A Segmentation-Recognition Approach with a Fuzzy-Artificial Immune System for Unconstrained Handwritten Connected Digits

Hocine Merabti
LabSTIC Laboratory, 8 May 1945 University, BP-401, Guelma, 24000. Algeria
E-mail: merabti.dr@gmail.com

Brahim Farou and Hamid Seridi
LabSTIC Laboratory, Computer Science Department,
8 May 1945 University, BP-401, Guelma, 24000. Algeria
E-mail: farou@ymail.com, seridihamid@yahoo.fr

Keywords: pattern recognition, optical characters recognition, handwritten digit recognition, handwritten numeral string segmentation, artificial immune system (AIS), fuzzy logic

Received: October 24, 2017

In this paper, we propose an off-line system for the segmentation and recognition of the unconstrained handwritten connected digits. The proposed system provides new segmentation paths by finding two types of structural features. The background and foreground features points are found from the input string image. The possible cutting paths are generated from these features points. Each candidate component is evaluated individually based on its features points and its height. The output of the segmentation module is evaluated using the fuzzy-artificial immune system (Fuzzy-AIS). The latter performs a decision function on the resulting segments, and then the hypothesis that has the best score is regarded as the global decision. The experimental results on the ell-known handwritten digit database NIST SD19 show the effectiveness of the proposed system compared with other methods in both segmentation and recognition.

Povzetek: Razvit je sistem za segmentiranje in prepoznavanje ročno pisanih števk.

1 Introduction

The handwritten numeral string recognition has become a very open research area since their introduction in a wide range of application areas such: indexing and automatic processing of documents, automatic processing of bank checks, and automatic location of addresses and postal codes [1]. The aim of these applications is to reduce the manual effort involved in these tasks.

Handwriting recognition can be divided, according to the nature of the input, into two categories: on-line and off-line [2]. In the on-line case, the handwriting is produced by a pen or a mouse on an electronic surface and acquired as a time-dependent signal. In the off-line case, the handwriting is scanned on paper. Due to the variation in writing styles and the presence of overlapping and touching characters, the off-line recognition presents a good deal of challenging problems.

For building such off-line recognition system, the first step is the acquisition of the numeral string image followed by pre-processing operations on this image. Afterward, each numeral string is segmented into individual isolated digits. Finally, these digit images are sent to the classifier which assigns the corresponding class [3, 4, 5]. The segmentation of a string into isolated digits becomes one of the important challenges of handwritten recognition systems. Indeed, a very good recognition system can be practically useless when text identification and segmentation are performed poorly [6].

The segmentation problems are mainly related to several factors. First, the slope of the images or the noise introduced by the scanner. The variability in writing style and the inking defects caused by scripters. The variability and complexity of the character string shapes illustrated in the overlapping or the joining of two consecutive digits. Second, we do not know the number of characters in the string, and consequently, the optimal boundary between them is unknown [6].

To overcome these problems, many proposed solutions combine the segmentation and recognition processes. Performing a correct segmentation of an image involves knowing what it contains. On the other hand, if the recognition of the content of an image is correct, it means that the system has all the necessary information for the segmentation process.

The segmentation process can be divided into two classes: segmentation-then recognition and recognition-based [7] (Fig. 1). In the first class, the segmentation module tries to separate the connected characters by building a segmentation path. The latter contains a unique sequence hypothesis, and each subsequence should contain a single character to be submitted for recognition [8, 9]. In the second class, the process provides a set of segmentation hypotheses and defines the segmented digits by performing recognition of each provided segmentation hypothesis [10, 11]. This kind of approach gives good results because it provides several hypotheses that increase the classifier choice to find the correct recognition [6]. The segmentation can also be either explicit or implicit, as seen in Fig. 1. In the explicit methods, the segmentation is carried out prior to the recognition to provide candidate digits for the classifier [12, 13]. However, in the implicit methods, the segmentation is embedded in the recognition process, and it is performed simultaneously with recognition [14, 15]. Several works have proposed segmentation algorithms based on these two methods in the last few years. The literature has also shown that implicit segmentation offers very interesting perspectives, but explicit segmentation achieves better results [6, 16].

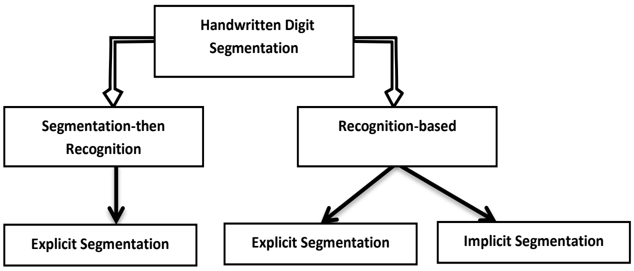

Figure 1: Segmentation and recognition of digits string methods (adapted from [6]).

Usually, segmentation can be conducted by the examination of the following three cases: connected digits, overlapped digits or distinct digits (as shown in Fig. 2). From these problems and in most instances, the connected and the overlapped digits are the most frequent situations observed in handwriting. Also, many algorithms have been proposed to deal with these situations [6, 17]. Some of them are based on features extracted from background pixels in the image [18], and others on features extracted from foreground pixels in the image [12]. Recently, several algorithms have used a combination of both these features [19, 20].

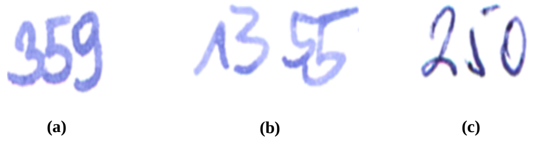

Figure 2: Main difficult examples; (a) connection (the 3 and the 5), (b) overlapping (the first and the second 5), (c) disjunction (in the 5).

Practically, to build a robust system for segmentation and recognition of connected handwritten digits, it is necessary to find: in the first, new methods to select or reduce the number of segmentation points which optimize the number of resulting segmentation hypotheses. In the second, new methods to eliminate the unnecessary segmentation paths which decrease the rejection rate. Finally, using the accurate classifiers on this type of data for keeping or increasing the recognition performance.

