2011 International Conference on Document Analysis and Recognition

An MRF Model for Binarization of Natural Scene Text

Anand Mishra *, Karteek Alahari † and C.V. Jawahar *
* International Institute of Information Technology Hyderabad, India
† INRIA - Willow, ENS, Paris, France
Email: anand.mishra@research.iiit.ac.in, karteek.alahari@ens.fr, jawahar@iiit.ac.in

***Abstract*—Inspired by the success of MRF models for solving object segmentation problems, we formulate the binarization problem in this framework. We represent the pixels in a document image as random variables in an MRF, and introduce a new energy (or cost) function on these variables. Each variable takes a foreground or background label, and the quality of the binarization (or labelling) is determined by the value of the energy function. We minimize the energy function, i.e. find the optimal binarization, using an iterative graph cut scheme. Our model is robust to variations in foreground and background colours as we use a Gaussian Mixture Model in the energy function. In addition, our algorithm is efficient to compute, and adapts to a variety of document images. We show results on word images from the challenging ICDAR 2003 dataset, and compare our performance with previously reported methods. Our approach shows significant improvement in pixel level accuracy as well as OCR accuracy.**

***Keywords*-MRF, GMM, Graph Cut, Binarization**

I. Introduction

Binarization is one of the key preprocessing steps in any document image analysis system. The performance of the subsequent steps like character segmentation and recognition are highly dependant on the success of binarization. Document image binarization is an active area of research for many years. *Is binarization a solved problem?* Obviously not, especially, due to the emerging need for recognition of text in video sequences, digital-born (Web and email) images, old historic manuscripts and natural scenes where the state of art recognition performance is really poor. In this regard, designing a powerful binarization algorithm can be considered as a major step towards robust text understanding. The recent interest of the community by organising a binarization contest like DIBCO 2009 [1] at 10th International Conference on Document Analysis and Recognition (ICDAR 2009) also supports our claim. Note that DIBCO 2009 had 43 submissions which shows active interest in this research area.

We, in this work, focus on binarization of natural scene text. Natural scene texts contain numerous degradations not usually present in machine printed ones such as uneven lighting, blur, complex background, perspective distortion, multiple colours etc. Methods such as interactive graph cut by Boykov *et al.* [2] and thereafter GrabCut [3] have shown promising performance in foreground/background segmentation of natural scenes in recent years. We formulate the binarization problem in this framework (where text is foreground and anything else is background), and define a novel energy (cost) function such that the quality of the binarization is determined by the energy value. We minimize this energy function to find the optimal binarization using an iterative graph cut scheme. The graph cut method needs to be initialized with foreground/background seeds. To make the binarization fully automatic, we obtain initial seeds for graph cuts by our auto-seeding algorithm. At each iteration of graph cut, seeds and binarization are refined. This makes it more powerful compared to one shot graph cut algorithm. Moreover, we model foreground and background colours in a GMMRF framework [4] to make the binarization robust to variations in foreground and background colours.

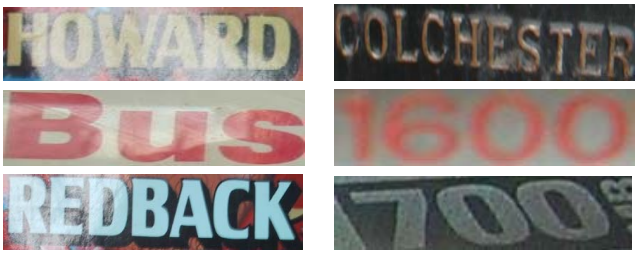

Figure 1. Some samples images we considered in this work

The remainder of the paper is organised as follows. We discuss related work in Section II. In Section III, the binarization problem is formulated as a labelling problem, where we define an energy function such that its minimum corresponds to the target binary image. This section also briefly introduces graph cut method. Section IV explains proposed iterative graph cut based binarization scheme. It also elaborates the method of finding auto-seeds for the graph cut. Section V describes experiments and results based on the challenging ICDAR 2003 word dataset. Some sample images of this dataset are shown in Figure 1. We finally conclude the work in Section VI.

