Performance Comparison of Different Classifiers for Hungarian Handwriting Recognition

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

***Abstract*— Offline cursive handwriting recognition is an ongoing challenge due to the different styles used by different persons. The difference in the handwriting styles brings about the hardship for segmentation of the characters hence the overall accuracy of the recognizer is highly dependent on the style. In Hungary, there is a tradition of using cursive handwriting and the alphabet contains some letters with punctuation. Therefore, Hungarian handwriting recognition is a challenging task to perform. In this study, we compare the performance of different classifiers on a small data set (1750 characters, 50 samples for each letter in the alphabet) that has previously been generated for our study. The data set only consists of lower case Hungarian letters (35 letters excluding the ones which consist of two letters). In our study we compared the performance of four classifiers namely, Neural Networks, Support Vector Machines (SVM), Rough Sets Theory (RST) and Bayesian Networks (BN) using the WEKA machine learning tool. The results indicated that in terms of classification accuracy, neural networks performed the best followed by BN, SVM and RST respectively. However, in terms of the time taken to build the model neural networks performed the poorest. BN took the shortest time to build followed by SVM and RST respectively.**

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 recognized 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 compares the performance of four classifiers applied to a small dataset which was created by the researchers earlier. The classifiers adopted are Neural Networks, Support Vector Machines (SVM), Rough Sets Theory (RST) and Bayesian Networks (BN). The next section provides the properties of Hungarian Handwriting with reference to its challenges. In the following sections the adopted dataset, feature extraction, classification phases are explained and the results are provided. Finally the conclusion is presented.

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 letter. These characters of the language generate a challenge for recognition purpose such as removal of the punctuation at the noise removal phase.

Another challenge in recognizing 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. However, this study does not include the segmentation of the Hungarian handwriting. The dataset adopted is already segmented into the characters. However, due to the nature of cursive handwritings, the characters are not as readable as in discretely written texts. The characters may be distorted and written in a personal way which is not clearly readable. In addition to the challenges, there are not many studies conducted for the purpose of Hungarian Handwriting Recognition.

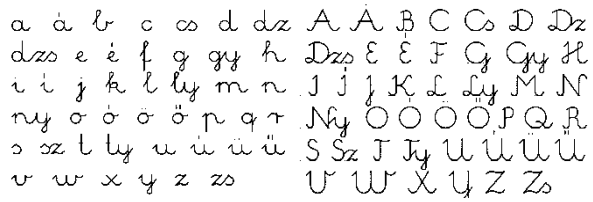


Figure 2 Hungarian Alphabet [4]

II. DATA SET

The adopted dataset was previously created by the researchers. It includes 1750 characters (50 samples of 35 lower case Hungarian characters excluding the characters which consist of more than one letter). Each character in the data set is normalized to 28x28 pixels and in the skeleton form as can be seen in the Figure 3.These characters are the output of the previous stages of study and in this study they are used as input of classification.

Figure 3 Sample characters from the dataset

In order to create the data set, Hungarian handwritings were collected from multiple users on paper. Then the papers were scanned with 300 dpi and saved in the png format in order to avoid information loss. Consecutively, the documents were pre-processed. Pre-processing phase included binarization, skew correction, slant removal and noise removal. Thereafter, the lines, words and characters were segmented respectively. Finally, size of segmented characters was normalized into 28x28 pixels.

III. FEATURE EXTRACTION

In the feature extraction phase, significant features of a character are extracted. The result of the classification is directly affected by the features extracted since the feature vectors are going to be the input for the classifier. Therefore, it is crucial to extract the key features.
It is possible to group the features into three categories namely distribution of points, structural analysis and transformations and series expansions. In our work, features were extracted using distribution of points and structural analysis features.

A. Distribution of Points:

In this category, features are extracted based on the statistical distribution of points. These features are usually tolerant to distortions and style variations[5]. The feature extraction techniques used in this study which are based on distribution of points are represented below:
Projection Profiles: Profiles refer to the distance from the border of the image until the next white pixel. An example representation of projections of a character is given in the Figure 4. In our work, left, right, top and bottom profiles are used as feature vectors.

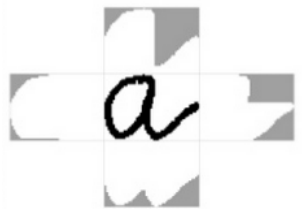


Figure 4 Right, left, top and bottom profiles of a character

Extremas of the character image: It returns the x and y coordinates of the 8 extremas of the image namely top-left, top-right, right-top, right-bottom, bottom-right, bottom-left, left-bottom and left-top as can be seen in Figure 5.