In this paper, we propose a new segmentation-recognition approach for handwritten numerical strings. Our work is focused on segmentation and recognition of connected digits, which present the main problematic in the segmentation through: selecting new features points for segmentation, evaluating the segmentation hypotheses to get more precise candidate segments, and using a good classifier for recognition. In segmentation process, we provide segmentation paths for separating the touched digits. This process is based on combining features from the background and foreground of the image. These features are used as segmentation points in the image. The fuzzy-artificial immune system (Fuzzy-AIS) is used for selecting the best segmentation hypotheses and properly classifying the separated digits. It also eliminates the arbitrary assignments in the decision phase when the dataset is overlapped, or the characteristics of objects are almost similar.

The paper is organized as follows. Section 2 presents a description of the proposed method. Section 3 is devoted to the experimental results. Finally, Section 4 concludes the paper.

2 Description of the proposed method

Our system consists of several stages: pre-processing, segmentation, feature extraction, and classification. The preprocessing module aims to remove the noise in the connected digits images and to simplify their further processing. The segmentation module allows providing the best set of candidate cutting paths for the input image and segmenting them into isolated individual digit images. The feature extraction module extracts some statistical and structural features from each digit image and represents them in a feature vector. Finally, the resulted features vectors are sent to the Fuzzy-AIS, and the corresponding class labels are assigned. An overview of the proposed system is shown in Fig. 3.

2.1 Pre-processing

The pre-processing module is applied to the strings image to eliminate or reduce the noise and to simplify the further processing. This module includes smoothing, binarization, dilation, and erosion.

- The smoothing is used to reduce the noise in the image.
- The binarization converts the image to a black and white.
- The dilation and erosion aim to close the disjoint edges and to smooth the global edges of the image.

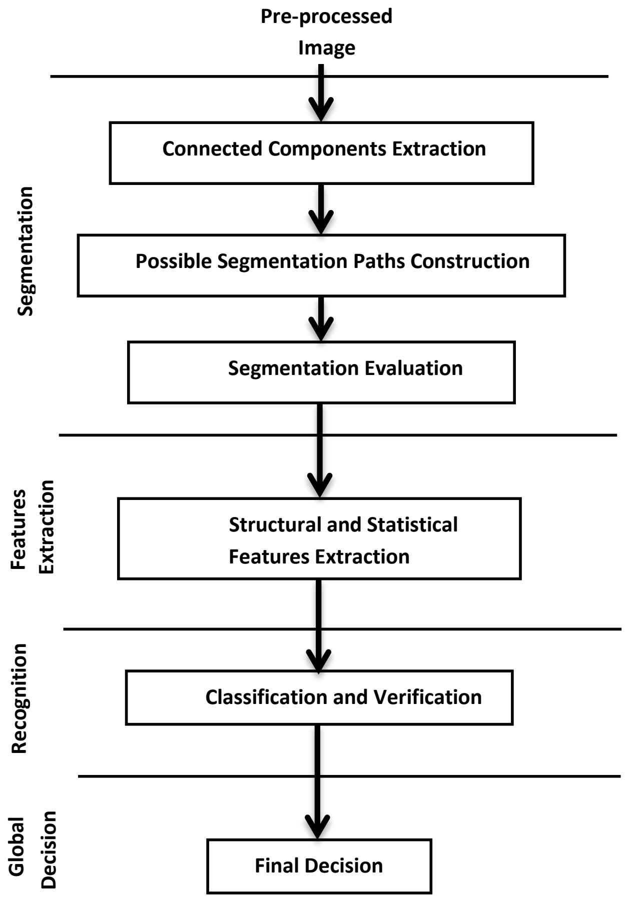
Pre-processed Image
Segmentation
Connected Components Extraction
Possible Segmentation Paths Construction
Segmentation Evaluation
Features Extraction
Structural and Statistical Features Extraction
Recognition
Classification and Verification
Global Decision
Final Decision

Figure 3: Flow diagram of the system.

Figure 4 shows a sample of the used database before and after the pre-processing stage.

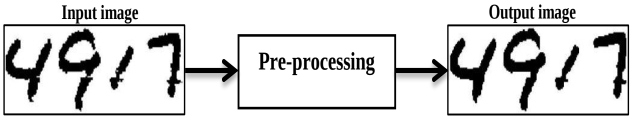

Figure 4: A sample before and after the pre-processing module.

2.2 Segmentation

The aim of the segmentation module is to segment the input string image into isolated digit images by providing the best set of candidate cutting paths. This module consists of three main steps: connected-components-extraction, touching connected-components-identification, and cutting paths constructing and evaluation of connected components (CCs) as seen in Fig. 5.

In the first step, the input image is separated into CCs. The second one allows detecting if a CC contains touched components (TC) or not by checking the following equation:

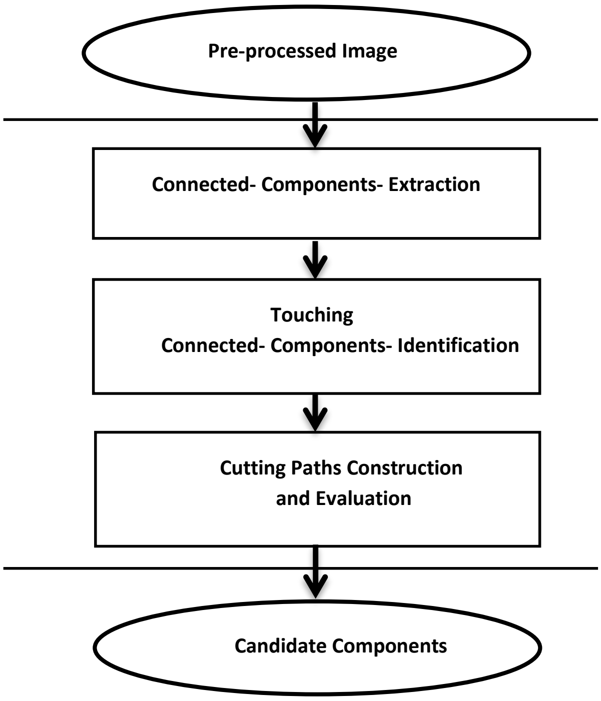
Pre-processed Image
Connected- Components- Extraction
Touching Connected- Components- Identification
Cutting Paths Construction and Evaluation
Candidate Components

Figure 5: Block diagram of the segmentation module.

$$TC = \begin{cases} 1, & \text{if } W_{CC} > \frac{\alpha * H}{100} \\ 0, & Otherwise \end{cases} \quad (1)$$

where, W_{CC} is the width of a CC, H is the height of the numeral string image, and α is a predefined parameter set in our case to 75.

