中央研究院
資訊科學研究所
Institute of Information Science, Academia Sinica • Taipei, Taiwan, ROC

TR-IIS-09-003

A Binarization Method with Learning-Built Rules for Document Images Produced by Cameras

Chien-Hsing Chou, Wen-Hsiung Lin, and Fu Chang

April 16, 2009 || Technical Report No. TR-IIS-09-003
http://www.iis.sinica.edu.tw/page/library/LIB/TechReport/tr2009/tr09.html

A Binarization Method with Learning-Built Rules for Document Images Produced by Cameras

Chien-Hsing Chou†, Wen-Hsiung Lin‡, and Fu Chang‡

†Department of Electronics Engineering, Vanung University, Taiwan

‡Institute of Information Science, Academia Sinica, Taipei, Taiwan

Email: {ister, bowler, fchang}@iis.sinica.edu.tw

Abstract

In this paper, we propose a novel binarization method for document images produced by cameras. Such images often have varying degrees of brightness and require more careful treatment than merely applying a statistical method to obtain a threshold value. To resolve the problem, our method divides an image into several regions and decides how to binarize each region. The decision rules are derived from a learning process that takes training images as input. Tests on images produced under normal and inadequate illumination conditions show that our method yields better visual quality and better OCR performance than three global binarization methods and four locally adaptive binarization methods.

Keywords: document image binarization, global threshold, image processing, local threshold, multi-label problem, non-uniform brightness, support vector machines

1. Introduction

Binarizing images of documents captured with camera-equipped electronic devices, such as PDAs or cellular phones, presents a new challenge. The captured content can be transformed in various ways. For example, French content can be translated into English, or information on a business card can be identified and stored in proper categories, such as name, address, telephone number, etc. The new challenge to binarization arises because such images are produced under illumination conditions that are inferior to those found in a scanning environment. As a

result, there are varying degrees of brightness over the images. If we simply apply a global threshold, as we do with scanned images, the binarized results could be too bright in one area and too dark in another area. A more effective way of binarizing such images is therefore desired.

Before discussing the problem in detail, we briefly review binarization methods proposed in the literature. Following Sezgin and Sankur [1], we classify the methods into six categories.

Histogram-based methods: These methods determine the binarization threshold by analyzing the shape properties of the histogram, such as the peaks and valleys (Sezan [2]), or the concavities (Rosenfeld and Torre[3]). Pavlidis [4] constructs a histogram by using gray-image pixels with significant curvature, or second derivative, values and then selects a threshold based on the histogram.

Clustering-based methods: The threshold is selected by partitioning the image's pixels into two clusters at the level that maximizes the between-class variance (Otsu [5]), or minimizes the misclassification errors of the corresponding Gaussian density functions (Kittler and Illingworth [6]).

Entropy-based methods: These methods employ entropy information for binarization (Kapur et al. [7]).

Object attribute-based methods: These methods select the threshold based on some attribute quality (e.g., edge matching of Hertz and Schafer [8]) or the similarity measure between the original image and the binarized image (Huang and Wang [9]).

Spatial binarization methods: These methods binarize an image according to the higher-order probability or the correlation between pixels (Abutableb [10]).

Locally adaptive methods: These methods compute a local threshold based on the information contained in the neighborhood of each pixel, or in the region of the image. In Bernsen's method [11], for example, the threshold is a function of the lowest and highest gray values; however, in Niblack's method [12], it is a function of the mean and the standard devi-

ation of the gray scales. Taxt et al. [13] apply the EM algorithm to compute the local threshold, while Eikvil et al. [14] apply Otsu's method to compute it. Mardia and Hainsworth [15] compute the local threshold based on the estimation of two-point spatial covariance. Chow and Kaneko [16], and Nakagawa and Rosenfeld [17], compute the local threshold by analyzing the bimodality of gray values. The threshold can also be determined by comparing a pixel's gray value with the average gray value of the pixels in its neighborhood (White and Rohrer [18]), or its local variance (Yasuda [19]). Sauvola and Pietikäinen [20] (also, Sauvola et al. [21]) first partition an image into windows and rapidly classify each window into background, picture and text. They then apply various binarization rules to the different types of window. Kim [22] modifies Sauvola and Pietikäinen's approach by introducing more than one window size for textual content. As an alternative to the above approaches, some special features extracted in the neighborhood of a pixel can be used to determine the local threshold. Examples are gradient information (Trier and Taxt [23], Parker [24], and Yanowitz and Bruckstein [25]), and character stroke width (Kamel and Zhao [26], Yang and Yan [27], and Ye et al. [28]).

If a binarization method computes a threshold for an entire image, it is called *a global method*. Trier and Taxt [29] evaluated four such methods ([5-7] and [10]) and concluded that Otsu's approach [5] outperforms the other three. On the other hand, if a method computes a threshold for the neighborhood around each pixel or for each designated block in the image, it is called a *local method*. Trier and Jain [30] evaluated some of these methods ([11-16], [18], and [23-25]) and concluded that those proposed by Bernsen [11], Niblack [12] and Eikvil et al. [14] are the top-ranked local threshold methods in terms of the error rate and rejection rate for character recognition, and also for the visual criterion. More complete surveys of image thresholding techniques can be found in [1] and [29-35].

As noted earlier, using cameras to produce document images creates a new challenge for document image binarization. To address the problem, Park et al. [36] proposed block adaptive binarization of business card images produced by a PDA camera. This method is very similar to

that of Eikvil et al. [14], which partitions an input image into blocks. For a given block, b, a larger concentric block, denoted as $L(b)$, is found and Otsu's method is applied to it. If the difference between the means of two classes, determined by Otsu's method, exceeds a certain threshold, block b is classified as a content block; otherwise, it is classified as a background block. Content blocks are binarized according to Otsu's thresholds, while background blocks are set directly to white or black based on the average of gray values found in them. The method in [36] differs from that of [14] in the way content blocks are differentiated from background blocks, and also in the way the sizes of b and $L(b)$ are set.

Our method also divides a document image into smaller areas, but differs from the methods proposed in [14] and [36] in a number of respects. For example, instead of dividing an image into fixed-size blocks, we divide it into $k \times k$ regions, using the value of k obtained in experiments. Dividing each image into the same number of regions ensures that the binarization effect is relatively invariant with respect to the resolution of the camera. Within each region r, one of the following four actions is applied: set the whole of r to black, set the whole of r to white, use Otsu's method to compute the threshold for r, or use the smallest Otsu threshold in the neighboring regions as the threshold for r. A learning process is used to establish the rules for deciding which of the above actions should be adopted for each region. The rules are expressed as decision functions, which take a number of features extracted from r as input. The experiment results demonstrate that the above factors have a significant impact on the successful performance of our method.

The crucial step in our approach is establishing rules to decide which action should be applied to each sub-divided region. To do this, we utilize a machine learning approach, namely, the support vector machine (SVM) classification method [37-38], which represents a major development in pattern classification research. Two innovations of SVM are responsible for its success: (1) the ability to find a hyperplane that divides samples into two groups with the widest margin between them; and (2) the extension of the concept in (1) to a high-

er-dimensional setting using a kernel function to represent a similarity measure on that setting. Both innovations can be formulated in a quadratic programming framework whose optimal solution is obtained in a reasonable amount of time. This makes SVM a practical and effective solution for many pattern classification problems. In this paper, we divide training images into a number of regions and label the appropriate actions for them, after which we commence the SVM learning process and construct the decision functions. Since the total number of regions is relatively small, compared to the alternative approach in which pixels are labeled, it is relatively easy for humans to label the regions using a graphic-user interface. The proposed binarization method thus constitutes an interesting application of the SVM approach in the area of image analysis.

The remainder of the paper is organized as follows. In Section 2, the proposed binarization method is introduced. In Section 3, we discuss the learning algorithm. Section 4 details the experiment results. Then, in Section 5, we present our conclusions.