II. Related Work

Traditional thresholding based binarization can be categorized into two categories: the one which uses global threshold for the given document (like Otsu [5], Kittler *et al.* [6]) and the one with local thresholds (like Sauvola [7], Niblack [8]). An exhaustive review of thresholding based

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.12

binarization is beyond the scope of this paper. The reader is encouraged to see [9] for this. Although most of these previous algorithms perform satisfactorily for many cases, they suffer from the problems like: (1) Manual tuning of parameters, (2) High sensitivity to the choice of parameters, (3) Handling images with uneven lighting, noisy background, similar foreground-background colours.

Recently, Markov Random Field (MRF) based binarization has been applied for degraded documents. In [10], Wolf *et al.* proposed binarization in an energy minimization framework and applied a less powerful and computationally expensive simulated annealing (SA) for energy minimization. In [11], authors classified document into Text Region (TR), Near Text Region (NTR) and Background Regions (BR) and then applied graph cut to produce final binary image. MRF based binarization for hand-held device captured document images was proposed in [12], where authors first used thresholding based technique to produce a binary image and then applied graph cuts to remove noise and smooth binarization output. However, these methods can not be directly applied to natural scene text images due to additional challenges like blur, hardly distinguishable foreground/background colours, variable font sizes, and styles.

Researchers have also shown interest in colour image binarization in recent years (see [13], [14]). But these methods lack a principled formulation of the binarization problem of complex colour documents, and hence can not be generalized.

III. The Binarization Problem

We define the binarization problem in a labelling framework as follows: the binarization of an image can be expressed as a vector of binary random variables $\mathbf{X} = \{X_1, X_2, ..., X_n\}$, where each random variable X_i takes a label $x_i \in \{0, 1\}$ based on whether it is text (foreground) or non-text (background). Most of the heuristic based algorithms take the decision of assigning label 0 or 1 to x_i based on the pixel value at that position or local statistics. Such algorithms are not effective in our case because of the variations in foreground/background colour distributions.

In this work, we formulate the problem in a more principled framework where we represent image pixels as nodes in a Markov Random Field and associate a unary and pairwise cost of labelling pixels. We then solve the problem in an energy minimization framework where the "Gibbs" energy function E of following form is defined:

$$E(\mathbf{x}, \boldsymbol{\theta}, \mathbf{z}) = E_i(\mathbf{x}, \boldsymbol{\theta}, \mathbf{z}) + E_{ij}(\mathbf{x}, \mathbf{z}), \quad (1)$$

such that its minimum corresponds to the target binary image. Here $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ is a set of labels at each pixel. $\boldsymbol{\theta}$ is the set of model parameters which is learnt from the foreground/background colour distributions and the vector $\mathbf{z} = \{z_1, z_2, ..., z_n\}$ denotes the colour intensities of pixels.

In Equation (1), $E_i(\cdot)$ and $E_{ij}(\cdot)$ corresponds to data term and smoothness term respectively. Data term $E_i(\cdot)$ measures the degree of agreement of the inferred label x_i to the observed image data z_i. The smoothness term measures the cost of assigning labels x_i, x_j to adjacent pixels and is used to impose spatial smoothness. A typical unary term can be expressed as:

$$E_i(\mathbf{x}, \boldsymbol{\theta}, \mathbf{z}) = -\sum_i \log p(x_i|z_i).$$

Similarly, the smoothness term most commonly used in literature is the Potts model:

$$E_{ij}(\mathbf{x}, \mathbf{z}) = \lambda \sum_{(i,j)\in\mathbf{N}} exp\frac{-(z_i - z_j)^2}{2\beta^2}\frac{[x_i \neq x_j]}{dist(i,j)},$$

where λ determines the degree of smoothness, $dist(i, j)$ is the Euclidean distance between neighbouring pixels i and j. The constant β allows discontinuity preserving smoothing. $\mathbf{N}$ denotes the neighbourhood system defined in MRF. Further, the smoothness term imposes cost only for those adjacent pixels which have different labels (i.e. $[x_i \neq x_j]$).