In the case where the CC does not contain TC, then this CC is very likely to be a piece of a broken digit or a single digit. Figure 6 shows the extraction and identification process.

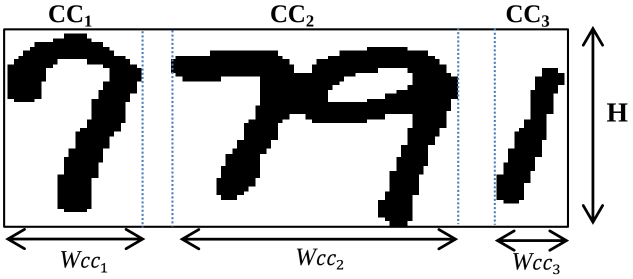

Figure 6: Extraction and identification of connected components; CC_2 with (W_{CC_2}) is higher than $\alpha\%$ of the height (H) of the numeral string image, and need further segmentation. CC_1 and CC_3 with (W_{CC_1}) and (W_{CC_3}) respectively, do not require any further segmentation.

The final step allows providing the optimum position for cutting a CC and extracting the correct candidate components. This step involves analyzing the foreground and background features of the CC to generate the segmentation points, followed by the generation of the possible cutting paths. The evaluation process is used to optimize the resulting segmentation paths and get more accurate results. In the following, we explain in detail, how to construct and evaluate the cutting paths for a CC.

2.2.1 Generating segmentation points

a. Profile features

The method of finding the profile features for a CC is as follows:

- Find the vertical upper and lower projection profiles of the CC, as seen in Fig. 7(b) and (c).
- Extract the upper and lower skeletons of these profiles, which are less and higher than the middle height (H) of CC (see Fig. 7(d) and (e)).
- Extract the end points (PFs) that have just one black neighbor pixel from the skeletons. The first and the last end points in each skeleton will not be considered (see Fig. 7(f)).

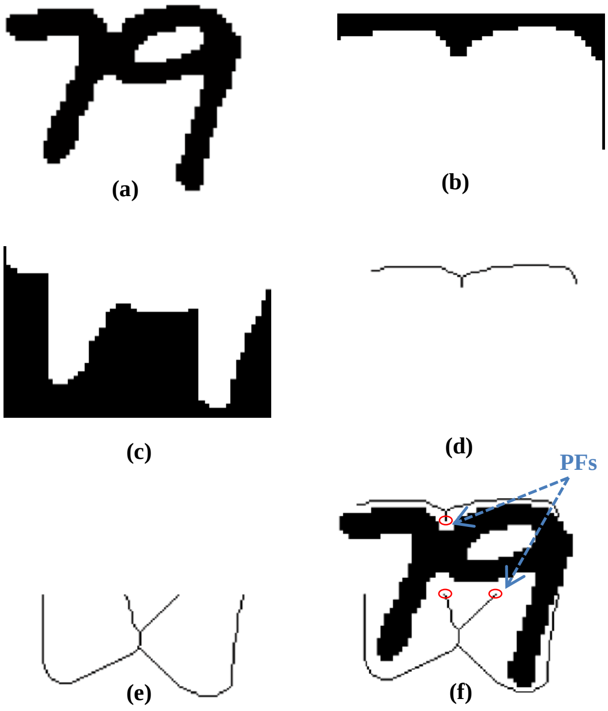

Figure 7: Profile features extraction; (a) Original image, (b) Upper projection profile, (c) Lower projection profile, (d) Upper skeleton profile, (e) Lower skeleton profile, (f) End points of the skeletons (PFs) (denoted by a red circle).

b. Skeleton and edge features

The following steps show how to find the skeleton and edge features:

- Extract the skeleton of the CC.
- Extract the intersection points (SFs) which have more than two black neighbor pixels from the skeleton (Fig. 8(b)).
- Extract the outer edge (upper/lower) from the CC, and add it to the skeleton image (Fig. 8(c)).
- Calculate the distance between the intersection point and the upper edge image, and select the points (EFs) that have the minimum value (Fig. 8(c)).
- Calculate the distance between the intersection point and the lower edge image, and select the points (EFs) that have the minimum value (Fig. 8(d)).

These feature points (SFs and EFs) show the proper location of segmentation regions.

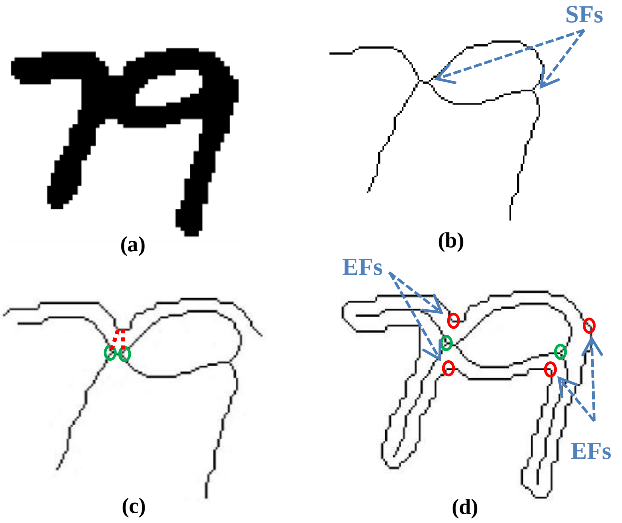

Figure 8: Skeleton (SFs) and Edge (EFs) features extraction; (a) Original image, (b) Skeleton of the CC with intersection points, (c) Upper edge of the CC superimposed on the skeleton image, (d) Edge points (denoted by a red circle).

2.2.2 Generating segmentation paths

All the feature points of the touching digits are found in the previous step. Now, the segmentation path can be generated from these points using two ways: from top to bottom, and from bottom to top. These feature points are connected together to construct the possible segmentation paths (Fig. 9). The two points P_1 and P_2 are connected according to the following equation:

$$| x_{P_1} - x_{P_2} | \leq \mu * (W_{CC}/2) \quad (2)$$

where, x_{P_1} and x_{P_2} are the horizontal coordinates of P_1 and P_2 respectively, μ is a constant parameter set empirically to 0.6, and W_{CC} is the horizontal width of the connected component.

