8th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS
S u c e a v a, R o m a n i a, M a y 25 – 27, 2 0 0 6

USING CHARACTER MOMENT BASED INVARIANT FEATURES TO IMPROVE OFF-LINE HANDWRITING RECOGNITION

Dan CIREŞAN, Dan PESCARU
"Politehnica" University of Timisoara, Faculty of Automation and Computers, Computers Department
Bd. V. Parvan no. 2, 300232 Timisora, Romania
cdanc@cs.upt.ro
dan@cs.upt.ro

Abstract. *Important progress has been made in handwrite recognition over the last few years. However, actual systems depend very much on quality of the input text. This paper presents a study addressing this problem by using moment based invariant features in conjunction with a LVQ neural network.*
Keywords: *off-line handwrite recognition, OCR, segmentation, features extraction, classification algorithms.*

Introduction

The constant development of computer tools involves new requirements for interfaces between man and computer. Considering these, Handwritten Character Recognition represents an advanced topic in this area [1]. It may apply to handwrite documents digitization, automatic Zip-code classification, signature recognition etc. For several years, the pressure coming from industry, which recognizes the importance of these applications, implies an intense research in this field.

On-line and Off-Line Handwrite Character Recognition

The main structure of such a recognition system is illustrated at the Figure 1.

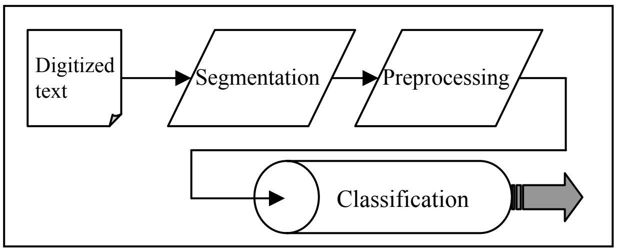

Figure 1. A typical recognition system structure.

The first step is the character segmentation, which consists in analyzing the digitalized image provided by a scanning device. Its purpose is to localize the limits of each character and to isolate them one from others. Despite of the writing constrains that does often exist on the original printed form, the segmentation process is not so easy in practice.

The goal of the second step of the recognition process is to extract relevant information's from an image of a character, as well as to reduce its dimension of representation. This reduction is required in order to make easier the design of the classification system, when relevant feature extraction allows presenting competently a character to the classifier only by using significant information.

Once relevant features have been extracted, they are submitted to a classifier whose task is to identify the character that they represent and to assign them the corresponding *ASCII* code. This represents the classification step and, for some years, artificial neural networks have shown good capabilities in performing this kind of tasks. This is due to the non-linearity that is included in these connectionist systems, and to the training phase that allows a fine-tuning through learning. However, their performance is strongly affected by the quality of the representation of the characters. This may require a large number of parameters to represent the character, which then results in difficulty in establishing the rules for recognition. In other words, the network becomes difficult to train. Moreover, the greater the size of the network affects directly the computation time. This can restrict their practical use. Therefore, it is necessary to perform efficient features extraction on the one hand, based on theirs relevance.

On-line handwriting recognition, in contrast to off-line one [2], refers to the situation where the recognition is performed concurrent to the writing process. This distinction has some consequences, regarding further differentiations between on-line and off-line recognition. Among others, these involve important aspects like data acquisition, data representation, recognition methods and typical applications.

On-line handwriting is a dynamic, digitized representation of a digital pen movement, generally describing sequential information about position, velocity, acceleration or even pen angles as a function of time [3].

In contrast to that, the off-line handwriting recognition is usually performed some time after creation. In this context, data is usually represented as a pixel matrix (scanned image).

Further improvements could be made using more complex information like dictionaries [4, 9].

Pre-processing

As input data for tests, the **NIST 19** database [5] was used. The **NIST 19** contains handwritten text from 3600 writers. The text is composed from separate digit, upper and lower case and free text. Over 800000 images with hand checked classifications are stored. Aside the images from database, tools for compression /decompression are present. Because all tools are written in **C** for **UNIX**, but not adapted for 80x86 architecture (consequently, Linux), they were adapted to this architecture.

The images from database are black and white. The size for images that represent symbols is 256 by 256 pixels. One pixel is stored on only one bit, so an image consume 256x256/8=8192 bytes of memory. All images are stored one by another and the file is compressed using an implementation of the **CCITT Group 4** algorithm. More information about the database can be found in [5].

We have developed tools for visualizing the images from database, for extracting specifically images from database and creating files that can be used as input for artificial neuronal networks, for resizing and normalizing the images, for features extraction.

