Separation of Text and Non-text in Document Layout Analysis using a Recursive Filter

Tuan-Anh Tran, In-Seop Na, Soo-Hyung Kim*
Department of Computer Science, Chonnam National University, Gwangju, 500-757, Korea
[e-mail: trtanh@hcmus.edu.vn, ypencil@hanmail.net, shkim@jnu.ac.kr]

*Corresponding author: Soo-Hyung Kim

Received April 14, 2015; revised June 30, 2015; accepted August 3, 2015; published October 31, 2015

Abstract

A separation of text and non-text elements plays an important role in document layout analysis. A number of approaches have been proposed but the quality of separation result is still limited due to the complex of the document layout. In this paper, we present an efficient method for the classification of text and non-text components in document image. It is the combination of whitespace analysis with multi-layer homogeneous regions which called recursive filter. Firstly, the input binary document is analyzed by connected components analysis and whitespace extraction. Secondly, a heuristic filter is applied to identify non-text components. After that, using statistical method, we implement the recursive filter on multi-layer homogeneous regions to identify all text and non-text elements of the binary image. Finally, all regions will be reshaped and remove noise to get the text document and non-text document. Experimental results on the ICDAR2009 page segmentation competition dataset and other datasets prove the effectiveness and superiority of proposed method.

Keywords: text detection, non-text identification, page segmentation, document layout analysis, OCR, recursive filter.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2015-018993) and this research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2015R1C1A1A02036495).

http://dx.doi.org/10.3837/tiis.2015.10.017 **ISSN : 1976-7277**

1. Introduction

In computer vision, document layout analysis is the process of identifying and categorizing the regions of interest in the scanned image of a text document. A reading system requires the segmentation of text zones from non-textual ones and the arrangement in their correct reading order [34]. We can see that the quality of the layout analysis task outcome can determine the quality and the feasibility of the next steps, and hence of the whole document processing activity.

In the field of document layout analysis, a number of approaches were proposed but the results them is still limited. Top-down method [7-10] look for global information on the entire document page and split it into blocks and then split the blocks into text lines and the text lines into words. These methods use different ways to separate the input document into distinguish regions and use many heuristic condition to identify non-text elements. This method proves the efficient when the document have Manhattan layout (Manhattan layout has region boundaries consisting of vertical and horizontal line segments). Bottom-up method [1-3], [5], [11], [13] start with local information and first determine the words, then merge the words into text lines, and merge the text lines into blocks or paragraphs. This method is more efficiency on non-Manhattan layout but it requires longer time in computation and space. Besides, some threshold of these method are complex and sometimes inaccurate. Hybrid method [19], [22] uses the combination of two methods above. However most of these methods are not really paying attention to the classification of text and non-text elements before grouping them. This leads to the result obtained without high precision.

As we can see, one of the main reasons of this limitation is due to these methods do not pay much attention to the classification of text and non-text components in the document, even thought the classification of text and non-text elements in the document layout plays a very important role. This lead to the wrong segmentation of text and miss classification of non-text elements.

Besides, the distribution of text and non-text elements in documents is very random and do not follow the rules, especially in non-Manhattan documents (non-Manhattan layout may include arbitrarily shaped document components and be in any arrangement). Therefore, the classification of text and non-text in the original document by bottom-up methods, or in the region obtained by top-down methods, even the homogeneous region also becomes complex and often provides unexpected results. These methods typically separate the original document into many different regions. Then use many filters to classify each region [5], [18], [20] (only one layer of homogeneous region is used). In addition to creating many filters, these methods only effective when the region is not too complicated. The winner of ICDAR 2009 page segmentation competition [21], the Fraunhofer Newspaper Segmenter also uses this approach, so that the precision of non-text identification is not satisfied.

On the other hand, there are some methods which uses the wavelet transform and multi-scale resolution [15], [16] to identify non-text elements. This method yielded positive results and the approach is very general; however, with this method, by using the multi scale resolution,

the computing time is quite long. In addition, when using wavelet methods difficulties rise in the case of documents with less information, or with structured overview together. Another difficulty when using wavelet transform is that it will create a lot of noise when the document size is changed. The most common mistake is the confusion of the classification of cases images that has structure similar to text, or small images with big text font size.

In [27] Chen and Wu proposed a multi-plane approach for text segmentation. This method prove an efficient in complex background document images. However, this method can not control the number of planes and uses many threshold so the computation time in some cases is quite long. Besides, this method only focus on text extraction without paying attention to the identification of non-text elements.

One of the important features of the document image is the large of information. In other words, there are many components inside the document. It will create favorable conditions for the algorithms that use statistical approach. In this paper, we propose an effective method to separate the input document to two distinguish binary documents, text document and non-text document. Our method use recursive filter on multi-layer of homogeneity to carry out the classification. Based on statistical approach, the proposed method provides the high efficiency of text and non-text classification. The motivation is as follows (**Fig. 1**).

Firstly, the colored input document is binarized by the combination of Sauvola technique and Intergral image. Then, the skew estimation [28], [31] is performed. This is an optional step for the skewn document. Secondly, based on the binary document f, we extract the connected component and get connected component properties. A heuristic filter is applied to identify some non-text elements, all these non-text elements are removed from the binary document by the label matrix to get the new image $\hat{f}$. Then, based on this image, the recursive filter is performed to identify all non-text components. This process is the combination of whitespace analysis with multi-layer homogeneous region. Non-text elements are eliminated layer-by-layer. All remain elements after recursive filter is the text elements. It will be reshaped by their coordinates to get the text-document. Finally, the non-text document can be obtained by the subtraction of original binary image f and the text document.

An overview of proposed method and its performance is given next. In Section 2, the method is described in detail. Section 3 presents the experimental results and the evaluation using ICDAR2009 page segmentation dataset [21] and six methods in this competition. Finally, the paper is concluded in Sections 4.

2. Proposed Method

The proposed method for text and non-text classification is described as follows: Given the colored image,

Step 1: Binarize document image by Sauvola algorithm with the efficient implementation using Integral image (Section 2.1).

Step 2: Connected component analysis and whitespace extraction (Section 2.2).

Step 3: Perform the heuristic filter (Section 2.3).

Step 4: Perform the recursive filter (Section 2.4).

Step 5: Reshape regions and post processing (Section 2.5).

2.1. Image Binarization

Like other image processing methods, our method also performs on the binary image, so we first need to convert the input colored document to binary document. Image binarization is the process that converts an input grayscale image $g(x, y)$ into a bi-level representation (if the input document is colored, its *RGB* components are combined to give a grayscale image). Generally, there are two approaches to binarize the grayscale image: the algorithms based on global threshold and algorithms based on local threshold. In our method, we refer to Sauvola technique [14] to obtain the local threshold at any pixel (x, y).

$$T(x, y) = m(x, y) \times \left[1 + k\left(1 - \frac{s(x, y)}{R}\right)\right] \quad (1)$$

where $m(x, y), s(x, y)$ are the local mean and standard deviation values in a $W \times W$ window centered on the pixel (x, y) respectively. R is the maximum value of the standard deviation and k is a parameter which takes positive values in the range $[0.2, 0.5]$.