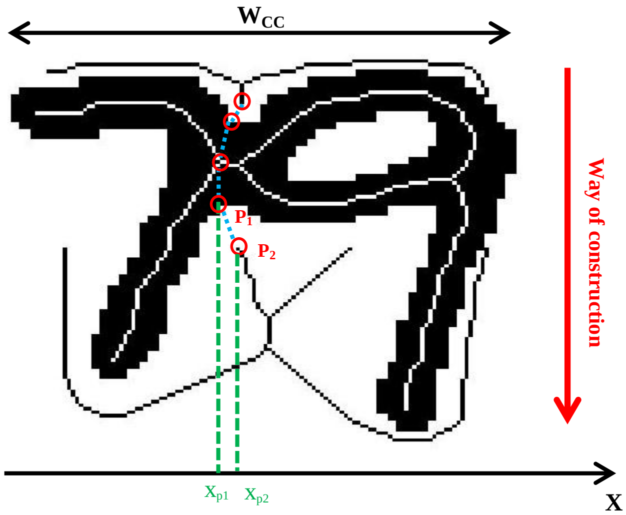

Figure 9: Construction of the segmentation path from feature points (from top to bottom).

The proposed method scans all possible relationships between PF, EF, and SF and generates the related segmentation paths according to the equation (2). Therefore, three hypotheses can be considered for the optimal segmentation path:

- **Hypothesis 1:** If the distance between the projection of the PF and the EF verifies the equation (2), then constructing a vertical segmentation path between these points (Fig. 10(a)).
- **Hypothesis 2:** If there is a skeleton path rather than one SF that linking both upper and lower EFs, this skeleton path is used as part of the vertical segmentation path (Fig. 10(b)).
- **Hypothesis 3:** If the CC does not contain SFs, the vertical segmentation path is constructed between PFs and the closest points PFs (Fig. 10(c)).

During the segmentation process, the segmentation paths may produce outliers: over-segmented parts (out-of-class) or under-segmented parts (non-digit patterns). The resulting segments that contain at least one outlier digit must be rejected using the evaluation of segmentation process.

2.2.3 Evaluation of the segmentation

After finding all possible segmentation paths, each one divides a CC into two new candidate connected components. At this stage, each candidate path is evaluated individually by using two constraints to evaluate our segmentation method and to get more precise results. The first constraint is related to the features points, while the second one is related to the height:

- **Constraint one:** if a candidate component is inside two possible segmentation paths with the same start and end points, then this candidate component is rejected (Fig. 11(a) and (b)).
- **Constraint two:** If the higher of a candidate component is lower than 20% of the height (H) of the image, then this candidate component is rejected (Fig. 11(c) and (d)).

Each segmentation hypothesis divides a CC into two or more new CCs. Now, all the new segments are normalized into a matrix of size 78 × 64 for preserving their aspect ratio. From each normalized segment, we extract a set of characteristics and represent them as a feature vector. The latter is introduced into the Fuzzy-AIS for the classification.

2.3 Features extraction

In this work, we extracted 39 statistical and structural features from the character. These features are based on Hu moments, zoning features, transitions histograms, and end and crossing points.

- **Hu moments:** seven invariant moments of Hu are computed from normalized and centralized moments up to order three of the segment [21]. They are invariant to translation, scaling, and rotation.
- **Zoning features:** this technique allows dividing the segment into several zones (a grid of $N \times M$), where the features are extracted from each zone [22]. We take the skeleton of each normalized segment, and we divide it into 3×2 zones. For each zone, we extract the density zoning and the gravity center. The density zoning represents the ratio of the number of black pixels on the total size of a zone [23]. The two coordinates of gravity center are used [24].
- **Transitions histograms:** this technique counts the number of transitions from foreground to Background in specified direction (horizontal, vertical and both diagonals 45°/135°). We extract the mean, the variance, and the max from each histogram.
- **End and crossing points:** the end point is a point that has just one black neighbor pixel. A crossing point connects three or more branches.

After extracting features and representing them in feature vectors, the resulting features vectors are sent to the Fuzzy-AIS for assigning the corresponding class labels.

2.4 Fuzzy-AIS for recognition and verification

An artificial immune system (AIS) is an adaptive system inspired by the principles and functioning of the natural immune system [25]. They are classes of algorithms that have properties and abilities very useful for pattern recognition, especially the classification problem [26, 27]. In our case, we coupled one of the best-known classification algorithms based on artificial immune systems, called the Artificial Immune Recognition System (AIRS) [28], with the Fuzzy-KNN approach.

The principle of AIRS algorithm is as follows: for a given training set of samples from a data class of interest (antigens); the AIRS returns a set of memory antibodies which are used to recognize this class. It is also characterized by:

