

Research Article

Practical Recognition System for Text Printed on Clear Reflected Material

Khader Mohammad¹ and Sos Agaian²

¹ *Ingram School of Engineering, Texas State University-San Marcos, 601 University Drive, San Marcos, TX 78666-4684, USA*

² *Electrical and Computer Engineering, University of Texas at San Antonio, One UTSA Circle, San Antonio, TX 78249, USA*

Correspondence should be addressed to Khader Mohammad, hajkhader@gmail.com

Received 19 May 2012; Accepted 30 July 2012

Academic Editors: A. Gasteratos, D. Hernandez, and A. Prati

Copyright © 2012 K. Mohammad and S. Agaian. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Text embedded in an image contains useful information for applications in the medical, industrial, commercial, and research fields. While many systems have been designed to correctly identify text in images, no work addressing the recognition of degraded text on clear plastic has been found. This paper posits novel methods and an apparatus for extracting text from an image with the practical assumption: (a) poor background contrast, (b) white, curved, and/or differing fonts or character width between sets of images, (c) dotted text printed on curved reflective material, and/or (d) touching characters. Methods were evaluated using a total of 100 unique test images containing a variety of texts captured from water bottles. These tests averaged a processing time of ~10 seconds (using MATLAB R2008A on an HP 8510 W with 4 G of RAM and 2.3 GHz of processor speed), and experimental results yielded an average recognition rate of 90 to 93% using customized systems generated by the proposed development.

1. Introduction

Recognition of degraded characters is a challenging problem in the field of image processing and optical character recognition (OCR). The accuracy and the efficiency of OCR applications are dependent upon the quality of the input image [1–3]. Security applications and data processing have increased the interest in this area dramatically. Therefore, the ability to replicate and distribute extracted data becomes more important [4, 5].

In [6], Jung et al. presented a survey of the term text information extraction (TIE) within an image, by assuming that there is no prior knowledge such as location, orientation, number of characters, font, color, and so on. They also noted that: (a) the variations of text due to differences in size, style, orientation, and alignment, as well as low image contrast and complex background make the problem of automatic text extraction extremely challenging; (b) variety of approaches to TIE from images and video have been proposed for specific applications, such as page segmentation, address block location, license plate location, and content-based image/video indexing. In spite of such extensive studies, it

still remains laborious to design a general-purpose OCR system [5] for the reason that there is an abundance of possible sources of variation when extracting text from a shaded or textured background, from low-contrast or complex images, or from images having variations in font size, style, color, and orientation [6]. These variations make the problem of automatic TIE extremely arduous and convoluted.

Recently, many sophisticated and efficient text recognition algorithms have been proposed [7–11]. A survey of text information extraction in images and video is presented in [6, 12, 13].

Babu et al. in [14] proposed degraded character recognition from historical printed books using a coupling of vertical and horizontal hidden Markov models. Other studies focused on text localization methods which directly measured pixel-by-pixel differences between two images and quantified them as the average of the squares using a log scale [15]. Despite their relative simplicity, these algorithms have the drawback of being unable to assess similarity across different distortion types [16] and possess a greater computational complexity [17].



FIGURE 1: Images white text on white bottle.

Prior researches have also examined image invariants that handle scaled, rotated, skewed, blurred, and noise-distorted images [18, 19]. One recent research study dealt with degraded character recognition [20], while another research has focused on single characters [21]. Another study has been done on shape-based invariants, which are most commonly used for character recognition [22]. Methods of text extraction from colored (TEFC) images are presented in [23].

The process of extracting text from binary images has been presented in [17, 24, 25]. The research by Likforman-Sulem and Sigelle illustrated an example of how to convert a gray image to a binary image and how to locate the text area using connected component techniques [20]. Some enhanced methods used morphology [26], while other approaches used algorithms with performance measures. Other researchers employed skew correction and normalization operations for enhancements and parallel classification [27–29].

These approaches have been adopted for common applications such as locating an address on postal mail or a courtesy amount on checks mainly because of the simplicity in implementation [30]. The current researches are focusing on developing OCR systems for limited domain applications such as postal address reading [31], check sorting [32], tax reading [33], and office automation for text entry [6] and others. However, there are several practical security applications where one needs to have an optical dotted lines/character recognition system scanned from the reflection of light and the curvature of water bottles images (see Figure 1).

Since the quality of clear reflected material images is much lower than printed or handwritten texts images that employ commonly used character segmentation and recognition algorithms, it is complicated to gain the high success rates by directly applying them to these images. Figure 1 illustrates some bottle text images. It continues to be a challenging problem for an automatic recognition system to detect text from these images because they may face unexpected conditions such as closing loops, spurious branches, and shiny or raised text. Other conditions could be related to the surface on which the text is printed, such as curvature, indentation, or matte finishes.

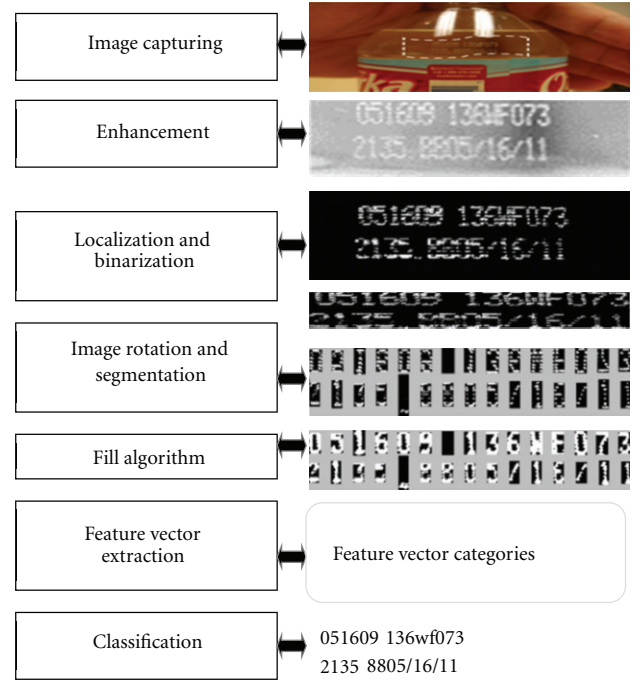


FIGURE 2: System overview.

As a result, new OCR tools should naturally be developed to use the unique properties of text in clear reflected material images.

This paper presents an application-oriented dotted text localization and character segmentation algorithms in clear reflected material images. We have successfully applied the proposed methods in developing text localization and character segmentation algorithms on curved plastic bottles images including Ozarka and Dasani bottled water brand names.

The rest of paper is organized as follows: Section 2 explicates the proposed system and covers the image capturing device (camera), enhancement, text localization and binarization; segmentation and Section 3 explains classification and recognition flow which includes the training set, feature vectors, and classification; Section 4 exhibits computer simulation results and comparison to other systems; Section 5 gives some concluding remarks and perspectives.

2. Proposed System

This section summarizes the proposed system along with components therein. System overview is summarized by the block diagram shown in Figure 2. It includes image capturing, preprocessing, localization, filling, rotation segmentation, and recognition.

2.1. Image Capturing. The capturing device selection satisfies the need for recognition application. The device resolution, focus, aperture, shutter speed, and price are the main factors for selection. Experimental testing shows that camera with a single lens reflex (SLR) has an extremely short shutter speed,

high resolution, and excellent close-up capability. In addition of low cost, one of the advantages of this kind of camera is that a user can visualize the image and send it remotely to computer station for analyses. This reduces the amount of memory storage on the device.

2.2. Preprocessing. The goal of preprocessing is to reduce irrelevant information such as noise and increase both contrast and sharpness of images. Different enhancement options are proposed to empower the system and to enable portability. Accuracy and speed of recognition stand as two of the highest concerns in any type of enhancement. The measure of enhancement method is shown in (1). It checks and compares between different options where the original image is divided into m blocks with $(k1 \times k2)$ size before processing each block. The variable α , $I_{\min k,l}^w$ and $I_{\max k,l}^w$ are the maximum and minimum intensity value in a given block, C is a small constant equal to 0.0001 to avoid dividing by 0.

$$\text{Mes_Enh} = \frac{1}{k_1 k_2} \sum_{l=1}^{k_2} \sum_{k=1}^{k_1} \alpha \left(\frac{I_{\max k,l}^w}{I_{\min k,l}^w + c} \right)^\alpha \ln \left(\frac{I_{\max k,l}^w}{I_{\min k,l}^w + c} \right). \quad (1)$$

This measure used for choosing the best parameters among set of images for the local enhancement algorithms. The process of enhancement goes through the three stages; the first one is a local one which is applied on the original image, followed by global, and finally background removal method.

