Pre-processing Techniques for Hungarian Handwriting Recognition

Gaye Ediboğlu Bartos*, Éva Hajnal**
* Anadolu University/Department of Computer Engineering, Eskişehir, Turkey
* Bilecik Şeyh Edebali University/Department of Computer Engineering, Bilecik, Turkey
** Óbuda University Alba Regia Technical Faculty, Székesfehérvár, Hungary
gayeediboglu@anadolu.edu.tr, gaye.ediboglu@bilecik.edu.tr
hajnal.eva@amk.uni-obuda.hu

Abstract— **The technology for offline handwriting recognition has reached a very successful level for Latin character recognition for discretely written characters. However, there are still issues for recognising cursive handwritings. In addition to that, there are only a few studies conducted for Hungarian handwriting recognition. Recognition of Hungarian handwritten texts is challenging for various reasons such as the tradition of using cursive handwriting and letters with punctuation. It is known that pre-processing phase is essential for a successful recognition and the outcomes of the pre-processing phase determine the accuracy of the recognition. This paper provides the most suitable pre-processing techniques for offline Hungarian handwriting recognition. All the methods applied are provided with the results and the most suitable methods are highlighted.**

I. Introduction

Optical Character Recognition (OCR) is conversion of scanned images of machine printed or handwritten text, numerals, letters and symbols into a computer processable format such as ASCII without any human intervention. There are two types of OCR namely online and offline recognition. In online recognition, the characters are recognised as they are drawn. Furthermore, the order of strokes are available and successive points are represented as a function of time [1][2]. On the other hand, in offline recognition optical recognition is performed after the writing or printing has been completed. In other words, its input is an image or a scanned document [3].

An OCR system consists of several components. Fig. 1 shows the components in a typical OCR system. As can be seen from the Fig. 1, firstly the document is scanned through an optical scanner. Secondly the crucial pre-processing phase is applied. Pre-processing is critical for an OCR system since the outcomes of this step are going to be recognized in the next step. Generally in the pre-processing phase binarization, noise removal, normalization, feature extraction and segmentation are performed. Finally in classification step, the recognition is performed. In addition to those steps, an extra post-processing phase could be adopted in which verification is performed in order to improve the accuracy rate.

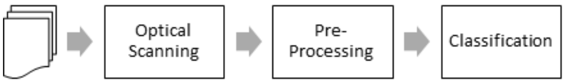


Figure 1 Components of a typical OCR system

This paper presents pre-processing techniques that applies to offline Hungarian handwriting recognition. Hungarian handwriting recognition is challenging for various reasons. The next section provides the properties of Hungarian handwriting with reference to why it is challenging. In the following sections pre-processing techniques that have been tested are explained and the results are provided. Finally the conclusion is presented in the final section.

A. Properties of Hungarian Handwriting

Hungarian language consists of 44 letters (Fig. 1). Some Hungarian letters are the same as English letters however other letters have punctuation and some consist of more than one character. These letters of the language generate a challenge for recognition purpose such as avoid removal of the punctuation at the noise removal phase.

Another challenge in recognising Hungarian handwriting is that in Hungary there is a tradition of using cursive scripts. Cursive character of the handwriting brings about the challenge to the segmentation phase. Tappert defines five difficulty levels for handwritten characters which are namely boxed discrete characters, spaced discrete characters, run-on (touching or overlapping) discretely written characters, pure cursive handwriting, and mixed cursive and discrete characters (Fig. 3) [4]. In our data sample, most of the handwritings belong to the most difficult categories which are pure cursive handwriting, and mixed cursive and discrete characters. Therefore, Hungarian handwriting recognition is very challenging compare to other Latin handwritings. Additionally, there are not many studies conducted for the purpose of Hungarian handwriting recognition.

