Genetic Algorithm Based Handwritten Numeric Strings Recognition

Maytham A. Shahed

Department of Computer Science, College of Science, University of Basrah
www.abualhail.com

Abstract

This paper presents an approach for the recognition of off-line handwritten numeric strings using genetic algorithm. The proposed scheme is divided in two parts. The first part is remove the image noise, then the vertical projection is used to segment the numeric strings at isolated digits and every digit will be presented separately to the second part. The second part using improved genetic algorithm to recognize isolated handwritten digit. The result of the recognition of the numeric strings will display at the exit of the global system.

Keywords: Handwritten digit, Genetic algorithms, Pattern recognition, vertical projection.

تمييز السلاسل الرقمية المعتمد على الخوارزمية الجينية

ميثم أبوالهيل شهيد
قسم علوم الحاسبات – كلية العلوم – جامعة البصرة
www.abualhail.com

الخلاصـــــة

يعرض البحث طريقة غير مباشرة لتمييز السلاسل الرقمية باستخدام الخوارزمية الجينية. تتكون التقنية المقترحة من جزئين: الأول، يقوم بإزالة تشويه الصورة، ومن ثم تقطيعها بطريقة الإسقاط العمودي إلى الأرقام المكونة للسلسلة المدخلة، وتقييسهم لغرض معالجتهم بعد ذلك. أما الجزء الثاني، فيستخدم خوارزمية جينية محسنة لتمييز نمط غير معروف واحد في كل مرة، ويكرر تطبيقها بعدد الأنماط المكونة للسلسلة المدخلة. نتيجة النظام المقترح طباعة الأرقام المكونة للسلسلة.

الكلمات المفتاحية: الأرقام اليدوية، الخوارزميات الجينية، تمييز الأنماط، طريقة الإسقاط العمودي.

1. Introduction

Automatic reading of numeric fields has been attempted in several application areas. One such area is the reading of courtesy amounts on bank checks. This application has been very popular in handwriting recognition research, due to the availability of relatively inexpensive CPU power, and the possibility to reduce considerably the manual effort, and errors involved in this task. Another application is the reading of postal ZIP codes in addresses written or typed on envelopes. The former is more difficult than the latter due to a number of differences in the nature of the handwritten material. For example, bank check systems have to take into account the great variability in the representation of a numerical amount and shape, e.g., the number of components to be identified, which is not necessary for a ZIP code system since the number of digits is fixed and known a priori. Another important requirement from a bank check system is its reliability. It has been estimated that such a system becomes commercially efficient only when the error rate is 1 percent or lower [1].

Methods for courtesy amount recognition belong to the class of digit recognition techniques although in many cases these amounts include also some non digit symbols such as commas, periods, strokes, currency names, etc. Strategies for digit string recognition can be divided into segmentation-then-recognition [1] and segmentation-based recognition [2]. In the first approach, the segmentation module provides a single sequence hypothesis where each subsequence should contain an isolated character which is submitted to the recognizer. This technique shows its limits rapidly when the correct segmentation does not fit with the predefined rules of the segmenter. Very often, contextual information is used during the segmentation process to improve the robustness of the system.

The second strategy is based on a probabilistic assumption where the final decision must express the best segmentation recognition score of the input image. Usually, the system yields a list of hypotheses from the segmentation module and each hypothesis is then evaluated by the recognition. Finally, the list is post processed taking into account the contextual information. Although this approach gives a better reliability than the previous one, the main drawback lies in the computational effort needed to compare all the hypotheses generated. Moreover, the recognition module has to discriminate various configurations such as fragments, isolated characters, and connected characters. In this strategy, segmentation can be explicit when based on cut rules [3,4] or implicit when each pixel column is a potential cut location [5,6].

According to the way handwriting data is generated, two different approaches can be distinguished [7]: on-line and off -line. In the former, the data are captured during the writing process by a special pen on an electronic surface. In the latter, the data are acquired by a scanner after the writing process is over. Off-line and on-line recognition systems are also discriminated by the applications they are devoted to. The off-line recognition is dedicated to bank check processing, mail sorting, reading of commercial forms, etc., while the on-line recognition is mainly dedicated to pen computing industry and security domains such as signature verification and author authentication.

