Text and Non-text Segmentation based on Connected Component Features

Viet Phuong Le*, Nibal Nayef*, Muriel Visani*, Jean-Marc Ogier* and Cao De Tran†
*Laboratory L3I, Faculty of Science and Technology, La Rochelle University, France
†College of Information and Communication Technology, Can Tho University, Vietnam
{viet_phuong.le, nibal.nayef, muriel.visani, jean-marc.ogier}@univ-lr.fr, tcde@cit.ctu.edu.vn

Abstract—**Document image segmentation is crucial to OCR and other digitization processes. In this paper, we present a learning-based approach for text and non-text separation in document images. The training features are extracted at the level of connected components, a mid-level between the slow noise-sensitive pixel level, and the segmentation-dependent zone level. Given all types, shapes and sizes of connected components, we extract a powerful set of features based on size, shape, stroke width and position of each connected component. Adaboosting with Decision trees is used for labeling connected components. Finally, the classification of connected components into text and non-text is corrected based on classification probabilities and size as well as stroke width analysis of the nearest neighbors of a connected component. The performance of our approach has been evaluated on the two standard datasets: UW-III and ICDAR-2009 competition for document layout analysis. Our results demonstrate that the proposed approach achieves competitive performance for segmenting text and non-text in document images of variable content and degradation.**

I. Introduction

Separating text and non-text in a document is an important layout analysis step in the Document Image Analysis and Recognition (DIAR) field. Such a step improves the accuracy rate as well as the running time within the OCR process or in some other DIAR tasks. Separating text and non-text is also useful for information spotting tasks in documents such as symbol spotting, word spotting, logo spotting, and so on, because it is easier and faster to perform spotting if the text and non-text are well segmented. Non-text in documents can be one of the following categories: halftone, drawing, math, logo, table, chart, separator, etc.

In literature, text/non-text segmentation approaches can generally be classified into three groups: (i) region (or block or zone) based segmentation [1], [2], (ii) pixel based segmentation [3], [4], and (iii) connected component based segmentation [5], [6], [7]. In region-based segmentation approaches, a page segmentation step on the document image is firstly done to get document zones, and then a classifier is applied to classify those zones [8]. Methods of this type of approach rely much on the accuracy of the segmentation step, which can be challenging in less-structured documents of complex layout. In pixel-based approaches, the classification is applied on each pixel in a document. Methods which follow this approach tend to be sensitive to noise and time consuming. The connected component-based approaches, on the other hand, are independent of block (zone) segmentation step, and are more robust due to considering the connected component level to discriminate between text and non-text.

The review done by Okun et al. [2] details the approaches of page segmentation and also of zone classification, summing up the main approaches used for document segmentation and region classification in the 1990s. Run Length Smearing Algorithm (RLSA), presented by Wong et al. [1], is one of the earliest approaches using region-based segmentation. RLSA analyses the spaces between black pixels in order to merge characters into blocks and then use an analysis technique to classify each block. Lin et al. [9] proposed a region-based approach. They used different texture-based GLCM (Grey Level Co-occurrence Matrix) features to divide a document into blocks of graphics, text and space zones, and used K-means for clustering the blocks into zones. They then used pre-learned heuristic rules for zone classification.

In connected component based segmentation, there are some noticed approaches, such as [5], [6], [7]. Fletcher and Kasturi [5] proposed a method based on Hough transform to group connected components into a text string and then classify them by using the analysis method. Tombre et al. [6] present a consolidation of the method proposed by Fletcher and Kasturi, with a number of improvements in the analysis method.

Bukhari et al. [7] have presented a method that is more related to the method proposed here. They use shape and context information of each connected component as a feature vector and then discriminate them into text and non-text classes by a self-tunable multi-layer perceptron (MLP) classifier. Each connected component and its surrounding context area are rescaled to 40x40 pixel window size for a 3204-dimension vector including four other features based on the connected component's size. However, rescaling a large connected component into a small window size will reduce their shape information. Thus, it will become a black window if the connected component is large and solid enough. Hence, Bukhari's method works only for separating text from halftone, one specific type of non-text. To solve the shortcomings raised by this, we extract – without component rescaling – features about shape information, size information, stroke width information, and context information. Our method is able to segment text versus all types on non-text.

