Here, we regard the value of $f(\boldsymbol{x})$ of (3) as estimating the degree of "character-likeness", and also assume that the

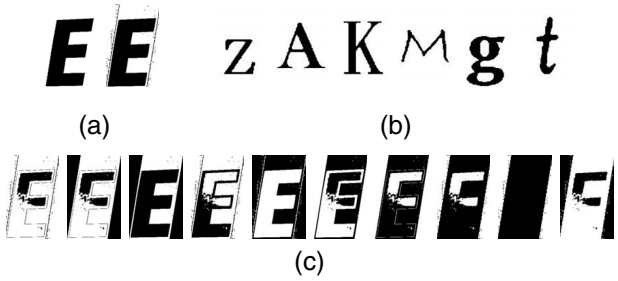

Figure 4. Examples of SVM training data . (a) Character class (correctly binarized images). (b) Character class (available fonts for English). (c) Non-character class (incorrectly binarized images).

larger the value of $f(\boldsymbol{x})$ is the more its character-likeness is.

Incidentally, the thus trained SVM achieved a high correct classification rate of 97.9% between character and non-character classes based only on the sign of $f(\boldsymbol{x})$ of (3).

VI. Selection of a Correctly Binarized Image via the Maximum Average of Character-Likeness

In this section we propose to estimate the degree of "character-string-likeness" by the average of "character-likeness" obtained for single-character-like images segmented from each tentatively binarized image.

First, we denote an input color image by F, a total of $(2^K - 2)$ tentatively binarized images by $\{B_i\}_{i=1}^{2^K-2}$, and two sequences of single-character-like images obtained for the ith tentatively binarized image B_i by $\{b'_{ij}\}_{j=1}^{n_1}$ and $\{b''_{ik}\}_{k=1}^{n_2}$. The numbers of n_1 and n_2 are determined for each individual tentatively binarized image.

Second, we denote an output value of the trained SVM described in Section V for each single-character-like image b by $f(b)$. The value of $f(b)$ specifies the degree of "character-likeness" of b.

Third, we denote the degree of "character-string-likeness" of the ith tentatively binarized image B_i by $S(B_i)$. and calculate the value of $S(B_i)$ as follows.

$$S(B_i) = \max\left(\frac{\sum_{j=1}^{n_1} f(b'_{ij})}{n_1}, \frac{\sum_{k=1}^{n_2} f(b''_{ik})}{n_2}\right). \quad (5)$$

Finally, we select the tentatively binarized image B_{i^*} with the maximum average of character-likeness or the maximum degree of "character-string-likeness" as an optimal binarization result according to

$$i^* = \underset{1 \le i \le 2^K-2}{\operatorname{argmax}} \; S(B_i) \quad (6)$$

VII. Experimental Results

We apply the proposed binarization technique to a total of 1000 images extracted from "TrialTrain" subset of the ICDAR 2003 robust word recognition dataset [6] containing JPEG character string images in natural scenes.

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, experimental results on generation of tentatively binarized images using K-means clustering showed that a correctly binarized image was included in a set of 30 tentatively binarized images individually generated for 878 color character string images. Namely, the correctly binarized image generation rate was 87.8%.

Fig. 5 shows examples for which correctly binarized images were not included in tentatively binarized images.

Figure 5. Examples for which correctly binarized images were not generated.

From Fig. 5, it can be seen that even humans cannot easily make a discrimination between figure and background against those examples.

Second, using the above-mentioned 878 color character string images we investigated the ability of selecting a correctly binarized image from a total of 30 tentatively binarized images based on the evaluation of "character-string-likeness" according to (5) and (6). Namely, a total of 30 tentatively binarized images were arranged in the decreasing order of the value of (5) as candidates of correctly binarized images.

Fig. 6 shows cumulative correct selection rates. The qth cumulative correct selection rate is an average rate at which the top q candidates contain a correctly binarized image.

From Fig. 6, it is found that the correctly binarized image selection rate or the 1st cumulative correct selection rate is 91.6%, and the 9th cumulative correct selection rate exceeds 99.9%.

Fig. 7 shows examples of successful and unsuccessful selection of correctly binarized images.