This paper deals with numeric strings recognition and presents an effective scheme based on genetic algorithm. Improved genetic algorithm will be used and genetic operators enable the current recognition scheme to do refinement process along with a classification. In order to evaluate the robustness of the scheme, the experiments are present on two different databases: isolated digits from MNIST database available from internet and numeric strings for four writers.

This paper is structured as follows: Section 2 provides a brief overview of the genetic algorithms. Section 3 explains the proposed scheme and Section 4 reports the experiments carried out. Finally, Section 5 presents the conclusions.

2. Genetic Algorithms [8,9]

Genetic Algorithms (GAs) are guided, yet random search algorithms for complex optimization problems and are based on principles from natural evolutionary theory.

GAs are computationally simple yet powerful and do not require the search space to be continuous, differentiable, unimodal or of a functional form.

The GA process is illustrated in Figure 1.

```
Procedure GA;
Begin
    Initialization(pop);
    Evaluation(pop);
    While (not termination condition)
        Parents = Selection(pop);
        Offspring = Crossover(parents);
        Mutation(offspring);
        Pop = Replacement(offspring);
        Evaluation(pop);
    End;
End
```

Figure 1: The GA process

To obtain solutions, the problem space is initially encoded into a relevant format, suitable for evolutionary computation. The parameters of the search space are encoded in the form known as *chromosomes* (individuals) and each indivisible parameter in a chromosome is called a *gene*. A collection of such strings is called a *population* and a number of chromosomes in population are called a *population size*. Initially, a random population is created, which represents different points in the search space. An *objective* and *fitness* function is associated with each string that represents the degree of goodness of the chromosome. Based on the principle of survival of the fittest, a few of the strings are reproduced and each is assigned a number of copies that go into the mating pool. Biologically inspired operators like *crossover* and *mutation* are applied on these strings to yield a new generation of chromosomes. The process of reproduction (selection), crossover, and mutation continues for a fixed number of generations or till a termination condition is satisfied.

GA is useful when a sub-optimal solution is sufficient. GAs were designed to efficiently search large spaces, they have been used for a number of different application areas such as camera calibration, signature verification, medical diagnosis, facial modeling, and handwritten recognition.

3. The proposed genetic-based numeric strings recognition system

The aim of proposed system is to recognize the handwritten numeric string using GA.

The following figure explian the general diagram of the proposed system.

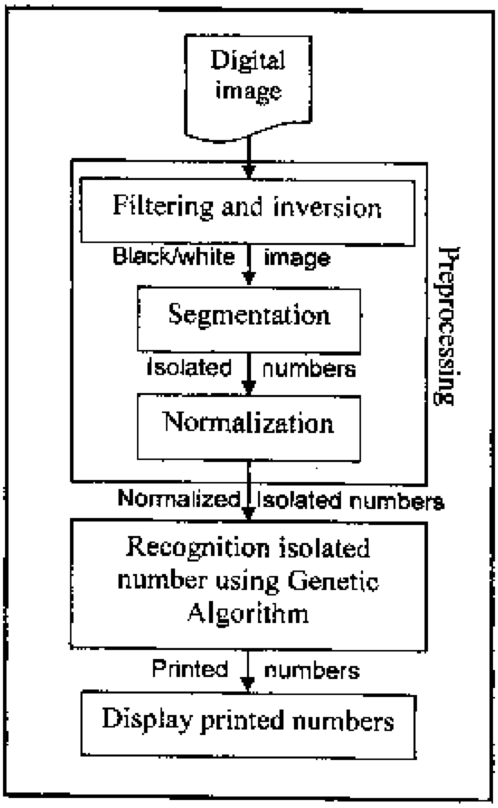

Figure 2: General diagram of the proposed system

The proposed system is divided into two main parts:

1. *Preprocessing*

The preprocessing attempts to eliminate some variability related to the writing process, such as the variability due to the writing environment, writing style, acquisition and digitizing of image [10,11]. In our case, the following preprocessing operations were used:

a). *Filtering and inversion of the gray levels*

During filtering the noises in the image due to different reasons (bad acquisition conditions, bad writing conditions, the writer's mood, .. etc.) was eliminated, in our case, some digits were marked *"peppers and salt"* noise [12], the application of the median filter on the digit image permitted us to eliminate this type of noise [13]. In order to reduce the amount of calculation we reversed the gray levels of the character image (black for the bottom and white for the object).

b). *Segmentation of the numeric strings*

This stage used the vertical projection to segment the numeric string into isolated numbers. The vertical projection for a given image is the number of the object pixels in every column of the image; it is obtained by the Equation 1 [10,13]:

$$V(j) = \sum_{i=1}^{n} IM(i, j) \qquad \ldots(1)$$

Where:

V(j): is the value of j order of the projection sequence.