This paper is organized as follows. In section II, we describe our proposed method and detail the different blocks in our framework: pre-processing, feature extraction, learning with Adaboosting decision trees and post-processing. In section III, we discuss the evaluation protocol for our method on the connected component-level and the pixel-level. In Section IV we present the experimental results and comparison to state-of-the-art methods. We draw conclusions in the last section.

978-1-4799-1805-8/15/$31.00 ©2015 IEEE

II. Our Proposed Method

Figure 1 shows a block-diagram of our proposed method for segmenting text and non-text in document images. We describe each of the blocks in detail in the following subsections.

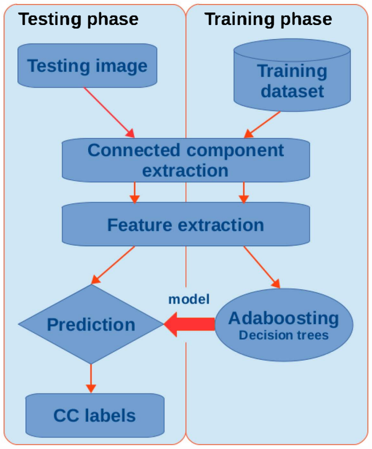


Fig. 1. Block diagram of our proposed method of text and non-text segmentation.

A. Pre-processing

Our method is a connected component-based method. Therefore, pre-processing is an important step to binarize images, extract connected components and remove noise. First of all, Otsu's binarization method is applied to binarize documents and then all connected components are extracted by the connected component labeling method. However, not all connected components are suitable for learning phase or/and testing phase such as noise, stains resulting from the scanning process, etc. Based on the characteristics of connected components such as size, shape and position, we apply some rules to remove such small noisy components and stain connected components which usually have long shapes and appear in the boundary of a document.

B. Feature Extraction

Many features can be extracted from connected components. However, if a selected feature is not good, it does not benefit classification. As shape and context are very important features with which humans recognize or segment and image, we extract features from size information, shape information, stroke width, and position of connected components. Bonakdar [10] has computed a set of such features for connected components. We use the stroke width features and the subset of features presented in [10]. Our selected set of features is presented in the following:

- $Elongation$ is the first feature used in our method. $Elongation$ is the height to width ratio of a connected component. It represents the square of connected component which has differences between texts and lines.

$$Elongation = \frac{min(height, width)}{max(height, width)} \qquad (1)$$

- $Solidity$ is the next selected feature. $Solidity$ is the number of black pixels divided by the area of the bounding box. It is selected for the reason that tables, borders and many graphical drawings has much different in solidity compared to texts.

$$Solidity = \frac{\#\_black\_pixels\_of\_component}{area\_of\_bounding\_box} \qquad (2)$$

- $Height$, $Width$ and $X - Y coordinates$ of a connected component are also selected into the set of features. Most of text connected components in a document have the same size or very small difference. Their position is also useful for classification, because unlike text, most of noise components, borders and frames are located near the boundaries of a document. X-Y coordinates is the center of a connected component. $Height$ and $Width$ is the size of the bounding box of a connected component.
- $Hu\ moments$ [11] are a set of good features to describe the shape of connected components because they are invariant to the image scale, rotation and reflection. In our experiments, the first four of seven moments are selected to our set of features.
- The $stroke\ width$ of connected components is a good feature for discriminating text from non-text. This is due to that text characters, line etc. have nearly constant stroke width while non-text do not [12]. The stroke width of an edge point of a connected component is computed based on the algorithm presented in [12]. The mean (mSW) and coefficient of variation $(CoefvariationSW)$ of the stroke width at all edge points are also considered as features in our method.