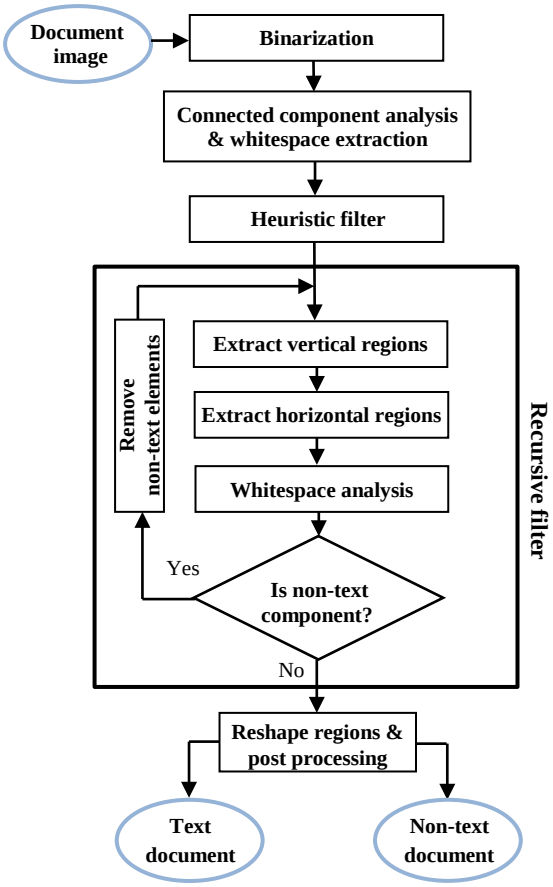


Fig. 1. Flowchart of Proposed System

In the field of document binarization, algorithms use local threshold often give better result than algorithms use global threshold. Nevertheless, they require a longer computation time.

For example, the computational complexity for an image in size $a \times b$ by Sauvola algorithm is $O(W^2ab)$. In order to reduce computation time of Sauvola algorithm; we use integral images for computing local means and variances. The integral image or summed area table, was first introduced to us in 1984 [24] but was not properly introduced to the world of computer vision till 2001 by Viola and Jones [6]. The value of integral image at location (x, y) is the sum of the pixels above and to the left of it. The value of integral image I at position (x, y) can be written as:

$$I(x,y) = \sum_{i=1}^{x}\sum_{j=1}^{y} g(x,y) \tag{2}$$

The integral image of any grayscale image can be efficiently computed in a single pass [6]. Then the local mean $m(x, y)$ and local standard deviation $s(x, y)$ for any window size W can be computed simply by formula [23]:

$$m(x,y) = \frac{1}{W^2}([I_{11} + I_{22}] - [I_{12} + I_{21}]) \tag{3}$$

$$\begin{aligned} s(x,y) &= \frac{1}{W}\sqrt{\sum_{i=x-W/2}^{x+W/2}\sum_{j=y-W/2}^{y+W/2} g^2(i,j) - m^2(x,y)} \\ &= \frac{1}{W}\sqrt{([I_{11}^2 + I_{22}^2] - [I_{12}^2 + I_{21}^2]) - m^2(x,y)} \end{aligned} \tag{4}$$

where

$$I_{11} = I\left(x + \frac{W}{2}, y + \frac{W}{2}\right), I_{12} = I\left(x + \frac{W}{2}, y - \frac{W}{2}\right), \tag{5}$$

$$I_{21} = I\left(x - \frac{W}{2}, y + \frac{W}{2}\right), I_{22} = I\left(x - \frac{W}{2}, y - \frac{W}{2}\right) \tag{6}$$

The computational complexity of binarize image now is only O(ab) and the computation time does not depend on the window size. In our algorithm, we choose window size $W = 1/2 \times \min(a, b)$. The parameter k controls the value of the threshold in the local window such that the higher the value of k, the lower threshold from the local mean $m(x, y)$. Experimented with different values and found that $k = 0.34$ gives the best results for the small window size [23], but the difference is small. In our system the window size is large so the efficient of parameter k is very small. In general, the algorithm is not very sensitive to the value of k. The experimental result shows that our system has the similar results with $k \in [0.2,0.5]$ and the best result with $k \in [0.2,0.34]$. We assign the pixels that belong to the foreground is a value of 1 and background pixels a value of 0. This means the binary image f (**Fig. 6(a), 6(d) and Fig. 7(a), 7(d)**) is calculated by:

$$f(x,y) = \begin{cases} 1, \text{if } g(x,y) > T(x,y) \\ 0, \text{elsewhere} \end{cases} \tag{7}$$

2.2. Connected component analysis and whitespace extraction

2.2.1. Connected component analysis

Connected components labelling is the process of extracting and labelling connected components from a binary image. All pixels that are connected and have the same value are extracted and assigned to a separate component. There are many algorithms that use this method, such as [5], [17-20]. Let L is the label matrix, CCs be all connected components and CC_i is the i_{th} connected component of f. Every CC_i is characterized by the following set of features:

- $B(CC_i)$ is the bounding box of CC_i with (Xl_i, Yl_i), (Xr_i, Yr_i) is the top-left and bottom-right coordinate, H_i and W_i is the height and width of $B(CC_i)$, see **Fig. 2.**
- $C_{size}(CC_i)$ is the number of pixel of CC_i.
- $B_{size}(CC_i)$ is the size of $B(CC_i)$, $B_{size}(CC_i) = W_i \times H_i$
- $C_{dens}(CC_i)$ is the ratio of $C_{size}(CC_i)$ and $B_{size}(CC_i)$,

$$C_{dens}(CC_i) = \frac{C_{size}(CC_i)}{B_{size}(CC_i)}, C_{dens} \in (0,1] \tag{8}$$

- $Ins(CC_i)$ is the number of $B(CC_j), i \neq j$ located inside the $B(CC_i)$. The $B(CC_j)$ is called inside the $B(CC_i)$, as in **Fig. 2** if

$$(Xl_i < Xl_j) \wedge (Yl_i < Yr_j) \wedge (Xr_i > Xr_j) \wedge (Yr_i > Yr_j) \tag{9}$$

- A_{HW} is the aspect ratio denotes the of width and height of CC_i, $A_{HW} \in (0,1]$

$$A_{HW} = \frac{\min(H_i, W_i)}{\max(H_i, W_i)} \tag{10}$$

- $H_{olap}(CC_i)$, $V_{olap}(CC_i)$ are the collection of connected components stay on the same column and same row with CC_i respectively [5].

$$H_{olap}(CC_i) = \{CC_j \in CCs \mid \max(Xl_i, Xl_j) - \min(Xr_i, Xr_j) < 0\} \tag{11}$$

$$V_{olap}(CC_i) = \{CC_j \in CCs \mid \max(Yl_i, Yl_j) - \min(Yr_i, Yr_j) < 0\} \tag{12}$$

2.2.2. Whitespace extraction

For every connected component CC_i in the region, we find the right nearest neighbor and left nearest neighbor. Then based on these neighbors we extract the length of whitespace (distance) between them. In our system, we use the technique was proposed in [19] to get the linear computation time. The $CC_j, j \neq i$ is called the right nearnest neighbor of CC_i if $V_{olap}(CC_i) \neq \emptyset, \mathrm{CC_j} \in V_{olap}(CC_i)$, CC_j does not locate inside CC_i, $Xl_j > Xr_i$ and

$$Xl_j - Xr_i = \min\{Xl_t - Xr_i > 0 | CC_t \in V_{olap}(CC_i)\} \tag{13}$$

where, $Xl_j - Xr_i$ is the whitespace (distance) between CC_i and CC_j. Besides, if CC_j is the right nearest neighbor of CC_i, then CC_i is the left nearest neighbor of CC_j.