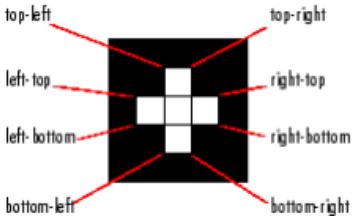


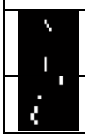
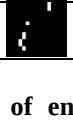
Figure 5 Extremas

Center of gravity: The (x, y) values of the center of gravity of the character image. In addition to those, the distance between the bottom of the character and the y coordinate of the center of gravity and the distance from the left end of the character and x coordinate of the center of gravity are also used as feature vectors.
Density: The density of the character image.
Area: Actual number of pixels in the region, returned as a scalar.
The proportion of width and length: The result of dividing the width of the character into the length of the character.
Number of regional minimas and maximas: After applying horizontal and vertical projections, the number of regional maximas and regional minimas for both vertical and horizontal projection are used as features.

B. Structural analysis

This type of features represents the geometric and topological structures of a character. The most common types include endpoints, loops and strokes[5][6]. It is worth mentioning that these types of features are highly affected by any noise in the data. As can be seen in Table 1 that any noise in the character image would cause a change in the feature vector thus it is crucial for the recognition that the data set is noise free. The feature extraction techniques used in this study which are based on the structural analysis are explained below:

Table 1 An example feature set of noise free character image and a noisy character image

Character image	#endpoints	#connected components	#isolated small areas
	4	2	2
	5	3	3

No of endpoints: It represents the number of pixels having only 1 connected neighbor in an 8 connected image.
No of branch points: It represents the number of pixels having at least 3neighbors that are 1s in an 8 connected image.
Euler number: The Euler number represents the total number of objects in the image minus the total number of holes in those objects.

Number of loops: The vector represents the number of holes in the image.
Number of small components: The feature vector represents the number of isolated areas with the area smaller than 7 in the character image.
Sum of the area of small components: It represents the sum of all small components with less than 7 pixel area.
No of connected components: It represents the number of connected components in the image.

IV. CLASSIFICATION

The data was classified by four classifiers namely, Neural Networks, Support Vector Machines, Rough Sets Theory and Bayesian Networks using WEKA machine learning tool[15]. A brief explanation of the classifiers is given below.

Neural Networks: Neural Network design which is made of parallel interconnection of adaptive processors[7]. Since it has the parallel connections, it has a better performance that the classical techniques. Additionally, its adaptive nature provides a better adaptability to changes in the data and an ease to learn the characteristics of input signal[8]. The structure of a neural network contains many nodes. The output of one node is input to another, thus the final output is a result of complex interaction of all nodes. Neural network architectures can be classified into two major groups which are feed-forward and feedback networks. In our work, the multilayer perceptron of the feed forward networks is adopted since it is the most popular for character recognition purposes.

Support Vector Machines: SVM classifier carries out the classification by mapping all the input data to a value in a higher dimensional space. The data is classified by coming up with an optimum N-dimensional hyper plane which separates data into positive and negative examples[9]. In our work SVM is used due to its ease of implementation and high performance.

Rough Sets Theory: Rough Sets is a mathematical tool which deals with uncertainty and vagueness[10]. The idea is based on the assumption that with every object of the universe of discourse, it is possible to associate some information. Objects characterized by the same information are indiscernible in view of the available information about them. The indiscernibility relation generated in this way forms the mathematical basis of the theory. The rough sets theory provides a technique of reasoning from imprecise data, discovering relationships in data and generating decision rules[11]. Not requiring any preliminary or additional information about data like probability distributions in statistics is rough sets theory's main strength[12]. In our work, the RST was adopted due to the above-mentioned strengths. Since the data set is relatively small, we believe that RST may be a good classifier in such conditions.

Bayesian Networks: Bayesian classifiers are the statistical classifiers based on Bayes Theorem. They are able to predict class membership probabilities such as the probability that a given tuple belongs to a particular class[13]. Bayesian networks are a model representing uncertain knowledge about a complex phenomenon and allowing real reasoning from data. They effectively represent a domain of knowledge, as a causal graph, permitting learning the dependency relationships that can help us make decisions and manage all incomplete data[14].

V. EXPERIMENTS AND RESULTS

The classification task was carried out with and without applying feature selection. For the same data set, a supervised feature selection algorithm provided by WEKA is applied to the features. Additionally, the results of the classification without any feature selection are also given. Finally the recognition is performed with three different cross validation values which are 5, 7 and 10 fold cross validation.

The classification accuracy and time taken to build the model for each classifier are given in the Table 2 and Table 3 respectively.