– Self-regulation: the ability of adaptation and learning,

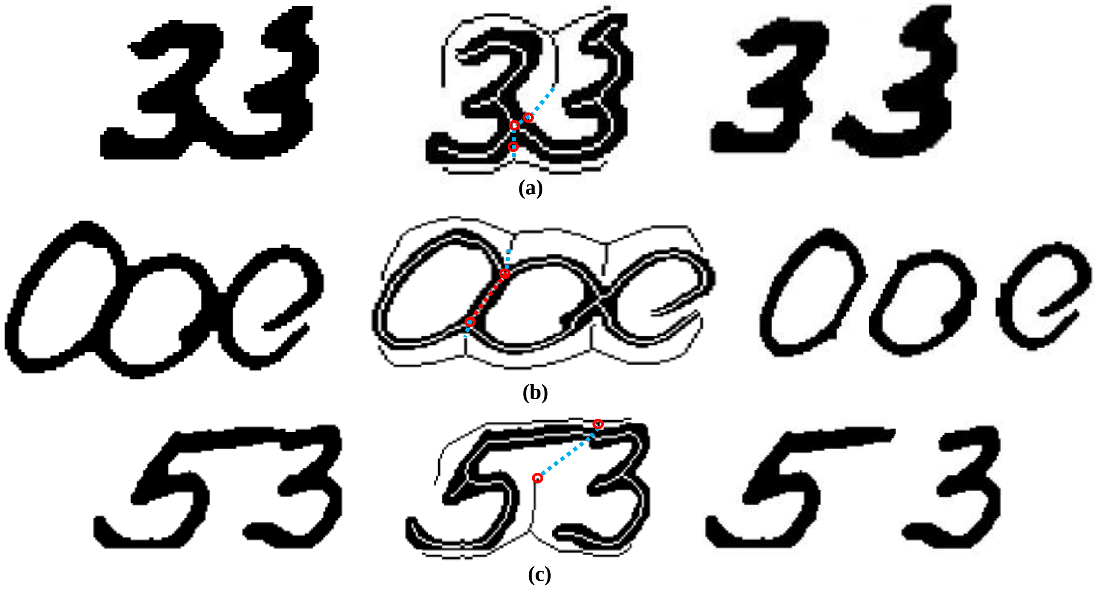

Figure 10: Hypotheses of the segmentation path; (a) Hypothesis 1, (b) Hypothesis 2, (c) Hypothesis 3.

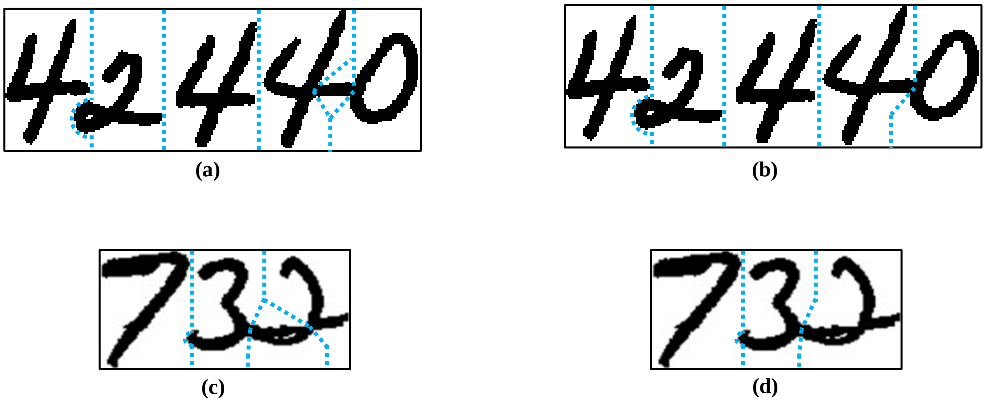

Figure 11: Effect of the evaluation method; (a) and (c) Cases of segmentation before evaluation, (b) and (d) Cases of segmentation after evaluation.

- Competitive performance: their results can be classified among the best works in the classification field,
- Generalization via data reduction: it allows reducing the database on a few training samples,
- Parameter stability: their parameter tuning on different data.

For more detail about this algorithm, the reader is referred to [28, 29].

The similarity measure is one of the most significant design choices in the development of an artificial immune system algorithm, and more precisely in their decision phase. The decision in most artificial immune systems algorithms is provided with the K-Nearest Neighbor approach. The latter has not the ability to correctly assign an object to a particular class when it belongs to other classes with the same value of similarity measure.

The decision will be random in the case when the dataset is overlapped, or the characteristics of the objects are almost similar. To overcome these limitations, the fuzzy concept is introduced in the decision phase, and it lies in the Fuzzy-KNN approach. It ensures that the arbitrary assignments are not made [30].

The Fuzzy-KNN approach finds the k Nearest Neighbors of the candidate component. Each candidate component D belongs to a class i with a membership value $mv_i(D)$. The latter depends on the class of its k Neighbors, and it is given by:

$$mv_i(D) = \frac{\sum_{j=1}^{k} mv_{ij}\left(\frac{1}{d(D,x_j)^{\frac{2}{(m-1)}}}\right)}{\sum_{j=1}^{k}\left(\frac{1}{d(D,x_j)^{\frac{2}{(m-1)}}}\right)} \quad (3)$$

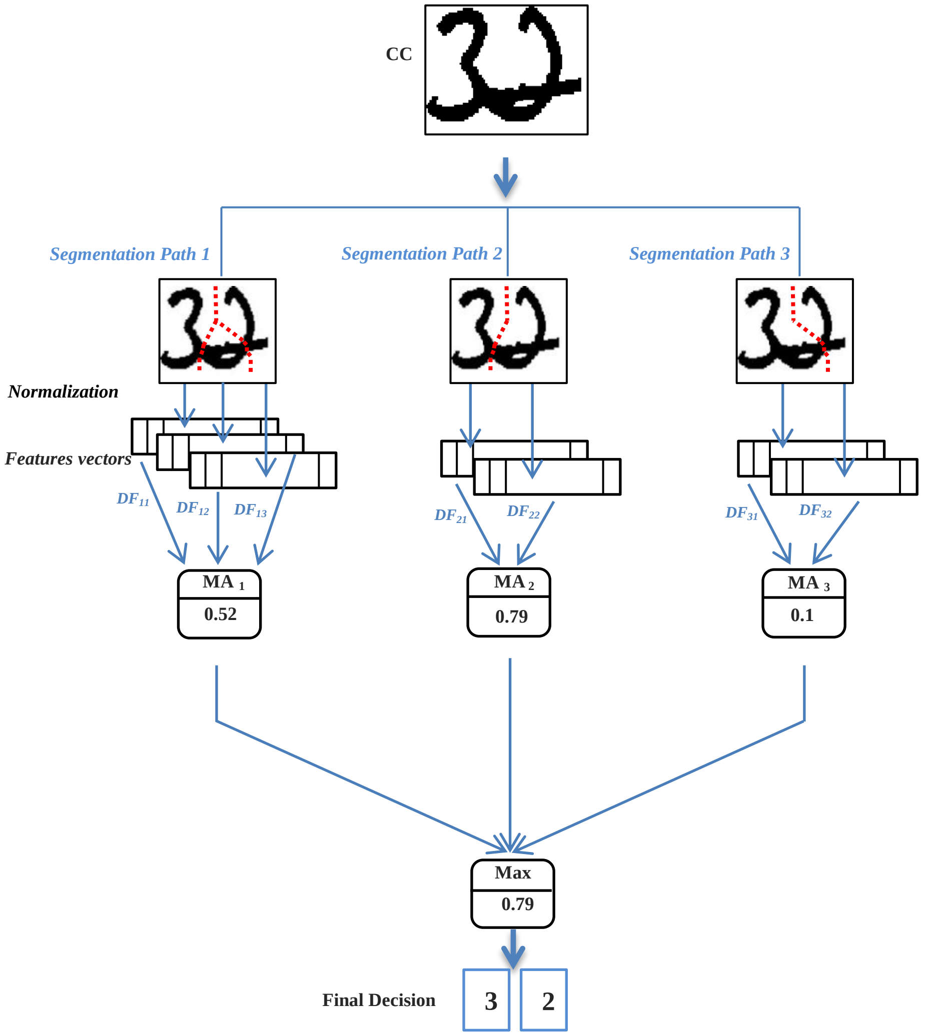

Figure 12: Segmentation followed by the Fuzzy-AIS as a recognition and verification strategy.