Normalizing the selected features is a necessary step. The size information and the position information of a connected component are normalized by considering all connected components of a document. Normalization overcomes the problem of documents of different resolutions. Therefore, Log-normal distribution is used to normalize the elongation, solidity and height of the connected components, while the height and the width are normalized with respect to the size of the document from which they are extracted. Log-normal distribution is a probability distribution of a random variable whose logarithm is normally distributed. It is defined as

$$F(x;\sigma;\mu) = \frac{1}{x\sigma\sqrt{2\pi}} \exp - \frac{(\ln x - \mu)^2}{2\sigma^2} \qquad (3)$$

where x is variable; σ and μ are the mean and standard deviation of the variable's natural logarithm for all connected components of the document respectively.

We additionally consider the context of connected components. According to [7], the surrounding context of a connected component provides important information for separating text and non-text. The ratio of height, width and stroke width between a connected component and its k-nearest neighbors are extracted as features (k=10 in our experiments). Table I shows the selected features. The last three features are computed from the k-nearest neighbor connected components as the surrounding context. The $mean\_of\_mSW$, $mean\_of\_height$ and $mean\_of\_width$ are the mean of mSW, $height$ and $width$ features of k-nearest neighbors.

TABLE I. THE SELECTED FEATURE SET FOR BUILDING OUR MODEL FOR TEXT AND NON-TEXT CLASSIFICATION.

Features	Formulas
Log-normal distribution of 1/elongation	Eq 3, $x = 1/elongation$
Log-normal distribution of 1/solidity	Eq 3, $x = 1/solidity$
Log-normal distribution of height	Eq 3, $x = height$
Normalized x's center	$x/doc.width$
Normalized y's center	$y/doc.height$
Logarithm Normalized of height	$\log(doc.height/height)$
Logarithm Normalized of width	$\log(doc.width/width)$
Logarithm of 1/elongation	$\log(1/elongation)$
Logarithm of 1/solidity	$\log(1/solidity)$
Logarithm of Hu's moment 1	$\log(Hu1 + 1)$
Logarithm of Hu's moment 2	$\log(Hu2 + 1)$
Logarithm of Hu's moment 3	$\log(Hu3 + 1)$
Logarithm of Hu's moment 4	$\log(Hu4 + 1)$
Coefficient of variation of SW	$CoefvariationSW$
The ratio of mSW	$mSW/mean\_of\_mSW$
The ratio of height	$height/mean\_of\_height$
The ratio of width	$width/mean\_of\_width$

C. Learning by Adaboosting Decision Trees

Boosting, presented in [13], is a supervised and a powerful learning tool. The main idea of this learning technique is to combine the performance of many ”weak” classifiers (such as naive Bayes or Decision trees) to improve their performance. Different variants of boosting are known as Discrete Adaboost, Real AdaBoost, LogitBoost, and Gentle AdaBoost which have very similar overall structure [14]. In this paper, we use Discrete Adaboost with Decision trees because Decision trees is a simple learning method for our set of features, and it provides fast and good results. A two-class Discrete Adaboost model is trained as follows:

1) Given N samples (x_i, y_i) with $x_i \in \Re^K$, $y_i \in \{-1, +1\}$
2) Initialize a weight w_i of each sample.
3) For each weak classifier $T_m, m = 1, .., t$
   - Fit the classifier $T_m(x) \in \{-1, +1\}$, using weights w_i on the training data.
   - Compute error and scaling factor.
   - Update all weights w_i based on error and scaling factor so that the weights are increased for training samples that have been misclassified and vice versa.
4) The final classifier is the sign of the weighted sum over the individual weak classifiers.

