

An Overview of Different Binary Methods for Documents Based on Their Features

Fatemeh Fazeli, Abdolhossein Sarrafzadeh, and Jamshid Shanbehzadeh

Abstract—this paper surveys binarization of document images. The main role of binarization is dimension and noise reduction. Binarization is one of the most important steps in preprocessing of document image understanding and compression. Image binarization means to classify image pixels into two classes, background and foreground. The input of this classification is a feature vector based on intensity values of image pixels. The new features are extracted from the first input vector and, according to the extracted features a cost function as a classifier is constructed. The intensity value that maximizes the cost function is considered as the boundary line of two classes. This paper divides the binarization algorithms into three groups. The first considers one input feature vector including intensity values of each pixel. The second one considers an input feature vector for each pixel based on the intensity value of the pixel and its neighbors. The third group is based on a combination of the first and second group of schemes.

Index Terms—Binarization, thresholding, feature extraction, cost function, image preprocessing, image compression

I. INTRODUCTION

THERE exist surveys for binarization [1] [2]. This paper reviews document image binarization based on image features with a new paradigm. Binarization is classification of image pixels based on their attributes. The input of binarization is a feature vector containing the intensity of image pixels. Depending on the selected method, new features containing the best properties of the image are extracted from the input vector. Thus, the feature space is changed from the intensity values of each pixel to the new extracted features and, the cost function is created based on the new feature space. The output of binarization is the intensity value that maximizes the cost function and it is called threshold value. After determining the thresholds, image pixels are classified into two classes.

Binarization algorithms can be divided into three groups. The first one employs a feature vector for the entire image that includes the intensity of each pixel. Section II studies this group of algorithms. The second group of algorithms has a feature vector for each pixel of image that contains the

intensity values of the pixel and its neighbors. Section III discusses these algorithms. The third class of algorithms has both a general feature vector and adaptive ones. Section IV explores the third group.

II. ALGORITHM WITH AN INPUT FEATURE VECTOR FOR THE ENTIRE IMAGE

The initial feature vector of these algorithms contains intensity values of every image pixel. Feature extraction is performed on the input vector and the threshold is calculated by maximizing the cost function. The classifying criterion for these algorithms is the type of extracted features. The characteristics are all very different and lead to different evaluation functions. These algorithms are described in the following five categories.

A. Feature extraction based on histogram

In this group, feature extraction process is based on histograms. Below we describe some of these techniques.

The Prewitt and Mendelson method which has been explained in [3], focused on images in which object and background pixels are separable. The extracted features correspond to the location of the peaks. The deepest point intensity of the valley between the two peaks is introduced as the threshold value. If the condition is not true, finding the threshold value from the histogram is difficult because the histogram does not separate into two peaks and the valley may be a flat area. This method is also called the mode method.

The next method is Rosenfeld [4]. In this method a convex hull is the extracted feature and is used for finding the concavity of the histogram. Suppose that HS is the histogram of the gray levels, g_0, g_1, \dots, g_{l-1} . The height of histogram at each level, $(h(g_0), h(g_1), \dots, h(g_{l-1}))$, is defined as:

$$h(g_i) \neq 0 \quad \text{for all } i \quad (1)$$

Convex hulls (\overline{HS}) are used to characterize the concavity of the histogram. The convex hull is the smallest and nearest convex polygon that surrounds HS . Every two arbitrary points of \overline{HS} can be connected with a straight line and the farthest point from the line is the threshold. To calculate the histogram's concavity, use $\overline{HS} - HS$ as a cost function. The possible threshold values are the levels at which their $\overline{h(g_i)} - h(g_i)$ value has local maxima.

All the maximums are inappropriate coordinates for thresholds because the large concavity may be due to noise. Rosenfeld called spurious concavities to these points.

Manuscript received 4th of Feb 2013.

Fatemeh Fazeli is an M.Sc. student with the Department of Computer Engineering, Faculty of Engineering, Kharazmi University, Tehran, I.R. Iran (e-mail: ff6739@gmail.com).