2. The Proposed Binarization Method

As our approach involves the computation of thresholds using Otsu's method, we begin by giving a brief summary of that method. Given a gray-scale image, Otsu's method sets a threshold *thrl* as follows:

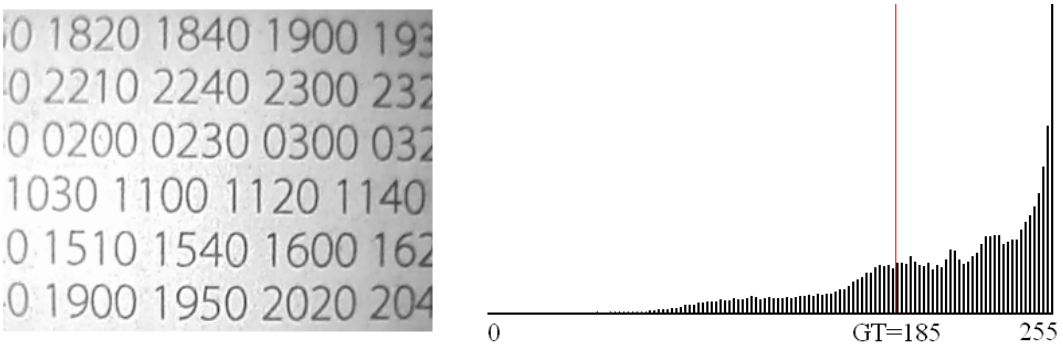
$$thrl = \underset{v}{\operatorname{argmax}} \sum_{i=1}^{2} \pi_i(v)(\mu_i(v) - \mu_T)^2 , \tag{1}$$

where $\pi_1(v) = \sum_{m \le v} p_m$, $\mu_1(v) = \sum_{m \le v} mp_m$, $\pi_2(v) = \sum_{m > v} p_m$, $\mu_2(v) = \sum_{m > v} mp_m$,

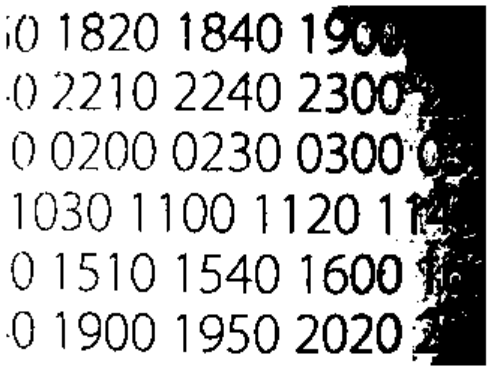
$\mu_T = \sum_{m=1}^{256} mp_m$, and p_m is the proportion of pixels in the image whose gray scale is m for $m = 1, 2, \ldots, 256$. It then designates a pixel p as black, if p's gray scale is less than or equal to *thrl*; or white otherwise.

When the background and foreground intensities are well separated, Otsu's method yields good binarized results. However, if the image intensities are inseparable, the resulting thre-

shold value is unsuitable. Fig. 1a shows a document image taken by a camera. From the histogram of the image, shown in Fig. 1b, we observe that the gray scales associated with the foreground pixels are mixed with those associated with the background pixels; thus, it is difficult to determine a good threshold value for binarization. In fact, when applying Otsu's method, we find that the threshold value is 185, causing part of the image to become blurred. Clearly a different binarization solution is required.



(a) (b)

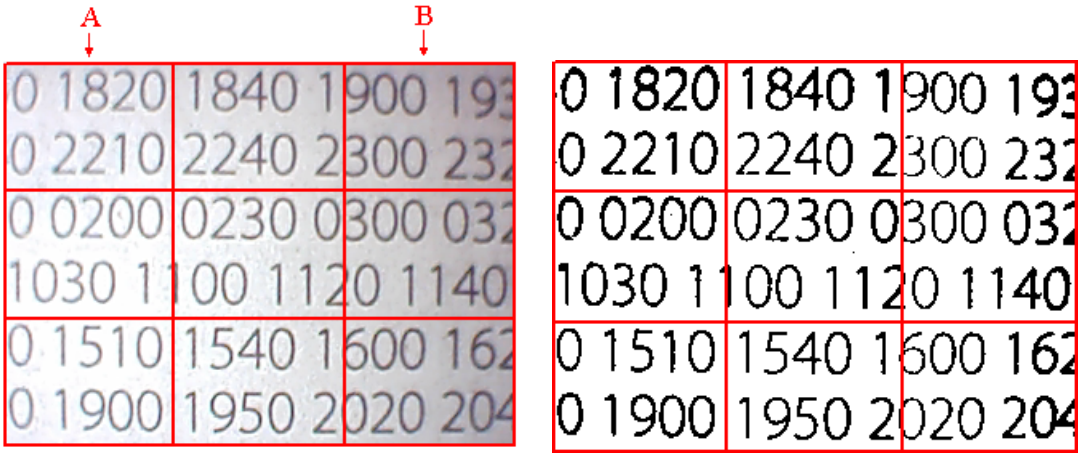


(c)

Fig. 1. (a) A document image obtained by a camera under the inadequate illumination condition. (b) The histogram of (a). (c) The document image binarized using Otsu's method.

An immediate solution is to divide an image into several regions and apply a thresholding method to each region separately. However, the image should not be divided according to the layout structure of the document, since the whole image (shown in Fig. 1a) falls within the same text region of the layout structure, while the brightness varies extensively over the region.

Instead, we divide the image into equal-sized regions. For example, we divide the image in Fig. 2a into 3×3 regions and apply Otsu's method to obtain a local threshold (LT) for each region. The Otsu thresholds usually vary according to the regions. For example, in Fig. 2b, the thresholds are 204 over region A and 156 over region B. The resulting binarized image, shown in Fig. 2b, is more satisfactory than the result shown in Fig. 1c.



(a) (b)

Fig. 2. (a) A raw image partitioned into 3×3 regions. (b) The binarized image using Otsu's method to find the local threshold for each region.

Even so, using Otsu threshold as a local threshold can yield poor results for regions containing background pixels only. For example, region C in Fig. 3a would be improperly binarized, as shown in Fig. 3b, if the Otsu threshold (= 162) were employed as the LT. In fact, setting the LT to 0 would yield a satisfactory result, as shown in Fig. 3c. A reasonable condition for setting the LT to 0 is to take reference of the μ and σ values of a given region, which are the mean and variance of C's gray values respectively. When the σ value is high, there is good mixture of black and white pixels, so we can use the Otsu threshold as the LT. On the other hand, when the σ value is low, we can set the LT to 0, provided the μ value is high. Fig. 3 is an example of the latter case, where the σ value of region C (= 9.3) is low. Thus, it is reasonable to set the LT of C to 0, since its μ value is 158 – a rather high value.

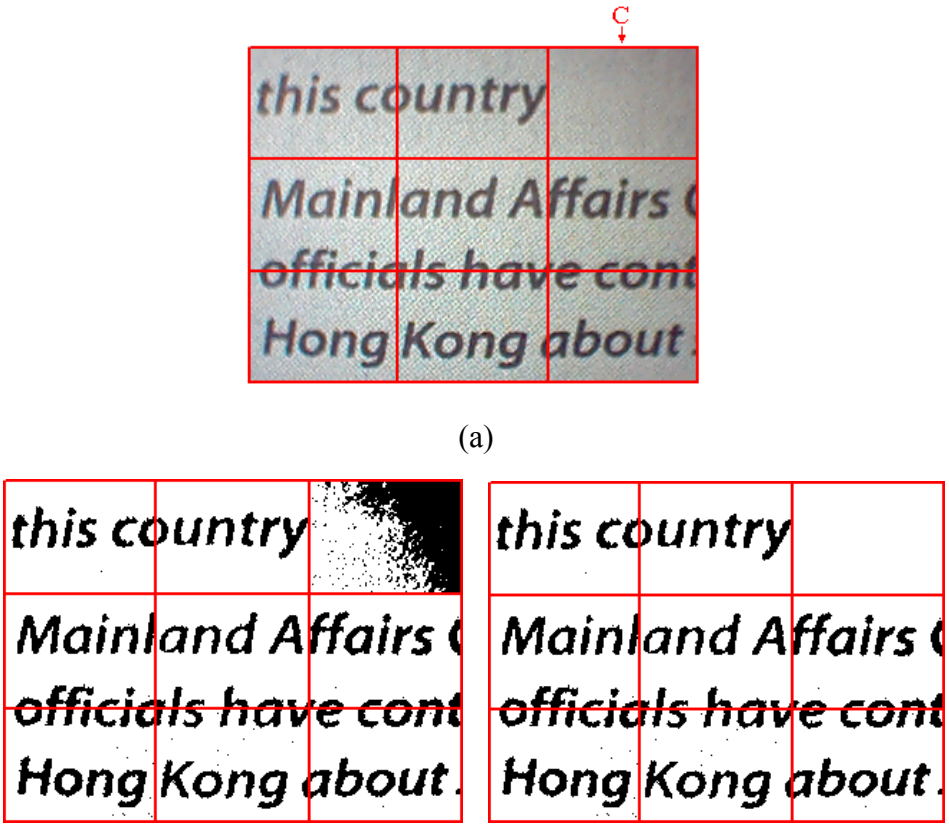


Fig. 3. (a) An original gray-scale image. (b) The image binarized using Otsu's method to find a local threshold for each region. (c) The local threshold is set to 0 for region C.

If the σ and μ values of a region are both low, it is reasonable to set the LT to 255. Fig. 4a provides such an example, in which the σ values of regions D, E, and F (= 7.1, 5.1, 4.6, respectively) are low and their μ values (= 34, 32, 29, respectively) are also low, since these regions are parts of a figure. Using Otsu's threshold as the LT would turn some of the pixels white, as shown in Fig. 4b. Instead, 255 could be a better LT, as shown in Fig. 4c.