Table 2 Classification accuracies for different classifiers

	No Feature Selection			**Feature Selection**		
	5 fold	**7 fold**	**10 fold**	**5 fold**	**7 fold**	**10 fold**
SVM	91.1 %	92.5 %	92.1 %	95.0 %	95.4 %	95.6 %
RST	86.1 %	88.8 %	88.1 %	88.4 %	91.1 %	90.3 %
BN	89.0 %	88.8 %	89.4 %	95.8 %	96.0 %	95.4 %
NN	92.2 %	92.9 %	92.7 %	96.6 %	96.8 %	96.6 %

As provided in Table 2, Neural Networks give the highest accuracy with and without feature extraction compare to the other classifiers. It is followed by BN, SVM and RST respectively. In addition to that, it is possible to say that feature selection increases the accuracy as well as the time taken to build the model. Finally, a 7 fold cross validation appears to be the most suitable value for this data set since it is fastest and provides the most accurate classification results.

Table 3 Time taken to build the model (seconds)

	No Feature Selection			**Feature Selection**		
	5 fold	**7 fold**	**10 fold**	**5 fold**	**7 fold**	**10 fold**
SVM	1.81	0,84	0,8	0,69	0,69	0,71
RST	21,01	19,8	20,02	14,09	10,95	11,52
BN	0,11	0,06	0,09	0,2	0,06	0,03
NN	2301	2158	2265	1567	1452	1504

Although, NN gives the best accuracy, it is clearly the slowest when it comes to the time taken to build. There is almost 99% difference in speed with the second slowest classifier RST. Although there is only about 1%

difference in accuracy with the second best classifier, the time taken to build the model is almost 99% times slower.

VI. Conclusion

Hungarian Handwriting recognition is a challenging field considering the tradition of using cursive handwriting and the letters with punctuations. In this study, we performed a Hungarian Handwriting classification using a small data set with four different classifiers. The results of the different classifiers are compared in terms of their performances.

Classification of handwritten characters includes several crucial steps. It is possible to say that feature extraction is one of the most important steps for the recognition of the characters since the distinctive and characteristic features must be extracted. In our work, several feature extraction methods were adopted. Consecutively, the character images from the data set were classified by four different classifiers.

The results were interesting considering the difference in the time taken to build different classifiers. NN performed the best in terms of accuracy, followed by BN, SVM and RST. However, NN was significantly slower than any other classification with around 99% difference in speed with the second slowest RST.

VII. Future Work

A deeper understanding of the results of feature extraction methods may be useful with representation of data such as which method is more distinctive for which characters and which characters are more likely to be misclassified. Additionally, it would be beneficial to apply the same method to a bigger data set. We believe the greater the data set, the better the accuracies are going to be. For example, RST was applied considering its nature to work well with only a little data available. However, it performed one of the poorest in both accuracy and time taken to build the model. It would be necessary to compare the differences with a bigger data set.

VIII. 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] Bergamott, “Írott és nyomtatott ábécé,” 2010. .

[5] L. Eikvil, “Optical character recognition,” *Citeseer. Ist. Psu. Edu/142042. Html*, vol. 3, no. 1, pp. 4956–4958, 1993.

[6] N. Arica and F. T. Yarman-Vural, “An overview of character recognition focused on offline handwriting,” *IEEE Trans. Syst. Man Cybern. Part C (Applications Rev.*, vol. 31, no. 2, pp. 216–233, 2001.

[7] 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.

[8] N. Arica and F. T. Yarman-Vural, “An overview of character recognition focused on off-line handwriting,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 31, no. 2, pp. 216–233, 2001.

[9] J. Taylor, S. Kumar, and I. Khaimovich, “Cursive Handwriting Segmentation and Character Recognition,” 2007.

[10] Z. Pawlak, “Rough Sets,” *Int. J. Comput. Inf. Sci.*, vol. 11, no. 5, pp. 341–356, 1982.

[11] J. J. Shuai and H. L. Li, “Using rough set and worst practice DEA in business failure prediction,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3642 LNAI, pp. 503–510, 2005.

[12] B. Mağden and S. Telçeken, “Probabilistic Rough Sets in Turkish Optical Character Recognition,” in *6th World Conference on Soft Computing*, 2016, no. 3, pp. 170–173.

[13] K. M. Lakshmi, K. Venkatesh, G. Sunaina, D. Sravani, and P. Dayakar, “Hand Written Telugu Character Recognition Using Bayesian Classifier,” *Int. J. Eng. Technol.*, vol. 9, no. 3S, pp. 37–42, 2017.

[14] K. Jayech, M. A. Mahjoub, and N. Ghanmi, “Application of Bayesian Networks for Pattern Recognition : Character Recognition Case,” in *International Conference on Sciences of Electronics, Technologies of Information and Telecommunications*, 2012, vol. 3, no. March, pp. 748–757.

[15] F. Eibe, M. A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", *Morgan Kaufmann*, Fourth Edition, 2016.