2.3. Algorithm for Local Image Enhancement. Local image enhancement: the logarithmic model as presented in [26, 34] can be used for both contrast and sharpness of the image. It effectively enhances details in very dark or very bright areas (adjust image sharpness) of the image. The following are the proposed formulas used for computation:

$$\log(\bar{f}(i, j) = (\alpha - \beta) \log(\bar{a}(i, j)) + \beta \log(\bar{f}(i, j)), \quad (2)$$

$$\log(\bar{f}'(i, j) = \beta [\log(\bar{f}(i, j)) - \log(\bar{a}(i, j))],$$

$$f \oplus g = f + g - \frac{fg}{\gamma(M)}, \quad (3)$$

where f and g are the generated vectors from (2), $\gamma(M) = AM + B$, A and B are constant parameters [35], and $\bar{f}(i, j)$ and $\bar{f}'(i, j)$ are the normalized original image and the normalized enhanced image. The arithmetic mean of a (3×3) window of the original image is $\bar{a}(i, j)$, where the parameters α and β controls the contrast. Therefore, from (2) when $\beta > 1$, the sharpness of the image increases as β increases, conversely if $\beta < 1$, the resulting image is blurred.

This flexibility is not available with many conventional methods including histogram equalization. The flexibility and quality of a logarithmic image-processing algorithm make it a very effective technique for text recognition applications. Figure 3 displays the proposed algorithm example test case results using a window size of 10×10 with $\alpha = 5$

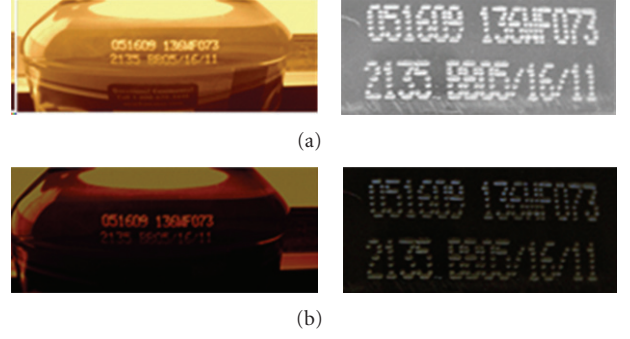


FIGURE 3: Enhancement test case images (1st row is original, 2nd row is enhanced image).

and $\beta = 8.5$. The results demonstrate that the algorithm works better for the gray images. Using larger window size will generate an image with sharper edges, which may be desirable for edge detection applications. On the other hand, some of the details of the original image get lost as the window size increases. Larger window size has a disadvantage as the algorithm becomes more computational and time consuming.

2.4. Algorithm for Global Image Enhancement. The proposed global enhancement method is based on converting a typical 24-bit RGB colored image to a grayscale image by performing six conversions methods. Note that each pixel component (e.g., R in RGB) ranges in value from 0 to 255 and is represented by an eight-bit binary number. As well, a digital color image may be easily converted from one color coordinate system to another. This method differs from what was presented by Adolf and Agaian in [23, 36] by factoring color intensity log function and applying logarithmic operations.

This paper employs the following conversions methods to map and generate gray images out of red (R), green (G), and blue (B) colors using different weight factors (a_K for red, b_K for green, C_K for blue, where $k = 1 : 6$).

To avoid division by zero, the corresponding value of the color component of the received image is biased upward by one. A crucial aspect to the text extraction process is the proposed conversion of the received color image into a grayscale in a manner that maximizes the contrast between any text in the image and the rest of the image. The proposed process uses different combinations of weight factors and logarithmic functions before the gray images are converted to binary images using a defined threshold (T) as shown below.

The following conversion process applies with respect to pixels in the following manner.

- (i) The first gray image (C_{1ij}) is based on the average of red (R_{ij}) and green (G_{ij}) components, as in (4). The “ i ” and “ j ” represent the row and column pixel location for all images.

$$C_{1ij} = \text{avg}(R_{ij}, G_{ij}) = (a_k * R_{ij} + b * G_{ij}). \quad (4)$$

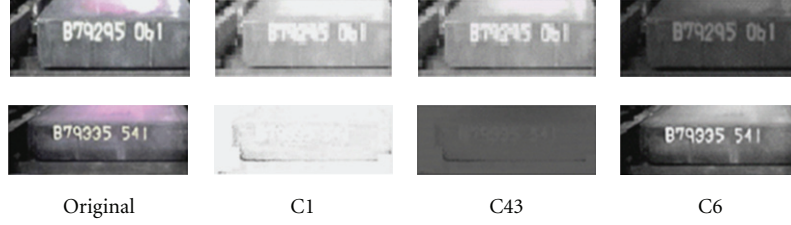


FIGURE 4: Samples results based on color image algorithm.

- (ii) The second gray image is a result of dividing the minimum of the three color components values by the average of the color components values, as in (5).

$$C2_{ij} = \frac{\min(R_{ij}, G_{ij}, B_{ij})}{\text{avg}(R_{ij}, G_{ij}, B_{ij})}. \quad (5)$$

- (iii) The third image is a result of dividing the minimum values of the three color components by the maximum values of the color components, as in (6).

$$C3_{ij} = \frac{\min(R_{ij}, G_{ij}, B_{ij})}{\max(R_{ij}, G_{ij}, B_{ij})}. \quad (6)$$

- (iv) The fourth set selected consists of 6 colors images, which results from all possible combinations of the colors by dividing one of the values by the other values, as in (7), and using the log for the other set of colors.

$$C4n_{ij} = \frac{\text{one}(R_{ij}, G_{ij}, B_{ij})}{\text{another}(R_{ij}, G_{ij}, B_{ij})},$$

$$C41_{ij-\log} = \frac{\log(R_{ij})^{a_k}}{G_{ij}^{b_k}}, \quad C44_{ij-\log} = \frac{\log(G_{ij})^{c_k}}{B_{ij}^{c_k}},$$

$$C42_{ij-\log} = \frac{\log(R_{ij})^{a_k}}{B_{ij}^{c_k}}, \quad C45_{ij-\log} = \frac{\log(B_{ij})^{c_k}}{R_{ij}^{c_k}},$$

$$C43_{ij-\log} = \frac{\log(G_{ij})^{b_k}}{R_{ij}^{a_k}}, \quad C46_{ij-\log} = \frac{\log(B_{ij})^{c_k}}{G_{ij}^{b_k}}, \quad (7)$$

where $n = 1 : 6, k = 1 : 6$.

One means one of the colors (take log function), another means another log function of another color as follows. The fifth set of gray images (6 images) is a result of dividing a sum of a combination of two values of (R, G, and B) by the sum of all of the values, as in (8).

$$C5n_{ij} = \frac{\text{comb}(R_{ij}, G_{ij}, B_{ij})}{\text{sum}(R_{ij}, G_{ij}, B_{ij})}. \quad (8)$$

TABLE 1: Weight factors.

K	C1	C2	C3	$C4n$	$C5n$	$C6n$
a_k	0.1	0.7	0.1	0.2	0.4	0.5
b_k	0.1	0.2	0.2	0.2	0.1	0.2
c_k	0.8	0.1	0.7	0.6	0.5	0.3

The other newly proposed set of images (6 totals) was also generated based on $C4_{ij}$. This new set is defined as shown in the (9), where $n = 1 : 6$.

$$C6n_{ij} = \frac{\{C4n_{ij} * \text{average of } (3 \times 3) \text{ block of } C41(i, j)\}}{\{C4n_{ij} + (\text{average of } (3 \times 3) \text{ block of } C41(i, j))\}}. \quad (9)$$

For C1 and $C4n$ images, various weight factors are used, and the optimal factors are selected for each set of images experimentally. A nested loop is implemented to find the best combination factors of a , b , and c . The best output image is selected or used, based on the highest intensity difference between text and background. Table 1 shows set of factors for the images we used.

For all generated grayscale image as discussed is converted to a new image C_{new} as shown in (10), with an optimal factor of $p = 8$. This factor is needed so that the nearest values (close pixel intensity) are spread out.

$$C_{\text{new}} = \text{floor} \left\{ \frac{(2^p - 1)[Cn - \min(\text{allpixelvalues})]}{\max(\text{allpixelvalue}) - \min(\text{allpixelvalues})} \right\}. \quad (10)$$

Figure 4 shows samples of enhancement results. The last column in the figure shows the best result (in terms of image quality and intensity difference between text and background). In comparison, the new methods illustrate that conventional enhancement (histogram, linearization, and local deviations) does not yield successful results even for fairly clean images, as shown in Figure 5.

To summarize, the following algorithm is what is been used as default system enhancement.

Background Removal Enhancements Algorithm. This is the last step for enhancement. The steps for the proposed algorithm are as follows.