IM(i,j): the value of the pixel whose coordinates are i and j in the image of the numeric strings.

n: is the number of lines of the numeric strings image.

The segmentation of the numeric strings in digits is achieved by exploiting the transition black/white (or white/black) when we sweep for the last line the image columns of the vertical projection.

c). *Normalization of the digit image*

Knowing that the images of the digits have variable sizes, this operation consists at normalizing the image size at 12×8 pixels by using nearest neighbor interpolation [13].

2. *Recognition of the numeric strings*

The objective of this part of the proposed system is to recognize an unknown pattern. It uses genetic algorithm to achieve this task. Every obtained digit by the normalization will be presented separately to the entry of the GA. We proposed an *improved GA* called IGA as shown in Figure 3.

```
Procedure IGA;
Begin
  Initialization(pop);
  Evaluation(pop);
  Sort(pop);
  While (not termination condition)
    N = population_size DIV 2;
    newpop(1) = pop(1);

    for i= 1 to N-1
      Parent1 = pop(i);
      Parent2 = pop(i+1);
      Offspring = Crossover (parent1,
                            parent2);
      Mutation(offspring);
      newpop(i+1) = offspring;
    Endfor;

    for j= 1 to N
      Parent1 = pop(j);
      Parent2 = pop(N+j);
      Offspring = Crossover (parent1,
                            parent2);
      Mutation(offspring);
      newpop(N+j) = offspring;
    Endfor;

    pop = newpop;
    Evaluation(pop);
    Sort(pop);
  End;
End
```

Figure 3: The proposed improved GA (IGA)

The improved GA (IGA) works as follows: after the initial population is generated and evaluated. The population is sorted, and a new population is created by applying the genetic operators of selection, crossover, and mutation on chromosomes of the current population as the following: the best chromosome will be passed to the new population without modification. After that the sorted population is divided in to two equal groups. From the fist group, we select every two adjacent chromosomes as parents for crossover and mutation to generate new offspring for the new population. This process will be created N-1 offspring (where N is the half of population size). Form both groups we select the chromosomes with the same position in their group as parents for crossover and mutation to generate new offspring for the new population. This process will be created N offspring. The generated population is evaluated and sorted. This process repeated several generations till termination conditions are satisfied.

The difference between the traditional GA and the proposed IGA is the selection scheme only. The selection scheme of IGA has the following good characteristics, which helps to increase the population diversity and improves the GA's performance, as well as preservation the population evolution:

- Giving the worst chromosomes the chance to participate in generating new population as the best chromosomes. This was done because there is a possibility of finding good characteristics in the worst chromosomes that can not be found in the best chromosomes.
- Prevents the repetition of the selected parents in the same generation and gives all chromosomes in population at least one chance always to participate in generating new population.

- Ensures that the fittest chromosome of each generation always propagate into the next generation.

The various steps of a GA for numeric strings recognition are described bellow:

a). *Representation scheme*

An integer encoding is used in our system. Each chromosome in the population is a possible solution, it is implemented as a vector of bytes and each gene represents the equivalent decimal number for the row (each row was considered as a binary number of 8 bit) in the unknown pattern (digit), therefore the length of chromosome is equal to 12. The representation of chromosome in the population is shown in Figure 4.

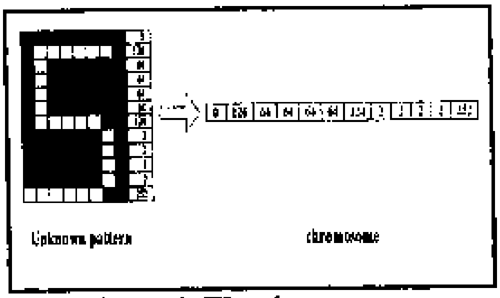

Figure 4: The chromosome representation

b). *Non-random initial population*

The initial population is generated depending on the input digit (unknown pattern) as in the following: the first chromosome in the population is the unknown pattern after it was transformed into a vector of bytes, and the other chromosomes are derived from the same unknown pattern with making random changes in random positions according to suitable probability or by shifting the unknown pattern horizontally and/or vertically with a certain number of dots, after that it is transformed into a vector of bytes.