2.3. Heuristic filter

In this filter, we find the elements that cannot be the texts without attention to its relative position in the document image. Clearly, these conditions must be precise and very stringent, because they have a strong influence in whether we are looking at a separate region or not. This filter not only reduces the computational time of whole process but also increases the accuracy of the proposed system.

Let CCs be all the connected components of the input binary document f, CC_i is the i_{th} connected component and $B(CC_i)$ is the bounding box of it. In order to improve our performance, the CC_i is considered as a non-text component if it satisfies one of following conditions:

- $C_{size}(CC_i)$ is too low (area of CC_i less than 6 pixels). With very small size, it is very difficult for the OCR system to be able to identify, or even for human eyes.
- $Ins(CC_i) > 3, B(CC_i)$ contains many (greater than 3) other bounding boxes. This is correct not only for English language but also for the Korean or Chinese language.
- $C_{dens}(CC_i)$ is too low (less than 5%). When the density of CC_i is too low, it can be a diagonal or noise element (the normal density of text element is greater than 20%)
- $A_{HW}(CC_i)$ <6%, the ratio between the height and the width is not too low or too high, even though it is the letter "I".

These thresholds in the above conditions are carefully calculated and checked many times with many a variety of document types. It also is considered carefully in the case of binary images with noise. Call CCs' is the set of non-text elements that were found by conditions above, $\forall CC_i \in CCs'$

$$\hat{f}(x,y) = \begin{cases} 0, \text{if } f(x,y) \in CC_i \\ f(x,y), \text{elsewhere} \end{cases} \tag{14}$$

Then, $CCs = CCs \backslash CCs'$

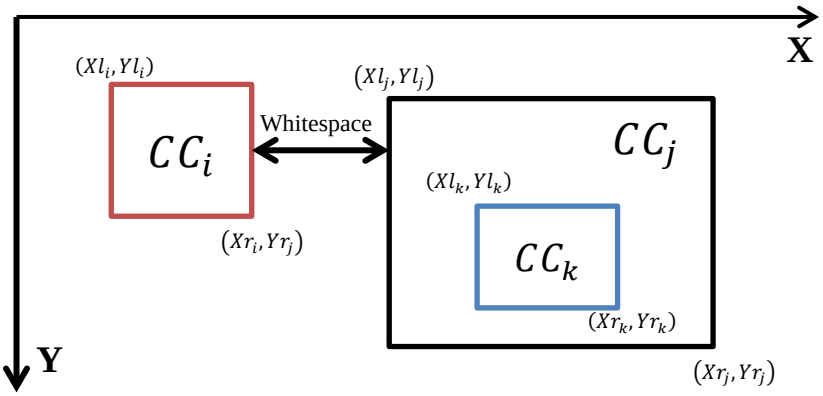


Fig. 2. Example of connected conponent analysis and whitespace extraction.

2.4. Recursive filter

This is the most important process of our method. After heuristic filter step, some non-text elements are identified and eliminated. However, there still exists many non-text elements in the document and these components are often not too different than text elements. In this

section, based on the statistical method we proposed an efficient filter to identify non-text elements in document, see **Fig. 1**. Let $\hat{f}$ is the binary document and CCs be all connected components obtained after heuristic filter. This is an iterative method with three main steps:

Firstly, we extract the homogeneous regions of $\hat{f}$. To do this, vertical homogeneous regions are extracted by vertical projection and then each vertical region are segmented horizontal again to get the homogeneous regions HR^k ($k = \overline{1,m}$, m is the number of region).

Seconly, the whitespace analysis process is performed to identify the non-text components and its label in all homogeneous regions HR^k. Call $CCs' \subset CCs$ is the set of them. Once again, these components are removed by label matrix to deduce the new text binary document $\hat{f}^*$.

Repeat two steps above until we cannot find any non-text component or $CCs' = \emptyset$. At this time, all regions HR^k are text homogeneous region HR^{k^*} and will be reshaped by their coordinate to get the text document $\hat{f}^*_{text}$ and the non-text document is the subtraction of f and $\hat{f}^*_{text}$.

2.4.1. Homogeneous regions extraction

Document is usually divided into various regions, including the text regions (or paragraphs), image regions, lines, etc. Moreover, in the paragraph, the text horizontally or vertically is often homogeneous and white spaces between them are almost the same. Based on these properties, we will segment the input document to many different regions (we call homogeneous regions). In this section, we present a method to segment the input document into many homogeneous regions. There are two kinds of homogeneity, horizontal homogeneity and vertical homogeneity. The different between them is the direction in which we get the projection.

Suppose $\hat{f}$ is the considering binary image with $a \times b$ as the size of it. In order to get the horizontal homogeneous region, we perform the following steps:

Step 1: Find histogram of horizontal projection P,

$$P = \left\{ p_x \middle| p_x = \sum_{y=1}^{b} \hat{f}(x,y), 1 \le x \le a \right\} \tag{15}$$

Step 2: Convert the value of projection to bi-level value LP,

$$LP = \left\{ lp_x \middle| \begin{cases} lp_x = -1 & \text{if } p_x > 0 \\ lp_x = \ \ 0 & \text{if } p_x = 0 \end{cases}, p_x \in P \right\} \tag{16}$$

Step 3: Extract the large of white line and black line.

Firstly, we use Run Length Encoding (abbreviate to *RLE*) to calculate the run length of all elements in *LP*. *RLE* is a data compressing method in which a sequence of the same data can be presented by a single data value and count. It is especially useful in the case of binary image data with only two values: zero and one (in our case, we use -1 instead of 1). For example, the result of *RLE* of sequence $\{-1,-1,-1,0,0,-1,-1,0,0,0,-1,-1,-1,-1\}$

is $RLE = \{-1,3,0,2,-1,2,0,3,-1,4\}$. Let b_i and w_i are the large of i_{th} black line and white line. Let

$$B = \{b_i \in RLE | b_i > 0 \wedge b_{i-1} < 0\} \tag{17}$$

$$W = \{w_i \in RLE | w_i > 0 \wedge w_{i-1} = 0\} \tag{18}$$

Step 4: Estimate the homogeneity of the input region.

Suppose μ_b, μ_W is the mean and n_B, n_W is the number of *B* and *W* respectively. The variance of black line and white line is as follow,

$$V_B = \frac{\sum(b_i - \mu_B)^2}{n_B} \tag{19}$$

$$V_W = \frac{\sum(w_i - \mu_W)^2}{n_W} \tag{20}$$

If the value of V_B and V_W is low, it means that the region is homogeneous. Conversely, if the variance of V_B or V_W is high, our region is heterogeneous and it should be segmented. In our method within skew consideration, we choose the threshold value for V_B and V_W are 1.3. This mean, if $V_W > 1.3$ or $V_B > 1.3$ the region should be segmented. Actually, this threshold is still depends on the type of the language (Korean or English) because the several of length and height in each letter of each language is also different. However, the impact of this threshold is not much, the experimental results show that we can fix the threshold for both language.