The problem of binarization is now to find the global minima of the Gibbs energy, i.e.,

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} E(\mathbf{x}, \boldsymbol{\theta}, \mathbf{z}). \quad (2)$$

The global minima of this energy function can be efficiently computed by graph cut [15] subject to fulfilling the criteria of sub modularity [16]. For this a weighted graph $G = (V, E)$ is formed where each vertex corresponds to an image pixel, and edges link adjacent pixels. Two additional vertices source (s) and sink (t) are added to the graph. All the other vertices are connected to them with weighted edges. The weights of all the edges are defined in such a way that every cut of the graph is equivalent to some label assignment to the energy function. Note that the cut of the graph G is a partition of set of vertices V into two disjoint sets S and T and the cost of the cut is defined as the sum of the weights of edges going from vertices belonging to set S to T (see [16]). The min cut of such a graph corresponds to the global minima of the energy function. There are efficient implementations available for finding min cut of such a graph [15].

In [2], the set of model parameters $\boldsymbol{\theta}$ describe image foreground/background histograms. The histograms are constructed directly from the foreground/background seeds which are obtained with user interaction. However, the foreground/background distribution in our case (see images in Figure 1) can not be captured efficiently by a naive histogram distribution. Rather, we assume each pixel colour is generated from a Gaussian Mixture Model (GMM). In this regard, we are highly inspired by the success of the GrabCut [3] for object segmentation. But at the same time, we want to avoid any user interaction to make the binarization fully automatic. We achieve this by our auto seeding algorithm

which we describe in the Section IV-A. Furthermore, iterative graph cut based binarization is also more suitable for our application as it refines seeds and, binarization output at each iteration and thus produces a clean binarization result even in case of noisy foreground/background distributions.

IV. Iterative Graph Cut based binarization

In GMMRF framework [4], each pixel colour is generated from one of the 2c Gaussian Mixture Models (GMMs) (c GMMs for foreground and background each) with mean μ and covariance Σ i.e. each foreground colour pixel is generated from following distribution:

$$p(z_i|x_i, \boldsymbol{\theta}, k_i) = \mathcal{N}(\mathbf{z}, \boldsymbol{\theta}; \mu(x_i, k_i), \Sigma(x_i, k_i)), \qquad (3)$$

where $\mathcal{N}$ denotes a Gaussian distribution, $x_i \in \{0, 1\}$ and $k_i \in \{1, ..., c\}$. To model foreground colour using above distribution, an additional vector $\mathbf{k} = \{k_1, k_2, ..., k_n\}$ is introduced where each k_i takes one of the c GMM components. Similarly, background colour is modelled from one of the c GMM components. Further, the likelihood probabilities of observation can be assumed to be independent from the pixel position. Thus can be expressed as:

$$p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}, \mathbf{k}) = \prod_i p(z_i|x_i, \boldsymbol{\theta}, k_i)$$
$$= \prod_i \frac{\pi(x_i, k_i)}{\sqrt{det(\Sigma(x_i, k_i))}} \times$$
$$exp(-\tfrac{1}{2}(z_i - \mu(x_i, k_i))^T \Sigma(x_i, k_i)^{-1}(z_i - \mu(x_i, k_i))).$$

Here $\pi(\cdot)$ is Gaussian mixture weighting coefficient.

Due to the introduction of GMMs the energy function in Equation (1) now becomes:

$$E(\mathbf{x}, \mathbf{k}, \boldsymbol{\theta}, \mathbf{z}) = E_i(\mathbf{x}, \mathbf{k}, \boldsymbol{\theta}, \mathbf{z}) + E_{ij}(\mathbf{x}, \mathbf{z}), \qquad (4)$$

i.e. the data term depends on its assignment to GMM component. It is given by:

$$E_i(\mathbf{x}, \mathbf{k}, \boldsymbol{\theta}, \mathbf{z}) = -\sum_i \log p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}, \mathbf{k}). \qquad (5)$$