Abdolhossein Sarrafzadeh is an Associate Professor and Head of Department of Computing, Unitec Institute of Technology, New Zealand (email:hsarrafzadeh@unitec.ac.nz).

Jamshid Shanbehzadeh is an Associate Professor with the Department of Computer Engineering, Faculty of Engineering, Kharazmi University (Tarbiat Moallem University of Tehran), Tehran, I.R. Iran (phone: +98 26 34550002; fax: +98 26 34569555; e-mail: jamshid@tmu.ac.ir).

Equation (2) is a balance measurement for removing spurious concavities:

$$E_i = \left\{ \sum_{j=g_0}^{g_{i-1}} h(j) \right\} \left\{ \sum_{j=g_i}^{g_{i-1}} h(j) \right\} \quad (2)$$

E_i measures the histogram balancing of level g_i . The balance measures the total amount of height of histogram levels, before and after the candidate threshold. Then it considers the other gray levels around the candidate points and decides which points are maximum spurious and should be removed. The method's criterion is the distance of each histogram point from the convex hull. The other algorithms also fit into this category, for example, Ramesh [5] and Sezán [6].

B. Feature extraction based on clustering

This group considers the image pixels as two classes. New features are extracted for each class and, accordingly, the cost function is calculated. Some of the algorithms are reviewed.

Otsu introduces a method to calculate the goodness of each threshold in [7]. Otsu's algorithm considers L as the number of gray levels, n_i as the number of pixels of each level, and N as the total number of pixels:

$$N = n_1 + n_2 + \dots + n_L \quad (3)$$

Equation (4) shows the probability distribution of intensity i :

$$p_i = n_i/N, \quad p_i \geq 0, \quad \sum_{i=1}^L p_i = 1. \quad (4)$$

Assume that all pixels are divided into two classes, C_0 and C_1 (background and object), and the threshold is the k^{th} level. As new features, they extract probability, mean, and variance of each class and create the following cost functions:

$$\lambda = \sigma_B^2 / \sigma_W^2, \quad \kappa = \sigma_T^2 / \sigma_W^2, \quad \eta = \sigma_B^2 / \sigma_T^2 \quad (5)$$

Where, σ_W^2 is the within-class variance, σ_B^2 is the between-class variance, and σ_T^2 is total variance of levels. Based on the relation between cost functions, it is sufficient to calculate the value of k which maximizes η . This method is non-parametric and unsupervised, and because it selects the optimal threshold value simply, automatically and stable, it has high generalization and can be used in other types of classification.

The other method is by Kittler and Illingworth [8]. This method considers the gray level histogram as an estimate of probability density function of the object and background pixel levels, and it is shown by $p(g)$. The extracted features are the mean, standard deviations, and probability density functions of each class, where $p(g | i)$ is distributed with the average μ_i , standard deviation σ_i , and priori probability p_i . The cost function is created as follows:

$$J(t) = 1 + 2\{P_1(t) \log_e \sigma_1(t) + P_2(t) \log_e \sigma_2(t)\} - 2\{P_1(t) \log_e P_1 + P_2(t) \log_e P_2(t)\} \quad (6)$$

The optimal threshold is the value that minimizes $J(t)$.

$$t^* = \text{ArgMin } J(t), t \in G \quad (7)$$

This method calculates the new features based on the probability density functions, and the binary problem changes into minimizing the error of the Gaussian density function problem.