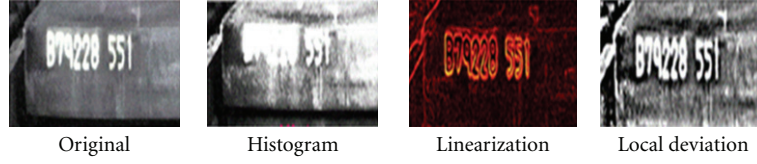


FIGURE 5: Test cases for nonworking enhancement using conventional methods.

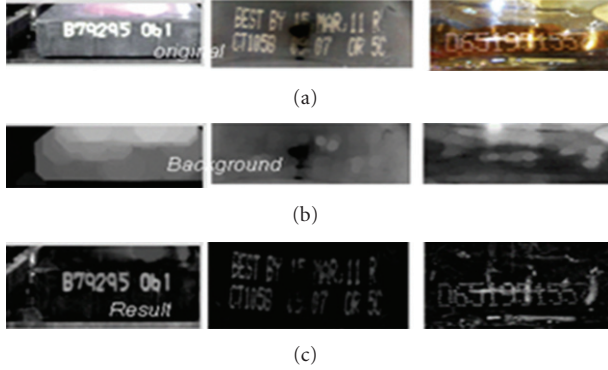


FIGURE 6: Background removal enhancement (1st row original, 2nd row background, and 3rd row the enhanced images).

Input Image (output of global enhancement)

Step 1: Examine adjacent pixels. If the pixel is significantly darker than the adjacent pixels, it is converted to black; otherwise it is converted to white (I_{nbkgd}). Using this method the new value is determined based upon local conditions and not on a single value. The new image (I_{new}) is generated using background image (I_{nbkgd}) which is generated from (I_{grad}) image.

Step 2: In this final step, the enhanced image is generated by combining the newly generated image ($I_{new2} = I_{gray} + I_{new}$) with the average Image ($I_{avg} = (I_{new} + I_{gray})/2$).

Output Image: Enhanced image.

Figure 6 demonstrates the proposed background removal enhancements example. The background removal is best suited for this system and is used as a default enhancement.

2.5. Finding the Region of Interest (Text Localization). In this section, a text location algorithm with candidate location is presented. Intensity of a plastic image is a main information source for text detection, but it maintains sensitivity to lighting variations. On the other hand, the gradient of the intensity (edge) is less sensitive to lighting changes. An edge-based feature is used for detection phase with the following assumptions.

- (i) The text is designed with high contrast to its background in both color and intensity images.
- (ii) The characters in the same context have almost the same size in most of the cases.



FIGURE 7: Edge detection process test case for water bottles.

- (iii) The characters in the same context have almost the same foreground and background patterns.

Most approaches for text identification refer to gray or binary document images. Only recently, some techniques have been proposed for text identification and extraction in color documents. Recently, Wang et al. [37] proposed a color text image binarization technique based on a color quantization procedure and a binary texture analysis; however, this technique neglects the identification of the text region stage that does not include a page layout analysis technique. Thillou and Gosselin [38] proposed a color binarization technique for complex camera-based images based on wavelet denoising and a color clustering with K-means; however, in this approach the technique does not include any text extraction stage and is applied only for document images containing already detected text.

The proposed approach uses edge detection and pixel intensity for localizing the edges of the text. After the edges are computed, the number of edges in the x and y directions is calculated. If the result is higher than a certain threshold, it will be considered as a text area. Then, each text area will be binarized using the luminance values. The noise around the digit will greatly affect this method, so a high quality preprocessing step is crucial for the localization approach. Having the right threshold eradicates noise and improves the binary image quality. In this research, gray image is used as the default color map for water bottles. Binarization is based on gray threshold values for a localized area.

Edge detection algorithms have been chosen for the water bottle system. Figure 7 shows examples of test cases for water bottles where (a) represents original image, (b) is the text

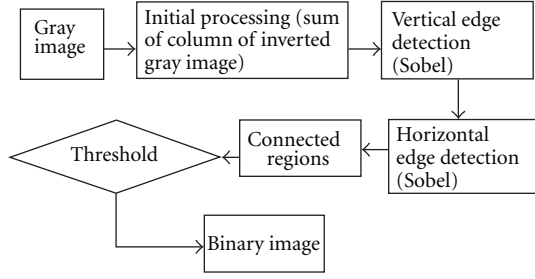


FIGURE 8: Text localization and binarization algorithm.

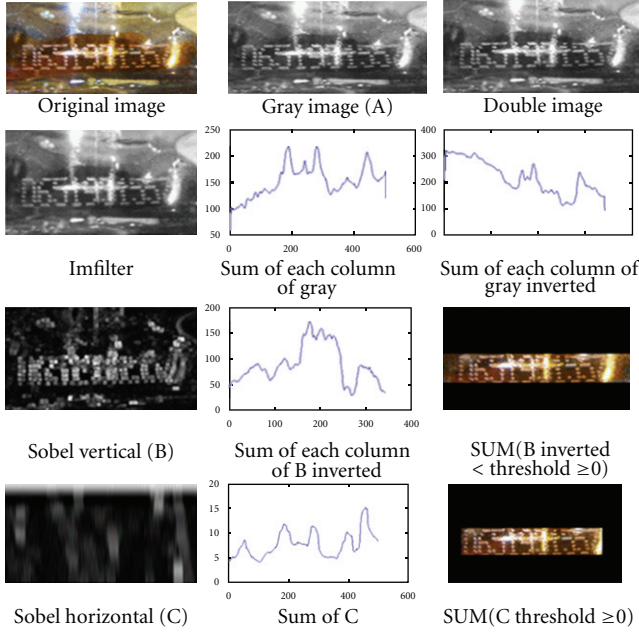


FIGURE 9: Simulation results using text localizations example.

area, (c) is the enhanced image, and (D-I) represents Gaussian, Sobel, Laplacian, log, average, and unsharp detection, respectively. Finding the region of interest algorithm is listed below and shown in Figure 8. Figure 9 show examples of test cases for slabs and water bottles, respectively.

Input: The candidates' image with text.

Step 1: The original image is converted to gray. This conversion is done using the standard conversion formula that is used for calculating the effective luminance of a pixel as shown in (11).

$$\text{Intensity} = 0.2989 * \text{red} + 0.5870 * \text{green} + 0.1140 * \text{blue}. \quad (11)$$

Step 2: Enhancing the gray image and calculating the sum of column of inverted image.

Step 3: Applying the log (laplacian of Gaussian) edge detection vertically and horizontally.

Step 4: Identifying the connected regions. This is based on scanning the image, pixel-by-pixel from

top to bottom and left to right in order to identify connected pixel regions, that is, regions of adjacent pixels which share the same set of intensity values. We use the procedure below for detecting text region of horizontal and vertical direction.

$$\text{Horizontal} = \frac{\text{Sum of horizontal edges in selected region}}{\text{Selected region size}}, \quad (12)$$

$$\text{Vertical} = \frac{\text{Sum of vertical edges in selected region}}{\text{Image width}}. \quad (13)$$

Step 5: Calculating local threshold to get a binary image.

The original image is segmented into 6 images to get more accurate threshold. The threshold (T) is defined by (14) in which the result of multiplying the optimal factor with the standard deviation of the gray image is subtracted from the maximum intensity value of the gray image.

$$T = \max(C_{mij}) - p * \text{std}(C_{mij}). \quad (14)$$

Output: The created region of interest.

2.6. Text Segmentation Approach. This section details the new algorithm for rotation, segmentation, and filling. Before applying the segmentation approach, the rotation and curved text handling approaches are discussed.

After the completion of the edge detection process, the utilization of the connected region procedure occurs before the binary image is generated.

2.6.1. Image Rotation. The goal in this process is to use geometry tools and operations to automatically change the orientation of the images appropriate for classification. Dealing with degraded dotted text on clear plastic poses a challenge for the segmentation of dotted and angled text. To address this problem, the image is binarized and rotated (if necessary). Then, text filling, localization, and segmentation procedures are applied. In the proposed system, the localized binary text image is used as input to the system.

The text in the image can be at different angles. Below is the algorithm pseudocode flow for determining the angle of rotation. Figure 10 summarizes a proposed rotation flow with an example.

Input: The candidates binarized image

Step 1: The algorithm identifies the location of the first nonbackground pixel (white, binary value of "1") in each binary image column starting from the bottom.

Step 2: It then identifies the location of the first nonbackground pixel (white pixel "1") from the top of each column in the previously generated image.