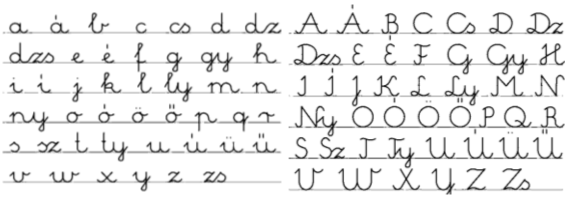


Figure 2 Hungarian Alphabet

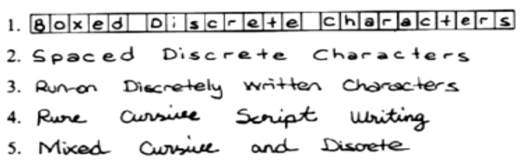

1. Boxed Discrete Characters
2. Spaced Discrete Characters
3. Run-on Discretely Written Characters
4. Pure Cursive Script Writing
5. Mixed Cursive and Discrete

Figure 3 Five stages of handwritten word recognition problem

II. PRE-PROCESSING

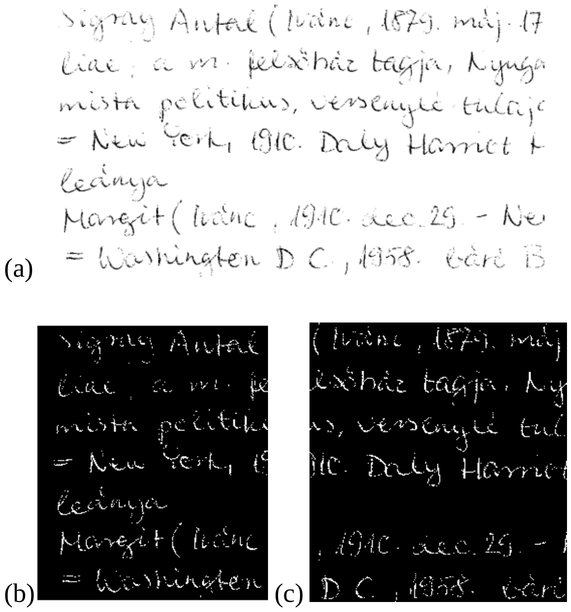
Raw data must be pre-processed in order to make it more suitable and useful. This process establishes the path between handwriting and recognition. The pre-processing for offline handwriting recognition consists of many steps which usually include binarization, noise removal, normalisation, skew correction, slant removal and segmentation. According to different needs, different steps may be included or excluded.

Our data sample was created by scanning handwritten papers in 300 dpi and storing them in grayscale image in png format. Then below mentioned pre-processing methods were applied respectively in the MATLAB 8.3 environment.

A. *Binarization (Thresholding)*

Image binarization is used in order to convert a gray-scale handwritten image into a binary image. This process also helps to reduce the size of the image and therefore speed the image processing up. There are two types of thresholds namely global and local (adaptive) thresholds [5]. In global thresholding, one threshold is used for the entire document. There are several techniques in the literature for picking the optimal threshold for example Otsu’s thresholding technique is amongst the most popular ones [6]. On the other hand, local thresholding uses different threshold values for each pixel according to the local area information [7].

For our work we tried two different global thresholding methods. Firstly, the threshold value was given manually by looking at the histogram of the image. Then, Otsu’s Global Thresholding was adopted [6]. As can be seen from the Fig. 4, there was not a visible difference in the results and the results were both good therefore no more methods were applied. It is worth pointing out that, no prior normalization was applied to the image. However, after applying the same noise removal techniques, it became visible that Otsu’s thresholding gave better results. Therefore, Otsu’s thresholding was adopted for the later processes.



Sigray Antal (Ivánc, 1879. máj. 17
liac; a m. felsőház tagja, Nyuga
mista politikus, versenyistálló tulajd
= New York, 1910. Daly Harriet H
leánya
Margit (Ivánc, 1910. dec. 25. – Ne
= Washington D C., 1958. báró B

Figure 4 Global Thresholding Applied to the Image: (a) original image, (b) manual thresholding, (c) Otsu's Thresholding

B. *Skew Correction and Slant Removal*

Skew correction is used when the document needs to be aligned with the coordinate system. Inaccuracies in the scanning process and the style of handwriting may result in tilted or curved lines within the image. Most popular approaches for skew correction includes correlation, projection and Hough transform [8], [9].

The slant of handwritten texts varies from person to person. Several methods are proposed for slant removal purpose. Examples of popular techniques are calculation of the average angle of near-vertical elements and Bozinovic- Shriari Method (BSM) [10], [11].

The data used for the study did not include a significant amount of skewed lines or slanted words. Therefore, this step was skipped in this study.