The other image binarization method is an optimization-based approach [9]. This method uses the sum of squared errors in data clustering. Suppose L is the number of gray levels, n_l is the number of k^{th} level pixels, and the image histogram is divided into two clusters s_1 and s_2 . The centers of gravity of each cluster are calculated as new features by using (8) and (9):

$$m_i = \frac{1}{a_i} \sum_{l \in S_i} n_l l, \quad i = 1, 2 \quad (8)$$

$$d_i = \sum_{l \in S_i} n_l, \quad i = 1, 2 \quad (9)$$

Then, the sum of the squared errors function is calculated as the cost function by using (10):

$$f(m_1, m_2) = \sum_{i=1}^2 \sum_{l \in S_i} n_l (l - m_i)^2 \quad (10)$$

The problem is to find s_1 and s_2 to minimize function f and this is performed iteratively to minimize function f . This algorithm is, essentially, similar to Otsu's method but it has less complexity. For example, with two clusters, the complexity of Otsu's algorithm is $O(L^2)$ and the complexity of this algorithm is $O(LQ)$, where Q is the number of iteration.

The other algorithm in this group is complicated image binarization based on the method of maximum variance [10]. Suppose g_1, r_1 are the intensity value and pixel ratio in the character area, and g_2, r_2 are the intensity value and pixel ratio in the background area ($0 \leq g_1, g_2 \leq 255, 0 \leq r_1, r_2 \leq 1, r_1 + r_2 = 1$).

First, the mean (M) and variance (C^2) of intensities are calculated as new features by using (11) and (12):

$$M = r_1 \times g_1 + r_2 \times g_2, \quad g_1 < M < g_2 \quad (11)$$

$$C^2 = r_1(g_1 - M)^2 + r_2(g_2 - M)^2 \quad (12)$$

If the character is white and background black, the threshold value is T_1 and, otherwise, the threshold value is T_2 .

$$T_1 = M - \sqrt{\frac{r_1}{r_2}} C, \quad T_2 = M + \sqrt{\frac{r_1}{r_2}} C \quad (13)$$

Equation (14) shows the cost function.

$$S\{k\} = \sigma_B^2(k) / \sigma_A^2(k), \quad k \in [T_0 - 10, T_0 + 10] \quad (14)$$

Where $S\{k\}$ is the cost function, T_0 is the probable threshold, σ_A^2 is the within-class variance and σ_B^2 is the between-class variance. The k value that maximizes the cost

function is suitable for the threshold. The other algorithms in this category are Lloyd [11] and Yanni [12].

C. Feature extraction based on entropy

This group extracts features based on entropy. Pun [13] is one of these algorithms. This method supposes f_1, f_2, \dots, f_n as the frequency of gray levels, and p_i as the probability of each gray level by using (15):

$$p_i = \frac{f_i}{N}, \quad \sum_{i=1}^n f_i = N, \quad i = 1, 2, \dots, n \quad (15)$$

Where N is the number of all pixels and n is the number of gray levels. Equation (16) calculates the posteriori entropy as a new feature.

$$H_n = -P_s \ln P_s - (1 - P_s) \ln(1 - P_s) \quad (16)$$

$$P_s = \sum_{i=1}^s P_i, \quad 1 - P_s = \sum_{i=s+1}^n P_i \quad (17)$$

Equation (18) defines the entropy:

$$H_s = -\sum_{i=1}^s p_i \ln p_i, \quad H_n = -\sum_{i=1}^n p_i \ln p_i \quad (18)$$

Then the cost function is created based on new features by using (19):

$$f(s) = \frac{H_s}{H_n} \frac{\ln p_s}{\ln[\max(p_1, p_2, \dots, p_s)]} + \left(1 - \frac{H_s}{H_n}\right) \frac{\ln(1-p_s)}{\ln[\max(p_{s+1}, p_{s+2}, \dots, p_n)]}, \quad (19)$$

$$\frac{H_s}{H_n} \leq f(s) \quad (20)$$

The s value that maximizes function f is introduced as the threshold. This method uses the entropy as a new feature, which represents the irregularity rate of intensity values of image pixels. The difference between this method and other methods in this group is the use of posteriori entropy. In other words, consider probability p for some pixels and $1 - p$ for others.