In summary, the proposed method achieved a correct binalization rate of 80.4%, which equals a correctly binarized image generation rate of 87.8% times a correctly binarized image selection rate of 91.6%.

From these results, we can say that the proposed method provides a very promising tool for binarizing multicolored

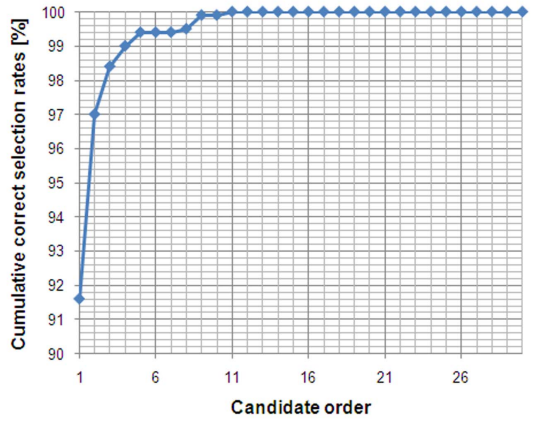

Figure 6. Cumulative correct selection rates.

Figure 7. Examples of successful and unsuccessful selection of correctly binarized images. (a) Successful ones. (b) Unsuccessful ones.

character strings with a variety of image degradations and complex backgrounds.

Future topics are as follows:

(1) adaptive selection of the number of clusters to point distributions in the color space,
(2) optimal segmentation of each binarized image into a sequence of "single-character-like" images, and
(3) more efficient evaluation of "character-likeness" via appropriate feature selection and SVM.

In particular, regarding (2) the proposed segmentation method is rather naive and simple and is to be much reinforced. Also, pruning the tentatively binarized images of noisy ones can not only reduce the processing time but also improve the binarization accuracy.

VIII. Conclusion

Binarization of color, low-quality character strings 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 four steps; generation of tentatively binarized images via K-means clustering in the HSI color space, segmentation of every binarized image into a sequence of "single-character-like" images, evaluation of the degree of "character-likeness" of each single-character-like image using SVM, and selection of a single binarized image with the maximum average of "character-likeness" as an optimal binarization result.

Experiments using the ICDAR 2003 robust word recognition dataset containing color character string images in natural scenes showed that the proposed method achieved a correct binarization rate of 80.4%.

It is interesting and challenging to combine this technique with text extraction/location and character recognition modules so as to build a total system of robust reading as applied to natural scene images.

References

[1] D. Doermann, J. Liang, and H. Li. "Progress in camera-based document image analysis". *Proc. of Seventh Int. Conf. on Document Analysis and Recognition*, pages 606–616, Edinburgh, Aug. 2003.

[2] O. Trier and A. K. Jain. "Goal directed evaluation of binarization methods". *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-17:1191–1201, 1995.

[3] S. Wu and A. Amin. "Automatic thresholding of gray-level using multi-stage approach". *Proc. of Seventh Int. Conf. on Document Analysis and Recognition*, pages 493–497, Edinburgh, Aug. 2003.

[4] K. Wang and J. A. Kangus. "Character location in scene images from digital camera". *Pattern Recognition*, 36:2287–2299, 2003.

[5] K. Kita and T. Wakahara. "Binarization of color characters in scene images using k-means clustering and support vector machines". *Proc. of Twentieth Int. Conf. on Pattern Recognition*, pages 3183–3186, Istanbul, Aug. 2010.

[6] The ICDAR 2003 Robust Reading Datasets. http://algoval.essex.ac.uk/icdar/Datasets.html.

[7] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[8] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. "ICDAR 2003 robust reading competitions". *Proc. of Seventh Int. Conf. on Document Analysis and Recognition*, pages 682–687, Edinburgh, Aug. 2003.

[9] K. Ashida, H. Nagai, M. Okamoto, H. Miyano, and H. Yamamoto. "Extraction of characters from scene images". *IEICE Trans. D.*, J88-D-II:1817–1824, 2005 (in Japanese).

[10] T. Joachims. "Making large-scale SVM learning practical". *Advances in Kernel Methods: Support Vector Learning*, B. Schölkopf, C. J. Burges, and A. J. Smola (eds.). Chap. 11, MIT Press, 1998.