In order to make the energy function robust to low contrast colour images we modify the smoothness term of the energy function by adding a new term which measures the "edginess" of the pixels as follows:

$$E_{ij}(x, z) = \lambda_1 \sum_{(i,j)\in\mathbf{N}} [x_i \neq x_j] exp(-\beta||z_i - z_j||^2)$$
$$+\lambda_2 \sum_{(i,j)\in\mathbf{N}} [x_i \neq x_j] exp(-\beta||w_i - w_j||^2). \qquad (6)$$

Here w_i denotes the magnitude of gradient (edginess) at pixel i and $\mathbf{N}$ denotes the neighbourhood system defined for the MRF model. The two neighbouring pixels with similar edginess values are more likely to belong to the same class. The edginess term enforces this constraint. The constants λ_1 and λ_2 determine the relative strength of the colour and edginess differences respectively. Parameters λ_i and β are learnt automatically from the image.

The Gaussian Mixture Models, in Equation (5), need to be initialized with foreground/background seeds. Since our objective is to make the binarization fully automatic, we initialize GMMs by foreground-background seeds obtained from our auto seeding algorithm. Then, at each iteration, the seeds are refined and new GMMs are learnt from them. It makes the algorithm more powerful and allows it to adapt to the variations in foreground/background.

A. Auto-seeding

To perform automatic binarization we need to compute foreground and background seeds for graph cut. Given an image we first convert it to an edge image using Canny edge operator and then find the foreground and background seeds as follows:

1) Foreground seeds: Our foreground seeding algorithm is highly motivated from the fact that there exist a parallel edge curve (line) for every edge curve (line) in a character i.e. if an edge pixel has gradient orientation θ then in direction of θ there exists an edge pixel whose gradient orientation is $\pi - \theta$

Step 1: Let p be a non-traversed edge pixel with gradient orientation θ. For every such edge pixel p we traverse the edge image in direction of θ until we hit an edge pixel q whose gradient orientation is $(\pi - \theta) \pm \frac{\pi}{36}$ (i.e. approximately opposite gradient direction). We mark this line segment $\overline{pq}$ as foreground seed candidate and store the length of it. We repeat this process for all the non-traversed edge pixels. After finding all foreground seed candidates, we remove all those line segments whose length is too high or too low with respect to the majority of seed candidates. The remaining line segments are marked as foreground seeds.

Step 2: Handling images with light text on dark background: When we have such image we rarely get parallel edge curves (lines) with the above mentioned traversal, rather many line segments $\overline{pq}$ start hitting the image boundary. We automatically detect such situations and subtract π to the original orientation and then follow the same process as Step 1.

2) Background seeds: For background seeding we adopt the following scheme: Given an edge image we find out the horizontal/vertical line having no edge pixel. We mark that line as background. When we do not get background seeds in the above method then we relax our criteria and mark all those regions as background which are accessible (without hitting an edge pixel) from at least two sides of image boundary. In practice, for some cases we do not get enough background seeds even after relaxation. For such cases we traverse the edge image from all four sides of the image boundary till we hit an edge. We mark all these regions as background seeds. Figure 2 shows typical initial seeds for the iterative graph cut.

(a) (b)

Figure 2. (a) Input Image (b) Its foreground-background seeds, Red and blue colour shows foreground and background seeds respectively (Best viewed in colour).

Figure 3. Images where auto-seeding fails

Although the proposed auto-seeding method performs satisfactorily well, it tend to fail in cases where Canny edge operator produces too many noisy or broken edges. In such cases some foreground regions are falsely marked as background and vice-versa, which leads to poor binarization. We show two such example in Figure 3, where our auto-seeding algorithm fails to mark foreground-background regions appropriately.

In summary, once we obtain initial seeds, GMMs for foreground and background colours are learnt. Then, based on the data and smoothness terms in Equation (5) and (6) respectively, the graph is formed. We use standard graph cut algorithm [15] to obtain initial binarization result. We then re-estimate GMMs using an initial binarization result and iterate the graph cut over new data and smoothness term, until convergence. This refines the binary image at each iteration and finally produces a clean binary image.