D. Post-processing

The post-processing step is necessary to correct some connected components which are labeled incorrectly by the classifier. For example, some text connected components like lines such as ”l”, ”I”, ”-”, ”—”, ”_” and some connected components coming from more than one character which are assigned to incorrect class labels. In addition, the broken parts of non-text connected components also have to be fixed. To refine the class label of each connected component, we use the average text and non-text probabilities of nearest neighbors and the analysis of size and stroke width to update the classified labels of connected components. That is, if a connected component classified as non-text appears within high text probabilities of nearest neighbors and if it has the similar size as well as stroke width, it will be reclassified as text connected component and vice versa. Figure 2 shows a case before and after using post-processing step.

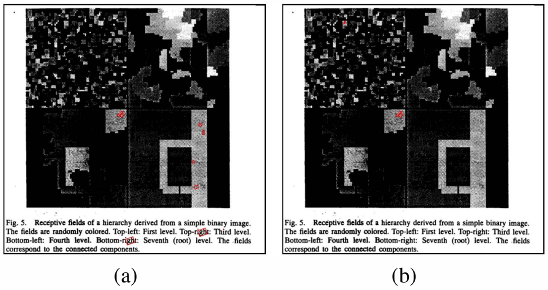


Fig. 5. Receptive fields of a hierarchy derived from a simple binary image. The fields are randomly colored. Top-left: First level. Top-right: Third level. Bottom-left: Fourth level. Bottom-right: Seventh (root) level. The fields correspond to the connected components.

(a)

Fig. 5. Receptive fields of a hierarchy derived from a simple binary image. The fields are randomly colored. Top-left: First level. Top-right: Third level. Bottom-left: Fourth level. Bottom-right: Seventh (root) level. The fields correspond to the connected components.

(b)

Fig. 2. A document region (a) before and (b) after post-processing step. Connected components shown in red colored boxes are incorrectly labeled by our method.

III. PERFORMANCE EVALUATION PROTOCOL

A. Databases

We experiment with two datasets. The first dataset is a subset of the standard UW-III dataset. We select 250 images from 1600 images which contain non-text regions, page-header regions, text-body regions, page-footer regions, etc. We relabel those regions as text or non-text, because we are mainly concerned by the ability to distinguish text from non-text regardless of the type of non-text. Figure 3 shows example documents from the UW-III dataset.

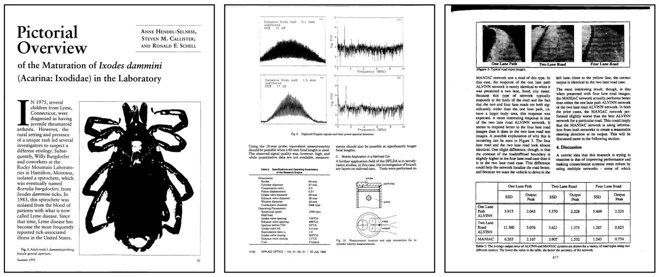


Fig. 3. Example documents of UW-III dataset, including text, image, graph and table regions.

The second dataset is ICDAR-2009 page segmentation competition dataset [15] for layout analysis of contemporary colored documents. In this dataset, there are a total of 55 images with different types of regions as as text, separator, graph, image, line art and noise. However, In our work, we only consider two classes text and non-text to perform segmentation. Therefore, we relabel the connected components in the ground truth text regions as text, and the components in all other regions regions as non-text. Figure 4 shows example documents and their region outlines of ICDAR-2009 dataset.

B. Evaluation

In this section, we present two methods for evaluation, one at the connected component-level and another at pixel-level. The evaluation at connected component-level assumes that the importance of all connected components is the same, while the evaluation at pixel-level uses weights for small and large connected components. The weight represented here as the area of connected components. *Precision*, and *Recall* are used to evaluate the performance of our segmentation method:

$$Precision = \frac{tp}{tp + fp} \tag{4}$$

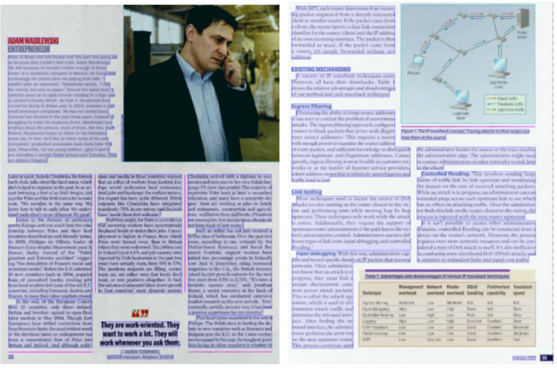


Fig. 4. Example documents of ICDAR-2009 dataset and their region outlines (blue: text, green: image, brown: table, magenta: separator)

$$Recall = \frac{tp}{tp + fn} \quad (5)$$

Evaluation based on connected component-level: each connected component has either text or non-text label based on ground-truth regions and a label predicted by our method. In that case, tp, tn, fp and fn are the number of connected components which are true positives, true negatives, false positives, and false negatives respectively.

Evaluation based on pixel-level: each pixel contains either text or non-text label based on ground-truth regions or predicted connected components. In that case, tp, tn, fp and fn are the number of pixels which are true positives, true negatives, false positives, and false negatives respectively. Moreover, as we would like to compare our method to that of [7], we additionally use the same notation they used. In [7], the metrics for performance evaluation are defined such as *Text classified as text*, *Non-text classified as non-text* and *Segmentation accuracy*. In fact, those are derived from *Recall*.

- *Text classified as text*: the ratio of intersection of text pixels in both segmented and ground truth image over the total number of text pixels in ground truth image. It is the *Recall* of text class.
- *Non-text classified as non-text*: the ratio of intersection of non-text pixels in both segmented and ground truth image over the total number of non-text pixels in ground truth image. It is the *Recall* of non-text class.
- *Segmentation accuracy*: average ratio of text classified as text accuracy and non-text classified as non-text accuracy.

IV. Experimental Results and Discussion

In the first experiment, to compare with the two popular classifiers Support Vector Machine (SVM) and MLP, we use a set of connected components selected randomly from ICDAR-2009 dataset with 10000 connected components (5000 text and 5000 non-text). We use 5-fold cross-validation method. Table II shows that Adaboosting Decision Trees is better and faster in our proposed method.

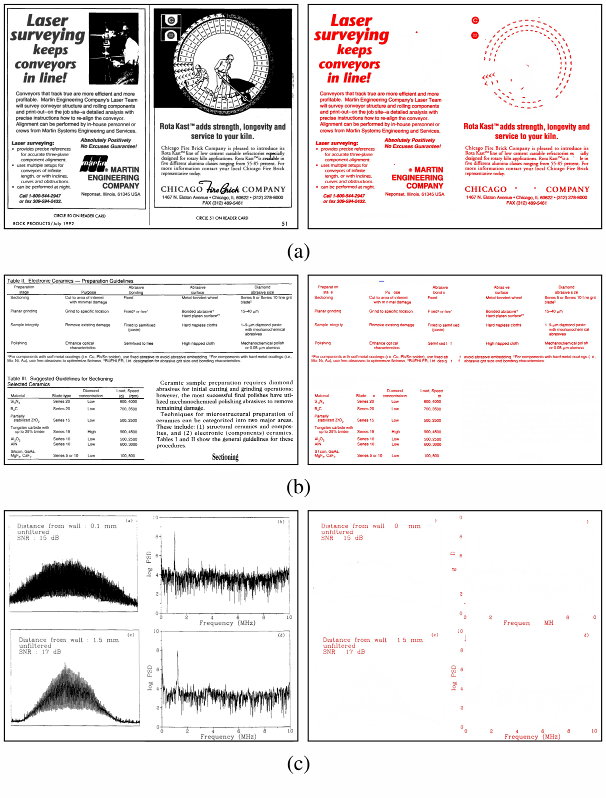
In the second experiment, we test on the subset of UW-III with 250 documents which contain halftones, drawings, and tables. We use 5-fold cross-validation method meaning 200 documents are selected for training randomly and the remaining 50 documents for testing. For training, we keep all non-text connected components and select randomly the number of text connected components twice as many as that of non-text connected components (because the number of text connected components is much higher than non-text components, which cause imbalanced training). For testing, we keep all text and non-text connected components. Table III shows the average results of multiple runs of our method, every time randomly selecting the set of documents for training and testing. The Recall of non-text is about 82.83%, meaning that nearly 18% non-text connected components are classified as text. However, most of them are actually text connected components but they are located in the non-text zones of ground-truth such as text in the drawing, tables, charts etc. Figure 5 shows three examples of texts contained in images, in tables and in charts where the actually text are labeled as non-text in ground-truth but they are predicted as text in our method.