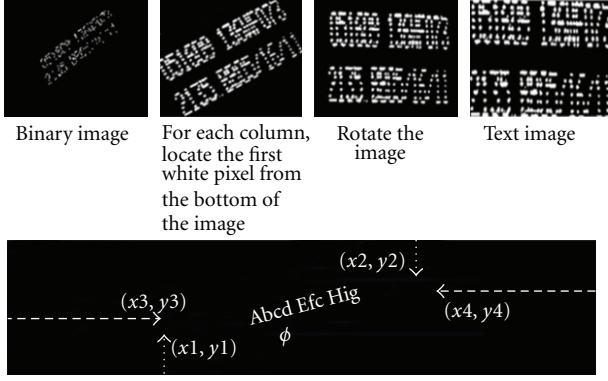


FIGURE 10: Rotation algorithm example.

TABLE 2: Rotation results/success rate for Dasani.

	D1	D2	D3	D4	D5	Average
Set1	96	97	96	99	97	97
Set2	96	100	98	99	100	99
Set3	94	100	100	97	99	98
Set4	95	99	100	10	100	98
Set5	97	98	100	99	98	99
Set6	99	100	100	99	100	98
Set7	97	100	100	98	100	99
Set8	97	99	100	96	99	98
Set9	98	100	97	99	100	99
Set10	99	97	100	96	100	98
Average	99	99	99	98	99	98.5

Step 3: Next, it identifies the four corners (top, left, bottom, and right), where the text touches.

Step 4: The rotation slope (angle ϕ) is determined using these four corners and ϕ is calculated as:

$$\begin{aligned}\phi_1 &= \sin^{-1}(\phi_1) = \frac{y_2 - y_1}{x_2 - x_1}, \\ \phi_2 &= \sin^{-1}(\phi_2) = \frac{y_4 - y_3}{x_4 - x_3}, \\ \phi &= \frac{(\phi_1 + \phi_2)}{2}.\end{aligned}\quad (15)$$

Once the angle ϕ is known, the text is rotated back to the horizontal orientation and the image is refurbished by using the select character procedure to remove all background outside the text.

Output Image: Rotated image

The rotation methods have been tested for two sets of water bottle images where the text is rotated with different angles. Tables 2 and 3 illustrate the results obtained when testing for orientation or rotation invariance for 50 Dasani images and 40 Ozarka images. Each set of images is selected for different bottles, for example set 1 has the bottle with clear day light, and set 2 during night time.

TABLE 3: Rotation results/success rate for Ozarka.

	z1	z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10
Set1	94	98	95	97	99	96	95	94	99	96
Set2	96	100	95	97	99	96	95	99	98	97
Set3	97	94	95	97	95	96	97	94	96	95
Set4	99	97	95	97	99	96	95	94	99	97

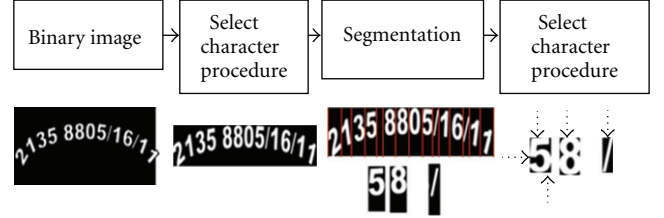


FIGURE 11: Curvature handling algorithm.

2.7. Proposed New Curved Text Handling Approach. Text images can have curved text from the curvature of the bottle or the printing of the characters. Figure 11 shows the algorithm along with an example wherein the original data is curved in the text image. The idea is to segment the image based on the black column between characters and then crop only text pixels; the boundary of the segmented character image is then touching the characters from four sides.

2.8. Segmentation. Extraction of degraded text in an image remains problematic for current algorithms to resolve. One such problem, as mentioned in [2], is the occurrence of touching characters, and this invalidates the assumption that character repetition patterns in the input text match that of a language mode. Figure 12(a) shows the new algorithm for segmentation from a binary image (text image). The segmentations steps are

Input (Binary image after rotation)

Step 1: Extract text image (text touches the 4 corners) using curved text handling approach.

Step 2: Segment vertically: Based on number of lines, split the image vertically

$$\text{Image}(j) = \frac{H}{j}, \quad (16)$$

where H is the image height and j is the number of lines.

Step 3: Segment horizontally: for each segmented image in Step 2, segment the image $\text{Image}(j)$ based on number of black columns.

$$\text{Number of characters}(i, j) = \frac{W(j)}{(N - 1)}, \quad (17)$$

where $W(j)$ is the width of each line image and N is the number of black column between characters.

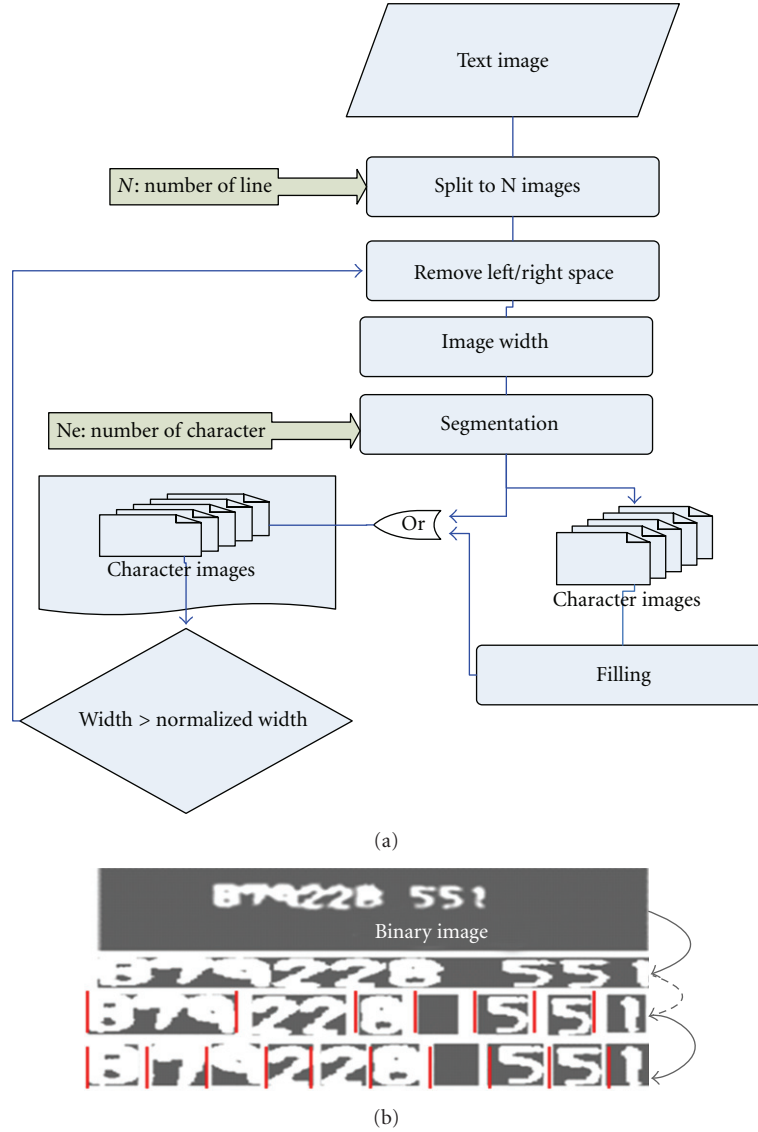


FIGURE 12: (a) Segmentation algorithm. (b) Segmentation of connected characters.

Step 4: In this step, every segmented character is checked. If it is larger than a normalized character with by 15% (this value is predetermined based on the experiment), then it is segmented again based on the normalized width, which means the characters are connected.

Output image: Characters.

The proposed segmentation in a localized binary text image is based on the black column between characters, the character width, and the text box width extracted from the text image. Based on the width of the new text image, the width of each separated image is reviewed. If the width of a segmented character is greater than a normalized width, then the segmented character is known to be composed of several characters and needs further segmentation. A new value for the normalized character width is chosen based on average width of the segmented characters. Figure 12(b)

shows segmentation of solid-connected characters from binary image.

2.9. Proposed Fill Dotted Characters. Dotted characters in the segmented images complicate the recognition process. Morphological operations such as dilation, imfill [39], and other methods of interpolation using spline curves or spline fills [39, 40] are evaluated and tested to generate solid characters. The larger the gap, the more likely these fill methods yield an erratic result. These methods did not rectify the problem because the character holes/dots are not spaced close enough together. The proposed fill algorithm is applied after segmentation to solidify characters. The proposed fill algorithm is shown in Figure 13. The proposed method is based on the shift and combine operation.