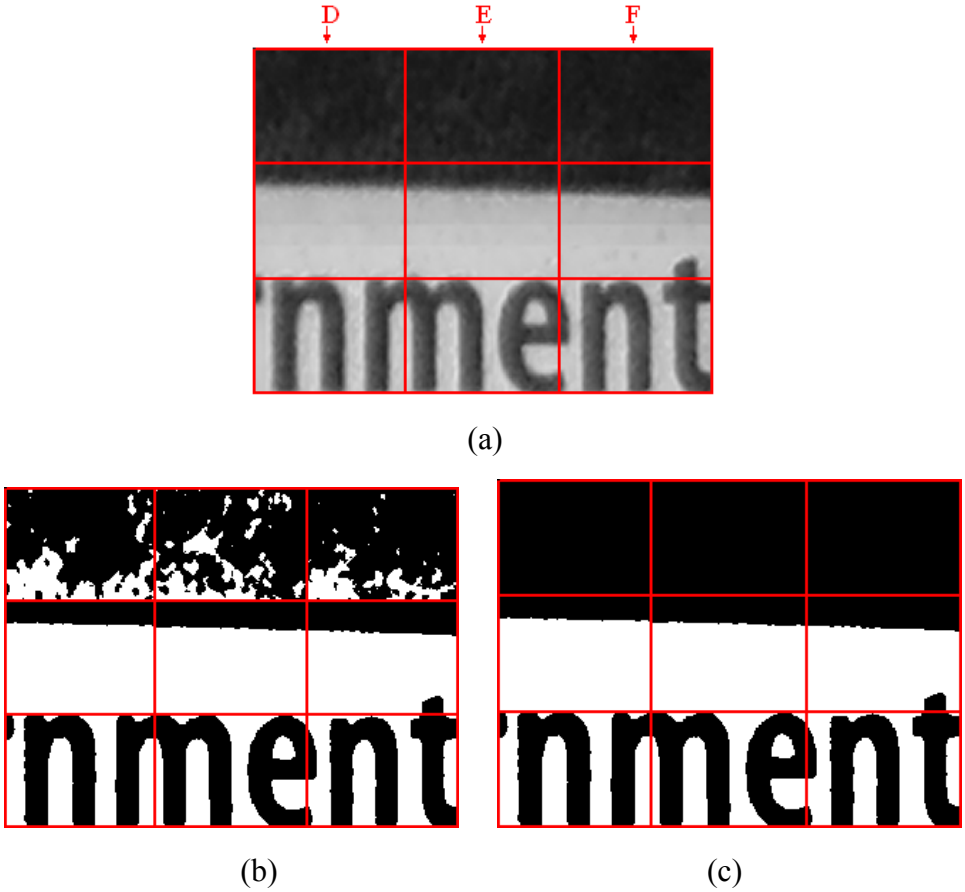
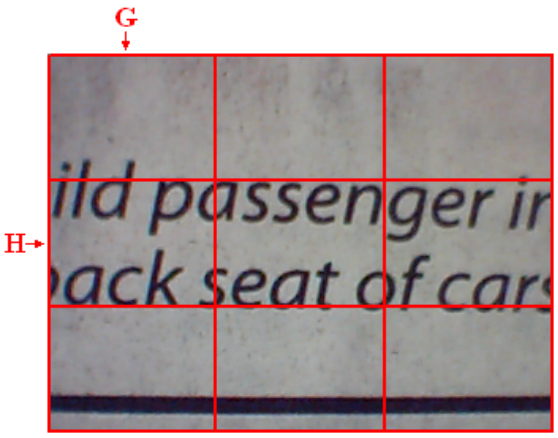
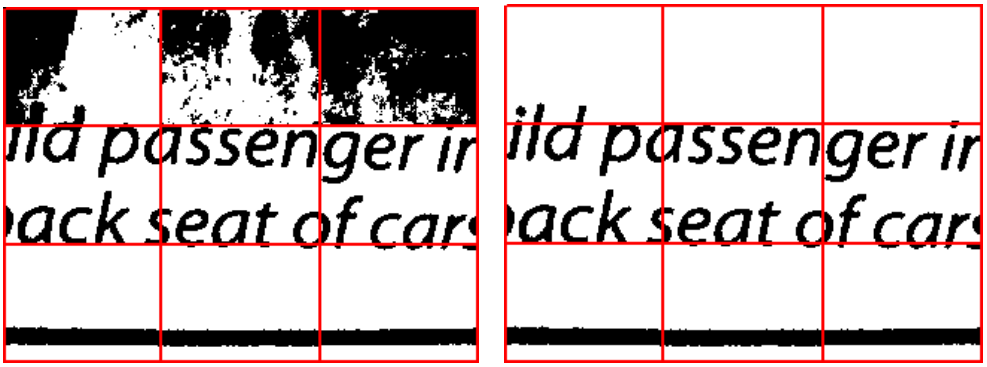


Fig. 4. (a) An original gray-scale image. (b) The image binarized using Otsu's method to find a local threshold for each region. (c) The local thresholds for regions D, E, and F are set to 255.

In Fig. 5a, region G is filled largely with white pixels, and some neighboring regions are also dominated by white pixels. In this case, using Otsu's threshold (= 161) as the LT would turn many white pixels black, as shown in Fig. 5b. Instead, the lowest neighboring Otsu threshold (= 106) serves as a better LT. Here, the minimum value is obtained from region H, which has a good mixture of black and white pixels, as shown in Fig. 5c.



(a)



(b) (c)

Fig. 5. (a) An original gray-scale image. (b) The image binarized using Otsu's method to find a local threshold for each region. (c) The local threshold is set to the minimum of the neighboring thresholds for region G.

The above examples show that there are four possible actions we can take for each region *r,* namely: set the LT of *r* to 0, 255, $T_{\text{Otsu}}(r)$, or $T_{\min}(r)$, where $T_{\text{Otsu}}(r)$ is the Otsu threshold for *r*, and

$$T_{\min}(r) = \min\left\{T_{\text{Otsu}}(r), \min_{s\in\Lambda(r)} T_{\text{Otsu}}(s)\right\}, \tag{2}$$

where $\Lambda(r)$ is the set of neighboring regions of *r.* The following features help determine which action would be appropriate for each region: $T_{\text{Otsu}}(r)$-$T_{\min}(r)$, $\mu(r)$, and $\sigma(r)$; the last two terms are, respectively, the mean and the standard deviation of the distribution of gray-values in *r*.

In summary, we extract the following features from each region *r*:

Feature 1: $T_{\text{Otsu}}(r)$-$T_{\min}(r)$

Feature 2: $\mu(r)$;

Feature 3: $\sigma(r)$.

From these features, we must decide which of the following actions would be the most appropriate for binarizing region r:

Action 1: set LT of r to 0;

Action 2: set LT of r to 255;

Action 3: set LT of r to $T_{\text{Otsu}}(r)$;

Action 4: set LT of r to $T_{\min}(r)$.

3. Constructing Adaptive Binarization Rules Using Support Vector Machines

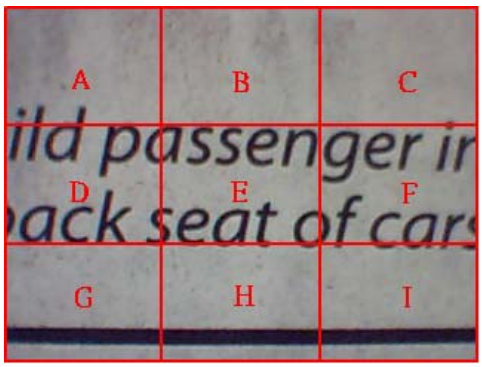
Having specified the three features, we use the SVM method to determine which binarization action to take for each region. SVM provides a learning algorithm that constructs decision functions from training data. We assume that a set of labeled training samples $(\mathbf{x}_1, y_1)$, …, $(\mathbf{x}_n, y_n)$ is given, where $\mathbf{x}_i$ is a vector in the d-dimensional Euclidean space R^d and y_i is the label, or class type, of $\mathbf{x}_i$ for $i = 1, 2, \ldots, n$. A decision function is of the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) + \beta, \tag{3}$$