Features Extraction

All neuronal networks need training and test data. Because the complexity and the time required for training is growing exponentially with the size of the images, aside offering to the input the entire images, alternative methods must be found.

Using as input the images in their brute form, 256 by 256 pixels is unacceptable. First step is to resize the images to much lower dimensions, like 32 by 32 pixels. Even if the dimensions are smaller, the information is enough to represent clearly the symbols. Still, the input to the network is composed from almost one thousand numbers that imply a very long training time.

Considering the fact that the central line preserves the topology of the initial image and decreases the number of relevant pixels in image, implicitly decreasing the computation time, the algorithm from [10] is applied on all characters before all other processing.

In order to lower the training time, the size of the input must suffer further reduction. One way is to use as feature the so-called moment based invariant features [6]. The idea of these features comes from physics, specifically from mass centre and from moment of inertia. The general formula for moments is

$$M_{pq} = \iint x^p y^q f(x,y)dxdy, \qquad (1)$$

where $f(x,y)$ is a function of two independent variables. The $f(x,y)$ function represents the intensity of the pixel (x,y) on the image. The parameters p and q are positive integers and the integration is over the entire plane. In the case of discrete pixels, x and y are coordinates and the integrals is replaced with summation. According to [7], $f(x,y)$ is completely determined by the infinite sequence M_{pq}. Indeed, it is impossible to work with infinite

sequence and a small number of moments are sufficient. The moments offer a very small size for features vector and, in the same time, they contain sufficient information for recognition. In addition, they can easily normalized and they are unaffected by position, height, width and orientation of the original image.
For binary, discrete images, equation (1) is rewritten using summations and the formula becomes

$$M_{pq} = \sum_{x} \sum_{y} x^p y^q f(x, y), \quad (2)$$

where $f(x, y) \in \{0,1\}$.
The mass centre is easily computed with:

$$\bar{x} = M_{10} / M_{00}, \ \bar{y} = M_{01} / M_{00} \quad (3)$$

Equation (2) does not offer moments invariable to translation. Using the equations for the mass centre, the new formula for moments invariable to translation turn to:

$$m_{pq} = \sum_{x} \sum_{y} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (4)$$

Moments with lower order are independent of the image and, consequently, useless for recognition purpose. Indeed, the only moments that contain information have the order greater than 2. In addition, the moments of large order are to particulars to respective sample image. Hence, we will work with moments with order smaller than seven, i.e. $2 < p + q < 7$. Twenty-two moments of this type exist.

Test Environment Settings

The tests were divided in three main groups, as presented in Table 1. The purpose was to see how the network is behaving with various sets of symbols.
The first group contains five sets of similar symbols, i.e. all symbols from a set have resembled shapes. More exactly, for the set S2, all symbols contain a loop. In the case off *c*, the loop in not closed, but is almost a loop there.
The set S3 is composed from symbols that have a long, vertical stroke. This group is intended to see how well a Self Organizing Network based on LVQ algorithm [8] is able to classify very resembling symbols.
The second group is composed from four sets of symbols; each set containing two subsets of similar symbols, but the subsets are different each from other. Specifically, for T3, the first subset is formed from *i* and *l*, i.e. symbols with a long and vertical stroke, and the second subset is formed from *u* and *y*, i.e. symbols with a concavity pointing down. The purpose of this group is to see the behavior of network for simultaneous classifying similar and dissimilar symbols.
The third group contain two sets of symbols, each set composed from dissimilar symbols. In the case of set D1, we have *a* – loop like symbol, *l* – long and vertical stroke, *u* – concavity pointing down, and *z* – horizontal and oblique strokes. These sets are intended to test the recognition ability of the LVQ network for very dissimilar symbols.

Table 1. The test groups

Group 1		**Group 2**		**Group 3**	
Set ID	Symbols	Set ID	Symbols	Set ID	Symbols
S1	a, o, b, q	T1	a, o, i, j	D1	a, l, u, z
S2	d, b, p, c	T2	a, o, u, v	D2	1, 3, 8, 4
S3	a, e, c, o	T3	i, l, u, y		
S4	i, j, l, t				
S5	v, u,y, w				

Experiment and Results

The results for group 1 witch contains similar characters are presented in figures 2 thru 6, and the overall recognition rates are plotted in figure 7. For each set, we used 1000 samples for training and 240 samples for testing.
The moments with order smaller or equals than five offer the best result.

