Binarization of Colored Document Images using Spectral Clustering

Enas M. Elgbbas*, Mahmoud I. Khalil*, and Hazem M. Abbas*
Email: enas.elgbbas@yahoo.com
*Faculty of Engineering, Ain Shams University, Egypt

***Abstract*—In this paper, we propose a hybrid method for text binarization of historical documents. Proposed method incorporates the advantages of Otsu and spectral clustering algorithm. In text binarization problem, there are noise and faint text challenges. To overcome the noise problem, a preprocessing step is applied to the colored document. After that, the resulted image is binarized using Otsu, producing a global binary image. As a final step, the spectral clustering algorithm is locally applied to the original image, with the aid of the global binary image, in order to retrieve faint text. The proposed design of spectral clustering provides a significant reduction of the similarity matrix computing time and size use, without affecting the quality of clustering. This method is suitable for colored document images. The efficiency of this method is illustrated by experiments using DIBCO images.**

I. Introduction and Related Work

Several methods are presented in the literature to find the binarization text in the document images. In general, binary methods are categorized into global, local and hybrid images (combination of global and local). A survey of these methods is presented in this section.

Global methods: In global thresholding techniques, such as Otsu[1] and Kapur[2], an optimal threshold is computed for the entire image. In Otsu method, the threshold is determined automatically by maximizing the between-class variance of foreground and background pixels, and in Kapur method, it is calculated by maximizing the entropy of two partitioned sub-images. These techniques require little calculations and produce suitable binarized results for images possessed under the normal lightness condition, but in poor contrast, variable intensity of foreground-background these methods fail to binarize the image. Using Otsu, faint texts exist in the dark background are missed or some of the faint characters become broken.

Local methods: In local binarization techniques, such as Niblack[3], Sauvola[4], Nick[5], and Bernsen[6], a local threshold value is computed for every individual pixel, by sliding a rectangular window over the entire image.

Niblack[3] calculates the local threshold using the mean and standard deviation of gray levels within each window. This method performs well in low contrast images, and in case of variation in text thickness size or low contrast image with a thin pen stroke text. However, it produces big binarization noise in the windows that do not contain text.

Sauvola[4] is a modulation of Niblack. This method does better in the images with a background containing uneven illumination and too variation or light texture, but it cannot remove the strong background degradation and fail in case of bleed-through, or when pen stroke text is thin. In the case of low contrast images, the faint text is breaking up or removed.

Nick[5] solves the problem of noise for empty windows in Niblack method and the low contrast problem in Sauvola method. This is done by reducing the thresholding value. Yet, it fails in case of too small contrast or in case of thin pen stroke text.

The threshold of Bernsen[6] is the mean of minimum and maximum gray values in the local window. This method does not perform well in complex background images. For the local method, choosing an inappropriate window size and unsuitable parameters result in poor performance.

In Shi[7] method, a local thresholding algorithm that exploiting stroke width as shape information, is presented. This method failed in faint text recovering, and stain or shadow removing.

Hybrid methods: In Lech[8] method, Otsu and Kapur methods are combined with their local versions and applied for blocks using the weighting coefficient, but the size of the block and weighting coefficient are manually selected. It is used for uniform images.

Singh[9] method composed of contrast analysis, contrast stretching for the window contains text pixels, global threshold, and noise removal. This method is applied to low illuminated and variable contrast images. Noise is produced in the following cases: in bleed-through degradation, when there is less text in a document very dark background shades text and produces broken characters in case of faint text.

Chiu[10] method consists of two stages. In the first stage, a proper window size determined based on the local variation of pixel intensities. In the second stage, noise is suppressed by contrasting two binarized images produced using two thresholding schemes which incorporate information on local mean gray and gradient values. This method is applied on images with a low variation on background and foreground.

The organization of this paper is as follows. Spectral clustering algorithm is summarized in section(II). The proposed method is explained in section(III). Experimental results and conclusions are presented in section(IV) and section(V) respectively.