where, mv_{ij} is the membership in the ith class of the jth vector of the training set, $d(D, x_j)$ is the distance between D and its jth nearest neighbor x_j. The parameter m determines how heavily the distance is weighted when calculating the class membership.

In this stage, for each candidate component D, the classifier gives a membership to every class and assigns to it the class which has the highest membership value mv. For this reason, the Fuzzy-AIS classifier allows performing a set of decision functions (DF) on the segments of CC according to the following equation:

$$DF = \begin{cases} B * mv, & \text{if } mv < 0.5 \\ mv, & Otherwise \end{cases} \tag{4}$$

where, B is a predefined parameter set empirically to 0.75.

Afterward, the classifier calculates the average (MA) of DFs provided by each hypothesis. Finally, the maximum of these averages is regarded as the final decision function of the classifier, as seen in Fig. 12.

3 Experimental results

To evaluate the proposed method, we perform our experiments on the standard database NIST SD19, which contains unconstrained handwritten numeral strings with various lengths [31]. Our experiments were performed on two stages. In the first, the digit classifier was trained with isolated digit samples. Secondly, the digit classifier was applied to numeral string recognition.

3.1 Isolated handwritten digit recognition

In this stage, we divided the used database into two sets: a set of 2000 isolated digits used for the Fuzzy-AIS learning, and a set of 1500 isolated digits used for testing. The first stage of the Fuzzy-AIS learning consists in performing several tests to initialize the parameters: clonal_rate, hyper_clonal_rate, hypermutation_rate, mutation_rate and Affinity_threshold_scalar. These parameters are necessary for calculating the clones number, the ARBs resources, and the mutation function. The parameters selection of our classifier is shown in Table 1.

Fuzzy-AIS Parameters	**Values**
Clonal_rate	10
Hyper_clonal_rate	4
Mutation_rate	0.1
Hypermutation_rate	15
Affinity_threshold_scalar	0.01

Table 1: Parameters selection for Fuzzy-AIS.

After the training process, we obtained a recognition rate of 98.70% on the testing set. The main target of this work is to evaluate the performance of foreground and background features with the Fuzzy-AIS. Indeed, we are not trying to train the classifier with no digits, to optimize their accuracy or to compare the result with other works. In the next stage, we will discuss these issues and compare the performance of our system with other works.

3.2 Handwritten numeral string recognition

Our experiments were performed in two phases. In the first, we examined the performance of our segmentation module without using classification information. In the second, the segmentation is integrated with the recognition process to construct a segmentation-recognition system.

• In the first phase, we perform some experiments on the 3000 string images of the NIST SD19 database for evaluating our segmentation module. All images contain touching pairs of digits, but the module does not know the length of the string. Figure 10 shows some of the results of our segmentation module and Table 2 illustrates their performances.

As shown in Table 2, after the segmentation module, we made a visual analysis and verified in 95.86% of cases, the best segmentation path is among the paths generated by the module. In this case, the module does not know the length of the input digits string and some images produce more than one cutting path (see Fig. 13(a) and (b)). In 1.77 % of cases, the correct segmentation path is not among the produced paths, so we consider these cases as errors (see Fig. 13(e) and (f)). In 2.37 % of cases, the segmentation path is not produced on images; we consider these cases as rejected images (see Fig. 13(g) and (h)). The error and rejection cases are related to the overlapping connected digits. Among 95.86% of the correct segmentation paths, 87.3 % of them have only one segmentation path (see Fig. 13(c) and (d)).

Cases of segmentation path	**Visualization (%)**
Correct segmentation path	95.86 %
Errors	1.77 %
Rejection	2.37 %
Exactly one segmentation path	87.3 %

Table 2: Performances of handwriting pairs digit segmentation with our method on 3000 images of NIST SD19 database.

A comparison of this result with several segmentation algorithms proposed in the literature in the last few years is shown in Table 3.

Approaches	**2-digit Strings Number**	**Results (%)**
StR [32]	2000	88.70
Rb [33]	1000	93.77
Rb [9]	3287	94.8
Rb [34]	2069	95.84
Our Approach	**3000**	**95.86**

Table 3: Performance comparison of several works on touching pairs of digits. Rb: Recognition-based, StR: Segmentation-then Recognition.

Table 3 summarizes a set of segmentation algorithms, declares the number of samples used for testing, and shows their accuracy on touching pairs of digits.

As shown in Table 3, our approach gives good segmentation results in pairs of digits compared with others works.

• In the second phase of our experiments, the recognition module is introduced. It is based on the Fuzzy-AIS approach. We used 2000 images as training samples from the NIST SD19 Database, 200 images per class. For the testing stage, we randomly selected 1500 images. For each string length from 2, 3, 4, 5, 6, and 10, we took 250 images. To determine the performances of the proposed approach, we tested the influence and effectiveness of both: the evaluation method in the segmentation module, and the fuzzy concept in the recognition module. The performance results of our segmentation-recognition system are shown in Table 4.