The extracted character image goes through three shift operations: (1) shift up m pixels, (2) shift down with

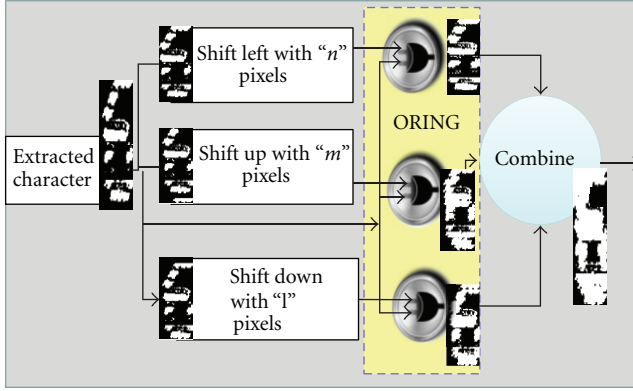


FIGURE 13: Character fill algorithm with an example.

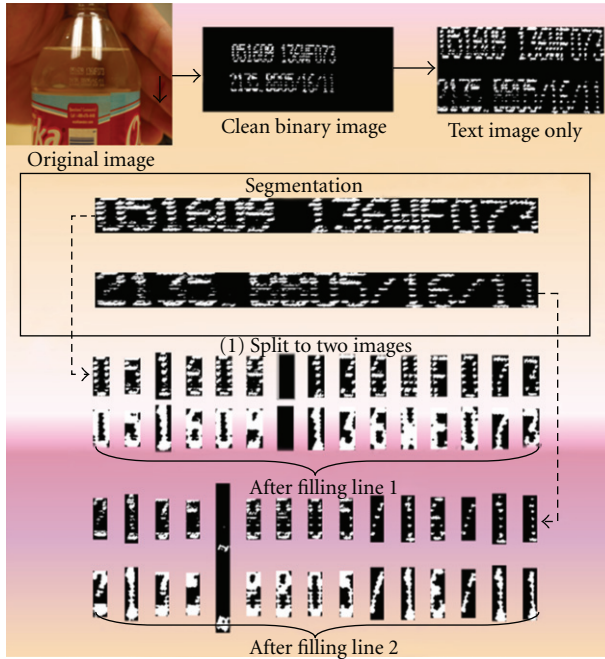


FIGURE 14: Complete test case for Ozarka bottle.

l pixel, and (3) shift left n pixels. Subsequently, a logical check of the original image with the three-shifted images occurs before combining them. The values of l , m , and n are chosen for specific systems after experimental results. Figure 14 shows results of segmentation with filling using Ozarka bottles, this example shows segmentation with filling after an image is split into N images, where N is the number of lines.

The segmentation methods have been tested for the two sets of images that total 90 images. Table 4 displays the results obtained from Ozarka images with a segmentation rate of 96.5%, while Table 5 displays the results obtained from Dasani images with a segmentation rate of 98.1%. The

TABLE 4: Segmentation rates for Ozarka.

	z1	z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10
Set1	92	96	93	95	97	94	96	92	97	94
Set2	96	199	96	98	98	95	97	97	95	97
Set3	98	95	96	98	96	97	98	95	97	97
Set4	99	99	99	98	97	96	99	98	96	98
Average	96	97	96	97	97	95	98	96	96	97

TABLE 5: Segmentation rates for Dasani.

	D1	D2	D3	D4	D5	Average
Set1	99	96	98	98	97	98
Set2	97	100	98	99	100	99
Set3	99	100	100	98	95	98
Set4	97	97	100	99	96	97
Set5	96	96	98	100	100	98
Set6	97	100	98	97	97	98
Set7	98	96	96	99	100	98
Set8	98	98	100	100	99	99
Set9	97	100	97	97	96	97
Set10	99	100	99	98	100	99
Average	98	98	98	98	98	98

TABLE 6: Recognition rate comparisons with fill/no fill and spline.

	z1	z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10
No fill	36	39	12	55	48	12	57	34	80	79
Imfill	36	39	25	56	49	20	56	32	31	70
Spline	78	84	89	86	88	84	70	90	92	85
Proposed	86	92	97	94	96	92	78	98	99	93

segmentation and recognition equation (18) demonstrates the evaluation rates.

Segmentation

$$= \frac{\text{Total number of correct segmented characters}}{\text{Total number of images in the text per image}},$$

Recognition

$$= \frac{\text{Total number of correctly recognized character}}{\text{Total number of detected character in an image}}. \quad (18)$$

Table 6 summarizes recognition using the proposed fill algorithm, the morphological fill algorithm, no fill algorithm, and spline fill.

3. Proposed Classification and Recognition Flow

The proposed recognition process uses a structural method for identifying each character. After skeletonization of the extracted character, the feature vector is extracted and compared to the database. The objective of the feature extraction

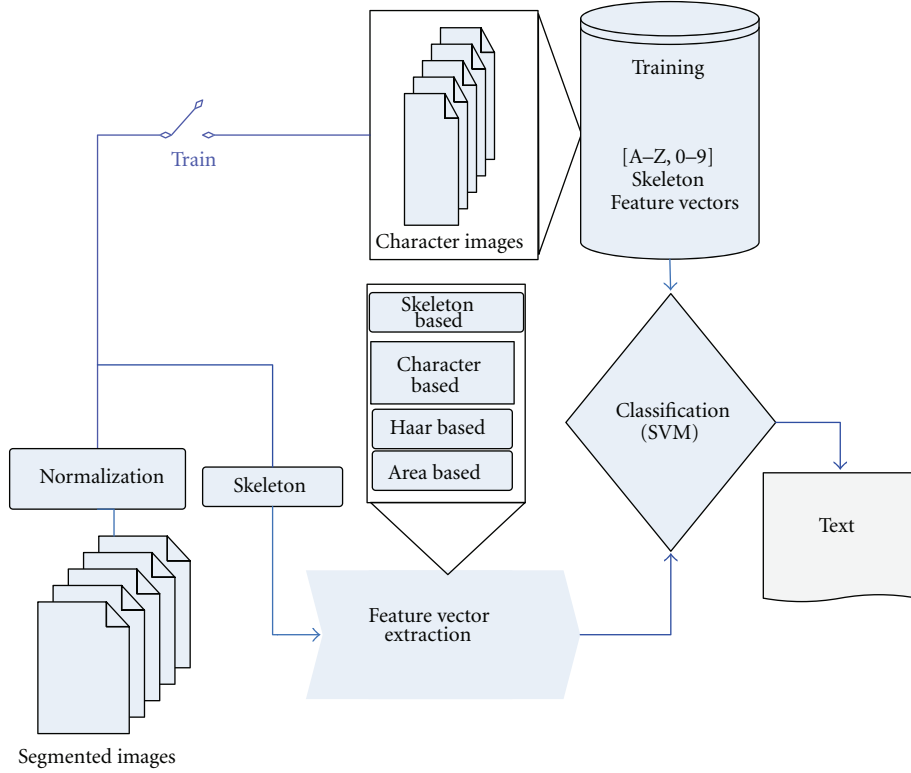


FIGURE 15: Classification and recognition.

phase is an organized pull out of a set of important features that reduce redundancy in the word image while preserving the key information for recognition. The first feature set is based on global and geometric properties. The second feature set is based on the analysis of the local properties.

Figure 15 details the recognition algorithm. An advantage of this structural method is its ability to describe the structure of a pattern explicitly. The recognition step extracts the image invariant features and applies the character recognition algorithm using set of binary support vector machine (SVM) classifiers. This is a widely used discriminative classification algorithm (51).

3.1. Feature Vectors. After isolating the characters in an image, a set of properties for each of these characters is determined.

The new set of features for single characters based on their geometric properties. The character image is segmented as shown in Figure 16 into blocks. Each block has its own feature value. The solid lines show bigger blocks and the dotted lines show the smaller blocks. The feature value is calculated in (19), where m and n are the vertical and horizontal group numbers. These values are selected based on experimental result where m and n equal 40. As the process is driven by character properties, the total number of block feature vector (BFV) is based on 26 groups per character; each group is generated based on different areas as

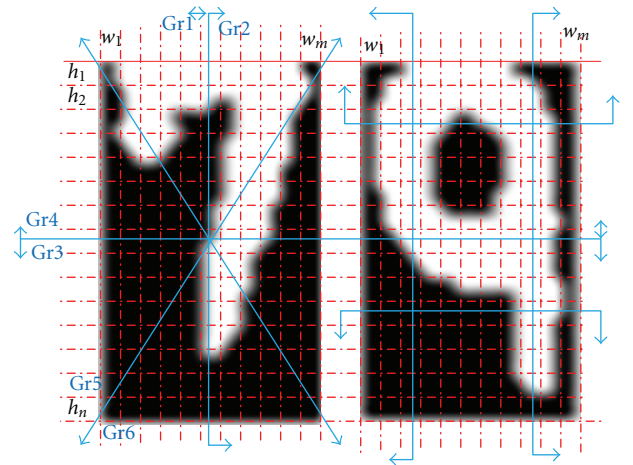


FIGURE 16: New feature vector group creation example.

seen in (20). A comparison was done from each set of features (all generated groups) and those which varied by less than 30% were removed.

$$\text{BFV} = \frac{\sum_{h=1}^n \sum_{w=1}^m \text{WhitePixel}}{\sum_{h=1}^n \sum_{w=1}^m \text{blackPixel}}. \quad (19)$$

- 1) Read character input image
- 2) Resize (normalized to 42×42)
- 3) Extract feature: $FV = \{\text{set of feature vectors}\}$
- 4) Divide characters into groups based on slope of the feature vector example:
 Group1 = set of characters $\{O, Q, G\}$
 Group2 = set of characters $\{L, I, F\}$
- 5) Divide sub groups characters into another subgroups based on slope of the new group feature vector.
- 6) Repeat 4, 5 until you have one character per group

FIGURE 17: Pseudocode algorithm for feature vector re-grouping.