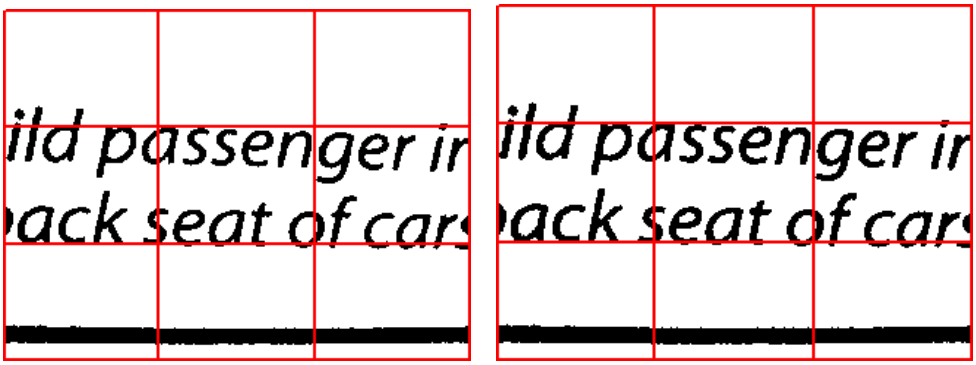
where $\mathbf{x} \in R^d$ is a test sample, α_i is a positive quantity derived in the learning process, β is a bias term, and $k(\mathbf{x}_i, \mathbf{x})$ is a kernel function (see [37]), for $i = 1, 2, \ldots, n$. The training samples $\mathbf{x}_i$ for which $\alpha_i > 0$ are called support vectors. In fact, if we only retain the vectors on the right-hand side of (3), we get exactly the same decision function. SVM is especially effective for binary classification problems in which an object is assigned one of two labels. In such cases, a decision function assumes only two values, which are the same as the two labels in question. It is thus called a binary decision function. Assuming that the two labels are –1 and 1, a binary decision function works in such a way that if $f(\mathbf{x}) \geqq 0$, $\mathbf{x}$ is assigned label 1; otherwise, it is assigned label –1.

To apply SVM to the binarization problem, we must first map the problem to the SVM setting. We do this by dividing each image into $k \times k$ regions, which constitute our training samples. How to find an appropriate value of k is considered in Section 4.2. From each region r, we extract the three features, $T_{\text{Otsu}}(r)$-$T_{\min}(r)$, $\mu(r)$, and $\sigma(r)$, to form a 3-dimensional feature vector; thus, the dimension of our Euclidean space is fixed at 3. The four actions are then taken as four labels so that if an action y is deemed appropriate for r, the latter is assigned the label y.

Two other problems must also be addressed. First, for certain regions, there may be more than one appropriate binarization action. We can observe this problem by examining the image in Fig. 6a, which is divided into 9 regions from A to I. Among the regions, A and C carry a single label, $T_{\min}$, D to I carry labels T_{Otsu} and $T_{\min}$, and B carries 0 and $T_{\min}$. Details are given in Table 1. To demonstrate that multiple labels are reasonable for some of these regions, in Fig. 6b, we show the binarized result using a common threshold $T_{\min}$ as the LT of each region. Meanwhile, Fig. 6c shows the binarized result using the alternative threshold to $T_{\min}$ as the LT of those regions for which two options are allowed. Both binarized results are acceptable. Thus, when preparing training data for the learning process, we allow multiple labels to associate with the training samples. We find that, among the 1,098 regions obtained in our data set, 352 regions carry a single label and the remaining 746 regions carry multiple labels. A similar situation can be found in text categorization (Joachims [39], Schapire and Singer [40]) or scene classification (Moutell [41]). For example, a news article can belong to two categories, such as *earn* and *trade* in the Ruthers-21578 dataset (see [39-40]).



(a)



(b) (c)

Fig. 6. (a) An original gray-scale image. (b) The image binarized using T_{min} as the LT of all regions. (c) The image binarized using the alternative to T_{min} as the LT of those regions for which two options are allowed.

Table 1. The labels of regions A to I and the values of T_{Otsu} and T_{min} for each region.

Region	Label	T_{Otsu}	T_{min}
A	T_{min}	138	106
B	0, T_{min}	151	106
C	T_{min}	147	106
D	T_{Otsu}, T_{min}	106	97
E	T_{Otsu}, T_{min}	109	97
F	T_{Otsu}, T_{min}	107	99
G	T_{Otsu}, T_{min}	97	97
H	T_{Otsu}, T_{min}	100	99
I	T_{Otsu}, T_{min}	99	99

In addition to the multi-label problem, we also need to deal with the multi-class problem. That is, there are four labels or class types in our application, but SVM can only deal with two class types at a time. To resolve these two problems we follow the solution proposed in [39] by using SVM to construct as many decision functions as there are labels (see also Bottou [42]).

Therefore, assuming that the four labels in our application are {1, 2, 3, 4}, we use SVM to construct four binary decision functions $\{f_1, f_2, f_3, f_4\}$.

To construct f_i for $i = 1, 2, 3, 4$ in the training phase, we divide the training samples into two groups. The first group, called the positive group, consists of samples with label i, and the second group, called the negative group, consists of samples without label i. Thus, if a sample carries labels j and k, it will be assigned to the positive group associated with f_j and the positive group with f_k, but not to the negative group associated with f_j or f_k. In the testing phase, when a test sample $\mathbf{x}$ is given, we compute $f_i(\mathbf{x})$ for $i = 1, 2, 3, 4$ and assign label l to $\mathbf{x}$ when

$$l = \arg\max_{i=1,2,3,4} f_i(\mathbf{x}) . \tag{4}$$

4. Experiment Results

In this section, we discuss how to prepare training data, how to divide document images, and how to use SVMs for deriving decision functions. Furthermore, we compare the results obtained by our method with three global methods and four locally adaptive methods. Finally, we conduct sensitivity studies on the parameters and the types of images used in the experiments.

4.1 Data Preparation

We collected 122 hardcopy documents from newspapers and magazines, and used an ORITE I-CAM 1300 one-chip color camera, with a resolution of 1,300,000 pixels, to photograph them. We then stored the photographs as gray-scale images consisting of 320×240 pixels. The images were produced under two conditions: the normal illumination condition and the inadequate illumination condition. In the former, the room light was on and there were no obstructions between the light and the documents, resulting in more or less uniform brightness across the images. In the latter, although the room light was on, humans or objects cast shadows over the documents, so that the shadowy area appears darker than the rest of the image. In total,

60 images were produced under normal illumination and 62 were produced under inadequate illumination.

We use three measures to evaluate a given binarization method's performance on the 122 camera images. Let A = the number of characters in the 122 camera images (there are actually 3,559 characters); B = the number of characters detected by ABBYY, an OCR software system; and C = the number of characters correctly recognized by ABBYY. The three measures, expressed in percentages, are:

Recall rate = C/A;

Precision rate = C/B; and

F_1 score, derived by the following formula [43-44]

$$F_\beta = \frac{(\beta^2+1)\times \text{recall rate}\times \text{precision rate}}{\beta^2\times(\text{recall rate}+\text{precision rate})}, \tag{5}$$

where β is set to 1.

4.2. Determining the Number of Regions

We divide each image into $k\times k$ regions and apply our binarization method to each region. To determine an appropriate value for k, we experimented with various values for k = 2, 3, …, 10. One possible way to find the best value of k would be to apply SVM training to each k and then use ABBYY to evaluate the binarized results of each k. However, this would force us to label the regions that are generated in all possible divisions, which would be rather demanding. An alternative approach is to apply a simple but reasonable binarization scheme to evaluate the binarized results for each k, and then choose the k that yields the best OCR performance.

We propose the following simple binarization scheme. For a given region r, if $\sigma(r)$ is larger than a specified threshold σ_0 (we set σ_0=15), we simply set LT = $T_{\text{Otsu}}(r)$; otherwise, we classify the entire r as white if $\mu(\underline{r})>\mu_0$, or black if $\mu(\underline{r})\leqq\mu_0$, where μ_0 is 128 (= 256/2). Having binarized all 122 images by means of this simple scheme, we apply ABBYY to the binarized images. Table 2 shows the recall rate, precision rate, and F_1 score of the OCR results. All three

measures suggest that dividing an image into 3×3 regions produces the best results.

Note that this simple binarization scheme sets an entire region to white or black, according to two thresholds, $\mu_0 = 128$ and $\sigma_0 = 15$ respectively. There is no doubt that the value of 128 is robust for μ_0, since a totally white or black region usually has a mean value far away from 128. To test the robustness of 15 for σ_0, we changed σ_0 to 10 and found that the 3×3 division still achieves the best OCR performance using the new value.

Having determined that $k = 3$ is the most appropriate value, we fetch the regions derived by the 3×3 division of all 122 images. There are 1,098 such regions, each comprised of 107×80 pixels. These regions constitute our training samples.

Table 2. The OCR accuracy derived by applying the simple binarization scheme to the 122 images, sub-divided into $k \times k$ regions for $k = 2, 3, \ldots, 10$.