II. OVERVIEW OF SPECTRAL CLUSTERING ALGORITHM

Clustering process is defined as an unsupervised method that is used for data pattern extraction. The main objective of clustering is to group similar samples in the same cluster, based on distance measures. Abundant clustering methods are robustly restricted to Euclidean distance, assuming that clusters have convex regions. Spectral clustering methods hold a larger range of geometries. So that, it is extensively used in document clustering and image segmentation. The objective of spectral clustering is to segment the data points into k classes. The concept is to form a similarity matrix, calculate the Laplacian matrix and eigenvectors. The second eigenvector of the normalized graph Laplacian minimizes the normalized cut on a graph. The complexity of this method takes $O(n^2)$ in time to compute, and in space to store, where n is the number of points. In large-scale problems, it is difficult to apply the spectral clustering algorithm, due to its high computational complexity[11],[12], these methods lose information about the original data, thus resulting in a poor performance.

In this section, a brief description of spectral clustering algorithm is presented [13], [14], [15], [16]. Assuming a group of n points $x1, x2, ..., xn$, spectral clustering forms an undirected graph $G = (V, E)$ with vertex set $V = v1, ..., vn$ corresponds to the n data points. For each edge $E = e1, e2, ..., en$ we associate a weight $w(i, j)$ that encodes the affinity (or similarity) of the points xi and xj. This graph is represented by its adjacency matrix W

$$W = w(i, j)_{i,j=1,...,n} \tag{1}$$

For not connected vertices (vi and vj), $w(i, j)$ is equal to 0, and $w(i, j) = w(j, i)$ such that G is unidirectional. The Gaussian similarity function is usually used in the fully connected graph. Edges between far points have low weights, while near points have high weights. The Gaussian parameter σ dominances how quickly the similarity W decreases with the distance between xi and xj. The degree matrix D is a diagonal matrix whose elements are columns or row sums of $w(i, j)$

$$D(i, i) = \sum_{j}^{n} w(i, j) \tag{2}$$

Let

$$L = D^{(-1/2)} * (D - W) * D^{(-1/2)} \tag{3}$$

which is called graph Laplacian. Spectral clustering then uses the top k eigenvectors of L corresponding to the k smallest eigenvalues, where the eigenvectors of a square matrix are the non-zero vectors that, after being multiplied by the matrix, remain parallel to the original vector. Finally, the k-means method is applied to cluster the data.

The aim of this paper is to binarize the document images, using an improved version of the spectral clustering algorithm in speed and complexity. This is accomplished by locally applying the spectral clustering algorithm on the text objects that are detected using a global binarization method. Inputs for the spectral clustering algorithm are reduced by selecting the pixels of interest in each local windows, based on their colors.

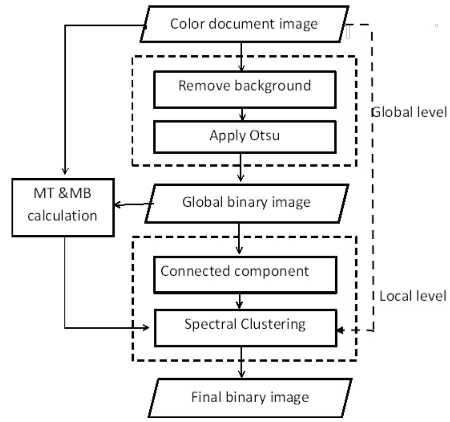


Fig. 1: A flowchart summarizes the proposed method

III. THE PROPOSED METHOD

The proposed method consists of two levels. In the first level (Global level), a high contrast color image is obtained in order to remove the most of background noise. Otsu binarization method is applied to the resulted high contrast image, producing a *global binary image*. That image may contain broken characters. In order to improve the *global binary image*, the contrast of the colored image is calculated by using *global binary image* and used in the second level. In the second level (Local level), the designed spectral clustering method is locally applied on connected components objects in the grayscale image with the aid of the *global binary image* and the calculated means for text and background. Fig. (1) summarizes flowchart of the proposed program.