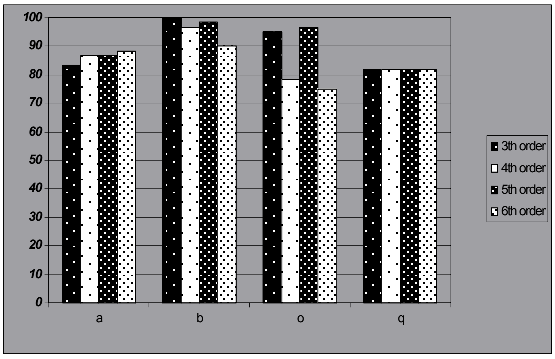

Figure 2. Recognition rate for set S1.

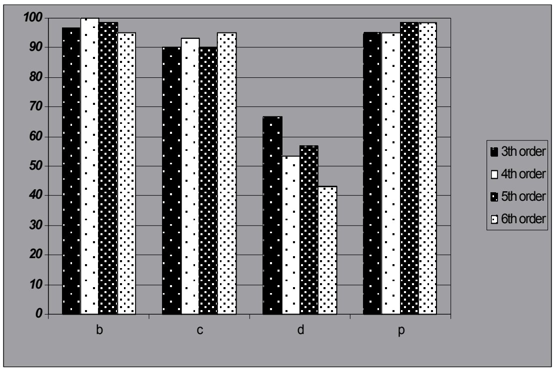

Figure 3. Recognition rate for set S2.

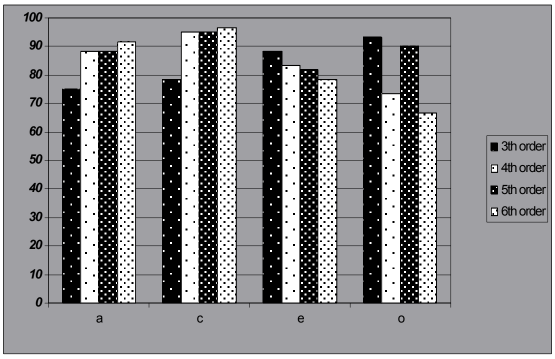

Figure 4. Recognition rate for set S3.

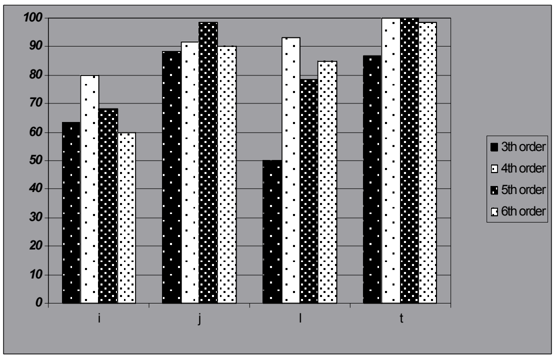

Figure 5. Recognition rate for set S4.

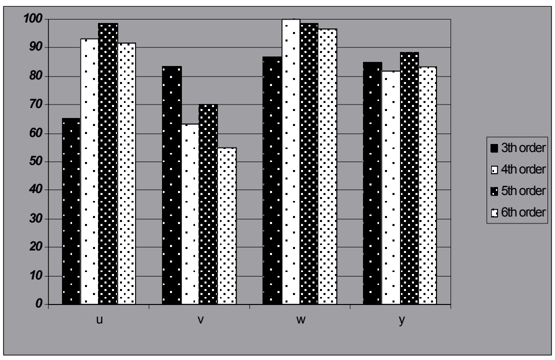

Figure 6. Recognition rate for set S5.

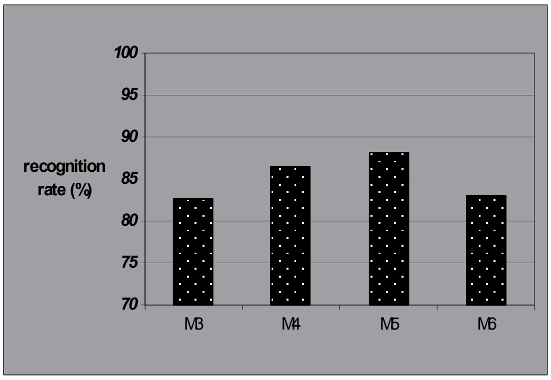

Figure 7. Overall recognition rate for group 1.

Figures 8, 9 and 10 show the recognition rates for second group – two different subgroups, each containing similar characters. Analysing this group we get similar result. The moments with order smaller or equal than five offer the best recognition rate, as concluded by figure 11.