c). *Fitness function*

In order to evaluate the goodness of a chromosome, five different templates were taken for each digit (0..9) and stored as group. The similarity (number of corresponding bits) between the unknown pattern and the stored templates are calculated. Then, the average of similarity for each group was computed and the largest average is the fitness value for this chromosome, as follows:

$$\text{Fitness} = \max\left(\frac{\sum_{j=1}^{5} F_i^j}{5}\right), \quad i = 0, 1, 2, \ldots, 9 \qquad ..(2)$$

Where:

F_i^j : the degree of similarity between the stored template j for the digit i and the unknown pattern.

i: digit index (i = 0, 1, 2,. . ., 9)

j: digit template index (j = 1,2,.., 5)

Here, the maximum value for fitness is (12×8 = 96).

d). *Genetic operators*

Now the next generation is created by applying the genetic operators, as follow:

Uniform crossover (UX) is applied, which is a probabilistic process that shuffles information between two parent chromosomes for generating new offspring The UX here uses two chromosomes to produce one chromosome in the crossover process. This process is summarized by a cancellation of any uncoincident pixels for both of chromosome patterns.

Mutation operator (1m) is applied, which is randomly picks up a gene in the chromosome and change its value properly in order to allow the GA to escape from a local optimal to search for a global optimal.

The genetic operators were applied on the current population to generate a new population.

e). ***Termination criteria***

The generation of new populations continues iteratively until that at least one of the following termination criteria is met:

1- The current generation reached to the 150 generation.
2- Obtain fitness greater than 90.
3- Repeat the best chromosome more than 30 sequence generations.

After the GA stopped, the fitness of the best chromosome was checked. If it is greater than 90 the system arrives to associate one and only one prototype to the recognized digit. Otherwise, the system doesn't take any decision of classification (rejected the digit).

After processing all the segments (digits), the system displays the result of the recognition of the handwritten numeric strings. A detailed example of the proposed system implementation is given by the Figure 5.

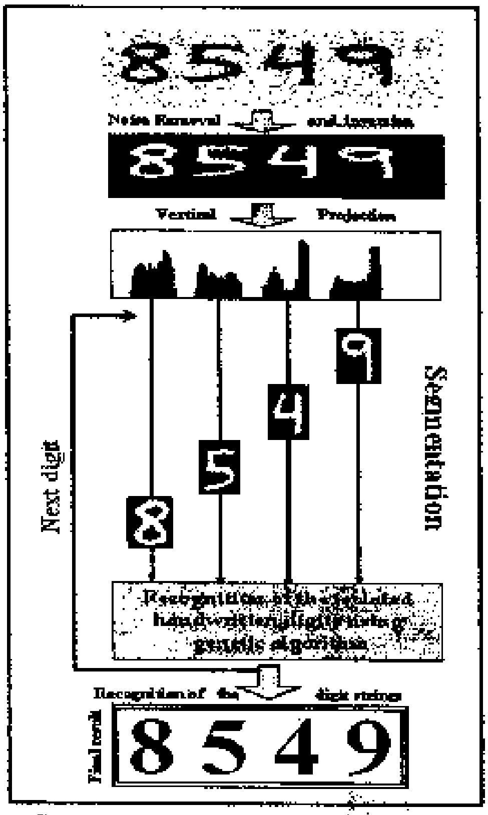

Figure 5: Detailed example of the proposed system implementation

4. Experiments and Results

In order to test the effectiveness of the proposed system, t has been applied to recognize different numeric strings. In recognizing handwritten numbers, there are some published experimental results. Since these results came from different datasets, authors, and evaluation metrics, we cannot compare the performance of the experimented classifiers. In this paper, the experiments were performed in two different contexts: isolated digits (10000 digits are derived from the original MNIST database [14]) and string of digits (500 samples database from four writers). Figure 6 show some examples of MINIST digits that used in tested our system.