TABLE II. Comparison of performance between Adaboosting Decision Trees and MLP, SVM on 10000 connected components.

	Mean Squared Error	Build Model Time (s)
LibSVM	0.3963	15.04
MLP	0.3283	23.44
Adaboosting Decision Trees	0.2184	11.04

TABLE III. Performance evaluation results of our method on a subset of the UW-III dataset (250 documents)

Evaluation method	Text		Non-text	
	Precision	Recall	Precision	Recall
CC-level	95.76%	99.25%	96.58%	82.83%
Pixel-level	94.69%	99.26%	99.72%	98.07%


Laser surveying keeps conveyors in line!
Rota Kast™ adds strength, longevity and service to your kiln.
CHICAGO Fire Brick COMPANY
MARTIN ENGINEERING COMPANY
Table II. Electronic Ceramics — Preparation Guidelines
Table III. Suggested Guidelines for Sectioning Selected Ceramics
Sectioning
Distance from wall 0.1 mm unfiltered SNR : 15 dB
Distance from wall 1.5 mm unfiltered SNR : 17 dB
log PSD
Frequency (MHz)

(a)

(b)

(c)

Fig. 5. Three examples of (a) text contained in images, (b) in tables, and (c) in charts. In each example, left shows original images (a part of a document) and right shows the connected components that are non-text in ground-truth but classified as text by our method.

In the third experiment, to compare our method with the method in [7], [16], 136 documents containing halftones are selected from UW-III dataset in order to try to ensure that it covers 95 documents selected by [7], [16]. The same scheme as in the second experiment is used. Table IV shows that our method is significantly better and on a larger set of documents.

TABLE IV. COMPARISON OF PERFORMANCE BETWEEN OUR METHOD AND THE METHODS IN [7], [16] ON UW-III DATASET (NOTE THAT OUR SUBSET HAS MORE DOCUMENTS THAN [7], [16]: 136 COMPARED TO 95 DOCUMENTS)

	Method in [7]	Method in [16]	Our method
non-text classified as non-text	98.91%	98.41%	99.49%
text classified as text	95.93%	99.42%	99.21%
segmentation accuracy	97.42%	98.92%	99.35%

Finally, we test on the challenging ICDAR2009 dataset. The same scheme for training phase and testing phase as in the second experiment is used. The average results of multiple runs are shown in Table V. There is a big difference of the non-text's recall between two evaluation methods because ICDAR-2009 dataset come from magazines. Therefore, there are many broken parts of natural images which are like as text. In this experiment, we also compare the performance of our system to three well-known state-of-the-art systems (ABBYY FineReader Engine 8.1, OCRopus 0.3.1 and Tesseract [17]) on F-measure. Table VI shows that our method achieves the best result on text at 98.98% and a good result at 64.99% on non-text. The performance can be improved by adopting sophisticated preprocessing techniques for grouping the broken connected components in non-text regions before applying our method.

TABLE V. THE RESULTS OF PERFORMANCE EVALUATION OF OUR METHOD ON ICDAR2009 DATASET (55 DOCUMENTS).