Table 4 summarizes the recognition rates of our system on numeral strings recognition of lengths 2, 3, 4, 5, 6, and 10 digits. The results in Table 4 show that the use of the evaluation method in the segmentation module improves the performance of the proposed system (Fig. 11). This improvement is visible in both classifiers (AIS and Fuzzy-AIS). The system segment and recognize 96.55% of string samples with the use of AIS classifier, and 96.79% with the use of Fuzzy-AIS classifier. From these results, we notice that the introduction of the evaluation method increased the recognition rate by 11.6% in the case of AIS and 11.4% in the case of Fuzzy-AIS. However, the changeover from the AIS to the Fuzzy-AIS gave a slight improvement by 0.24% in the recognition rate. This is due to the efficiency of segmentation method. To discuss and compare the effective-

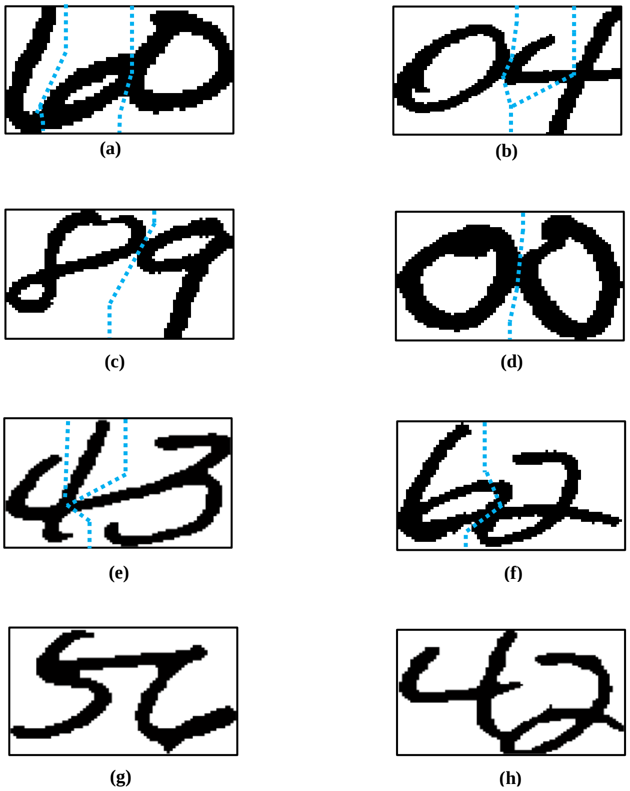

Figure 13: Some results of the segmentation module; (a) and (b) Case of correct segmentation, (c) and (d) Case of exactly segmentation, (e) and (f) Case of error, (g) and (h) Case of rejection.

ness of the proposed approach, we compare our results with others recent approaches on the same database (Table 5).

The results in Table 5 indicate that our system is promising and compare favorably with the other works.

4 Conclusion

In this paper, we proposed a new system to recognize unconstrained handwritten digit strings. We used a segmentation-recognition strategy for handwritten connected digits based on structural features and the Fuzzy-artificial immune system. First, we combined the background and foreground analysis for extracting the feature points. For the background features, we applied a thinning procedure to the vertical projection profile of the image. For the foreground features, we applied a thinning procedure on the connected component and their edge. These feature points are linked to generate the possible segmentation paths in connecting digits. The resulted candidate segmentation paths are evaluated for removing the useless among them and keeping the best. The evaluation process is based on two main constraints. The first one is related to the features points of the candidate segmentation paths and the second one is related to its height. Finally, we introduced the Fuzzy-AIS classifier for ranking all possible segmentation paths and considering the best of them as the global decision. The introduction of both the evaluation process in the segmentation module and the fuzzy concept in the decision phase allowed increasing the recognition rate.

Our experiments on the NIST SD19 database show that our system gets good results in both segmentation and recognition and compare favorably with other works in the same database.

References

[1] Gayathri, P. and Ayyappan, S. (2014) 'Off-line handwritten character recognition using Hidden Markov Model', *in Proceeding of the International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, pp.518–523.

[2] Lacerda, E. B. and Mello, C. A.(2013) 'Segmentation of connected handwritten digits using Self-Organizing Maps', *Expert Systems with Applications*, Vol. 40, no. 15, pp.5867–5877.

[3] Saba, T., Rehman, A. and Elarbi-Boudihir, M. (2014) 'Methods and strategies on off-line cursive touched characters segmentation: a directional review', *Artificial Intelligence Review*, Vol. 42, pp.1047–1066.

String Length	Recognition Rate (%)			
	Without Evaluation Method		With Evaluation Method	
	AIS	Fuzzy-AIS	AIS	Fuzzy-AIS
2	84.33	85.66	97.33	98.00
3	88.44	88.88	97.11	97.33
4	84.83	85.33	96.33	96.66
5	84.00	84.26	96.00	96.13
6	80.00	80.11	95.22	95.33
10	88.12	88.12	97.33	97.33
Average rates	**84.95**	**85.39**	**96.55**	**96.79**

Table 4: Experimental results of our segmentation-recognition approach.

String Length	Recognition Rate (%)				
	Approaches				
	[16]	[35]	[13]	[34]	**Our approach**
2	96.88	94.8	98.94	98.57	98.00
3	95.38	91.6	97.23	96.28	97.33
4	93.38	91.3	96.16	96.12	96.66
5	92.40	88.3	95.86	94.73	96.13
6	93.12	89.1	96.10	95.02	95.33
10	90.24	86.9	94.25	90.46	97.33
Average rates	93.57	90.33	96.42	95.63	**96.79**

Table 5: A comparison with others works.

[4] El Kessab, B., Daoui, C., Bouikhalene, B. and Salouan, R. (2014) ‘A Comparative Study between the K-Nearest Neighbours and the Multi-Layer Perceptron for Cursive Handwritten Arabic Numerals Recognition’, *International Journal of Computer Applications (0975–8887)*, Vol. 107, No. 21.

[5] El Kessab, B., Daoui, C., Bouikhalene, B. and Salouan, R. (2015) ‘A comparative study between the support vectors machines and the k-nearest neighbors in the handwritten latin numerals Recognition’, *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 8, No. 2, pp.325–336.

[6] Ribas, F. C., Oliveira, L. S., Britto Jr, A. S. and Sabourin, R. (2013) ‘Handwritten digit segmentation: a comparative study’, *International Journal on Document Analysis and Recognition (IJDAR)*, Vol. 16, no. 2, pp.127–137.

[7] Casey, R. G. and Lecolinet, E. (1996) ‘A survey of methods and strategies in character segmentation’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, no. 7, pp.690–706.

[8] Shi, Z. and Govindaraju, V. (1997) ‘Segmentation and recognition of connected handwritten numeral strings’, *Pattern Recognition*, Vol. 30, no. 9, pp.1501–1504.