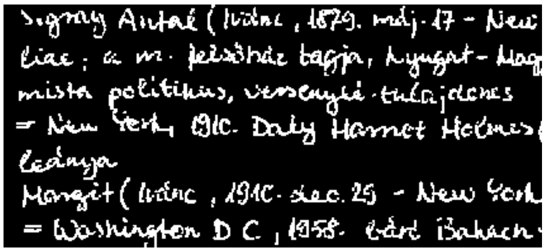
C. *Noise Removal*

The quality of the document is going to be improved by using noise reduction methods such as filtering and morphological operations (erosion, dilation etc.). Normalization also provides a tremendous reduction in data size by thinning and extracts the shape information of the characters [9], [12].

Most popular techniques for noise removal include filtering and morphological operations. Filters are usually aimed at smoothing, sharpening, thresholding, removing background and adjusting contrast [13]. Commonly used filters for noise removal include linear filtering, median filter, adapting filtering and averaging filter. As for the morphological operations, they replace the convolution operation by the logical operations. There are numerous morphological operations designed for noise removal such as thickening and thinning the characters; connecting the broken strokes and decomposing the connected strokes[10], [13].

In our work, standard MATLAB morphological operations were applied to the binary image. Firstly, the holes were filled by using ‘imfill’ operation. Subsequently, ‘dilate’ operation was adopted for thickening the characters. One point to worth mentioning is the result of performing a ‘dilate’ operation after applying an ‘erode’ operation was better than a merely ‘dilate’ operation. Hence, ‘dilate’ was performed after an ‘erode’ operation. Finally, a ‘thin’ operation was performed for thinning.

Median filtering and adapting filtering were applied to the image. However, the results were not satisfying since the noise in the image is not a real white noise or black and white noise. The filters removed the punctuation while removing the noise. Therefore, no filters were adopted. Instead, ‘bwareaopen’ was used to remove small noise from the image. The image after noise removal can be seen in Fig. 5.



Sigray Antal (Ivánc, 1879. máj. 17 – New
liac; a m. felsőház tagja, Nyugat-Mag
mista politikus, versenyistálló-tulajdonos
= New York, 1910. Daly Harriet Holmes
leánya
Margit (Ivánc, 1910. dec. 25 – New York
= Washington D C., 1958. báró Bakach-

Figure 5 Binary Image After Morphological Operations

Hungarian is a very challenging language for noise removal. The reason is that many letters contain punctuation and those punctuation marks are hard to discern with noise and they are highly likely to be removed at the noise removal phase.

D. Segmentation

Segmentation step is considered as the most crucial step for character recognition considering that the success of the recognition highly depends on a successfully segmented image[14], [15].

Since the input of the system is an image, segmentation is applied for multiple purposes. Firstly, line segmentation should be applied in order to divide the lines. Secondly word segmentation is adopted to extract words from each line. Finally character segmentation is performed.

1) Line Segmentation

In our study, in order to perform line segmentation different methods have been applied such as smearing, Hough Transform (HT), Watershed Transform (WT) and Horizontal Projection. WT did not perform well since it tried to segment not only lines but also words and individual characters at the same time. The image is segmented into many regions which are not necessarily meaningful. In order to overcome this, WT was applied to a smeared image in which regions are more apparent. In order to perform the smoothing with smearing, run length smoothing algorithm was applied to the image. This algorithm smears consecutive white pixels along the horizontal direction [16]. WT was tried on different smearing levels. However, the results were not satisfactory to be adopted in our system. The results of the WT can be seen in the Fig. 6.