V. Results and Discussions

We use sample images from the ICDAR 2003 Robust Word Recognition dataset [17] for our experiments. It consists of 171 natural scene text images. These images have several degradations due to uneven lighting, complex background, blur and similar foreground background colours. To evaluate the performance of proposed binarization algorithm, we compare it with the well-known thresholding based binarization techniques like Otsu [5], Sauvola [7], Niblack [8], Kittler *et al.* [6]. We also compare our binarization algorithm with colour thresholding based method proposed in [14]. Note that these classical binarization algorithms produce white text on black background in case of images with light text on dark background. On the contrary, our binarization algorithm works in object segmentation framework and thus produces black text on white background always. However, for fair comparison we reverse the colour of binarized output of the classical methods if they produce white text on black background.

For the proposed binarization algorithm we used 10 GMM components (5 each for foreground and background). We empirically determine the number of iteration for graph cuts as 8, since no significant change in binarization, was observed beyond 8 iterations. We also show our results with and without edginess difference in the the pairwise term. (Note that by edginess difference term we mean, energy function with gradient magnitude difference in addition to difference in RGB colour space). For parameter sensitive algorithms like [7] and [8] we use the parameters from which we obtain the best OCR accuracy.

All the implementations of the proposed method are done using C++ graph cut code [15] and Matlab. The proposed method takes 32 seconds on average to produce final binary result for an image on system with 2 GB RAM and Intel® Core™ 2 Duo CPU with 2.93 GHz processor system.

A. Qualitative evaluation

First we compare the proposed binarization algorithm with thresholding based methods intuitively in Figure 4. Samples of images with uneven lighting, hardly distinguishable foreground/background colours, noisy foreground colours, are shown in this figure. We observe that our approach produces clearly readable binary images. Further, our algorithm produces lesser noise compared to the local thresholding based algorithms like [7], [8], which also helps to improve the OCR accuracy.

B. Quantitative evaluation

Quantitative evaluation of binarization is one of the biggest challenge for document image community [9]. In this work, we demonstrate the performance of binarization not only based on OCR accuracy but also in terms of pixel level accuracy.

1) OCR accuracy: We test OCR accuracy to verify robustness of our algorithm. For this we fed the binarization result of all algorithms to commercial OCR ABBYY fine reader 9.0 [18]. The word and character recognition accuracies are summarized in Table I. Since this dataset consists of images of tight word boundaries, global methods (like [5], [6]) performs better than popular local methods. Furthermore, OCR fails to perform well in case of noisy binarization output (as in the case of Sauvola and Niblack). Otsu followed by colour thresholding binarization proposed in [14] improves the word recognition accuracy but not significantly. However, since the proposed algorithm produces clean binary images, it shows significant improvement in OCR accuracy.

2) Pixel level accuracy: For comparing various binarization algorithms based on *pixel accuracy*, we picked 30 images from the ICDAR 2003 word dataset and produced pixel level binarization ground truth for it. These images

Figure 4. Comparison of thresholding based algorithms and the proposed method (From left to right Original, Otsu, Sauvola, Niblack, Kittler, Proposed (with edginess difference)).

Table I
OCR ACCURACY(IN %)

Method	Word accuracy	Character accuracy
Otsu + CT [14]	45.03	51.98
Otsu [5]	41.52	51.74
Sauvola [7]	39.77	51.63
Niblack [8]	39.18	42.31
Kittler *et al.* [6]	41.52	49.88
Proposed (without edginess diff.)	49.12	55.94
Proposed (with edginess diff.)	**52.04**	**60.14**

Table II
BINARIZATION RESULTS WITH RESPECT TO WELL-KNOWN EVALUATION MEASURES (AVERAGE)

Method	Prec.	Recall	F-score	Specificity	GA
Otsu + CT [14]	0.76	0.89	78.12	0.78	0.81
Otsu [5]	0.77	0.87	79.32	0.81	0.83
Sauvola [7]	0.69	0.90	73.87	0.62	0.66
Niblack [8]	0.73	0.85	76.86	0.80	0.82
Kittler *et al.* [6]	0.74	0.86	72.89	0.68	0.67
Proposed without edginess	0.88	**0.91**	87.84	0.92	**0.91**
Proposed with edginess	**0.90**	**0.91**	**88.64**	**0.93**	**0.91**