A. Global Level

In this level, the *global binary image* is presented and obtained as follow. First, text in the original image $I(x, y, z)$ is eliminated by using the Gaussian filter, producing $G(x, y, z)$. The high contrast colored image $C(x, y, z)$ is obtained by subtracting the filtered image from the original image using the following equation.

$$C(x, y, z) = I(x, y, z) - G(x, y, z) \tag{4}$$

These steps are illustrated by Fig.(2). Fig.(2a) shows the original image. Fig.(2b) shows the image after text elimination using Gaussian filter. Fig.(2c) is the resulted high contrast image. It is clear that noise in the background is removed. Fig.(2c) is the *global binary image*. Although *global binary image* may have discontinuities on regions of weak ink, most of the noise in the background is removed. *Global binary image* is used as a mask to calculate text and background intensity means of the original colored image, as

Fig. 2: (a) is the original image, (b) is the image after applying Gaussian filter, (c) is the high contrast image produced by subtracting the filtered image from the original image, and (d) is the $global\ binary\ image$ produced by applying Otsu on the high contrast image

follow. First, intensity for text and intensity for background are calculated using equations (5) and (6).

$$T(x,y,z) = \begin{cases} I(x,y,z) & \text{if } O(x,y) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$B(x,y,z) = \begin{cases} I(x,y,z) & \text{if } O(x,y) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Where $I(x,y,z)$ is the original image, $O(x,y)$ is the ($global\ binary\ image$) that resulted by applying Otsu on the high contrast image $C(x,y,z)$. $T(x,y,z)$ and $B(x,y,z)$ are the results of applying the $global\ binary\ image$ ($O(x,y)$) as a mask for producing the intensity of text and background, respectively. The summations of intensity values for text and background are given by equations (7), (8) respectively. Number of background and text pixels are n_B and n_T given by equations (9), (10) , respectively. MT and MB given by equations (11) and (12) are means of intensity for text and background.

$$SumT(z) = \sum_{x,y} T(x,y,z) \tag{7}$$

$$SumB(z) = \sum_{x,y} B(x,y,z) \tag{8}$$

$$n_B = \sum_{x,y} O(x,y), forO(x,y) = 1 \tag{9}$$

$$n_T = x * y - n_B \tag{10}$$

$$MT(z) = \frac{SumT(z)}{n_T}, MT = \frac{MT(z)}{3} \tag{11}$$

$$MB(z) = \frac{SumB(z)}{n_B}, MB = \frac{MB(z)}{3} \tag{12}$$

Fig.(3) illustrates how to calculate the background intensity mean in the colored image. Fig.(3a) shows the mask ($global\ binary\ image$) that is used for background extraction. Fig.(3b) shows the background of colored image. Fig.(3c) shows the mean of that background. Fig.(3d) shows the background mean in the background locations. Fig.(4) illustrates

Fig. 3: Steps of calculating the background intensity mean: (a) is the mask used to extract the background, (b) is the background of the colored image, (c) is the background mean and (d) shows the background replaced by its mean.

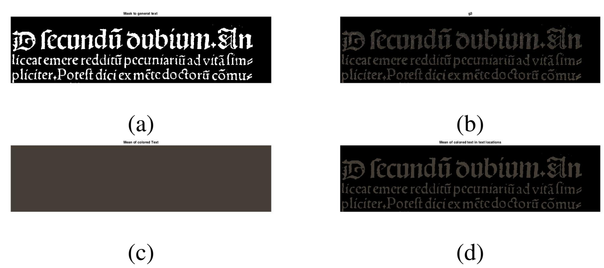


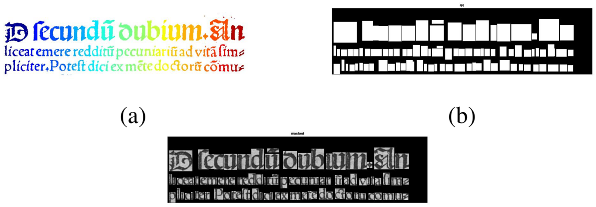
Fig. 4: Steps of calculating the text intensity mean: (a) is the mask used to extract the text, (b) is the text of the colored image, (c) is the text mean and (d) shows the text replaced by its mean

how to calculate the text intensity mean in the colored image. Fig.(4a) shows the mask used for extracting the image text. Fig.(4b) shows the text of colored image. Fig.(4c) shows the mean of text. Fig.(4d) shows text mean in text locations.

B. Local Level

In this level, the $global\ binary\ image$ that is produced in the previous section is improved. This is done by locally binarizing each object in the grayscale image. In this section, first, we introduce how to obtain the dynamic windows that fit each object. After that, we explain our proposed spectral clustering method.