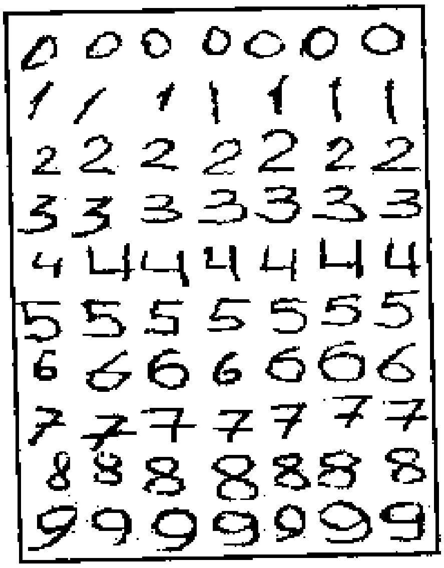

Figure 6: Examples of test digits from MINIST database

For all reported results, the following definitions of the recognition rate, error rate, rejection rate and reliability rate were used. Let B be a test set with N_B string images. If the recognition system rejects N_{rej}, classifies correctly N_{rec} and misclassifies the remaining N_{err}, then [15,16]:

$$\text{Recognition Rate (Rec\_R)} = \frac{N_{rec}}{N_B} \times 100 \quad \cdots (3)$$

$$\text{Error Rate (Err\_R)} = \frac{N_{err}}{N_B} \times 100 \quad \cdots (4)$$

$$\text{Rejection Rate (Rej\_R)} = \frac{N_{rej}}{N_B} \times 100 \quad \cdots (5)$$

$$\text{Reliability (Rel)} = \frac{\text{Rec\_R}}{\text{Rec\_R} + \text{Err\_R}} \times 100 \quad \cdots (6)$$

Therefore, the recognition rate, error rate and rejection rate sum up to 100%. In general, the recognition rate is a valid rate, but to characterize the quality of the classifier for practical issues, this rate is not appropriate. A more suitable rate for classifier examination in real applications is the reliability rate, which gives an impression of the classifier behavior in several different situations. Such a rate takes not only the error rate, but also the reject rate into consideration.

For all experiments, the following GA parameters are used:

- Population size: 16
- Probability of crossover (Pc): 0.9
- Probability of mutation (Pm): 0.005

Tables 1 and 2 summarize the results obtained by the proposed system for tested images with resolution 75 dpi and 300 dpi respectively.

Table 1: Results and different rates for resolution 75 dpi

String length	*Rec_R %*	*Err_R %*	*Rej_R %*	Rel%
1	86.83	5.98	7.18	93.55
2	90.05	3.98	5.95	95.76
3	89.1	8.5	2.4	91.29
4	91.85	4.82	3.32	95.01
6	89.84	6.27	3.88	93.47
Avg	**89.53**	**5.91**	**4.54**	**93.81**

Table 2: Results and different rates for resolution 300 dpi

String length	*Rec_R %*	*Err_R %*	*Rej_R %*	Rel%
1	93.23	3.01	3.76	96.12
2	95.65	2.33	2.01	97.94
3	94.07	2.26	3.67	96.25
4	95.26	2.35	2.39	97.55
6	94.91	2.67	2.42	97.51
Avg	**94.62**	**2.52**	**2.85**	**97.07**

It is obvious that the proposed system gives a good recognition rate when the resolution increases.

The proposed system is able to recognize any length of numeric strings.

Figure 7 shows some examples of correctly and misclassified numbers resulted by the proposed system.