The other algorithm is Kapur [14]. This method, first, supposes p_1, p_2, \dots, p_n as the probability distribution of gray levels. Consider two probability distributions, A for discrete values of 1 to s , and B for values of $s + 1$ to n .

$$A: \frac{p_1}{p_s}, \frac{p_2}{p_s}, \dots, \frac{p_s}{p_s} \quad B: \frac{p_{s+1}}{1-p_s}, \frac{p_{s+2}}{1-p_s}, \dots, \frac{p_n}{1-p_s} \quad (21)$$

The entropy value of each distribution is calculated based on the extracted features by using (22) and (23):

$$H(A) = \ln P_s + \frac{H_s}{P_s} \quad (22)$$

$$H(B) = \ln(1 - P_s) + \frac{H_n - H_s}{1 - P_s} \quad (23)$$

Next, the cost function $\psi(s)$ that contains the sum of entropy values is calculated by using (24):

$$\Psi(s) = \ln P_s (1 - P_s) + \frac{H_s}{P_s} + \frac{H_n - H_s}{1 - P_s} \quad (24)$$

The purpose of this algorithm is to find the value of s as a threshold by maximizing function $\psi(s)$. This method also employs the entropy of extracted features but, it utilizes the probability distribution to show the probability of pixels of each gray level. The result is better than Pun's method that considers only two probability values of gray levels. Another algorithm in this group is Yen [15].

D. Feature extraction based on similar features between the original and binary image

This group extracts similar features between the binary image and the original one. In other words, the extracted features before and after binarization are similar.

Tsai [16] proposed one of these algorithms. This method extracts the first three moments of gray level image as new features by using (25):

$$m_i = (1/n) \sum_j n_j (z_j)^i = \sum_j p_j (z_j)^i \quad (25)$$

$$p_j = \frac{n_j}{n} \quad (26)$$

Where m_i is the i^{th} moment of image, n_j is the number of pixels with intensity value z_j , n is the number of all image pixels, and p_j is the probability of each intensity. The higher-order moments contain useful information of image, and, using them, we can retrieve the image with higher quality and lower error. This method attributes the value z_0 to all pixels with intensity less than the threshold value, and z_1 to all pixels with intensity greater than the threshold value. Equations (27-30) are considered as the cost function:

$$p_0 z_0^0 + p_1 z_1^0 = m_0 \quad (27)$$

$$p_0 z_0^1 + p_1 z_1^1 = m_1 \quad (28)$$

$$p_0 z_0^2 + p_1 z_1^2 = m_2 \quad (29)$$

$$p_0 z_0^3 + p_1 z_1^3 = m_3 \quad (30)$$

Where $m_0 = 1$, and $p_0 + p_1 = 1$. After calculating the values of p_0 and p_1 by solving the last equations, we find the value of t as the threshold by using (31):

$$p_0 = (1/n) \sum_{z_j \leq t} n_j \quad (31)$$

In other words, we look for the best threshold value based on the cost function, where there is no change in moment values of image after binarization. This method can retrieve an ideal image from its blurred version and finds several threshold values without search. The other algorithms of this group are Huang [17], Hertz [18], Ogorman [19] and Pikaz [20].

E. Feature Extraction Based on Spatial Features

This group extracts features that describe the location of pixels. This location can be obtained by co-occurrence probability and entropy, such as Pal's [21] method.

III. ALGORITHMS WITH AN INPUT FEATURE VECTOR FOR EVERY PIXEL IN THE IMAGE

This class of algorithms considers one neighborhood window for each pixel. The size of this window affects the output. The number of input vectors is equal to the number of image pixels, and the size of each input vector depends on the neighborhood window size. Feature extraction is applied on each input vector, and a new feature space is produced for each pixel. The cost function is calculated based on new features. The type of features and the cost function formula are the same for all pixels. For each neighborhood window, the intensity value that maximizes the cost function is considered as the threshold value. Some methods such as Parker [22] and Sauvola [23], have added a feature extraction step as preprocessing before binarization.