1) **Obtaining the dynamic windows**: In order to obtain the dynamic windows that are used in local binarization, the connected components of the $global\ binary\ image$ are labeled. Then, rectangles with variable window sizes that fit each object, are produced. Fig.(5) shows how to create the dynamic windows and apply it on the grayscale image for local binarization in the next section. Fig.(5a) shows the connected components labels, such that each object has a color. Fig.(5b) shows the dynamic windows that fit each connected component. Fig.(5c) shows masks applied to each object in the grayscale image.
2) **The proposed spectral clustering algorithm**: In this section, the details of the proposed spectral clustering algorithm are introduced.
   In order to reduce the complexity of spectral clustering, the pixels of interest (POI) are selected. For that purpose,



(a) (b)

(c)

Fig. 5: Steps of dynamic windows creation and applying it on the grayscale image: (a) The connected components labels, (b) The dynamic windows that fit each connected component, and (c) Masks applied to each object in the grayscale image.

intensity means for text and background (MT and MB) calculated in the previous section, are used. Pixels in the grayscale image are classified into three types, background pixels, text pixels, and (POI) that are classified using spectral clustering. For each local window applied to the grayscale image, do the following steps:

a) The pixels that have intensity values greater than background intensity mean (MB) are classified as background. Pixels that have intensity values less than text intensity mean (MT) are classified as text. The remaining pixels are (POI). These pixels have intensity values less than background mean and greater than text mean, given by the following equation.

$$MT < Pixel(i,j) < MB \tag{13}$$

b) Form a weighted graph G = (V, E) by considering each pixel of (POI) as a node, and use edges to link each pair of these pixels. The edge weight is an indicator that the two pixels belong to the same class or different classes. The graph edge weight that connecting the two nodes i and j can be defined by using only the brightness Int value of that pixels using the following equation.

$$w(i,j) = \begin{cases} e^{(-\|Int(i)-Int(j)\|/(2*\sigma^2))} & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

c) Construct the degree matrix D by using the following equation:

$$D = \sum_{j}^{n} w(i,j) \tag{15}$$

d) Calculate the normalized Laplacian by using the following two equations:

$$L = D - W \tag{16}$$

$$L = D^{(-1/2)} * L * D^{(-1/2)} \tag{17}$$

e) In order to bipartition the graph, the eigenvector to the second smallest eigenvalue ($EV2$) of the normalized Laplacian (L) is computed.

f) After that, an unsupervised algorithm is used to classify the POI into text pixel or background, where black pixels represent the text and white pixels represent the background. So that, the K-means clustering method [17] is applied to ($EV2$). k is chosen to be equal to two. By this way, the $final\ binary\ image$ is obtained.

g) In order to remove the remaining background noise, local windows that have low stranded devisions or have small size are classified as background.

Fig.(6) shows an example to illustrate how the $global\ binary\ image$ is improved after applying the proposed spectral clustering method. Fig.(6a) shows the grayscale image masked by a local window.

Fig.(6b) shows the pixels classification, where the white pixels are the pixels that have intensity values greater than background intensity mean (MB), so that they are classified as background. The black pixels are the pixels that have intensity values less than text intensity mean (MT) and classified as text. Brown and yellow pixels are the pixels of interest (POI), where the brown pixels are the pixels that classified as text and yellow pixels are the pixels that classified as background after the applying the proposed spectral clustering algorithm. Fig.(6c) and Fig.(6d) are the $global\ binary\ image$ and the $final\ binary\ image$, respectively. Where the yellow pixels are correctly classified as background, red pixels are correctly classified as text, black pixels are wrongly classified as background, and white pixels are wrongly classified as text. It is clear that the wrong classified pixels in $final\ binary\ image$ are less than in $global\ binary\ image$.

Fig.(7) shows the result of using the K-means algorithm in classifying the pixels of interest (POI) applied on the eigenvector corresponding to the second smallest eigenvalue ($EV2$). Fig.(8) shows the $final\ binary\ image$.