Groups: $Gr(1), Gr(2), \dots, Gr(n)$.

$$Gr(1) = \frac{\sum_{h=1}^n \sum_{w=1}^{m/2} \text{WhitePixel}}{\sum_{h=1}^n \sum_{w=1}^{m/2} \text{blackPixel}}, \quad (20)$$

$$Gr(2) = \frac{\sum_{h=1}^n \sum_{w=m/2}^m \text{WhitePixel}}{\sum_{h=1}^n \sum_{w=m/2}^m \text{blackPixel}}.$$

The set of groups generated is regrouped, Figure 17 shows the pseudocode used to regroup characters based on the extracted features and an appropriate slope within the same group is reached. This approach demonstrates that characteristics of similar characters/shapes have distinguishable feature vector values. For example, (S and 8) (O, Q), and (L, I, F) are shown in Figure 18. This figure shows that for very close shapes like (O, Q), segmented from different images, the features from set (h1) and set (h2) exhibit an acceptable difference to distinguish between similar shapes.

An example of the regrouping method of feature vector results is shown in Figure 19(a) and all groups are shown in Figure 19(b). The line shows the sum of the features used to distinguish characters from each other based on the slope.

Features vectors were selected to minimize the processing time while maintaining a high accuracy rate. To maximize the recognition accuracy, three other existing set methods were used along with the proposed approach: (1) the first one is based on using 4 levels of the Haar transform [41] with a total of 6 features, (2) the second one is based on skeleton lines of the image (49) with 55 features, and (3) the third one is based on features for the area with 79 features [42]. Using a total number of 155 features per character, a target feature vector matrix is built for each system. Having a generalized target matrix for all images can slow and reduce accuracy.

3.2. Recognition. Having many different type feature vectors, the recognition is a multiclass problem. Since SVM supports only two-class recognition, a multiclass system is constructed by combining two-class SVMs as mentioned in [43].

Let $\{x_i, y_i\}$, $i = 1, 2, \dots, k$, $y_i \in \{1, -1\}$ and $x_i \in IR^n$ be the training samples where the training vector is x_i and y_i its

corresponding target value. For input pattern x , the decision function of binary classifier is shown in (13).

$$f(x) = \text{sign} \left(\sum_{i=1}^k y_i \alpha_i K(x, x_i) + b \right), \quad (21)$$

where:

$$\text{sign}(u) = \begin{cases} 1 & \text{for } u > 0, \\ -1 & \text{for } u < 0, \end{cases} \quad (22)$$

where k is the number of learning patterns, the target value of learning pattern y_i , x_i , b is the bias, and $K(x, x_i)$ is a kernel function which high-dimensional feature space that equals $K(x, x_i) = \phi(x) \cdot \phi(x_i)$.

Polynomial kernel which is shown in (23) and the Gaussian radial basis functions kernel which is shown in (24).

$$K(x, x_i) = (x \cdot x_i + 1)^p, \quad (23)$$

$$K(x, x_i) = \exp \left(\frac{\|x \cdot x_i\|^2}{2\sigma^2} \right) + 1)^p. \quad (24)$$

In (24), if p is chosen as 1, the polynomial kernel is called linear and if p is chosen as 2, it is called a quadratic kernel. (This approach can be seen in [43]). In our study, for a Gaussian radial basis function kernel, the kernel width, σ^2 is estimated from the variance of the sample vectors.

The SVM is chosen in this research for its processing speed and flexibility in using various distance methods. The classifier is tested using a different set of data with varying rules (nearest, random, consensus) and varying distances (Euclidean, cityblock, cosine, correlation). For the concrete slab and Ozarka test case systems, the highest accuracy results as shown in Table 7, where the Euclidean distance with the nearest rule gives best results (95% matching).

3.3. Proposed Training Procedure. The recognition algorithm relies on a set of learned characters and their properties. It compares the features in the segmented image file to the features in the learned set. Figure 20 shows the training database procedure. Each image can have up to 999 vectors in the training set. The input or the trained image for each character in the training set is either resized or assumed to be 24 by 42 pixels to satisfy the procedure, regardless of whether or not the image is distorted. The training databases include vectors for images of all characters and numbers along with distorted characters for system training. Training images are normalized to a fixed size.

4. Computer Simulations and Comparison

Experiments were carried out on an unconstrained Dasani bottles for both Ozarka plastic bottled water images as well as concrete slabs, which were captured at a resolution of 300 dpi. The water bottle text consists of 2 lines, each with 15 characters for Ozarka and each with varying numbers of characters for Dasani. Methods were evaluated using a

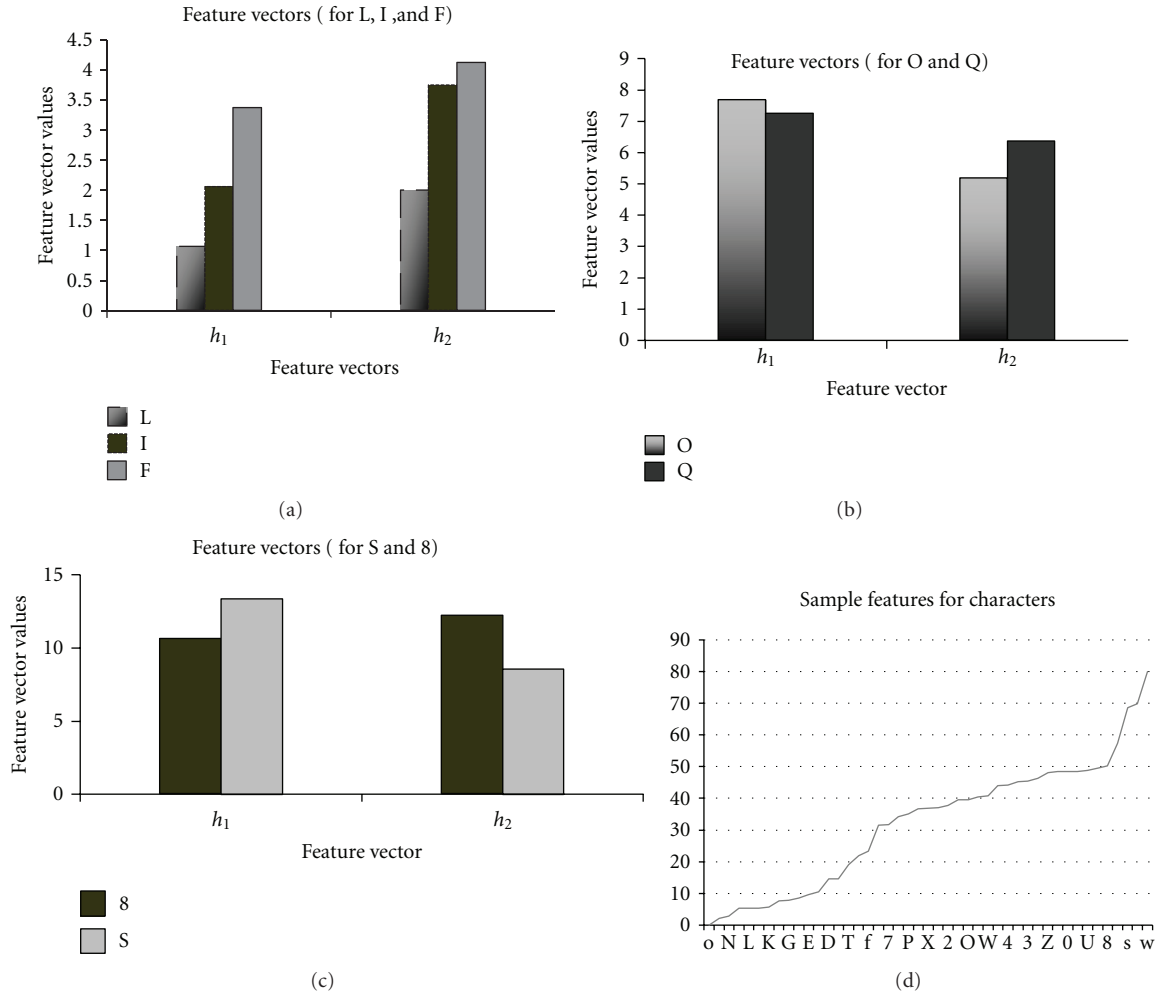


FIGURE 18: Feature vector example.