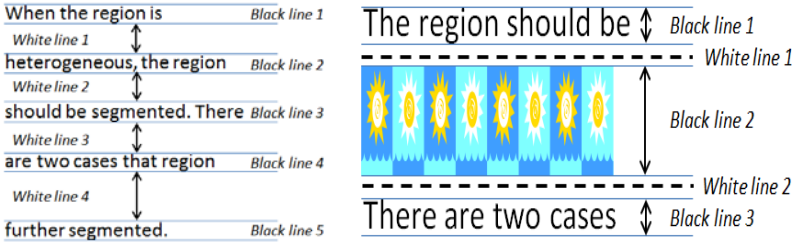
Step 5: Segment region.

When the region is heterogeneous, the region should be segmented. There are two cases that need to be further divided. First, in the region under consideration, there exists a white line with its width larger than other white line, **Fig. 3(a)**. Second, there exists a black line with its width is larger than another, **Fig. 3(b).**

In order to get the position to split, we need to find the most distinctive space of black or white. The splitting position is described as follow:

$$\begin{cases} \text{If } w_i > median(W) \wedge w_i = \max(W), \text{split } w_i \\ \text{If } b_i > median(B) \wedge b_i = \max(B), \text{split } w_i \text{ and } w_{i-1} \end{cases} \tag{21}$$

Repeat the steps above until the entire region obtained becomes homogeneous. In the same way, we can perform all steps on vertical and get the homogeneous region on this direction.



(a) (b)

Fig. 3. Example of: (a) white segment- *white line 4*, (b) black segment- *white line 1 and 2*

2.4.2. Whitespace analysis

As mentioned above, in the recursive filter we focus more carefully on the structure of the text and examine the relationship between the CC_i. The structure of the text is usually in rows or columns, so we can use two important properties in statistics, the median and

variance to classify them. The dispersion of the text in a homogeneous region is relatively low. Therefore, by using two important properties in statistics, me dian and variance, we can identify the nontext elements in each homogeneous horizontal region. Suppose HR^k is the region being considered, $CC^k \subset CCs$ is the set of connected components of HR^k. Put

$$\mathcal{L}_1 \equiv CC_{size}^k = \left\{CC_{size}\left(CC_j\right) \middle| CC_j \in CC^k\right\} \tag{22}$$

$$\mathcal{L}_2 \equiv H^k = \left\{H_j \middle| CC_j \in CC^k\right\} \tag{23}$$

$$\mathcal{L}_3 \equiv W^k = \left\{W_j \middle| CC_j \in CC^k\right\} \tag{24}$$

where $CC_{size}\left(CC_j\right), H_j, W_j$ is the area, height and width of CC_j respectively. In order to classify the text and non-text elements in a horizontal homogeneous region we can follow these steps.

Step 1: Find the non-text candidates.

The $CC_i \in CC^k$ is the non-text candidate if it satisfies

$$\begin{cases} C_{size}(CC_i) = \max\{\mathcal{L}_1\} \\ C_{size}(CC_i) > t_1 \times median\{\mathcal{L}_1\} \end{cases} \tag{25}$$

And one of two following conditions:

$$(H_i = \max\{\mathcal{L}_2\}) \wedge (H_i > t_2 \times median\{\mathcal{L}_2\}) \tag{26}$$

$$(W_i = \max\{\mathcal{L}_3\}) \wedge (W_i > t_3 \times median\{\mathcal{L}_3\}) \tag{27}$$

where $t_j, j = 1,2,3$ is the threshold fo the difference betwwen the size, height, width of CC_i and the median of them. The value of t_j can be chosen as,

$$t_j = \max\left\{\frac{median\{\mathcal{L}_j\}}{mean\{\mathcal{L}_j\}}, \frac{mean\{\mathcal{L}_j\}}{median\{\mathcal{L}_j\}}\right\};\ j = 1,2,3 \tag{28}$$

In each homogeneous region, the different between components is always small. In the other word, the variance of them is usually small if the region includes only text elements. The equation (28) is calculated t_j by considering the relationship of median and mean. Meanwhile, in statistic if the distribution has finite variance, then the distance between the median and the mean is bounded by one standard deviation [29].

Step 2: Classify non-text candidates.

Suppose CC_i is the non-text candidate that was found in step 1. Using the whitespace rectangle extraction (13), we find the left nearest neighbors $LNN(CC_i)$, right nearest neighbors $RNN(CC_i)$, the distance between CC_i and its left nearest neighbor $l_i = LNWS(CC_i)$, its right nearest neighbor $r_i = RNWS(CC_i)$.
The CC_i will be classified as non-text element if

$$\begin{cases} \min(l_i, r_i) > \max(medianWS, meanWS) \\ (\max(l_i, r_i) = maxWS) \vee (\min(l_i, r_i) > 2 \times meanWS) \end{cases} \tag{29}$$

Or

$$\max\left(num_{LN(CC_i)}, num_{RN(CC_i)}\right) \geq 3 \tag{30}$$

where $medianWS$, $meanWS$, $maxWS$ is the median, mean, max of whitespace in the considering region. $num_{LN(CC_i)}, num_{RN(CC_i)}$ is the total number of left nearest neighbor and

right nearest neighbor on the each row of CC_i. Experimentation showed that the use of the median generated desirable results, especially in the case that there are many connected components in considering region. In case our region has little information, we can further analyze the min-max and the variance of elements in the region.

2.4.3. Non-text elements removal

Call $CCs' \subset CCs, CCs' \neq \emptyset$ is the set of non-text elements in all regions $HR^k, k = \overline{1,m}$ that were found by whitespace analysis stage. Apply (14), we remove non-text elements and get new text document. $\forall CC_i \in CCs'$

$$\hat{f}^*(x,y) = \begin{cases} 0, \text{if } \hat{f}(x,y) \in CC_i \\ \hat{f}(x,y), \text{elsewhere} \end{cases} \tag{31}$$

and $CCs = CCs \backslash CCs'$.

2.5. Reshape regions and post processing

2.5.1. Reshape regions

When the whitespace analysis process cannot find any non-text component in all homogeneous regions or $CCs' = \emptyset$. This mean $HR^k, k = \overline{1,m}$ now contains only text elements and we call these regions are the text homogeneous regions $\left(HR^{k^*}\right)$. Reshape all regions by their coordinates to get the text document

$$\hat{f}^*_{text} = HR^{1^*} \cup HR^{2^*} \cup \ldots \cup HR^{m^*} \tag{32}$$

Then, the non-text document can be obtained by the logical "and" of the original binary document with text document.

$$\hat{f}^*_{ntext} = \begin{cases} 1, \text{if } \hat{f}^*_{text}(x,y) = 0 \wedge f(x,y) = 1 \\ 0, \text{elsewhere} \end{cases} \tag{33}$$

2.5.2. Post processing

Binarize images always contain a lot of noise especially when the original document has many figures and these figures have a big size. Besides, binary image often has many missing components or unexpected components (all components are stick together), especially for the low resolution document or nonstandard document.