Using the (POI) as inputs for the spectral clustering algorithm not only reduces the time and complexity but also produces more accurate outputs. This specially appears in local windows with very high text contrast. In this case, using all pixels of local windows as inputs classifies the darkest pixels as text and the remaining pixels as background. But using the (POI) as inputs, the very dark pixels, which are less than the text mean, are classified as text.

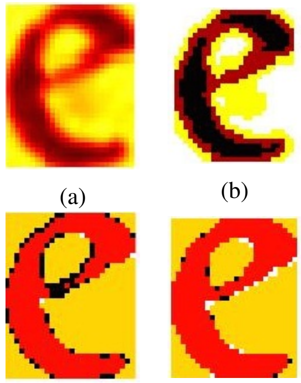
Fig.(9) shows an example for a local window whose text contrast is very high. The output of the spectral clustering algorithm when the inputs are all the local pixels is shown in Fig.(9b). When the inputs are the (POI) that is shown in Fig.(9c) the output is shown in Fig.(9d).

IV. Experimental results

In this work, we proceed experiments on images taken from Document Image Binarization Contest (DIBCO) [18].

- Evaluation measure

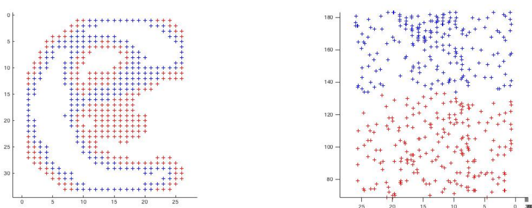
  F-measure[19] is used for performance evaluation. The



(a) (b)

(c) (d)

Fig. 6: Applying the proposed spectral clustering algorithm in a local window: (a) is the original image, (b) shows the pixels classification; white pixels are background pixels, black pixels are the text pixels, and the remaining pixels are the POI, (c) is a local of $global\ binary\ image$, and (d) is a local of $final\ binary\ image$



(a) The POI in (x,y) dimensions.

(b) The POI in (x,z) dimensions, where z is the intensity

Fig. 7: Pixels of interest classification where blue pixels are classified as text and red as background

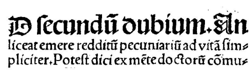


Fig. 8: The $final\ binary\ image$

F-measure is defined as follows:

$$F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

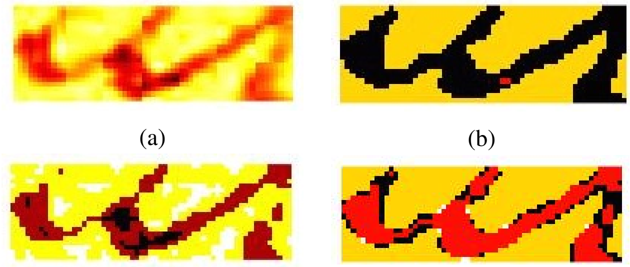
Where precision refers to background pixels that are classified as foreground, and the recall refers to foreground pixels that are classified as background. In the following, the experiments that we performed are describe and we discuss the results.

- Comparative Experiment

  The proposed method is evaluated against ten methods, Bernsen and its parameters are (window size= 31 and contrast threshold= 15), Niblack and its parameters are (window sizes=15 and k=-0.2), Nick and its parameters are (window sizes=15 and k=-0.2), Sauvola and its parameters are (window sizes=15 and k=0.3), Kapur, Otsu, Singh [9], Chiu [10], Shi [7], and Lech [8].

  Table (I) shows individual F-measure for 25 images from DIBCO. Shaded cells have the highest F-measure.

  Fig.(10) gives an example to compare between the proposed method and different binarization methods. Fig.(10a) Shows the input image. Fig.(10b) shows the output of Bernsen method. Fig.(10c) shows the output of Niblack method. Fig.(10d) shows the output of Nick method. Fig.(10e) shows the output of Sauvola method. Fig.(10f) shows the output of Kapur method. Fig.(10g) shows the output of Otsu method. Fig.(10h) shows the output of the proposed method. Bernsen method fails in case of complex background. Niblack works well in case of low contrast and thin pen stroke text but fails in case of empty windows. Nick fails in too low contrast images and thin pen stroke text. Sauvola fails in case of faint characters. Kapur fails poor contrast and variable intensity of foreground-background. Otsu fails in poor contrast, variable intensity of foreground-background and faint text. Singh method produces good results in case of low illuminated images, but fails in case of less text in the document image. Chiu method fails in low foreground or background variation. Lech fails in case of very faint text. Experimental results verify the effectiveness of our proposed method.