Number of Regions	Recall Rate	Precision Rate	F_1 Score
2×2	93.73	93.76	93.74
3×3	93.80	95.15	94.46
4×4	92.77	94.82	93.78
5×5	92.58	92.75	92.65
6×6	91.86	93.85	92.84
7×7	92.80	93.84	93.31
8×8	92.06	93.47	92.74
9×9	91.86	94.11	92.97
10×10	91.29	93.54	92.40

4.3. Constructing Decision Functions with SVM

As stated in Section 3, SVM is employed to construct the decision functions. In addition, we use the LIBSVM toolkit [45] to conduct SVM training. We need to perform four SVM training operations, each of which divides training samples into two groups: one group consists of all samples with label i, and the other consists of all samples without label i for $i = 1, 2, 3, 4$. We employ the soft-margin version of SVM (see [39] and [48]) and the RBF kernel function. The value range of the penalty factor C is set to $\{10^a: a = -1, 0, \ldots, 5\}$. The RBF function involves a parameter γ, whose value range is set to $\{10^b: b = -8, -7, \ldots, 0\}$. To find the best values

for (C, γ), we perform a cross validation operation whereby all samples employed in the experiment are partitioned into five folds. We conduct five tasks, using four folds in each task as training data to construct SVM classifiers and the remaining fold as test data. We then select the values of (C, γ) that maximize the average accuracy rates in the five tasks. By so doing, we find the optimal (C, γ) to be (10, 0.1), resulting in a 98.57% average accuracy rate (see Table 3).

The four decision functions $\{f_1, f_2, f_3, f_4\}$ derived from the SVM training jointly assign label l_0, defined in (4), to the test samples. A test sample is deemed correctly classified if one of its labels matches l_0. The binarized images generated by the above method were then tested on the ABBYY software. In this test, we obtained 97% of the F_1 score (see Table 3).

As well as constructing decision functions from multiple-labeled training samples, we also conducted an experiment in which each training sample was assigned a single label chosen randomly from the multiple labels the sample carries. From the experiment results, also shown in Table 3, we conclude that training SVM decision functions with *multiple-labeled* samples achieves a *better* cross-validation result and OCR performance.

Table 3. Results derived from multiple-label and single-label assignments.

Labeling Method	Average Accuracy Rate of Five-Fold Cross Validation	F_1 Score for OCR Performance
Multiple Labels	98.57	97.0
Single Label	96.56	95.7

4.4. Comparisons with Other Binarization Methods

In order to make comparisons, we implemented seven other binarization methods. Three of them are global threshold methods proposed by Rosenfeld and Torre [3], Pavlidis [4], and Otsu [5] respectively; the other four are locally adaptive methods proposed by Bernsen [11], Niblack [12], Taxt et al. [13], and Eikvil et al. [14] respectively. For the parameters involved in the three locally adaptive methods, we adopted the values suggested in [30]. We would have liked to implement the method proposed by Park et al. [36], as it had been explicitly applied to

camera images. However, we were not given the values of parameters for this method. We therefore implemented Eikvil et al.'s method instead, because there is a high degree of similarity between the two methods.

Figs. 7 to 10 show four camera images and their binarized results. The raw images are shown in Figs. 7a to 10a. The images in 7a and 8a were taken under the normal illumination condition, while those in 9a and 10a were taken under the inadequate illumination condition. Following each raw image, we show its binarized results. To determine the impact of the different binarization methods on character recognition, we fed all the binarized results into the ABBYY software. Table 4 shows the OCR performance on (i) images produced under a normal illumination condition, (ii) images produced under the inadequate illumination condition, and (iii) all images. The boldface figures in the table indicate the best performances.

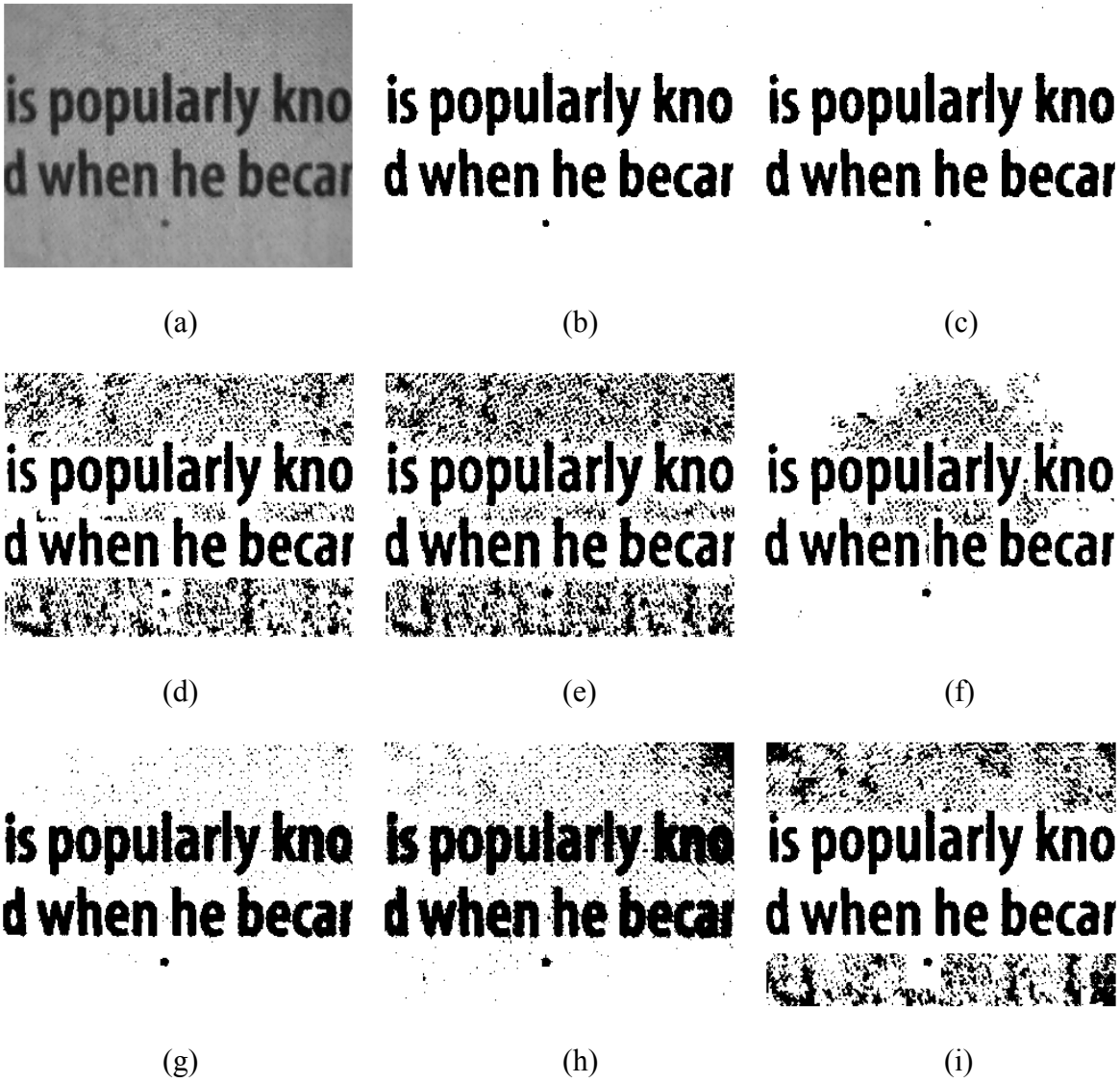
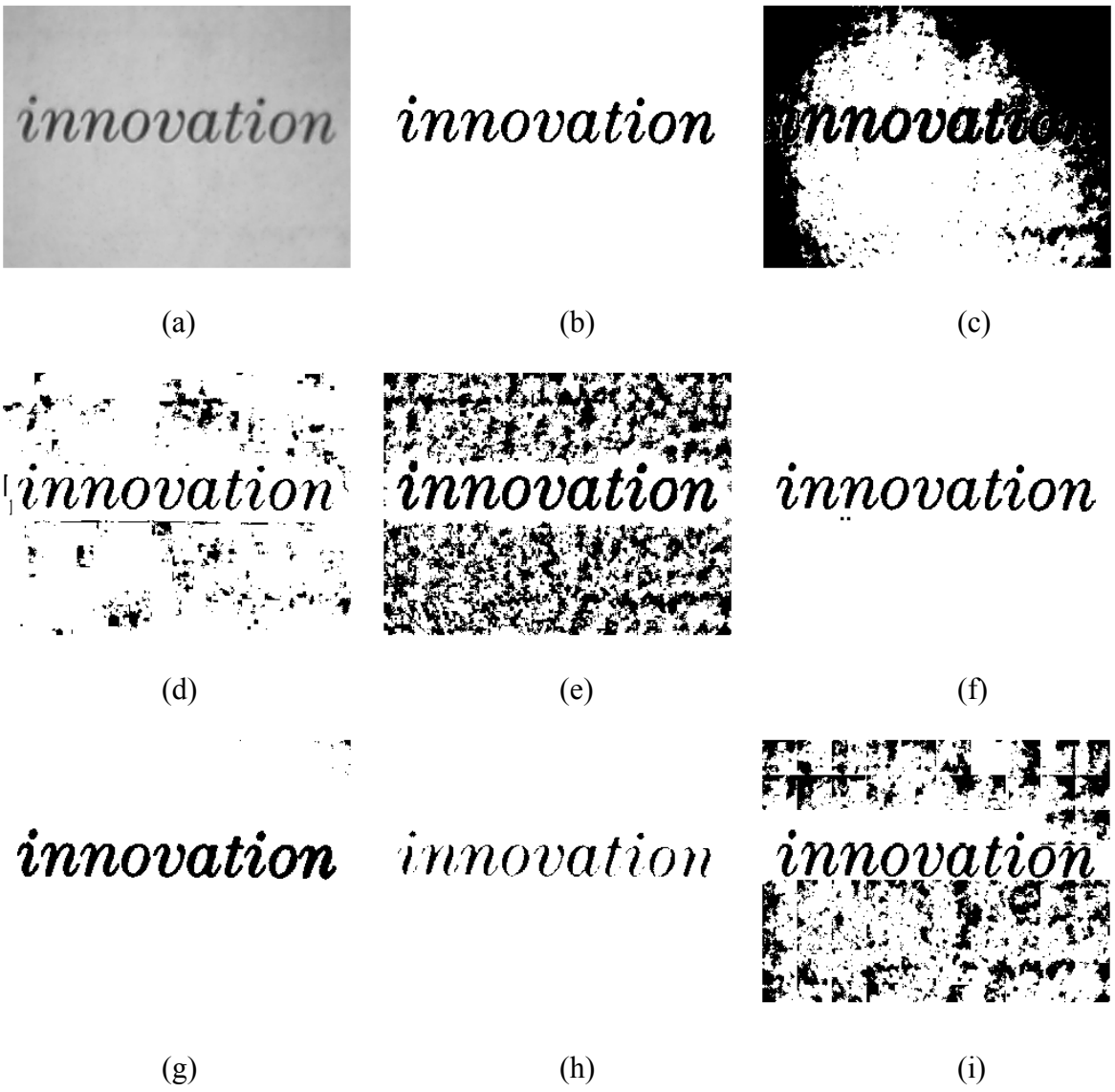