To reduce the noise, firstly, we apply a morphological with a small kernel for non-text document $\hat{f}^*_{ntext}$. Secondly, all holes inside this image will be filled to remove the noise. On the other hand, we extract the bounding box of all connected components $B(CC_i)$ in text document. Let CCs^{text} be the set of all connected components CC_i and $\hat{f}^B_{text}$ is the bounding box image of text document,

$$\hat{f}^B_{text} = \begin{cases} 1, \text{if } \hat{f}^*_{text}(x,y) \in B(CC_i), \forall CC_i \in CCs^{text} \\ 0, \text{elsewhere} \end{cases} \tag{34}$$

Let CCs^{ntext} be the set contains all connected component CC_j of non-text document. $\forall (x,y) \in a \times b$, if $\left(\hat{f}^B_{text} \in CC_i, \hat{f}^*_{ntext} \in CC_j\right) \wedge \left(\hat{f}^B_{text} = \hat{f}^*_{ntext} = 1\right)$

$$\begin{cases} CCs^{text} = CCs^{text} \setminus \{CC_i\} \\ CCs^{ntext} = CCs^{ntext} \cup \{CC_i\} \end{cases} \quad (35)$$

Then, the final output of non-text document $\hat{f}^{**}_{ntext}$ and the final output of text document $\hat{f}^{**}_{text}$ can be calculated by

$$\hat{f}^{**}_{ntext}(x,y) = \begin{cases} 1, \text{if } \hat{f}^{*}_{ntext}(x,y) \in \mathrm{CC_i}, \forall \mathrm{CC_i} \in \mathrm{CCs}^{ntext} \\ 0, \text{elsewhere} \end{cases} \quad (36)$$

$$\hat{f}^{**}_{text}(x,y) = \begin{cases} 1, \text{if } \hat{f}^{*}_{text}(x,y) \in \mathrm{CC_i}, \forall \mathrm{CC_i} \in \mathrm{CCs}^{text} \\ 0, \text{elsewhere} \end{cases} \quad (37)$$

3. EXPERIMENTAL RESULT

3.1. System Environment and Database

Our system was implemented in MATLAB R2014a on our workspace with a system configured with an Intel® Core™ i5-3470, 4G RAM, Windows 7 – 64 bits. Our database is generated from 114 document images with various layouts and document languages, as in **Fig. 4**. In there, 55 English documents of ICDAR2009 page segmentation competition dataset [21] and 59 Korean documents of the dataset of Diotek company [35].

3.2. Method for Performance Evaluation

The performance of our method is evaluated based on two measures: the pixel percentage of foreground from the ground-truth to be reserved (*Recall*), and the pixel percentage of remaining pixel after the process to belong to the foreground (*Precision*). The balance F-score (*F-measure*) shows the harmonic mean of these two measures,

$$Recall = \frac{Output \cap GroundTruth}{GroundTruth} \quad (38)$$

$$Precision = \frac{Output \cap GroundTruth}{Output} \quad (39)$$

$$Fmeasure = 2\frac{Precision \times Recall}{Precision + Recall} \quad (40)$$

3.3. Experimental Results and Evaluation Profile

Our method has been experimented on many different data sets and the results obtained are very encouraging. It is not dependent on the type of language and still gives good results when the input document has skew less than 5 degrees; see **Fig. 6, Fig. 7**. We can also apply a skew detection algorithm [28] after binarization step to estimate the skew of document and return the zeros horizontal angle of document before our algorithm is performed.

For the evaluation, firstly, we use the text documents and non-text documents (**Fig. 4**) datasets that are extracted from the ground truth of our database. The success rates of our method with two these datasets are shown in **Table 1, Table 2.** Secondly, **Table 3** and **Table 4** show the success rate of text region and non-text region (**Fig. 4**) that are given by the ground truth of our database. Before this evaluation process, our text documents and non-text documents result are smoothed by mathematical morphology. Firstly, in text document, we extract the homogeneous text regions (Section 2.4.1). Then, the kernel of morphology in each region is calculated by the mean of the height, the width of all elements in this area. On

the other hand, in non-text document, all elements are filled and smoothed by a small kernel (this kernel is depended on the size of each non-text element). **Table 5** shows the success rates of our method with overall foreground regions in the document.

Table 1. The success rates of text detection

Text extraction dataset	Precision (%)	Recall (%)	F-measure (%)
ICDAR2009 (English)	97.12	96.37	96.66
Diotek (Korean)	92.87	90.25	91.33

Table 2. The success rates of non-text detection

Non-text extraction dataset	Precision (%)	Recall (%)	F-measure (%)
ICDAR2009 (English)	92.26	91.20	91.12
Diotek (Korean)	85.73	83.25	84.06

Table 3. The success rates of text region

Text region dataset	Precision (%)	Recall (%)	F-measure (%)
ICDAR2009 (English)	93.18	94.46	93.80
Diotek (Korean)	88.54	89.88	89.16

Table 4. The success rates of non-text region

Non-text region dataset	Precision (%)	Recall (%)	F-measure (%)
ICDAR2009 (English)	83.22	85.28	84.19
Diotek (Korean)	82.83	81.25	82.01

Table 5. The success rates of overall region

Overall region dataset	Precision (%)	Recall (%)	F-measure (%)
ICDAR2009 (English)	92.67	93.82	93.22
Diotek (Korean)	87.39	88.63	88.01

According to the experimental results, the success rate of Korean document does not give results as expected. This cause by the Korean dataset contains many noise documents so the outcome of binary image is really bad in some cases. Besides, the ground-truth of this dataset is always try to return the rectangular shape (segmentation process) for each region, as in **Fig. 4(g)** whereas our algorithm only provide an efficient method for the classification of text and non-text. This also reduces the success rate when we evaluate our result especially with text evaluation.

The comparison of document classification is always complicated because they depend heavily on dataset and ground-truth. There are many different dataset and evaluation process. In our paper, we use dataset of ICDAR2009 page segmentation competition [21] for evaluation because this dataset had been published and very popular in our field. Even though this dataset is published in 2009, but it stills the basic for evaluating document analysis algorithms. Due to the complexity of the layout of the document in this dataset, not many algorithms are evaluated on this dataset. Up to now, the results of competitor's algorithms on this dataset are known as the best performances.