Niblack [24] is one of these algorithms. This method, first, sets a neighborhood window appropriately. This window should be small enough to keep detailed information and large enough to be less vulnerable to noise. The new features that are extracted for each of the areas are the mean and standard deviations of intensity values. The cost function is calculated by using (32):

$$T(x, y) = m(x, y) + k * s(x, y) \quad (32)$$

Where $m(x, y)$ represents the average, and $s(x, y)$ represents the standard deviation value of the window. K is a parameter of the boundary of object. The output of this process is a threshold value for each region and is applied to the related pixel. Zhang and Tan [25] improved the Niblack method. Equation (33) shows the cost function:

$$T(x, y) = m(x, y) * \left[1 + k * \left[1 - \frac{s(x, y)}{R} \right] \right] \quad (33)$$

K and R are empirical constant parameters to reduce the sensitivity of the Niblack method to noise.

Bernsen [26] is the other method in this group. This method extracts the maximum and minimum intensity values of each window as new features. The cost function is created by the new features for each window, according to (34):

$$T(i, j) = 0.5[I_{high}(i, j) + I_{low}(i, j)] \quad (34)$$

Where I_{high} and I_{low} are maximum and minimum intensity values of window pixels. If the value of this function for any pixel is greater than a threshold value, it will be considered as foreground, otherwise, the pixel will be background. One drawback of this method is determining the size of the neighborhood window. The size of the neighborhood window must be the same as the object size for correct image segmentation. If the size of the object is unknown, however, or the object is very large, the neighborhood window size cannot be set correctly and the segmentation will fail.

This method extracts the features quickly, because there is no need to estimate the histogram. Nevertheless, there are two disadvantages associated with large window size. First, it will increase the computational cost to find the maximum and minimum values. Second, small objects with low

resolution will be considered as background, especially in cases where they are near to the large size or high resolution objects.

Sauvola [23] is the other method in this group. This method, first, performs a preprocessing step to divide image pixels into two classes, background and text. Then two methods are introduced to select the best threshold value of each pixel. One of these methods is for the background pixels, and the other is for the text pixels. The size of the neighborhood window is 10-20 pixels, and the extracted features are the mean value of intensities and the transient difference for each window. The transient difference has three possible values, uniform, near uniform, and differing. The uniform and near uniform values are for the background area, and the differing values are for the text area. Based on the extracted features, we decide whether the pixel is text or background and a suitable method is applied on it.

IV. ALGORITHMS WITH TWO TYPES OF INPUTS

The third group of binarization algorithms consists of two steps. The first step considers a feature vector consisting of the intensity values of all image pixels as input. Then the new features for the entire image are extracted. The second step considers an input vector for each pixel by the use of neighborhood windows that contain the intensity values of the pixel and neighbors. Feature extraction is performed for each of the feature vectors individually. In other words for each pixel, the new features that have the same types but different values are extracted.

This group of algorithms considers the relation of all pixels together, and also the relation between neighboring pixels. In other words, the resulting feature space contains more and better information about pixels and introduces an efficient threshold value.

This paper divides the third group of binary algorithms into three classes. In the first class, the extracted features from step one are added to the feature space of each pixel. Then, the extracted features for each pixel in second step are added individually to the feature space of the corresponding pixel. Then, the cost function and the threshold value are calculated for each pixel. The result of this process is a binary image.

In the second class, after extracting the new features in the first step, the cost function is created and a threshold value is calculated for all the image pixels. Next, we consider the extracted features in the second step and a threshold value based on the characteristics of the neighborhood windows is generated for each pixel. Last, for each point, it is decided which of the two thresholds is suitable for binarization.

In the third class, first, based on one of the feature extraction strategies (for the entire image or for each pixel), the new features are extracted, and, after calculating the cost function, the threshold value is found and is applied to the image. Next, the resulting binary image from the previous section is considered as an input image, and the type of feature extraction that was not used before is performed and the cost function is calculated. The threshold value is then found and applied to the pixels. In other words, in this class we perform binarization twice.