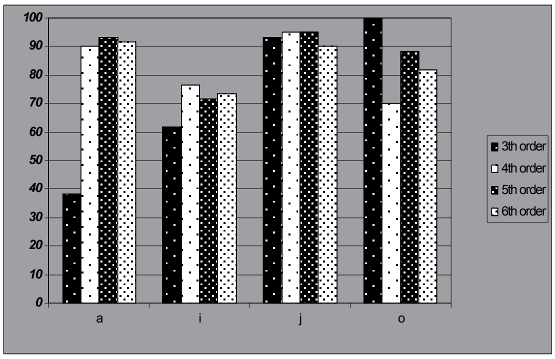

Figure 8. Recognition rate for set T1.

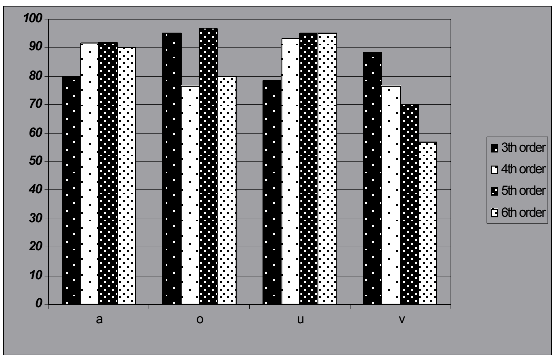

Figure 9. Recognition rate for set T2.

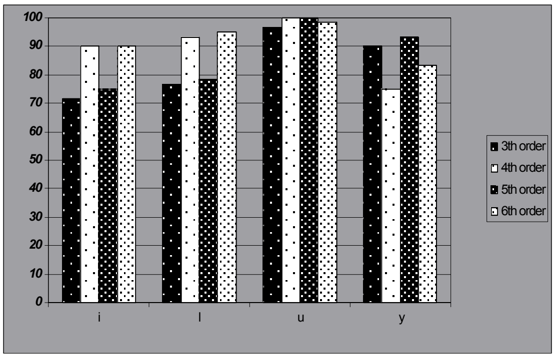

Figure 10. Recognition rate for set T3.

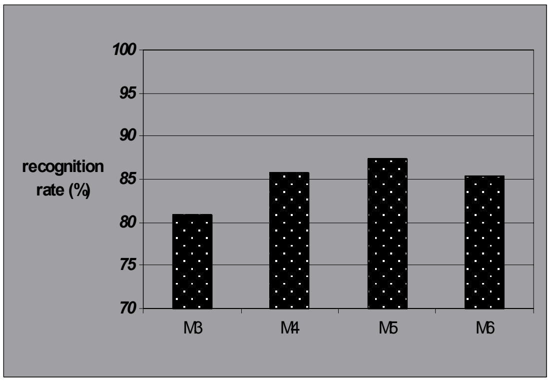

Figure 11. Overall recognition rate for group 2.

The recognition rates for sets D from the third group are plotted in figures 12 and 13. The overall recognition rate is presented in figure 14. Indeed, also here, the moments with order smaller or equal than five offer the best recognition rate.

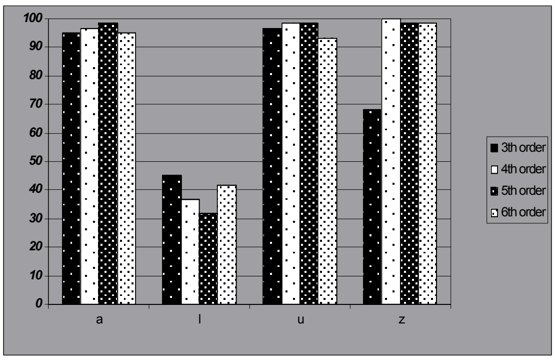

Figure 12. Recognition rate for set D1.

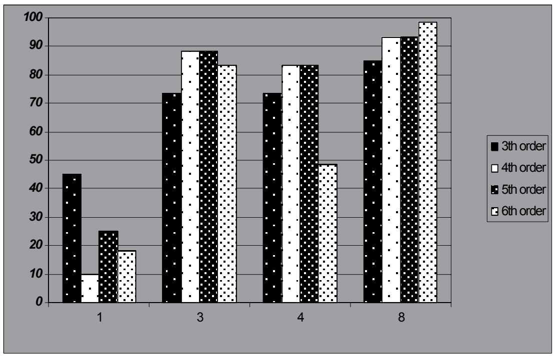

Figure 13. Recognition rate for set D2.