[9] Yu, D. and Yan, H. (2001) ‘Separation of touching handwritten multi-numeral strings based on morphological structural features’, *Pattern Recognition*, Vol. 34, no. 3, pp.587–599.

[10] Gattal, A. and Chibani, Y. (2015) ‘SVM-Based Segmentation-Verification of Handwritten Connected Digits Using the Oriented Sliding Window’,*International Journal of Computational Intelligence and Applications*, Vol. 14, no. 1, pp.1550005.

[11] Fujisawa, H., Nakano, Y. and Kurino, K. (1992) ‘Segmentation methods for character recognition: from segmentation to document structure analysis’, *Proceedings of the IEEE*, Vol. 80, no. 7, pp.1079–1092.

[12] Pal, U., Belaid, A. and Choisy, Ch. (2003) ‘Touching numeral segmentation using water reservoir concept’, *Pattern Recognition Letters*, Vol. 24, no. 1, pp.261–272.

[13] Sadri, J., Suen, C.Y. and Bui, T.D. (2007) 'A genetic framework using contextual knowledge for segmentation and recognition of handwritten numeral strings', *Pattern Recognition*, Vol. 40, no. 3, pp.898–919.

[14] Procter, S., Illingworth, J., and Elms, A.J. (1998) 'The recognition of handwritten digit strings of unknown length using hidden markov models', *In Proceedings of the 14th International Conference on Pattern Recognition*, pp.1515–1517.

[15] Choi, S. M. and Oh, I. S. (1999) 'A segmentation-free recognition of two touching numerals using neural networks', *In Proceedings of the 5th International Conference on Document Analysis and Recognition (ICDAR'99)*, IEEE, Bangalore, India, pp.253–256.

[16] Oliveira, L. S., Sabourin, R., Bortolozzi, F. and Suen, C. Y. (2002) 'automatic recognition of handwritten numerical strings: a recognition and verification strategy', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, no. 11, pp.1438–1454.

[17] Kulkarni, R. V. and Vasambekar, P. N. (2010) 'An overview of segmentation techniques for handwritten connected digits', *In Proceedings of the International Conference on Signal and Image Processing (ICSIP)*, IEEE, pp.479–482.

[18] Ayat, N.E., Cheriet, M. and Suen, C.Y. (2000) 'Un systme neuro-flou pour la reconnaissance de montants numriques de chques arabes', *In Colloque international francophone sur l'crit et le document*, pp.171–180.

[19] Oliveira, L. S., Lethelier, E., Bortolozzi, F. and Sabourin, R. (2000) 'A new approach to segment handwritten digits', *In Proceedings of the 7th International Workshop on Frontiers in Handwriting Recognition*, Amsterdam, The Netherlands, pp.577–582.

[20] Sadri, J., Suen, C. Y. and Bui, T. D. (2004) 'Automatic segmentation of unconstrained handwritten numeral strings', *In Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR-9)*, IEEE, Tokyo, Japan, pp.317–322.

[21] Cash, G. L. and Hatamian, M. (1987) 'Optical character recognition by the method of moments', *Computer Vision, Graphics and Image Processing*, Vol. 39, no. 3, pp.291–310.

[22] Hirabara, L. Y., Aires, S. B., Freitas, C. O., Britto Jr, A. S. and Sabourin, R. (2011) 'Dynamic zoning selection for handwritten character recognition', *In Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*,Pucn, Chile, pp.507–514.

[23] Parker, J.R. (1993) *Practical computer vision using C*, John Wiley and Sons, Inc., New York.

[24] Gorgevik, D. and Cakmakov, D. (2004) 'An efficient three-stage classifier for handwritten digit recognition. In Pattern Recognition', *In Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, IEEE, pp.507–510.

[25] Timmis, J., Andrews, P.S., Owens, N. and Clark, E. (2008) 'An interdisciplinary perspective on artificial immune systems', *Evolutionary Intelligence*, Vol. 1, no. 1, pp.5–26.

[26] De Castro, L. N. and Timmis, J. (2002) 'Artificial Immune Systems: A Novel Paradigm to Pattern Recognition', *Artificial Neural Networks in Pattern Recognition*, Vol. 1 , pp.67–84.

[27] Yang, Y. (2011) 'Application of artificial immune System in handwritten Russian Uppercase character recognition', *In Proceedings of the International Conference on Computer Science and Service System (CSSS)*, IEEE, pp.238–241.

[28] Watkins, A., Timmis, J. and Boggess, L. (2004) 'Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm', *Genetic Programming and Evolvable Machines*, Vol. 5 , no. 3, pp.291–317.

[29] Watkins, A. and Boggess, L. (2002) 'A New Classifier Based on Resource Limited Artificial Immune Systems', *In Proceedings of the 2002 Congress on Evolutionary Computation CEC'O2, Part of the World Congress on Computational Intelligence.*, IEEE, Honolulu, HI, USA, pp.1546–1551.

[30] Keller, J.M., Gray, M.R. and Givens, J.A. (1985) 'A Fuzzy K-Nearest Neighbor Algorithm', *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-15, no. 4, pp.580–585.

[31] Grother, P. J. (1995) 'NIST Special Database 19; Handprinted Forms and Characters Database', *National Institute of Standards and Technology (NIST).*

[32] Suwa, M. and Naoi, S. (2004) 'Segmentation of Handwritten Numerals by Graph Representation', *In Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition*, IEEE, Tokyo, Japan, pp.334–339.

[33] Ciresan, D. (2008) 'Avoiding segmentation in multi-digit numeral string recognition by combining single and two-digit classifiers trained without negative examples', *In Proceedings of the 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'08)*, Timisoara, Romania, pp.225–230.

[34] Cavalin, P. R. (2006) 'An implicit segmentation-based method for recognition of handwritten strings of characters', *In Proceedings of the 2006 ACM Symposium*

on Applied Computing, ACM, Dijon, France, pp.836–840.

[35] Britto Jr, A. D. S., Sabourin, R., Bortolozzi, F. and Suen, C.Y. (2003) 'The recognition of handwritten numeral strings using a two-stage HMM-based method', *International Journal on Document Analysis and Recognition*, Vol. 5, No. 2-3, pp.102–117.