The adaptive thresholding technique for the document image analysis method in [27] is one of the first class of algorithms. In the first step, the mean and standard deviation values are extracted as new features for the whole image, and are added to the new feature space of each pixel. In the second step, the mean and the standard deviation values are calculated for each neighborhood window and are added to the feature space of the corresponding pixel. Equation (37) shows the cost function based on the new feature space for each pixel:

$$k = -0.3 * \frac{(m_g(i,j)*\sigma_g(i,j)-m_l(i,j)*\sigma_l(i,j))}{\max((m_g(i,j)*\sigma_g(i,j)-m_l(i,j)*\sigma_l(i,j)))} \quad (37)$$

Where m is the mean value, σ is the standard deviation value, the index l represents the extracted features from the feature vector corresponding to each pixel, and the index g represents the extracted features from the input vector for the whole image. The coefficient -0.3 holds the k value in range $[-0.3, 0.3]$ and reduces the influence of the standard deviation in the Niblack formula. After calculating the cost function for each pixel, the threshold value is found for each pixel by using the Niblack formula.

A novel image binarization method using hybrid thresholding in [28] is one of the algorithms of second class of this group. The first step considers a vector that contains the intensity value of all image pixels as input feature vectors. Then, as for the Otsu method, the mean, weight, and variance values are extracted as new features and the threshold value is estimated. In the next step, a neighborhood window is considered appropriately and the mean value of each window is extracted as a new feature for each pixel. Then, by the combination of the Sauvola and Niblack methods, the threshold value is computed.

Valizadeh et al. [29] proposed a novel binarization algorithm in the third class of this group. This method performs a preprocessing step and divides pixels into three classes: text, background, and their combination. The feature extraction process is performed on the input vector of each pixel and extracts the mean value of each window. Then, based on the features of the classes, Valizadeh et al. introduce a transformation function as the cost function for mapping pixels into the range of gray levels. This function is sigmoid and it has a large slope in the range of gray levels. Equation (38) shows the cost function:

$$y(i,j) = \frac{1}{1 + \exp\left(\frac{I(i,j) - M(i,j)}{\alpha * M(i,j) + k}\right)} \quad (38)$$

Where $y(i,j)$ is the intensity of the pixels of the transformed image, $M(i,j)$ is the mean intensity of each window, $I(i,j)$ is the intensity value of pixel (i,j) , α and k are the parameters set to 0.5 and 10 for an image with 256 gray levels, respectively. The threshold for each pixel is applied and the image is binarized. In the next step, the resulting binary image is considered as the original image and the next type of inputs (a feature vector for the entire image) are used. A threshold value is calculated for all image pixels by using the Otsu method.