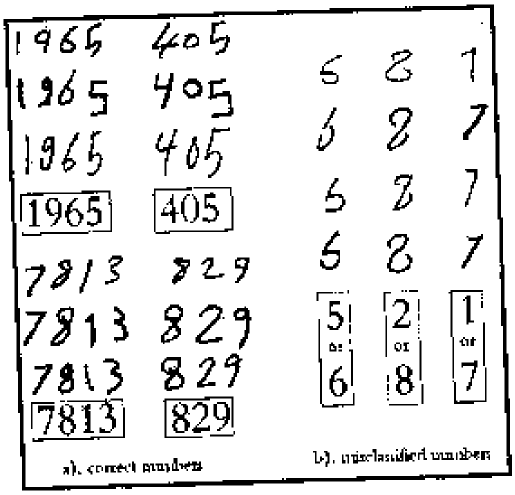

Figure 7: examples of (a).correctly recognize and (b).misclassified numbers

5. Conclusions

1. The using of GA as handwritten recognizer gives encouraging results.
2. The proposed system needs minimum preprocessing stages because the GA operators (crossover and mutation) enable the pattern recognition system to do refinement process along with classification.
3. The recognition rate depends on the resolution of the scanned pattern as shown in Tables 1 and 2.
4. Using an integer encoding in the proposed system minimized the chromosome length to 12 genes instead of 96 genes when using binary encoding. This helps to decrease the calculation time.
5. The IGA has the following good characteristics compared with traditional GA:
   - Giving the worst chromosomes the chance to participate in generation new population as the best chromosomes. This helps to increase the diversity of population. Which increase the IGA's chance to find the best solution.
   - It has an efficient selection method, which prevents the repetition of the selected parents in the same generation. This implies to increase the diversity of population.
   - Ensures that the best fittest chromosome of each generation will propagate into the next generation always; to preservation the good characteristics of the current population from lost or destroyed in the next generations.

But, the IGA take more times than traditional GA because it sorted the population in each generation.

6. References

[1]. M. Shridhar and A. Badreldin, **"Recognition of Isolated and Simply Connected Handwritten Numerals"**, Pattern Recognition, vol. 19, no. 1, pp. 1-12, 1986.

[2]. O. Matan and C. Burges, **"Recognizing Overlapping Hand-Printed Characters by Centered-Objects Integrated Segmentation and Recognition"**, Proc. Int'l Joint Conf. Neural Networks, pp. 504-511, 1991.

[3]. L.S. Oliveira, E. Lethelier, F. Bortolozzi, and R. Sabourin, **"A New Segmentation Approach for Handwritten Digits"**, Proc. 15th Int'l Conf. Pattern Recognition, vol. 2, pp. 323-326, 2000.

[4]. Y.K. Chen and J.F. Wang, **"Segmentation of Single- or Multiple-Touching Handwritten Numeral String Using Background and Foreground Analysis"**, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1304-1317, Nov. 2000.

[5]. A. Britto Jr., R. Sabourin, F. Bortolozzi, and C.Y. Suen, **"A Two-Stage HMM-Based System for Recognizing Handwritten Numeral Strings"**, Proc. Sixth Int'l Conf. Document Analysis and Recognition, pp. 396-400, 2001.

[6]. S.W. Lee and S.Y. Kim, **"Integrated Segmentation and Recognition of Handwritten Numerals with Cascade Neural Networks"**, IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 29, no. 2, pp. 285-290, 1999.

[7]. S. Ouchtati, B. Mouldi, and A. Lachouri, **"Segmentation and Recognition of Handwritten Numeric Chains"**, International Journal of Information Technology, vol. 4, no. 1, pp. 37-44, 2007.

[8]. R.L. Haupt and S. E. Haupt, **"Practical Genetic Algorithms"**, 2nd Edition, A John Wiley & Sons Inc., 2004.

[9]. M. Mitchell, **"An Introduction to Genetic Algorithm"**, MIT Press 1998.

[10]. R. Gonzales, and R. Woods, **"Digital Image Processing"**, 1st Edition, Addison Wesley PCI, 1992.

[11]. R. Steinert, M. Rehn, and A. Lansner, **"Recognition of handwritten digits using sparse codes generated by local feature extraction methods"**, European Symposium on Artificial Neural Networks ESANN'2006 proceedings, pp. 161-166, 2006.

[12]. Baker, **"Image Noise"**, Website: www.cs.nu.oz.au/~gavinb/download.php?file=noise.pdf

[13]. S. Shah, **"Introduction to Digital Image Processing"**, 2005, website: www.cs.uh.edu/~shah/teaching.

[14]. Y. Lecun. The MNIST database of handwritten digits. Available at http://yann.lecun.com/exdb/mnist/, 2004.

[15]. L. S. Oliveira, R. Sabourin, F. Bortolozzi, and Ch Y. Suen, **"Automatic recognition of handwritten numerical strings: A recognition and verification strategy"**, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 1483-1454, Nov. 2002.

[16]. L. S. Olivera, R. Sabourin, F. Bortolozzi, and C. Y. Suen, **"A Methodology For Feature Selection Using Multiobjective Genetic Algorithms For Handwritten Digit String Recognition"**, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 17, No. 6, pp. 903-929, 2003.