Evaluation method	Text		Non-text	
	Precision	Recall	Precision	Recall
CC-level	98.89%	99.09%	66.72%	63.35%
Pixel-level	98.37%	92.46%	92.37%	98.41%

TABLE VI. COMPARISON OF PERFORMANCE USING F-MEASURE BETWEEN OUR METHOD AND TESSERACT, FINEREADER AND OCROPUS ON ICDAR-2009 DATASET (55 DOCUMENTS).

	Text	Non-text
Tesseract [17]	92.50%	**74.23%**
FineReader	93.09%	71.75%
OCRopus	84.18%	51.08%
Our method	**98.98%**	64.99%

V. CONCLUSION

This paper has presented a method for segmenting the text and non-text in document images. The method is based on a set of powerful connected component features. Those features utilize size, shape, stroke width and position information of connected components. Adaboosting with decision trees trained on those features to obtain a model for labeling connected components. Our results show that the method is simple, fast and is really able to discriminate text from non-text, including the text that appears within graphical zones.

For future work, we are considering advanced preprocessing to resolve problems in distinguishing non-text in contemporary documents. Moreover, we will investigate the use of automatic feature learning for text versus non-text discrimination.

REFERENCES

[1] K. Y. Wong, R. G. Casey, and F. M. Wahl, "Document analysis system," *IBM journal of research and development*, vol. 26, no. 6, pp. 647–656, 1982.

[2] O. Okun, D. Dœrmann, and M. Pietikainen, "Page segmentation and zone classification: the state of the art," DTIC Document, Tech. Rep., 1999.

[3] M. Moll, H. Baird, and C. An, "Truthing for pixel-accurate segmentation," in *Document Analysis Systems, 2008. DAS '08. The Eighth IAPR International Workshop on*, Sept 2008, pp. 379–385.

[4] M. A. Moll and H. S. Baird, "Segmentation-based retrieval of document images from diverse collections," in *Electronic Imaging 2008*. International Society for Optics and Photonics, 2008, pp. 68 150L–68150L.

[5] L. A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 10, no. 6, pp. 910–918, 1988.

[6] K. Tombre, S. Tabbone, L. Pélissier, B. Lamiroy, and P. Dosch, "Text/graphics separation revisited," in *Document Analysis Systems V*. Springer, 2002, pp. 200–211.

[7] S. S. Bukhari, A. Azawi, M. I. Ali, F. Shafait, and T. M. Breuel, "Document image segmentation using discriminative learning over connected components," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, 2010, pp. 183–190.

[8] F. Shafait, D. Keysers, and T. Breuel, "Performance evaluation and benchmarking of six-page segmentation algorithms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 6, pp. 941–954, June 2008.

[9] M.-W. Lin, J.-R. Tapamo, and B. Ndovie, "A texture-based method for document segmentation and classification." *South African Computer Journal*, vol. 36, pp. 49–56, 2006.

[10] O. Bonakdar Sakhi, "Segmentation of heterogeneous document images: an approach based on machine learning, connected components analysis, and texture analysis," Ph.D. dissertation, Paris Est, 2012.

[11] M.-K. Hu, "Visual pattern recognition by moment invariants," *Information Theory, IRE Transactions on*, vol. 8, no. 2, pp. 179–187, 1962.

[12] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 5, pp. 970–983, May 2014.

[13] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Computational learning theory*. Springer, 1995, pp. 23–37.

[14] J. Friedman, T. Hastie, R. Tibshirani *et al.*, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[15] A. Antonacopoulos, S. Pletschacher, D. Bridson, and C. Papadopoulos, "Icdar 2009 page segmentation competition," in *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, July 2009, pp. 1370–1374.

[16] S. S. Bukhari, F. Shafait, and T. M. Breuel, "Improved document image segmentation algorithm using multiresolution morphology," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2011, pp. 78 740D–78 740D.

[17] R. Smith, "Hybrid page layout analysis via tab-stop detection," in *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, July 2009, pp. 241–245.