TABLE 7: Classify distance and rule test.

Actual text	Test result	Distance	Rule (nearest)
0516091363F073	051699 136wf073	Euclidean	0.93
0516091363F073	091699 136j6073	Cityblock	0.73
0516091363F073	051689/136wf073	Cosine	0.73
0516091363F073	051689/136wf073	Correlation	0.87
	Test result	Distance	Rule (random)
0516091363F073	051699 136nf073	Euclidean	0.80
0516091363F073	091699 136j6073	Cityblock	0.73
0516091363F073	1 136 0 3	Cosine	0.33
0516091363F073	031689/136jf073	Correlation	0.73
	Test result	Distance	Rule consensus
0516091363F073	1 136 3	Euclidean	0.20
0516091363F073	091699 136k6073	Cityblock	0.73
0516091363F073	091699 136k6073	Cosine	0.73
0516091363F073	1/36 3	Correlation	0.27

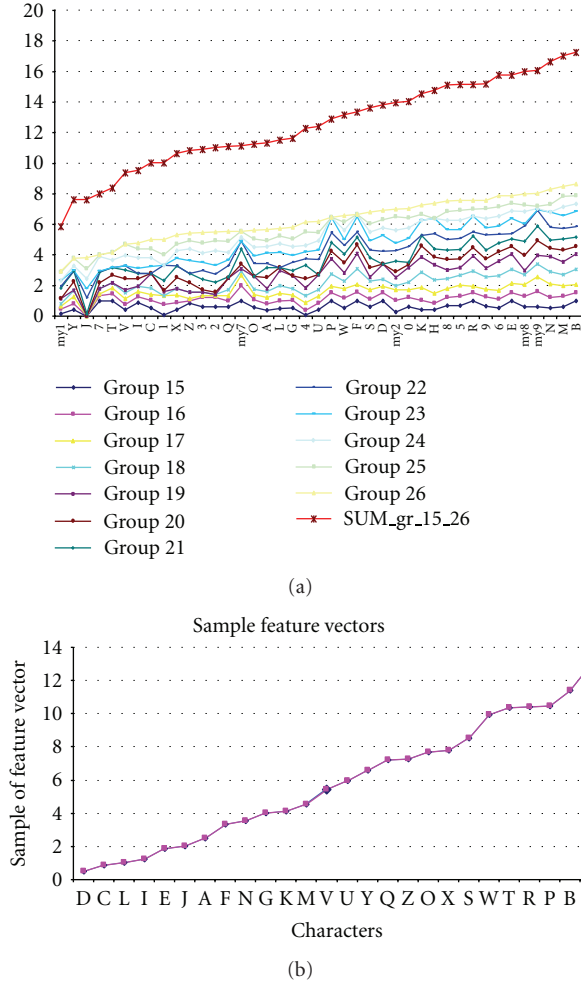


FIGURE 19: Sorted data and character regroup for feature vectors.

TABLE 8: Systems rotation accuracy with different method.

Type	Number of images	Recognition accuracy
Slab images	30	97.5
Ozarka	30	97.0
Dasani	50	98.5

total of 100 test images containing a variety of text captured from water bottles. For both Ozarka and Dasani water bottles, the text is comprised of dotted line characters, as shown in Figures 21 and 22. These were used as test case images from [6]. Images in Figure 23 were used for slab test cases. The system was built using MATLAB R2008A installed in a Compaq 8510W workstation with processor speed of 2.3 GHz and 4G of RAM.

Table 8 shows the summary results obtained from each set of pictures (systems) when tested for text localization and orientation. The segmentation results for all systems are shown in Table 9. The results in this table demonstrate impressive accuracy, especially when applying the fill algorithm. The results for slab images with varying line pictures are shown in Table 10.

TABLE 9: System segmentation accuracy with different method.

Type	Number of images	Original % accuracy	With fill accuracy %
Slab images	30	99	99
Ozarka	20	91	97
Dasani	50	80	99

TABLE 10: Recognition accuracy for slab imaging test case.

Number of images	Picture name used	Accuracy %
20	Slab 1 line picture	99
10	Slab 2 line picture	97
Average		98

The best accuracy with dotted characters was found with the Ozarka application. At a 93% accuracy rate, the elapsed time was 18.94 seconds. Dasani water bottles yielded an average accuracy of 90%, with an elapsed time of 29.5 seconds. Meanwhile, for the concrete slabs, the segmentation and recognition accuracy rate was an average of 98%.

The fill algorithm increased the accuracy by 10–20%. The greatest improvement after using the fill algorithm was for the Dasani bottles where the accuracy rate increased from 80% to 99%. The accuracy rate for Ozarka increased from 91% to 97%. Since the text in the slab images is solid lines, the fill algorithm did not affect the result.

All known research did not discuss the white text printed on reflected material, so comparison is applied to work presented in this research with relevant systems presented in other literatures [11–13, 44–51]. A comparison to their systems (as they reported), mostly from the last five years, as shown in Table 11 demonstrates that the presented system produced better results especially when compared with other related, same language, recognition systems as in [12, 44–48, 51].

5. Conclusion

In this paper, we presented a complete system for automatic detection and recognition of dotted text for application to clear reflected material images. This paper also poses novel tools for extracting text from an image with the following qualities: (a) poor background contrast, (b) white, curved, and/or differing fonts or character width between sets of images, (c) dotted text printed on curved reflective material, and (d) touching characters. Several improvements to current mechanisms were presented, including methods for:

- (i) text detection and segmentation of touching and/or closely printed characters,
- (ii) filling of dotted text,
- (iii) rotation and curvature handling.

We have successfully applied the proposed methods in developing text localization and character segmentation



FIGURE 23: Example poor in quality slab images [6].

development. We consider that our approach achieves good performance given that the data correspond to real text bottle images; however, it is hard to compare completely with other approaches since there is no similar investigation.

Disclosure

The authors do not have any financial relation with the commercial identity “Ozarka, and Dasani” mentioned in this paper.

Acknowledgments

The authors would like to thank the associate editor and anonymous reviewers for their invaluable comments and suggestions led to a great improvement on this paper. The authors also gratefully acknowledge Dr. Hani Saleh for his assistance and coding support.