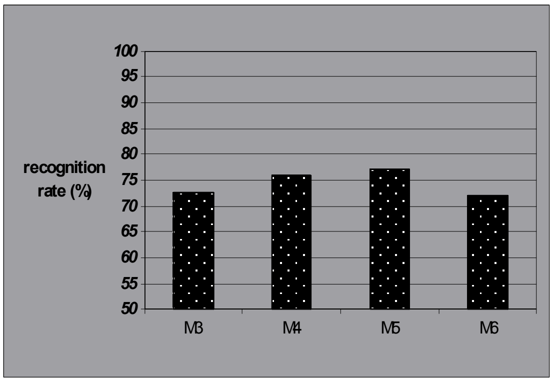

Figure 14. Overall recognition rate for group 3.

Table 2 depicted the global recognition rates considering all sets. In this case, the overall recognition rate is computed as average of all sets, considering all groups. Finally, we can conclude that the moments of order smaller or equal than five are most suitable.

Table 2. Overall recognition rates involving all groups

Set/ moments	**3th order**	**4th order**	**5th order**	**6th order**
S1	90.00	85.83	90.83	83.75
S2	87.08	85.42	85.83	82.92
S3	83.75	85.00	88.75	83.33
S4	72.08	91.25	86.25	83.33
S5	80.00	84.58	88.75	81.67
Group 1	**82.58**	**86.41**	**88.08**	**83.00**
T1	76.25	82.92	81.67	82.08
T2	69.17	68.75	72.50	62.08
Group 2	**72.71**	**75.83**	**77.08**	**72.08**
D1	73.33	82.92	87.08	84.17
D2	85.42	84.58	88.33	80.42
D3	83.75	89.58	86.67	91.67
Group 3	**80.83**	**85.69**	**87.36**	**85.42**
Overall	**80.08**	**84.08**	**85.66**	**81.54**

Conclusions

In this paper, we have studied the opportunity of using different orders moments, considering the effect on characters recognition rate. We used as input data for the classifier the moments computed for characters central lines. The experiment was performed on three main characters groups, each composed from sets of four characters. We aimed with these groups to studies the performance in various situations like four different shapes, two dissimilar subsets with two similar characters, and four characters with resembling shapes.

Despite of the fact that the results are quite spread on different orders, we can conclude that, overall, the moments smaller or equal than five have offered the best recognition rate. The moments smaller or equal than four give almost the same recognition rate, with the 40% less time needed for training.

Further studies will be devoted to combine the moments with another classifier based on topological information. The work will involve strokes and cycles extraction offering features like tilt, aspect ratio and chain codes. In addition, we will consider other recognition algorithms.

References

[1] Srihari, S. N., Srihari, R., Govindaraju, V. (1994) *Handwritten Text Recognition*, in Proceedings of the 4th International Workshop on Handwriting Recognition, Taipei, Taiwan, December, 1994

[2] Lallican, P.M., Viard-Gaudin, C., Knerr, S. (2000) *From Off-line to On-line Handwriting Recognition*, in proc. of the 7th International Workshop on Frontiers in Handwriting Recognition, Amsterdam, Netherlands, September 2000, pp. 303-312

[3] Bahlmann, C., Haasdonk, B., Burkhardt, H. (2002) *On-line Handwriting Recognition with Support Vector Machines--A Kernel Approach*, in proc. of the 8th International Workshop on Frontiers in Handwriting Recognition, Ontario, Canada, September 2002, pp. 49-54.

[4] Koerich, AL, Sabourin R, Suen CY (2003) *Large vocabulary off-line handwriting recognition using a constrained level building algorithm*, Pattern Analysis and Applications Journal, vol 6(2)/2003, pp. 97-121

[5] *NIST Special Database 19, Handprinted Forms and Characters Database*, `http://www.nist.gov/srd/nistsd19.htm`

[6] Pakchalakis S., Lee P. (1999) *Pattern recognition in gray level images using moment based invariant features*, IEE Conference on Image Processing and its Applications, Vol.465, pp. 245-249, 1999.

[7] Hu M.K. (1962) *Visual Pattern Recognition by Moment Invariant*, IRE Trans. Info. Theory, Vol. IT-8, pp. 179-187.

[8] Kohonen, T. (1995) *Self-Organizing Maps*, Springer-Verlag, Berlin Heidelberg.

[9] Ciresan, D., Cernazanu, C. (2004) *Linguistic modeling for automatic speech recognition in Romanian Language*, in Proc. of DAS04, 27-29 May, Suceava, Romania, pp. 490-497.

[10] Ahmed, M., Wart, R. (2002) *A rotation invariant rule-based thinning algorithm for character recognition,* IEEE TPAMI, Vol. 24, No. 12.