Fig. 7. (a) A raw image taken under the normal illumination condition. The image binarized by: (b) our method, (c) Otsu's method, (d) Bernsen's method, (e) Niblack's method, (f) Eikvil et al.'s method, (g) Rosenfeld and Torre's method, (h) Pavlidis's method, and (i) Taxt et al.'s method.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 8. (a) A raw image taken under the normal illumination condition. The image binarized by: (b) our method, (c) Otsu's method, (d) Bernsen's method, (e) Niblack's method, (f) Eikvil et al.'s method, (g) Rosenfeld and Torre's method, (h) Pavlidis's method, and (i) Taxt et al.'s method.

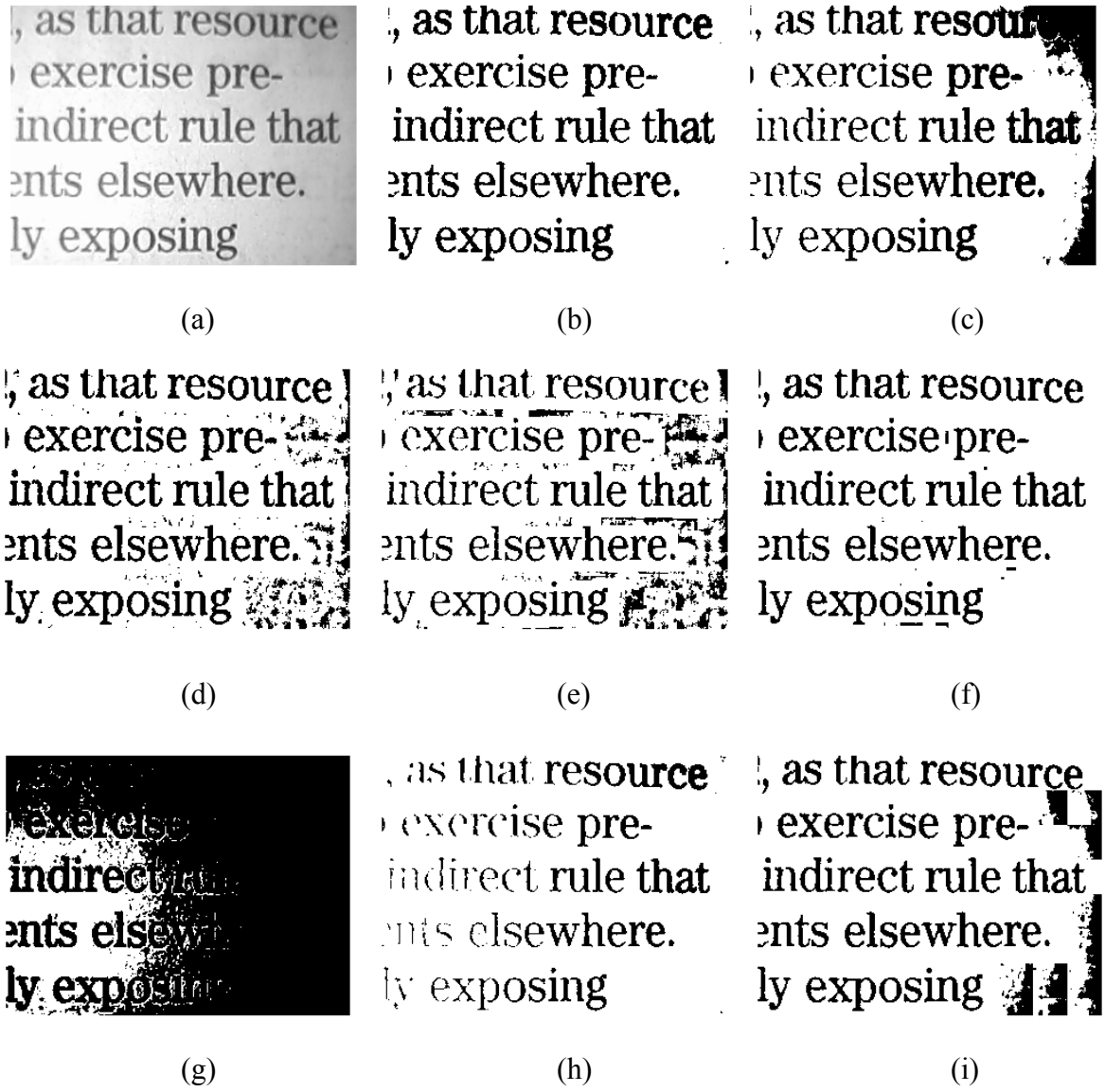


Fig. 9. (a) A raw image taken under the inadequate illumination condition. The image binarized by: (b) our method, (c) Otsu's method, (d) Bernsen's method, (e) Niblack's method, (f) Eikvil et al.'s method, (g) Rosenfeld and Torre's method, (h) Pavlidis's method, and (i) Taxt et al.'s method.

Fig. 10. (a) A raw image taken under the inadequate illumination condition. The image binarized by: (b) our method, (c) Otsu's method, (d) Bernsen's method, (e) Niblack's method, (f) Eikvil et al.'s method, (g) Rosenfeld and Torre's method, (h) Pavlidis's method, and (i) Taxt et al.'s method.

Table 4. OCR performance of the eight binarization methods.

Binarization Method		Ours	Otsu's	Bernsen's	Niblack's	Eikvil et al.'s	Rosenfeld and Torre's	Pavlidis's	Taxt et al.'s
Images Produced under the Normal Illumination Condition	Recall	**97.40**	94.00	87.50	77.21	91.54	82.13	82.58	88.45
	Precision	**97.08**	94.60	37.14	42.04	89.23	90.41	84.17	51.56
	F_1	**97.24**	94.50	53.12	54.44	90.37	86.07	83.37	65.15
Images Produced under the Inadequate Illumination Condition	Recall	**96.84**	84.61	85.31	85.72	89.06	57.62	78.03	89.62
	Precision	**96.68**	93.10	44.61	42.53	78.95	92.15	79.47	60.67
	F_1	**96.76**	88.64	58.58	56.85	83.70	70.90	78.74	72.36
All Images	Recall	**97.12**	89.31	86.41	81.47	90.30	69.88	80.31	89.04
	Precision	**96.88**	93.85	41.38	42.29	84.09	91.28	81.82	56.12
	F_1	**97.00**	91.52	55.96	55.67	87.04	79.16	81.06	68.84

Based on the above results, we make the following observations:

(1) In terms of OCR performance, our method outperforms the other methods by substantial margins. It also produces satisfactory binarized results for camera images taken under both normal and inadequate illumination conditions.