Table III
EFFECT OF NUMBER OF GMM COMPONENTS

Number of GMM components	Average F-score
1	83.12
5	**88.64**
10	86.43
15	86.22

(a) (b) (c)

Figure 5. Effect of edginess difference term: (a) Original image (b) Without edginess difference (c) With edginess difference.

are selected in such a way that the text has enough thickness for producing pixel level ground truth without any difficulty. However, this set covers most of the challenges present in this dataset, like low contrast, multi-colour, noisy foreground/background, hardly distinguishable foreground background colours etc. We compare the proposed algorithm with classical thresholding based algorithms using well-known measures like precision, recall, F-score, specificity and geometric accuracy (GA). High values for all these measures is desired for a good binarization algorithm. Readers can refer [1] for details of these measures. We also compare our method with Otsu followed by colour thresholding (CT) [14]. Table II summarizes this result, where we outperform thresholding based methods in all measures. Furthermore, with edginess difference term we get more accurate result in terms of pixel level accuracy, especially in case of low contrast images (one such example can be seen in Figure 5).

We also study the effect of number of GMM components in binarization output. Table III summarizes it. We see that the GMM with 5 foreground-background components yields better average F-score compared to the GMM with more or lesser foreground-background components.

VI. CONCLUSION

In this work we address the problem of binarization of natural scene text. The binarization problem is formulated in a principled framework to handle various kinds of noise commonly present in natural scene texts. Proposed binarization method is robust to variations in foreground - background colours as it uses GMM for modelling colour distributions. Moreover, it is powerful than one shot graph cut based binarization because it refines the seeds and binarization output at each iteration and produces clean binary images. Experimental results shows that we clearly outperform the well-known thresholding based methods both in terms of pixel accuracy and OCR accuracy.

ACKNOWLEDGEMENTS

This work is supported by Ministry of Communication and Information Technology, Government of India, New Delhi. Karteek Alahari is supported by the Quaero programme funded by the OSEO.

REFERENCES

[1] B. Gatos, K. Ntirogiannis, and I. Pratikakis, "ICDAR 2009 document image binarization contest (DIBCO 2009)," in *ICDAR*, 2009, pp. 1375–1382.

[2] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *ICCV*, 2001, pp. 105–112.

[3] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[4] A. Blake, C. Rother, M. Brown, P. Pérez, and P. H. S. Torr, "Interactive image segmentation using an adaptive GMMRF model," in *ECCV (1)*, 2004, pp. 428–441.

[5] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Systems, Man, and Cybernetics Society*, vol. 9, no. 1, pp. 62 – 66, 1979.

[6] J. Kittler, J. Illingworth, and J. Föglein, "Threshold selection based on a simple image statistic," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 2, pp. 125–147, 1985.

[7] J. J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, no. 2, pp. 225–236, 2000.

[8] W. Niblack, *An introduction to digital image processing*. New York: Prentice Hall, 1986.

[9] P. Stathis, E. Kavallieratou, and N. Papamarkos, "An evaluation technique for binarization algorithms," *J. UCS*, vol. 14, no. 18, pp. 3011–3030, 2008.

[10] C. Wolf and D. S. Doermann, "Binarization of low quality text using a markov random field model," in *ICPR (3)*, 2002, pp. 160–163.

[11] J. G. Kuk and N. I. Cho, "Feature based binarization of document images degraded by uneven light condition," in *ICDAR*, 2009, pp. 748–752.

[12] X. Peng, S. Setlur, V. Govindaraju, and R. Sitaram, "Markov random field based binarization for hand-held devices captured document images," in *ICVGIP*, 2010.

[13] K. Kita and T. Wakahara, "Binarization of color characters in scene images using k-means clustering and support vector machines," in *ICPR*, 2010, pp. 3183–3186.

[14] C.Thillou and B. Gosselin, "Color binarization for complex camera-based images," in *Electronic Imaging Conference of the International Society for Optical Imaging*, 2005.

[15] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, 2004.

[16] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, 2004.

[17] Robust word recognition dataset, http://algoval.essex.ac.uk/icdar/RobustWord.html.

[18] ABBYY Finereader 9.0. http://www.abbyy.com/.