References

- [1] F. Idris and S. Panchanathan, “Review of image and video indexing techniques,” *Journal of Visual Communication and Image Representation*, vol. 8, no. 2, pp. 146–166, 1997.
- [2] H. Li, D. Doermann, and O. Kia, “Automatic text detection and tracking in digital video,” *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 147–156, 2000.
- [3] R. Cattoni, T. Coianiz, S. Messoldi, and C. M. Modena, “Geometric layout analysis techniques for document image understanding a review,” ITC-IRST Technical Report #9703-09, 1998.
- [4] B. A. Yanikoglu, “Pitch-based segmentation and recognition of dot matrix text,” *International Journal on Document Analysis and Recognition*, vol. 3, no. 1, pp. 34–39, 2000.
- [5] H. Liu, M. Wu, G. F. Jin, and Y. Yan, “A post processing algorithm for the optical recognition of degraded characters,” in *Document Recognition and Retrieval VI*, vol. 3651 of *Proceedings of SPIE*, pp. 41–48, The International Society for Optical Engineering, San Jose, Calif, USA, January 1999.
- [6] K. Jung, K. I. Kim, and A. K. Jain, “Text information extraction in images and video: a survey,” *Pattern Recognition*, vol. 37, no. 5, pp. 977–997, 2004.
- [7] S. Choi, J. P. Yun, and S. W. Kim, “Text localization and character segmentation algorithms for automatic recognition of slab identification numbers,” *Optical Engineering*, vol. 48, no. 3, Article ID 037206, 2009.
- [8] K. Wang and J. A. Kangas, “Character location in scene images from digital camera,” *Pattern Recognition*, vol. 36, no. 10, pp. 2287–2299, 2003.
- [9] X. Chen, J. Yang, J. Zhang, and A. Waibel, “Automatic detection and recognition of signs from natural scenes,” *IEEE Transactions on Image Processing*, vol. 13, no. 1, pp. 87–99, 2004.
- [10] Y. Liu, S. Goto, and T. Ikenaga, “A contour-based robust algorithm for text detection in color images,” *IEICE Transactions on Information and Systems*, vol. E89-D, no. 3, pp. 1221–1230, 2006.
- [11] B. Zhu and M. Nakagawa, “Segmentation of on-line handwritten Japanese text of arbitrary line direction by a neural network for improving text recognition,” in *Proceedings of the 8th International Conference on Document Analysis and Recognition*, vol. 1, pp. 157–161, September 2005.
- [12] X. Liu, H. Fu, and Y. Jia, “Gaussian mixture modeling and learning of neighboring characters for multilingual text extraction in images,” *Pattern Recognition*, vol. 41, no. 2, pp. 484–493, 2008.
- [13] M. A. El-Shayeb, S. R. El-Beltagy, and A. Rafea, “Comparative analysis of different text segmentation algorithms on Arabic news stories,” in *Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI '07)*, pp. 441–446, August 2007.
- [14] D. R. R. Babu, M. Ravishankar, M. Kumar, K. Wadera, and A. Raj, “Degraded character recognition based on gradient pattern,” in *The 2nd International Conference on Digital Image Processing*, Proceedings of SPIE, February 2010.
- [15] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2002.
- [16] E. A. Silva, K. Panetta, and S. S. Agaian, “Quantifying image similarity using measure of enhancement by entropy,” in *Mobile Multimedia/Image Processing for Military and Security Applications*, vol. 6579 of *Proceedings of the SPIE*, April 2007, Paper #6579-32.
- [17] E. Wharton, K. Panetta, and S. Agaian, “Human visual system based similarity metrics,” in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '08)*, pp. 685–690, October 2008.
- [18] C. Fang and J. J. Hull, “A Modified character-level deciphering algorithm for OCR in degraded documents,” in *IS&T Conference on Document Recognition II*, vol. 2422 of *Proceedings of SPIE*, pp. 76–83, March 1999.
- [19] E. Y. Kim, K. Jung, K. Y. Jeong, and H. J. Kim, “Automatic text region extraction using cluster-based templates,” in *Proceedings of the of International Conference on Advances in Pattern Recognition and Digital Techniques*, pp. 418–421, 2000.
- [20] L. Likforman-Sulem and M. Sigelle, “Recognition of broken characters from historical printed books using Dynamic Bayesian Networks,” in *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR '00)*, 2000.
- [21] M. Yokobayashi and T. Wakahara, “Binarization and recognition of degraded characters using a maximum separability axis

- in color space and GAT correlation,” in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, pp. 885–888, August 2006.
- [22] D. J. Granrath, “The role of human visual models in image processing,” *Proceedings of the IEEE*, vol. 69, no. 5, pp. 552–561, 1981.
- [23] A. Cusmariu, “Method of extracting text present in a color image,” United State Patent, patent no 6519362b1, 2009.
- [24] S. Liang, M. Ahmadi, and M. Shridhar, “Segmentation of handwritten interference marks using multiple directional stroke planes and reformalized morphological approach,” *IEEE Transactions on Image Processing*, vol. 6, no. 8, pp. 1195–1202, 1997.
- [25] Y. K. Chen and J. F. Wang, “Segmentation of single- or multiple-touching handwritten numeral string using background and foreground analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1304–1317, 2000.
- [26] E. Wharton, S. Agaian, and K. Panetta, “Comparative study of logarithmic enhancement algorithms with performance measure,” in *Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning*, vol. 6064 of *Proceedings of SPIE*, January 2006, Paper #6064-12.
- [27] S. S. Agaian, K. Panetta, and A. M. Grigoryan, “Transform-based image enhancement algorithms with performance measure,” *IEEE Transactions on Image Processing*, vol. 10, no. 3, pp. 367–382, 2001.
- [28] P. Xiang, Y. Xiuzi, and Z. Sanyuan, “A hybrid method for robust car plate character recognition,” *Engineering Applications of Artificial Intelligence*, vol. 18, no. 8, pp. 963–972, 2005.
- [29] L. Xu, A. Krzyzak, and C. Y. Suen, “Methods of combining multiple classifiers and their applications to handwriting recognition,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 3, pp. 418–435, 1992.
- [30] D. Chen, J. Luetttin, and K. Shearer, “A survey of text detection and recognition in images and videos,” Institut Dalle Molle’Intelligence Artificielle Perceptive (IDIAP) Research Report IDIAP-RR, 2008.
- [31] S. N. Srihari, Y. C. Shin, V. Ramanaprasad, and D. S. Lee, “A system to read names and addresses on tax forms,” *Proceedings of the IEEE*, vol. 84, no. 7, pp. 1038–1049, 1996.
- [32] S. Gopisetty, R. Lorie, J. Mao, M. Mohiuddin, A. Sorin, and E. Yair, “Automated forms-processing software and services,” *IBM Journal of Research and Development*, vol. 40, no. 2, pp. 211–229, 1996.
- [33] N. Gorski, V. Anisimov, E. Augustin, O. Baret, D. Price, and J. C. Simon, “A2iA Check Reader: a family of bank check recognition systems,” in *Proceedings of the 5th International Conference on Document Analysis and Recognition*, 1999.
- [34] K. Mohammad, S. Agaian, and F. Hudson, “Implementation of Digital Electronic Arithmetics and its application in image processing,” *Computers and Electrical Engineering*, vol. 36, no. 3, pp. 424–434, 2010.
- [35] G. Deng, L. W. Cahill, and G. R. Tobin, “Study of logarithmic image processing model and its application to image enhancement,” *IEEE Transactions on Image Processing*, vol. 4, no. 4, pp. 506–512, 1995.
- [36] S. S. Agaian, “Visual morphology,” in *Nonlinear Image Processing X*, vol. 3646 of *Proceedings of SPIE*, pp. 139–150, January 1999.
- [37] B. Wang, X. F. Li, F. Liu, and F. Q. Hu, “Color text image binarization based on binary texture analysis,” *Pattern Recognition Letters*, vol. 26, no. 10, pp. 1568–1576, 2005.
- [38] C. Thillou and B. Gosselin, “Color binarization for complex camera-based images,” in *Proceedings of the Electronic Imaging Conference of the International Society for Optical Imaging*, pp. 301–308, January 2005.
- [39] M. Unser, “Splines: a perfect fit for medical imaging,” in *International Symposium on Medical Imaging: Image Processing (MI' 02)*, Proceedings of the SPIE, pp. 225–236, San Diego, Calif, USA, February 2002.
- [40] [http://en.wikipedia.org/wiki/Spline_\(mathematics\)](http://en.wikipedia.org/wiki/Spline_(mathematics)).
- [41] <http://cnx.org/content/m11089/latest>.
- [42] M. Pechwitz and V. Maergner, “Baseline estimation for Arabic handwritten words,” in *Proceedings of the 8th International Workshop of Frontiers in Handwriting Recognition (IWFHR '02)*, August 2002.
- [43] N. Kilic, P. Gorgel, O. N. Ucan, and A. Kala, “Multifont Ottoman character recognition using Support Vector Machine,” in *Proceedings of the 3rd International Symposium on Communications, Control, and Signal Processing (ISCCSP '08)*, pp. 328–333, March 2008.
- [44] Y. J. Song, K. C. Kim, Y. W. Choi et al., “Text region extraction and text segmentation on camera-captured document style images,” in *Proceedings of the Eight International Conference on Document Analysis and Recognition*, Seoul, Korea, August 2005.
- [45] S. Sharma, *Extraction of Text Regions in Natural Images*, Rochester Institute of Technology, Rochester, NY, USA, 2007.
- [46] D. Chen, H. Bourlard, and J. P. Thiran, “Text identification in complex background using SVM,” in *Proceedings of the IEEE International Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. II621–II626, December 2001.
- [47] Z. Saidani, *Image and Video Text Recognition Using Convolutional Neural Networks [Ph.D. thesis]*, LAP Lambert Academic, Saarbrücken, Germany, 2008.
- [48] V. Ganapathy and L. W. L. Dennis, “Malaysian vehicle license plate localization and recognition system,” *Journal of Systems, Cybernetics and Informatics*, vol. 6, no. 1, 2008.
- [49] X. Li, W. Wang, Q. Huang, W. Gao, and L. Qing, “A hybrid text segmentation approach,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '09)*, pp. 510–513, July 2009.
- [50] Q. Ye, W. Gao, and Q. Huang, “Automatic text segmentation from complex background,” in *Proceedings of the International Conference on Image Processing (ICIP '04)*, pp. 2905–2908, October 2004.
- [51] J. Gllavata, E. Qeli, and B. Freisleben, “Detecting text in videos using fuzzy clustering ensembles,” in *Proceedings of the 8th IEEE International Symposium on Multimedia (ISM '06)*, pp. 283–290, December 2006.