(2) Otsu's method produces acceptable binarized results for images taken under the normal illumination condition, provided that the foreground area makes up a relatively high proportion of the image, as shown in Fig. 7c; however, the method performs poorly on images containing only a few characters, as shown in Fig. 8c. It also has difficulty with images taken under inadequate illumination conditions. The other two global threshold methods (i.e., Rosenfeld and Torre's method and Pavlidis's method) work rather well on images containing only a few characters (Fig. 8g and 8h), but their performance is inconsistent on other images.

(3) Although the four locally adaptive methods perform well in foreground areas, the methods of Bernsen, Taxt et al., and Niblack may create pepper noise in background areas. Eikvil et al.'s method performs relatively better in background areas, but it has difficulty if the brightness of such areas varies beyond a certain level, as shown in Fig. 10f.

(4) By adjusting some parameters (the block size, for example) of the three locally adap-

tive methods, we can improve the visual quality of some binarized images at the expense of other images. This means that none of three methods can binarize all images well under a set of specific operating parameters.

4.5. Sensitivity Studies

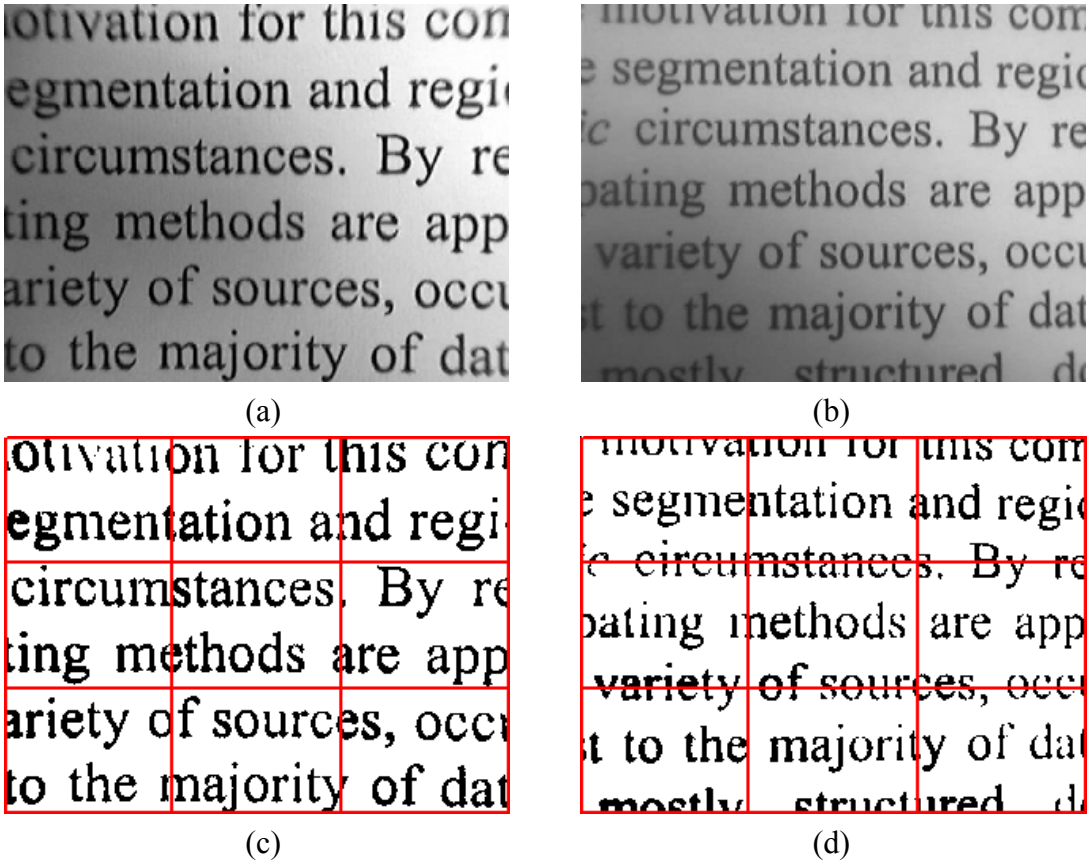
Our method relies on the results of two learning processes. The first process determines the value of k for the $k\times k$ division of images, while the second constructs four decision functions. It is natural to ask: How sensitive is our binarization method to the results of the learning? For example, what would happen if we construct the decision functions based on a 5×5 division of images in the training phase, and apply them to the 5×5 division of images in the testing phase? We show the OCR performance of this division in the second row of Table 5, with the OCR performance of the 3×3 division shown in the first row. Interestingly, the different decision functions yield almost the same OCR performance.

Table 5. The OCR performance of the decision functions trained on $k\times k$ divisions of images and applied to $k\times k$, where $k = 3$ or 5.

Division Number	Recall Rate	Precision Rate	F_1 Score
3×3	97.12	96.88	97
5×5	97.19	96.75	96.97

Since all our learning processes are based on images taken by one type of camera, we want to determine if we can obtain satisfactory binarized results by applying our decision functions to images produced by other types of camera. To find out, we produced two raw images using a Logitech QuickCam Pro 4000 and a Logitech QuickCam Zoom. The images, which were taken under the inadequate illumination condition, are shown in Figs. 11a and 11b respectively. We apply the same decision functions derived in the learning process using images produced by an ORITE I-CAM 1300 one-chip color camera, as noted earlier. The binarized results of the two images, shown in Figs. 11c and 11d respectively, appear to have fine visual quality. This demonstration by no means implies that the same decision functions can

yield good binarized results of images produced by all possible types of camera. However, it does suggest that the learned decision rules based on images created by one type of camera achieve satisfactory results when applied to images created by cameras manufactured with similar technology and specifications. This may be acceptable in terms of applications, since the binarization rules are likely to be run by an operating system embedded in the camera and manufacturers would like those rules to be tuned to their type of camera.

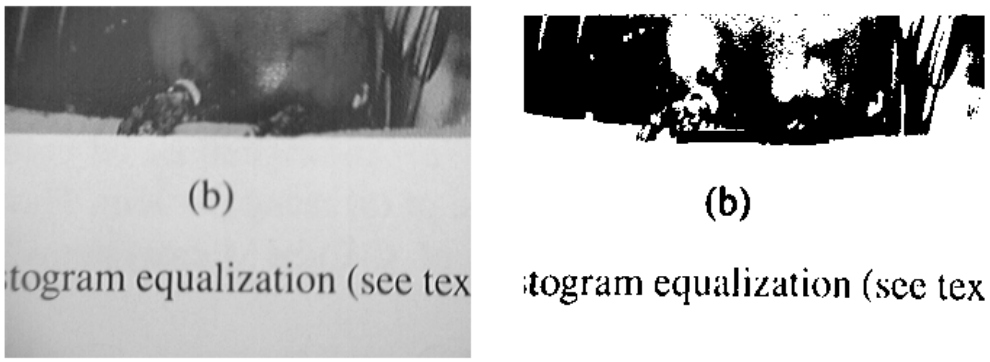


(a) (b) (c) (d)

Fig. 11. Binarization results of an image produced under the inadequate illumination condition. (a) Raw image generated by Logitech QuickCam Pro 4000. (b) Raw image generated by Logitech QuickCam Zoom. (c) Binarized result of (a). (d) Binarized result of (b).

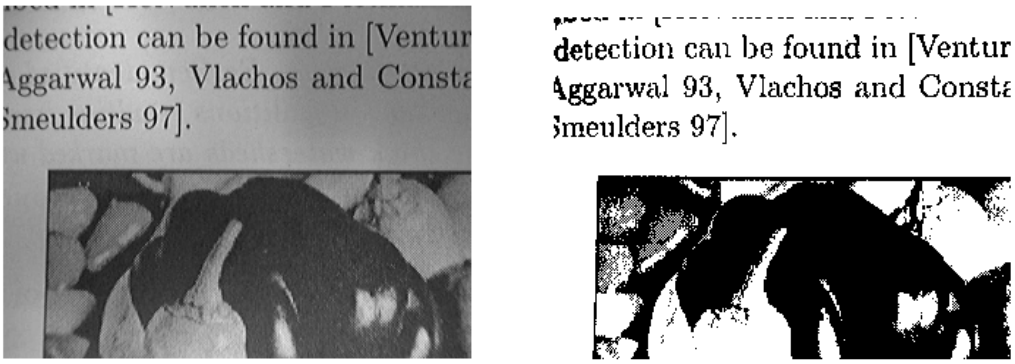
It would be reasonable to ask: How does our method handle documents that contain both text and pictures? In fact, the co-existence of text and pictures has relatively little effect on our method's performance. To demonstrate this assertion, Fig. 12a shows a document image with both types of content taken under a normal illumination condition, while Fig. 13a shows such

an image taken under the inadequate illumination condition. The results, shown in Fig. 12b and 13b, respectively, demonstrate that binarization is not affected by the occurrence of both text and picture content. Moreover, in Table 6, we detail the OCR performance of ABBYY on the binarized images to further demonstrate that pictures do not affect text recognition.