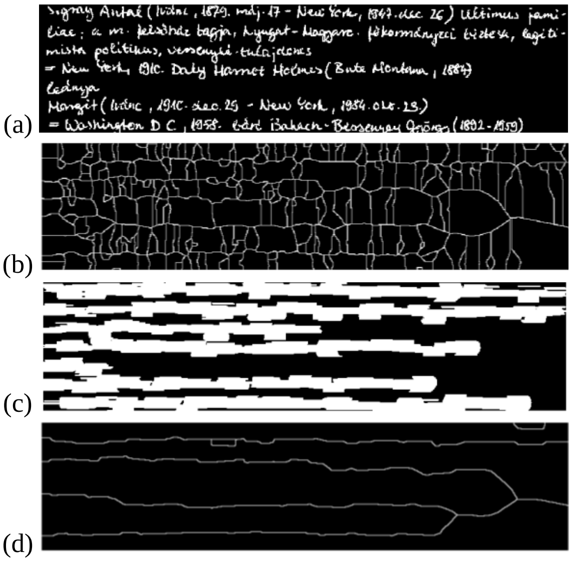


Figure 6 Segmentation with Watershed Transform: (a) original image, (b) segmentation of the original image with watershed transform, (c) smeared image, (d) segmentation of smeared original image with watershed transform

As another widespread method for several years, HT was applied to the image in order to find and then segment the lines [17], [18]. In contrary to WT, HT performed well for cursive handwritings. The results can be seen in Fig. 7. The green lines show the detected lines.

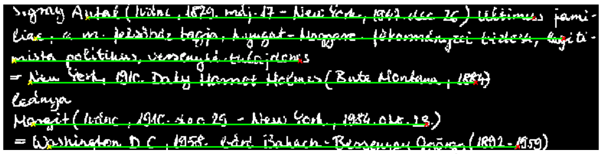


Figure 7 Line Detection with Hough Transform

Finally, Horizontal Histogram Projection was adopted for Hungarian Handwriting Recognition purpose. This method is generally used for character, word and line segmentation [19]–[21]. Horizontal Projection provides the histogram of the sum of the black or white pixels. It can be seen from the Fig. 8, the histograms clearly indicate where a line ends and another one starts in both images.

A threshold value can be used separate the lines. However, any set threshold value might led to cutting the tails of the words since there are overlaps between the lines caused by the tails of the letters. Instead a very effective and simple method was adopted. As can be seen from the Fig. 9, after smoothing the histogram, finding the regional minima gives a great result for segmentation. The first histogram (a) is the histogram of the number of white pixels in each row and the second histogram (b) is the smoothened histogram a, by using 'smooth' operator. The last one (c) is the histogram of the regional minimas. In MATLAB, 'imregionalmin' sets the regional minimas into 1 where and the rest into 0. This provides a straightforward way for finding the pixel where the lines should be separated.

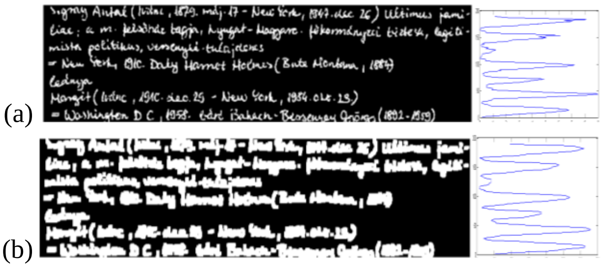


Figure 8 Horizontal Histogram Projection of the Images: (a) projection of the original image, (b) projection of the smoothened image

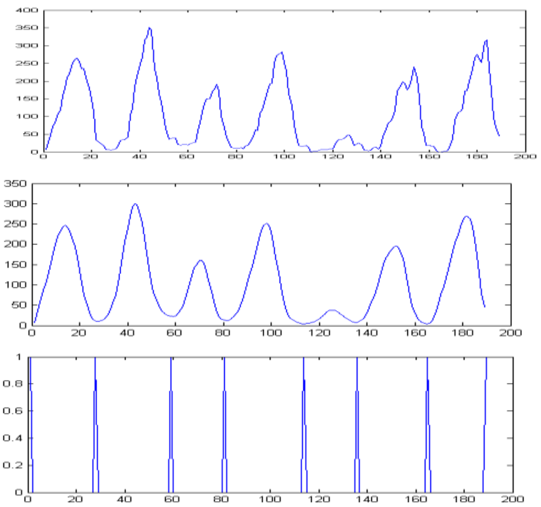


Figure 9 Histograms of the lines: (a) histogram of the number of white pixels, (b) smoothened Histogram (a), (c) histogram of regional minimas

2) Word Segmentation

Similarly to the line segmentation, a vertical projection method can be applied for word segmentation. When it comes to word segmentation phase, the challenges caused by the cursive handwriting starts to appear. As can be seen from the Fig. 10, vertical projection of an ordinary line would have zero pixel count at gaps between the words and within the words. In other words, any section of the line where there is not ink would be zero pixels at the vertical projection.