REFERENCES

- [1] P.K. Sahoo, S. Soltani, A.K.C. Wong, Y. Chen., A Survey of Thresholding Techniques, *Computer Graphics and Image Process*, 41, 233-260, 1988.
- [2] B.Sankur, M.Sezgin, Survey Over Image Thresholding Techniques and Quantitative Performance Evaluation, *Journal of Electronic Imaging*, 13(1), 146-165, 2004.
- [3] J.M.S. Prewitt and M.L. Mendelsohn, The Analysis of Cell Images, in *Ann. New York Acad. Sci. Vol. 128, 1035-1053*, New York Acad. Sci., New York, 1966.
- [4] A. Rosenfeld, P. De la Torre, Histogram Concavity Analysis as an Aid in Threshold Selection, *IEEE Trans System, Man and Cybernetics*, SMC-13 (1983) 231-235.
- [5] N. Ramesh, J.H. Yoo, I.K. Sethi, Thresholding Based on Histogram Approximation, *IEE Proc. Vis. Image, Signal Proc.*, 142(5) (1995) 271-279.
- [6] M.I. Sezan, A Peak Detection Algorithm and its Application to Histogram-Based Image Data Reduction, *Graphical Models and Image Processing*, 29 (1985) 47-59.
- [7] N. Otsu, A Threshold Selection Method from Gray-level Histograms, *IEEE Trans. Systems Man Cybernet*, 9 (1), 62-66, 1979.
- [8] J. Kittler and J. Illingworth, On Threshold Selection using Clustering Criterion, *IEEE Trans. Systems Man Cybernet*, SMC-15, 1985, 652-655.
- [9] Liju Dong, Ge Yu, An Optimization-Based Approach to Image Binarization, *IEEE*, 2004.
- [10] Jie Bai, Yao-Quan Yang, Rui-Li Tian, Complicated Image's Binarization Based on Method of Maximum Variance, *IEEE*, 2006.
- [11] D.E. Lloyd, Automatic Target Classification using Moment Invariant of Image Shapes, Technical Report, RAE IDN AW126, Farnborough-UK, December 1985.
- [12] M.K. Yanni, E. Horne, A New Approach to Dynamic Thresholding, *EUSIPCO-9: Eurpean Conference on Signal Processing*, Vol. 1, Edinburgh, 1994, 34-44.
- [13] T. Pun, A New Method for Gray-level Picture Thresholding using the Entropy of the Histogram, *Signal Processing*, 2, 223-237, 1980.
- [14] N. Kapur, P.K. Sahoo, A.K. Wong, A New Method for Gray-level Picture Thresholding using the Entropy of the Histogram, *Computer Vision Graphics and Image Processing*, 29,273-285, 1985.
- [15] J.C. Yen, F.J. Chang, S. Chang, A New Criterion for Automatic Multilevel Thresholding, *IEEE Trans. on Image Processing*, IP-4 (1995) 370-378.
- [16] W.H. Tsai, Moment-preserving Thresholding: A New Approach, *Graph. Models Image Process*, vol. 19, 377-379, 1985.
- [17] L.K. Huang, M.J. Wang, Image Thresholding by Minimizing the Measures of Fuzziness, *Pattern Recognition* 28(1), 41-51, 1995.
- [18] L. Hertz, R.W. Schafer, Multilevel Thresholding using Edge Matching, *Computer Vision Graphics and Image Processing*, 44 (1988) 279-295.
- [19] L. O'Gorman, Binarization and Multithresholding of Document Images using Connectivity, *Graphical Models and Image Processing*, 56 (1994) 494-506.
- [20] A. Pikaz, A. Averbuch., Digital Image Thresholding Based on Topological Stable State, *Pattern Recognition*, 29 (1996) 829-843.
- [21] N.R. Pal, S.K. Pal, Entropic Thresholding, *Signal Processing*, 16 (1989) 97-108.
- [22] J.R. Parker, Gray level Thresholding in Badly Illuminated Images, *IEEE PAMI*. 13 (8), 813-819, 1991.
- [23] J. Sauvola, M. Pietikainen, Adaptive Document Image Binarization, *Pattern Recognition* 33, 225-236, 2000.
- [24] W. Niblack, An Introduction to Digital Image Processing, Prentice Hall, Eaglewood Cliffs, 115-116, 1986.
- [25] Z. Zhang, C. L. Tan, Restoration of Images Scanned from Thick Bound Documents, *Proceedings of International Conference on Image Processing*, vol. 1, 1074-1077, 2001.
- [26] J. Bernsen, Dynamic Thresholding of Grey-level Images, *Proceedings of 8th ICPR*, 1251-1255, 1986.
- [27] Naveed Bin Rais, M. Shehzad Hanif, Imtiaz A. Taj, Adaptive Thresholding Technique for Document Image Analysis, *IEEE*, 2004.
- [28] Tien-Ying Kuo, Yun-Yang Lai, Yi-Chung Lo, A Novel Image Binarization Method using Hybrid Thresholding, *IEEE*, 2010.
- [29] M.Valizadeh, N.Armanfard, M.Komeili, E. Kabir, A Novel Hybrid Algorithm for Binarization of Badly Illuminated Document Images, *IEEE*, 2009.