(a) (b)

Fig. 12. (a) A raw image taken under a normal illumination condition. (b) The image binarized by our method.



(a) (b)

Fig. 13. (a) A raw image taken under the inadequate illumination condition. (b) The image binarized by our method.

Table 6. OCR performance on the images binarized by our method.

	Recall Rate	Precision Rate	F_1 Score
Binarized Image in Fig. 12b	96.43	96.43	96.43
Binarized Image in Fig. 13b	98.46	95.52	96.97

5. Conclusion

Document images with non-uniform brightness require binarization methods with delicate local thresholds that must be determined according to various conditions. For this purpose, we propose a region-based binarization method. We use the SVM method to construct decision functions from the information provided by training samples and use these rules to decide what binarization action to take for each region. The experiments produce favorable results, judged in terms of the visual quality of images and also in terms of the OCR performance.

References

[1] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146-168, 2004.

[2] M. I. Sezan, ''A peak detection algorithm and its application to histogram-based image data reduction,'' *Computer Vision, Graphics, and Image Processing*, vol. 49, no. 1, 36-51, 1990.

[3] A. Rosenfeld and P. de la Torre, "Histogram concavity analysis as an aid in threshold selection," *IEEE Trans. on System, Man, and Cybernetics,* vol. 13, pp. 231–235, 1983.

[4] T. Pavlidis, "Threshold selection using second derivatives of the gray-scale image," *Proc. ICDAR*, pp. 274–277, 1993.

[5] N. Otsu, "A thresholding selection method from gray-scale histogram," *IEEE Trans. on System, Man, and Cybernetics*, vol. 9, pp. 62-66, 1979.

[6] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognition*, vol. 19, no. 1, pp. 41-47, 1886.

[7] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Comput. Vision Graphics Image Process.*, vol. 29, pp. 273-285, 1985.

[8] L. Hertz and R. W. Schafer, ''Multilevel thresholding using edge matching,'' *Computer Vision, Graphics, and Image Processing*, vol. 44, pp. 279–295, 1988.

[9] L. K. Huang and M. J. J. Wang, "Image thresholding by minimizing the measures of fuzziness," *Pattern Recognition*, vol. 28, pp. 41-51, 1995.

[10] S. Abutableb, "Automatic thresholding of gray-level pictures using two-dimensional en-

tropy," *Comput. Vision Graphics Image Process.*, vol. 47, pp. 22-32, 1989.

[11] J. Bernsen, "Dynamic thresholding for gray-level images," *Proc. Eighth International Conf. Pattern Recognition*, pp. 1251-1255, Paris, 1986.

[12] W. Niblack, *An introduction to Digital Image Processing*, pp. 115-116, Englewood Cliffs, N.J.: Prentice Hall, 1986.

[13] T. Taxt, P. J. Flynn, and A. K. Jain, "Segmentation of document images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11 no. 12, pp. 1322-1329, 1989.

[14] L. Eikvil, T. Taxt, and K. Moen, "A fast adaptive method for binarization of document images," *Proc. ICDAR*, pp. 435-443, 1991.

[15] K. V. Mardia and T. J. Hainsworth, "A spatial thresholding method for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10 no. 8, pp. 919-927, 1988.

[16] C. K. Chow and T. Kaneko, "Automatic detection of the left ventricle from Cineangiorgrams," *Computers and Biomedical Research*, vol. 5, pp. 388-410 1972.

[17] Y. Nakagawa and A. Rosenfeld, "Some experiments on variable thresholding," *Pattern Recognition*, vol. 11, no. 3, pp. 191-204, 1979.

[18] J. M. White and G. D. Rohrer, "Imager segmentation for optical character recognition and other applications requiring character image extraction," *IBM J. Research and Development*, vol. 27 no. 4, pp. 400-411, 1983.

[19] Y. Yasuda, M. Dubois, and T. S. Huang, ''Data compression for check processing machines,'' *Proc. IEEE*, vol. 68, pp. 874–885, 1980.

[20] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognition,* vol. 33, pp. 225-236, 2000.

[21] J. Sauvola, T. Seppänen, S. Haapakoski, and Pietikäinen, "Adaptive Document Binarization," *Proc. ICDAR*, pp. 147-152, 1997.

[22] J. Kim, "Multi-window binarization of camera image for document recognition," *Ninth International Workshop on Frontiers in Handwriting Recognition*, pp. 323–327, 2004.

[23] Ø. D. Trier, and T. Taxt, "Improvement of 'integrated function algorithm' for binarization of document images," *Pattern Recognition Letters*, vol. 16, no. 3, pp. 277-283, 1995.

[24] R. Parker, "Gray level thresholding in badly illuminated images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 813-819, 1991.

[25] S. D. Yanowitz and A. M. Bruckstein, ''A new method for image segmentation,'' *Computer Vision, Graphical and Image Processing*, vol. 46, no. 1, pp. 82–95, 1989.

[26] M. Kamel and A. Zhao, ''Extraction of binary character/graphics images from grayscale

document images,'' *Computer Vision, Graphical and Image Processing*, vol. 55, no. 3, pp. 203–217, 1993.

[27] Y. Yang and H. Yan, "An adaptive logical method for binarization of degraded document images," *Pattern Recognition*, vol. 33, no. 5, pp. 787-807, 2000.

[28] X. Ye, M. Cheriet and C. Y. Suen, "Stroke-model-based character extraction from gray-level document images," *IEEE Trans. on Image Processing*, vol. 10, no.8, pp 1152-1161, 2001.

[29] Ø. D. Trier and T. Taxt, "Evaluation of binarization methods for document images for document images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.17, no. 3, pp. 312-315, 1995.

[30] Ø. D. Trier and A. K. Jain, "Goal-Directed Evaluation of Binarization Methods," *IEEE Trans. on Pattern Anal. Mach. Intell*, vol. 17, no. 12, pp. 1191-1201, 1995.

[31] S. Weszka and A. Rosenfeld, "Threshold evaluation techniques," *IEEE Trans on System Man and Cybernetics*, SMC-8, pp. 622-629, 1978.

[32] P. W. Palumbo, P. Swaminathan, and S. N. Srihari, "Document image binarization: Evaluation of algorithms," *Proc. of SPIE*, vol. 697, pp. 278–285, 1986.

[33] P. K. Sahoo , S. Soltani , A. K. C. Wong , Y. C. Chen, "A survey of thresholding techniques," *Computer Vision, Graphics, and Image Processing*, vol. 41, no. 2, pp. 233-260, 1988.

[34] S. U. Lee, S. Y. Chung and R. H. Park "A comparative performance study of several global thresholding techniques for segmentation," *Computer Vision, Graphics, and Image Processing*, vol. 52, 171-190, 1990.

[35] C. A. Glasbey, "An analysis of histogram-based thresholding algorithms," *CVGIP: Graphical Models and Image Processing*, vol. 55, no. 6, pp. 532–537, 1993.

[36] J. H. Park, I. H. Jang, and N. C. Kim, "Skew correction of business card images in PDA," *IEEE Pacific Rim Conference on Communications, Computers and signal Processing*, pp. 724-727, 2003.

[37] C. Cortes and V. Vapnik, "Support-vector network," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[38] V. Vapnik, *The Nature of Statistical Learning Theory*, New York: Springer Verlag, 1995.

[39] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," *Proceedings of 10th European Conference on Machine Learning*, pp. 137-142, Berlin, 1998.

[40] R. E. Schapire and Y. Singer, "BoosTexter: A boosting-based system for text categoriza-

tion," *Machine Learning*, vol. 39, no. 2/3, pp. 135-168, 2000.

[41] R. Moutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, pp. 1757-1771, 2004.

[42] Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, L. Jackel, Y. LeCun, U. Muller, E. Sackinger, P. Simard, and V. Vapnik, "Comparison of classifier methods: A case study in handwriting digit recognition," *Proc. Int. Conf. Pattern Recognition*, pp. 77–87, 1994.

[43] C. J. van Rijsbergen, *Information Retrieval*, Butterworths, London, 1979.

[44] D. D. Lewis, "Evaluating and optimizing autonomous text classification systems," Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 95), pp. 246–254, 1995.

[45] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415-425, 2002.