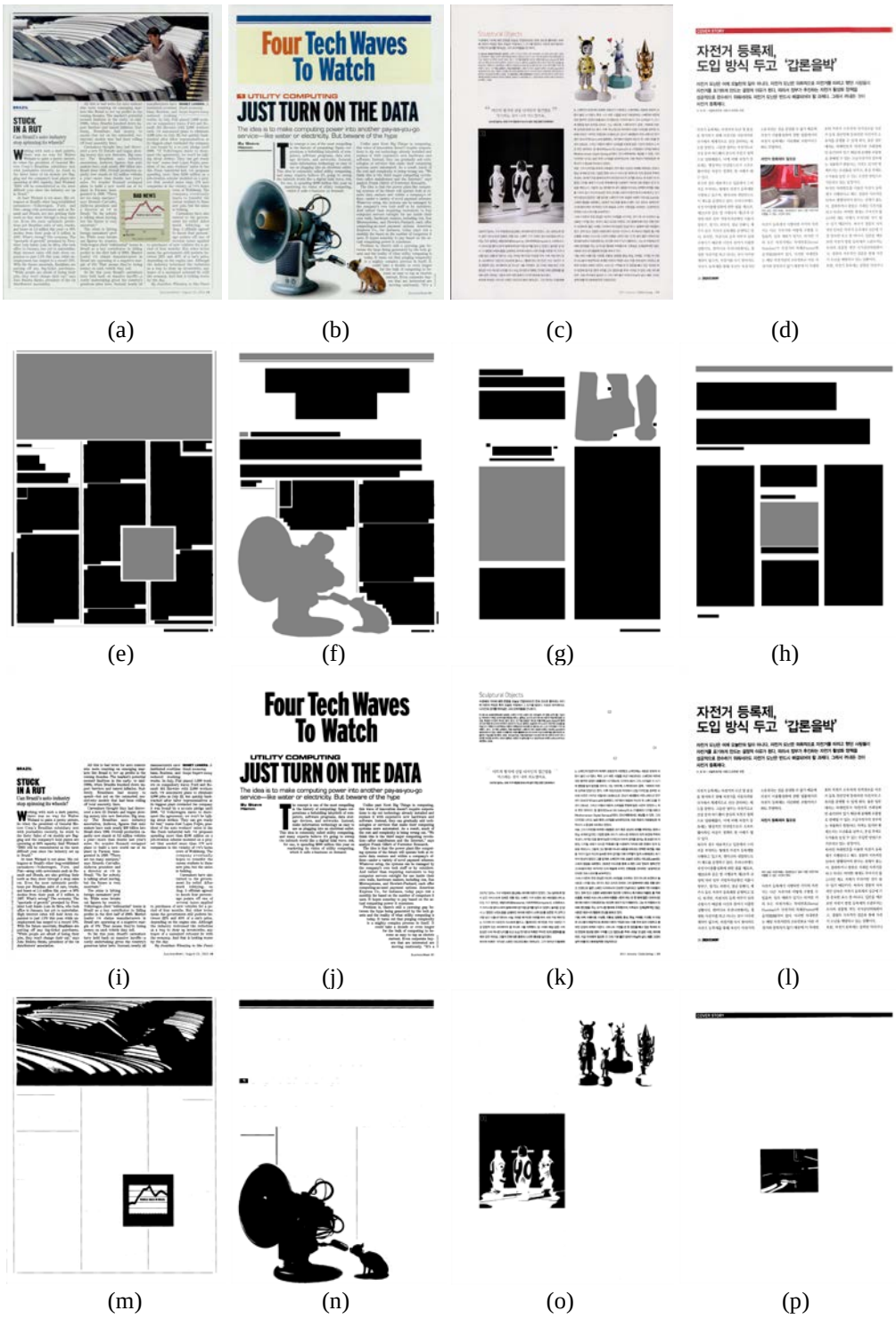
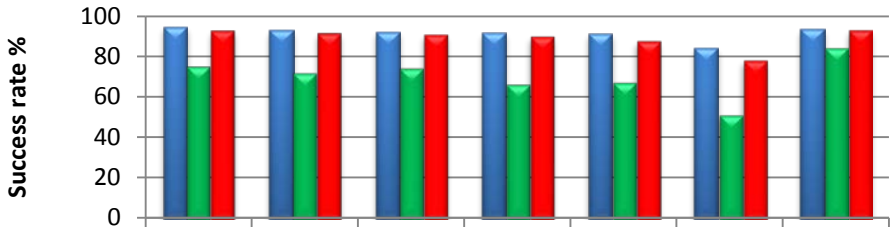


Fig. 4. Example of our database: (a,b,c,d) Colored document image; (e,f,g,h) ground truth of each region type – black: text, gray: non-text; (i,j,k,l) text elements extract from text region; (m,n,o,p) non-text elements extract from non-text region

In 2013, Chen et al. [18] also evaluated his method on this dataset but his result stills low and it cannot compare to the best performance of ICDAR2009 page segmentation competition.

We chose four algorithms and two method-of-arts get the highest results in [21] to compare with our algorithm. The F-measure evaluation profile is given in **Fig. 5.** Note that, all methods in ICDAR2009 page segmentation is the full system of document layout analysis.

After performing many evaluation process, we found that our method is very promising. The use of connected components analysis combine with multilevel homogeneity structure is an effective method to classify text and non-text elements. The success rate of non-text detection and non-text region is the highest. This demonstrates that our method can identify and classify most of the non-text elements in the document. Meanwhile, although we do not pay much attention to the page segmentation (just use a simple mathematical morphology), but the result of our method for text regions is very encourage. This will also create favorable conditions for us when we perform document layout analysis.

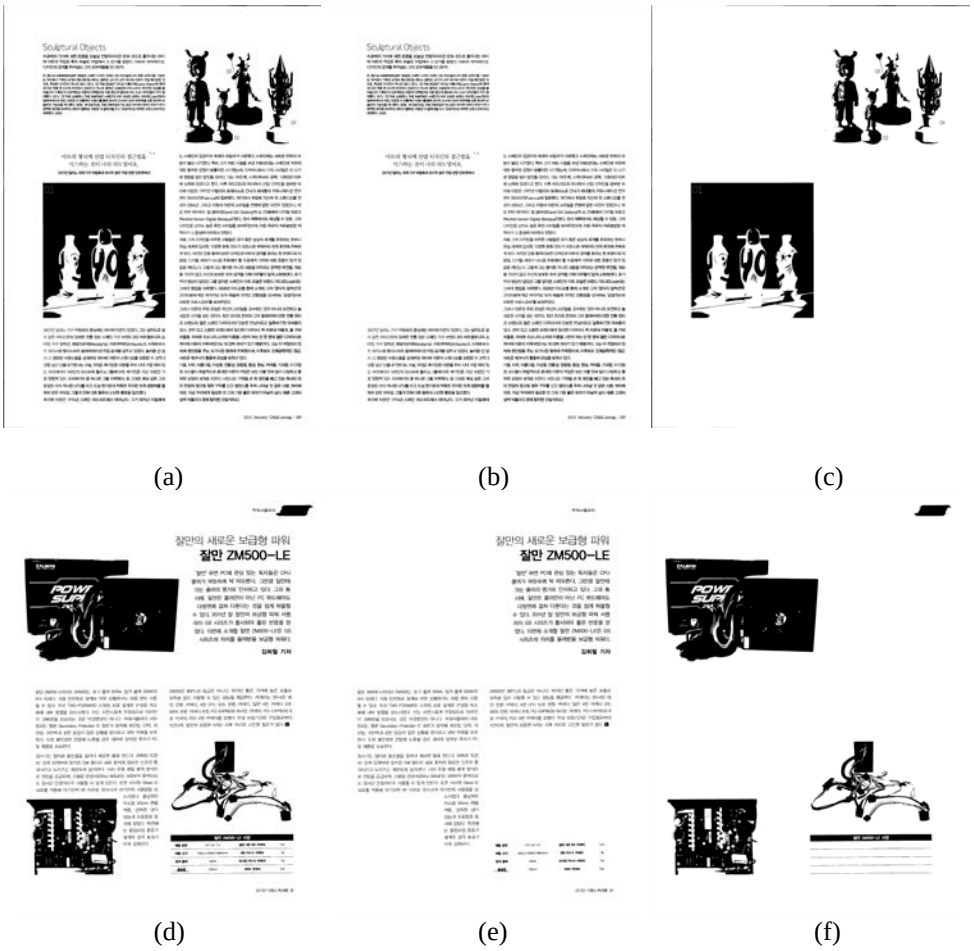


	Fraun-hofer	Fine-reader	Tess-eract	DICE	REGIM-ENIS	OCR-opus	Our method
Text	95.04	93.09	92.5	92.21	91.73	84.18	93.8
Non-text	75.15	71.75	74.23	66.22	67.13	51.08	84.19
Overall	93.14	91.9	91.04	90.09	87.82	78.35	93.22

Fig. 5. F-measure comparision of our method with four algorithm and two state-of-art methods have the highest performance.