(a) (b)

(c) (d)

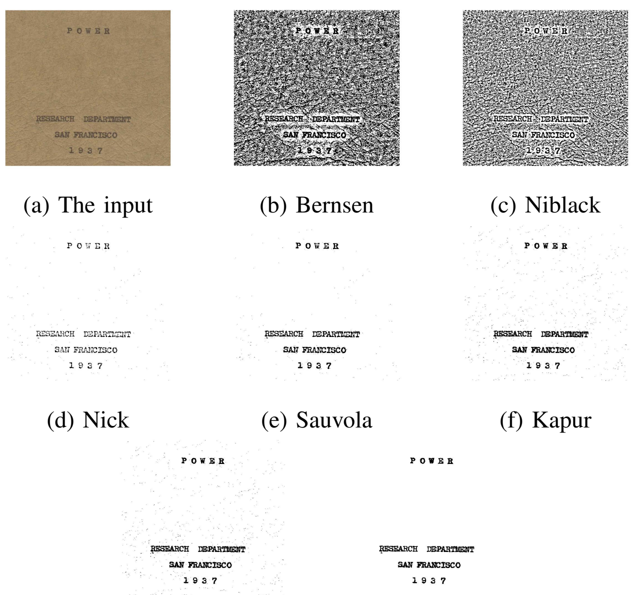
Fig. 9: An example for comparing the output when the inputs are all the pixels and the (POI): (a) is a local window whose text contrast is very high, (b) is the output when the inputs are all the local pixels, (c) shows the (POI), and (d) is the output when the inputs are the (POI).

V. CONCLUSION

In this paper, we presented a method to binarize pixels over the colored document images. Our algorithm consists of two main levels. First, a preprocessing method is applied to the input document images in order to remove the noise. Global binary image is obtained that is used in the second level for local binarization. We presented a spectral clustering algorithm for local binarization that reduces the running time and memory by approximating the similarity matrix. That method depends on pixel intensities for text and background detection.

TABLE I: Comparison between the proposed algorithm and different binarization methods

Image	Bernsen	Niblack	Nick	Sauvola	Kapur	Otsu	Singh	Chiu	Shi	Lech	Proposed
P02_09	68.8	63.5	83.4	87.7	90	96.6	-	95	95.3	-	**96.1**
H01_09	31.8	29.0	60.0	38.6	88.4	90.8	91.7	-	90.1	-	**92.6**
H01_10	38.8	37.1	6.9	35.7	89.5	91.2	81.8	91.3	-	86.4	**93.2**
H02_10	18.8	19.1	40.8	25.2	87.9	88.1	34.0	90.5	-	84.4	**91.2**
H03_10	36.1	35.0	69.6	68.2	86.3	84.6	80.1	-	-	**87.0**	85.7
H04_10	51.7	37.4	72.8	73.3	87.8	85.6	88.6	-	-	**90.1**	87.4
H06_10	28.4	26.9	67.4	66.1	82.2	80.2	76.1	-	-	**84.6**	80.9
H07_10	29.3	28.5	84.4	83.4	88.3	90.1	90.7	-	-	-	**91.6**
H09_10	39.9	20.5	66.5	55.7	**88.1**	81.0	80	-	-	82.2	84.9
H10_10	25.4	24.8	61.0	51.1	81.4	79.2	68.4	-	-	**86.1**	79.8
HW2_11	19.8	20.4	84.1	82.3	92.	88.9	90.4	-	-	-	**92**
PR7_11	9.4	9.9	71.3	86.9	86.4	86.4	59.6	-	-	-	**91.3**
PR8_11	57.4	50.9	71.7	76.8	87.4	82.2	79.6	-	-	-	**83.6**
H03_12	25.1	24.6	19.3	17.0	86.3	89.5	-	-	-	-	**89.5**
H04_12	29.4	21.4	87.9	88.8	82.9	89.4	-	-	-	-	**90.8**
H07_12	49.7	29.6	67.9	62.8	82.3	82.7	-	-	-	-	**84**
H12_12	58.1	27.0	47.1	35.9	88.6	88.3	-	-	-	-	**88.7**
HW1_13	48.0	17.4	67.0	55.1	**90.4**	83.0	-	-	-	-	88
HW3_13	33.2	29.8	62.2	59.0	79.1	74.8	-	-	-	-	**79.5**
PR1_13	22.0	22.1	78.3	83.2	84.5	87.7	-	-	-	-	**89.1**
PR7_13	72.9	67.9	83.8	87.8	92.7	93.5	-	-	-	-	**92.8**
H01_14	51.8	26.2	75.3	68.9	91.1	89.1	-	-	-	-	**92**
H02_14	55.7	35.5	81.6	82.1	**89**	86.3	-	-	-	-	87.7
H03_14	48.0	38.5	80.4	82.7	91.4	97.4	-	-	-	-	**97.6**
H04_14	61.3	54.6	64.4	71.2	93.3	94.2	-	-	-	-	**94.6**
H05_14	60.3	55.5	67.0	69.0	93.3	93.4	-	-	-	-	**94.5**