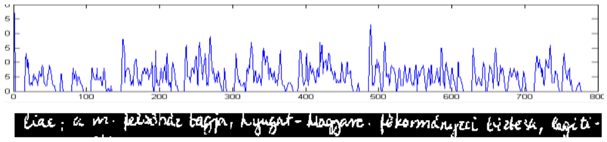


Figure 10 Vertical Projection of a Line

In order to overcome the challenge, the length of the consecutive zero pixels were taken into consideration. For this purpose, mean of five smallest consecutive zero counts was set as the threshold. As an alternative threshold, a set 7 pixels also gave very similar result. Subsequently to this process, the words were segmented. A point worth mentioning is by segmenting the words this way; punctuation marks and parenthesis are also segmented. However, elimination of those at this stage would be more convenient than trying to eliminate those at the recognition phase. Therefore, another threshold was adopted in order to ignore segments including the punctuation marks which are smaller than the threshold. An example histogram of a line containing punctuation mark and another histogram for a word containing a gap within the word can be seen in Fig. 11.

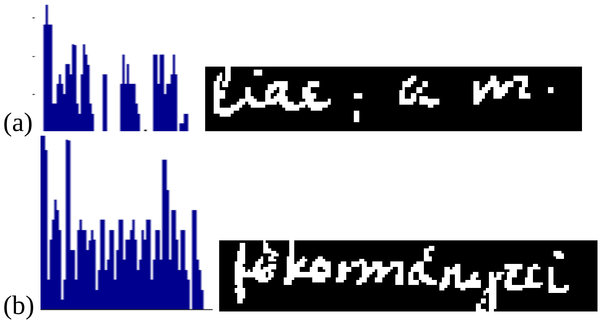


Figure 11 Examples of histograms of the words and punctuation marks: (a) an example histogram and a n image for punctuation marks, (b) an example for a word containing gap within the letters

3) Character Segmentation

Several methods for character segmentation have been adopted over years. It is possible to group those into two categories as explicit segmentation and implicit segmentation [22]. Explicit Segmentation refers to partition of the word into individual characters. Then, the recognition is performed according to the results of segmentation. Therefore, the accuracy of the segmentation is very important for a successful recognition. Implicit Segmentation on the other hand tries to match any input over the image with its alphabet. It basically searches all possible locations across the whole word signal [23]. The recognition is performed at the same time with segmentation.

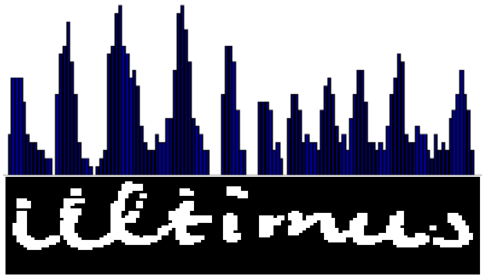


Figure 12 Vertical projection of a Word

In our work, explicit segmentation was adopted. Vertical Projection of the words was calculated and the individual characters were segmented according to the low pixel count at the point. However, when it comes to letters, a low number of pixels at a certain vertical line do not necessarily indicate a connection between letters. As can be seen from Fig. 12, it may also be the letter itself.

In order to distinguish between letters and the connecting lines, another measure was adopted before segmenting according to the threshold. The threshold is the centre line which is acquired by the horizontal projection of the word. And the line containing the highest pixel run is considered as the centre line [5]. If the pixel count is below threshold at a certain vertical line, then location of the white pixels are checked. If they belong to the upper part of the image, then there is no segmentation. On the other hand, if the white pixels are below or over the centre line then the segmentation is performed. This way, letters having low pixel count at their top are segmented correctly (h, m, n, r, t). Another addition to the algorithm is about the very last pixels. The algorithm ignores the low number of pixels in the very end of the image, in order to avoid segmentation of the ends of the letters into several pieces.