4. Conclusion

In this paper, we proposed an efficient text and non-text classification method based on the combination of whitespace analysis and multi-layer homogeneous regions. In our approach, we first label the connected component and get their properties. Then, a heuristic filter is performed to identify strictly non-text elements in the binary document. The third stage, all elements in each homogeneous region are classified carefully in the the recursive filter. Not only one, but multi-layer of homogeneous regions are used to identify the non-text component. Therefore, this filter demonstrate the effectiveness in classifying text and non-text components. The final result is obtained after that, all text homogeneous regions are reshaped by their coordinates to get the text document. A simple post processing step which uses mathematical morphology will help us to remove noise and increase the performance of proposed method.



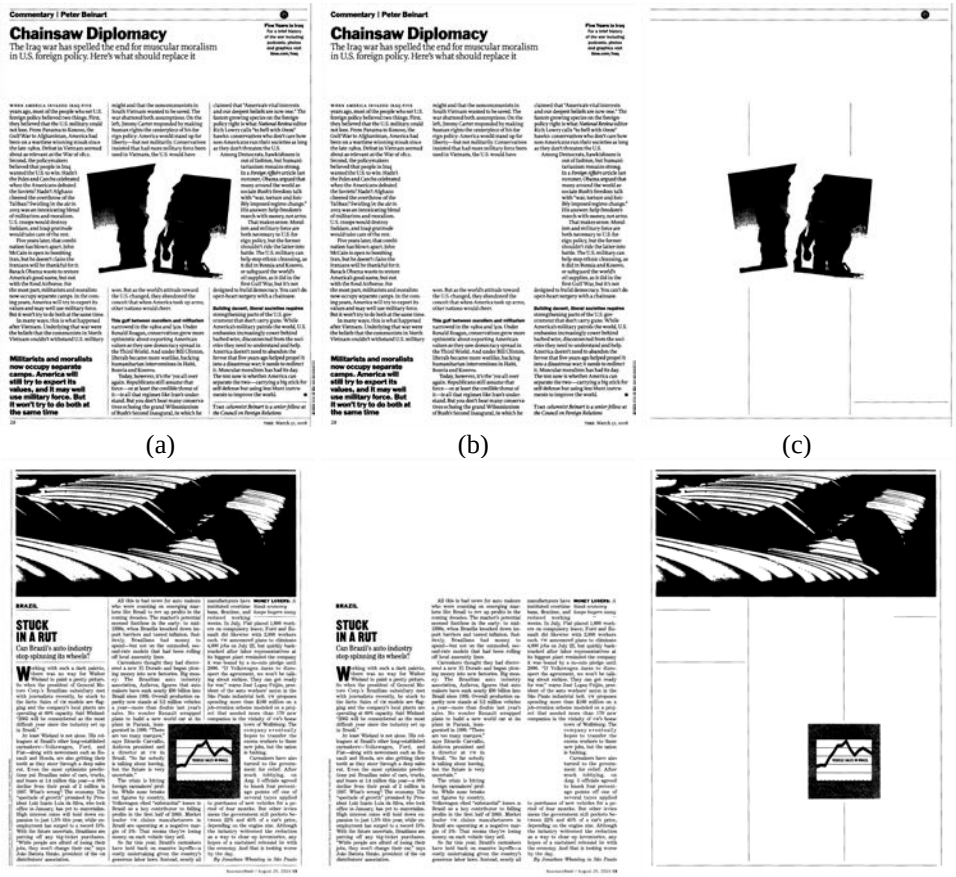
(a) (b) (c)

(d) (e) (f)

Fig. 6. Example of proposed method with Korean document: (a,d) Input binary document, (b,e) Text document, (c,f) Non-text document.

Our algorithm is not too complicated and easier to improve. Besides, our algorithm does not depend on the size of document, this means it can run on any resolution of image. Therefore, with the big size document image, we can reduce the resolution before implement our algorithm to improve the time consuming. The experimental results show that our algorithm get a higher precision with the image has the resolution greater than 1 Megapixel. Like most of document layout algorithm, our method is sensitive with the skew document because the top-down approach is used in our system. However, skew document is the other interest field. For example, in further process of document layout, the reading system OCR is also requires a non-skew document. Besides, all published and well-known dataset of document image is non-skew, and our goal is focus on the complex of document layout.

Experimental results on ICDAR2009 and other databases gave high performance and very encouraging. The proposed method not only has good results on English dataset but also on the Korean dataset (many other algorithms cannot do that). In future work, we are going to implement the page segmentation. Text document will be separate into corresponding text regions, all elements in non-text document will be classified in more detail (i.e. figure regions, table regions, separator regions, etc.).



(a) (b) (c)

(d) (e) (f)

Fig. 7. Example of proposed method with English document: (a,d) Input binary document, (b,e) Text document, (c,f) Non-text document.

References

[1] K. Kise, A. Sato, M. Iwata, "Segmentation of page images using the area Voronoi diagram," *Computer Vision Image Understanding*, vol. 70, no.3, pp. 370–382, 1998. Article (CrossRef Link)

[2] Agrawal, M., Doermann, D., "Voronoi++: A dynamic page segmentation approach based on Voronoi and Docstrum features," in *Proc. of 10th ICDAR*, pp. 1011-1015, 2009. Article (CrossRef Link)

[3] L. O'Gorman, "The document spectrum for page layout analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1162-1173, 1993. Article (CrossRef Link)

[4] B. Gatos, N. Papamarkos and C. Chamzas, "Skew Detection and Text Line Position Determination in Digitized Documents," *Pattern Recognition*, vol. 30, no. 9, pp. 1505-1519, 1997. Article (CrossRef Link)

[5] A. Simon, J. C. Pret, A. P. Johnson, "A Fast Algorithm for Bottom-Up Document," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no.3, pp. 273-277, 1997. Article (CrossRef Link)