(a) The input (b) Bernsen (c) Niblack

(d) Nick (e) Sauvola (f) Kapur

(g) Otsu (h) The proposed

Fig. 10: Comparison between the proposed algorithm and different binarization methods

REFERENCES

[1] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[2] J. N. Kapur, P. K. Sahoo, and A. K. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer vision, graphics, and image processing*, vol. 29, no. 3, pp. 273–285, 1985.

[3] W. Niblack, *An introduction to digital image processing*. Strandberg Publishing Company, 1985.

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

[5] K. Khurshid, I. Siddiqi, C. Faure, and N. Vincent, "Comparison of niblack inspired binarization methods for ancient documents.," *DRR*, vol. 7247, pp. 1–10, 2009.

[6] J. Bernse, "Dynamic thresholding of grey-level images," in *Proc. 8th Int. Conf. on Pattern Recognition, 1986*, pp. 1251–1255, 1986.

[7] J. Shi, N. Ray, and H. Zhang, "Shape based local thresholding for binarization of document images," *Pattern Recognition Letters*, vol. 33, no. 1, pp. 24–32, 2012.

[8] P. Lech, K. Okarma, and D. Wojnar, "Binarization of document images using the modified local-global otsu and kapur algorithms," *Przeglkad Elektrotechniczny*, vol. 91, no. 2, pp. 71–74, 2015.

[9] B. M. Singh, R. Sharma, D. Ghosh, and A. Mittal, "Adaptive binarization of severely degraded and non-uniformly illuminated documents," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 17, no. 4, pp. 393–412, 2014.

[10] Y.-H. Chiu, K.-L. Chung, W.-N. Yang, Y.-H. Huang, and C.-H. Liao, "Parameter-free based two-stage method for binarizing degraded document images," *Pattern Recognition*, vol. 45, no. 12, pp. 4250–4262, 2012.

[11] X. Chen, W. Hong, F. Nie, D. He, M. Yang, and J. Z. Huang, "Spectral clustering of large-scale data by directly solving normalized cut," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1206–1215, ACM, 2018.

[12] L. Wu, P.-Y. Chen, I. E.-H. Yen, F. Xu, Y. Xia, and C. Aggarwal, "Scalable spectral clustering using random binning features," *arXiv preprint arXiv:1805.11048*, 2018.

[13] U. von Luxburg, "A tutorial on spectral clustering,"

[14] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[15] T. Hastie, R. Tibshirani, and J. Friedman, "Unsupervised learning," in *The elements of statistical learning*, pp. 485–585, Springer, 2009.

[16] A. Challa, S. Danda, B. D. Sagar, and L. Najman, "Power spectral clustering," 2018.

[17] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[18] K. Ntirogiannis, B. Gatos, and I. Pratikakis, "Icfhr2014 competition on handwritten document image binarization (h-dibco 2014)," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pp. 809–813, IEEE, 2014.

[19] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, *et al.*, "Findings of the 2014 workshop on statistical machine translation," in *Proceedings of the ninth workshop on statistical machine translation*, pp. 12–58, 2014.