4) Evaluation of the Segmentation

Segmentation accuracy indicates how precise the prediction of the segmented image is. However, accuracy alone cannot be relied on in judging the performance of the given segmentation [24]. That is where the application of a confusion matrix can help. By setting the predicted segmentation against the true segmentation, not only the overall loss can be calculated, but the cost of different types of segmentation errors are also revealed by each cell of the matrix [25]. Table 1 shows the confusion matrix. In the table below, 'a' is the number of correct negative predictions, and 'b' shows the number of incorrect positive predictions, while in the following row 'c' is the number of incorrect negative predictions, and 'd' indicates the number of correct positive predictions [26].

TABLE I
CONFUSION MATRIX

		Predicted	
		Negative	*Positive*
Actual	*Negative*	a	b
	Positive	c	d

We adopted four measures for evaluation namely accuracy, precision, recall and f measure. Table II, represents those results of the line segmentation, word segmentation and character segmentation methods adopted. Line segmentation method worked very well with the data set. The accuracy for line segmentation is 96% for projection of histogram method. As can be seen in the Table II, HT and WT did not perform as well as the projection of histogram method.

TABLE II
RESULTS

	Line			Word	Character
Method	Projection of Histogram	HT	WT	Projection of Histogram	Projection of Histogram
Accuracy (%)	96	79	38	85	78
Precision	0,97	0,85	0,43	0,90	0,78
Recall	0,99	0,91	0,55	0,90	1
F-measure	0,98	0,88	0,55	0,90	0,88

In case of word segmentation, the accuracy was only 85% with 0,9 precision, recall and f measure value. Words written very close to each other created the biggest miss-segmented words group. This challenge comes with cursive handwriting. In the future, other methods will be adopted in order to overcome this issue. Finally, character segmentation phase was completed and the accuracy rate was only 78%. The method adopted did not perform very well on cursive handwritings. However, after modification certain difficulties were overcome. Segmentation of individual characters in cursive script is still problematic when it comes to real life documents with low quality. In the future, different methods are going to be employed for character segmentation purpose.

III. CONCLUSION

Pre-processing is a crucial step for Optical Character Recognition. At the same time, it is also very challenging especially in offline recognition since the quality of the input cannot be controlled. Recognition of real life documents is different from the ready databases as they can be in any form. Our data set was obtained by scanning handwritten lecture notes in Hungarian. The handwritings in the data set were all cursive or mixed cursive and discrete which are the most difficult handwritings for recognition [4].

In this paper, we proposed pre-processing techniques for Hungarian handwritings. The proposed method is also applicable to other languages with Latin characters. After completion of the binarization and noise removal, the images of pages were segmented into lines, words and finally characters respectively. The results were satisfying for line segmentation and word segmentation. However, character segmentation accuracy was lower than ideal.

The expected accuracy for character segmentation is very high since the accuracy of the recognition is highly dependent upon that. It is hard to compare the results of this work with others because different data sets were employed. In addition to this, there are not many research conducted for Hungarian Handwriting recognition. However, the results of the current work are going to be compared with the results of the future works.

IV. FUTURE WORK

Looking at the results, implementation of other word and character segmentation techniques seems to be beneficial. Alternatively, implicit segmentation methods can also be adopted for the same data set. Additionally, skew correction and slant removal phase was skipped due to a very little amount of skewed lines and slanted words in this data set. Therefore, those were ignored. However, in the future work, worse handwritings are going to be recognized. Hence, skew correction and slant removal phase is estimated to be included in the future work.

Subsequent to this study, a recognition step is going to be completed by using different techniques. After the completion of the recognition of Hungarian handwritings, more challenging Hungarian handwritings are going to be tested. If necessary, the methods are going to be chanced according to the needs of the new handwritings. As a result, a common recognition system for Hungarian handwriting will be provided.

REFERENCES

[1] J. Pena, S. Letourneau, and F. Famili, “Application of Rough Sets Algorithms to Prediction of Aircraft Component Failure,” in *Advances in Intelligent Data Analys*, no. i, 1999, pp. 473–484.

[2] R. Plamondon and S. N. Srihari, “On-Line and Off-Line Handwriting Recognition : A Comprehensive Survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 63–84, 2000.

[3] P. K. Charles, V. Harish, M. Swathi, and C. H. Deepthi, “A Review on the Various Techniques used for Optical Character Recognition,” *Int. J. Eng. Res. Appl.*, vol. 2, no. 1, pp. 659–662, 2012.