[6] P. Viola, and M. J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features,"

in *Proc. of 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 511-518, 2001. Article (CrossRef Link)
[7] G. Nagy, S. Seth and M. Viswanathan, "A prototype document image analysis system for technical journals," *Computer*, vol.25, no. 7, pp. 10-22, 1992. Article (CrossRef Link)
[8] H. Baird, S. Jones and S. Fortune, "Image segmentation by shape-directed covers," in *Proc. of 10th International Conference on Pattern Recognition*, pp. 820–825, 1990. Article (CrossRef Link)
[9] F. M. Wahl, K. Y. Wong and R. G. Casey, "Block segmentation and text extraction in mixed text/image documents," *Graphical Models and Image Processing*, vol.12, no.4, pp. 375-390, 1982. Article (CrossRef Link)
[10] G. Nagy, S. C. Seth and S. D. Stoddard, "Document Analysis with an Expert System," *Pattern Recognition in Practice*, vol. II, pp. 149-159, 1986. Article (CrossRef Link)
[11] S. Ferilli, T. M. A. Basile and F. Esposito, "A histogram based technique for automatic threshold assessment in a run length smoothing-based algorithm," *The ninth IAPR International workshop on document analysis system*, 2010. Article (CrossRef Link)
[12] C. Clausner, S. Pletschacher, A. Antonacopoulos, "Scenario driven in-depth performance evaluation of document layout analysis methods," in *Proc. of 11th ICDAR*, pp. 1404-1408, 2011. Article (CrossRef Link)
[13] H. M. Sun, "Page segmentation for Manhattan and non-Manhattan layout documents via selective CRLA," in *Proc. of 8th ICDAR*, pp. 116–120, 2005. Article (CrossRef Link)
[14] J. Sauvola and M. Pietikainen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, no. 2, pp.225-236, 2000. Article (CrossRef Link)
[15] S.-W. Lee and D.-S. Ryu, "Parameter - Free Geometric Document Layout Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1240-1256, 2001. Article (CrossRef Link)
[16] H. Cheng and C. A. Bouman, "Multi-scale Bayesian Segmentation Using a Trainable Context Model," *IEEE Transactions on Image Processing*, vol. 10, no. 4, pp. 511-525, 2001. Article (CrossRef Link)
[17] J. Ha, R. M. Haralick and I. T. Phillips, "Recursive X-Y Cut Using Bounding Boxes of Connected Components," in *Proc. of 3rd ICDAR*, pp. 952-955, 1995. Article (CrossRef Link)
[18] J. Liang, J. Ha, R. M. Haralick and I. T. Phillips, "Document Layout Structure Extraction Using Bounding Boxes of Different Entities," in *Proc. of 3rd IEEE Workshop on Applications of Computer Vision*, pp. 278-283, 1996. Article (CrossRef Link)
[19] K. Chen, F. Yin and C.-L. Liu, "Hybrid Page Segmentation with Efficient Whitespace Rectangles Extraction and Grouping," in *Proc. of 12th ICDAR*, pp. 958-962, 2013. Article (CrossRef Link)
[20] Y. Pan, Q. Zhao and S. Kamata, "Document Layout Analysis and Reading Order Determination for a Reading Robot," in *Tencon2010-2010 IEEE Region 10 Conference*, pp. 1607-1612, 2010. Article (CrossRef Link)
[21] A. Antonacopoulos, S. Pletschacher, D. Bridson and C. Papadopoulos, "ICDAR2009 page segmentation competition," in *Proc. of 10th ICDAR*, pp. 1370-1374, 2009. Article (CrossRef Link)
[22] R. Smith, "Hybrid Page Layout Analysis via Tab-Stop Detection," in *Proc. of 10th ICDAR*, pp. 241-245, 2009. Article (CrossRef Link)
[23] F. Shafait, D. Keysers and T. M. Breuel, "Efficient implementation of local adaptive thresholding techniques using integral images," *Document Recognition and Retrieval XV*, 2008. Article (CrossRef Link)
[24] F. C. Crow, "Summed-area texture mapping," in *Proc. of 11th Annual conference on computer graphics and iterative techniques*, pp. 207-212, 1984. Article (CrossRef Link)
[25] J. S. Lim, I. S. Na, and S. H. Kim, "Correction of Signboard Distortion by Vertical Stroke Estimation," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 9, pp. 2312-2325, 2013. Article (CrossRef Link)
[26] D. Phan, I. S. Na, S. H. Kim, G.-S. Lee and H.-J. Yang, "Triangulation Based Skeletonization and Trajectory Recovery for Handwritten Character Patterns," *KSII Trans. Internet and Information Systems*, Vol. 9, No. 1, pp. 358-377, Jan. 2015. Article (CrossRef Link)
[27] Y.-L. Chen, B.-F. Wu, "A multi-plane approach for text segmentation of complex document

images," *Pattern Recognition*, vol. 42, pp.1419-1444, 2009. Article (CrossRef Link)
[28] B. Gatos, N. Papamarkos, and C. Chamzas, "Skew Detection and Text Line Position Determination in Digitized Documents," *Pattern Recognition*, vol. 30, pp. 1505-1519, 1997. Article (CrossRef Link)
[29] C. Mallows: "Another comment on O'Cinneide," *American Statistician*, vol. 45, no.3, pp. 256-262, 1991.
[30] F. Shafait, D. Keysers and T. M. Breuel, "Performance Evaluation and Benchmarking of Six - Page Segmentation Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 941-954, 2008. Article (CrossRef Link)
[31] A. Papamdreou, B. Gatos, "A Novel Skew Detection Technique Based on Vertical Projections," in *Proc. of 11th ICDAR*, pp. 384–388, 2011. Article (CrossRef Link)
[32] T. A. Tran, I. S. Na, S. H. Kim, "Text and Non-text Classification in Page Segmentation using a Recursive Filter," *7th International Conference on Computer Research and Development*, CDPub, 2015.
[33] M. H. Lee, S. H. Kim, G. S. Lee, Sun-Hee Kim, H. J. Yang, "Correction for Misrecognition of Korean Texts in Signboard Images using Improved Levenshtein Metric," *KSII Transactions on Internet and Information Systems*, Vol. 6, No. 2, pp. 722-733, Feb. 2012. Article (CrossRef Link)
[34] H. S. Baird, "Anatomy of a Versatile Page Reader," in Proc. of the IEEE, Vol. 80, no. 7, pp. 1059-1065, 1992. Article (CrossRef Link)
[35] http://www.diotek.com/

Tuan Anh Tran received his B.S. degree in Mathematics and Computer Science, University of Science, Ho Chi Minh city, Viet Nam, in 2010 and the M.S. degree in Apply Mathematic in MAPMO, University of Orleans, France, in 2011. He is currently researching as a Ph.D. student at Electronics and Computer Engineering, Chonnam National University, Korea. His research interests include document layout analysis, pattern recognition, machine learning, and mathematics application**.**

In Seop Na received his B.S., M.S. and Ph.D.egree in Computer Science from Chonnam National University, Korea in 1997, 1999 and 2008, respectively. Since 2012, he has been a research professor in the Department of Computer Science, Chonnam National University, Korea. His research interests are image processing, pattern recognition, character recognition, and digital library.

Soo Hyung Kim received his B.S degree in Computer Engineering from Seoul National University in 1986, and his M.S and Ph.D degrees in Computer Science from Korea Advanced Institute of Science and Technology in 1988 and 1993 respectively. From 1990 to 1996, he was a senior member of research staff in Multimedia Research Center of Samsung Electronics Co., Korea. Since 1997, he has been a professor in the Department of Computer Science, Chonnam National University, Korea. His research interests are pattern recognition, document image processing, medical image processing, and ubiquitous computing.