[4] C. C. Tappert, “Adaptive on-line handwriting recognition,” in *7th International Conference on Pattern Recognition*, 1984, pp. 1004–1007.

[5] N. Arica and F. T. Yarman-Vural, “Optical character recognition for cursive handwriting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 801–813, 2002.

[6] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms,” *IEEE Trans. Syst. Man, Cybern.*, vol. 9, pp. 62–66, 1979.

[7] P. K. Sahoo, S. Soltani, and A. K. C. Wong, “A Survey of Thresholding Techniques *,” *Comput. Vision, Graph. Image Process.*, vol. 41, no. 2, pp. 233–260, 1988.

[8] N. Priyanka, S. Pal, and R. Mandal, “Line and Word Segmentation Approach for Printed Documents,” in *IJCA Special Issue on Recent Trends in Image Processing and*

Pattern Recognition-RTIPPR, 2010, pp. 30–36.

[9] B. Mağden and S. Telçeken, “Probabilistic Rough Sets in Turkish Optical Character Recognition,” in *6th World Conference on Soft Computing*, 2016.

[10] R. K. Nath and M. Rastogi, “Improving Various Off-line Techniques used for Handwritten Character Recognition : a Review,” *Int. J. Comput. Appl.*, vol. 49, no. 18, pp. 11–17, 2012.

[11] R. M. Bozinovic and S. N. Srihari, “Off-line cursive word recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 1, pp. 68–83, 1989.

[12] R. J. Rodrigues, A. Carlos, and G. Thom, “Cursive character recognition – a character segmentation method using projection profile-based technique,” in *The 4th World Multiconference on Systemics, Cybernetics and Informatics SCI 2000 and The 6th International Conference on Information Systems, Analysis and Synthesis ISAS*, 2000.

[13] S. Mo and J. Mathews, “Adaptive, quadratic preprocessing of document images for binarization,” *IEEE Trans. image Process.*, vol. 7, no. 7, pp. 992–999, 1998.

[14] B. Verma and M. Blumenstein, “Fusion of Segmentation Strategies for Off-Line Cursive Handwriting Recognition,” in *Pattern Recognition Technologies and Applications: Recent Advances*, New York: Information Science Reference, 2008, pp. 1–16.

[15] A. R. Khan and M. Zulkifli, “A Simple Segmentation Approach for Unconstrained Cursive Handwritten Words in Conjunction with the Neural Network,” *Int. J. Image Process.*, vol. 2, no. 3, pp. 29–35, 2008.

[16] K. Y. Wong, R. G. Casey, and F. M. Wahl, “Document analysis system,” *IBM J. Res. Dev.*, vol. 26, no. 6, p. 647–656., 1982.

[17] D. H. Ballard, “Generalizing the Hough transform to detect arbitrary shapes,” *Pattern Recognit.*, vol. 13, no. 2, p. 111–122., 1981.

[18] R. O. Duda and P. E. Hart, “Use of the Hough transformation to detect lines and curves in pictures.,” *Commun. ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[19] M. H. Glauberman, “Character recognition for business machines,” *Character Recognit. Bus. Mach. Electron.*, vol. 29, no. 2, pp. 132–136, 1956.

[20] R. Kasturi, *Image analysis applications.*, 24th ed. CRC Press, 1990.

[21] O. D. Trier, A. K. Jain, and T. Taxt, “Feature extraction methods for character recognition - A survey,” *Pattern Recognit.*, vol. 29, no. 4, pp. 641–662, 1996.

[22] A. Choudhary, “A Review of Various Character Segmentation Techniques,” vol. 4, no. 6, pp. 559–564, 2014.

[23] C. Viard-gaudin and É. Caillault, “Using Segmentation Constraints in an Implicit Segmentation Scheme for On-line Word Recognition,” in *Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006.

[24] N. Xue, “Chinese word segmentation as character tagging,” *Comput. Linguist. Chinese Lang. Process.*, vol. 8, no. 1, pp. 29–48, 2003.

[25] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*. MIT Press, 2001.

[26] “Confusion Matrix.” [Online]. Available: http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html. [Accessed: 07-Sep-2016].