An Enhanced HMM Topology in an LBA Framework for the Recognition of Handwritten Numeral Strings

Alceu de S. Britto Jr.[1,2], Robert Sabourin[1,3,4], Flavio Bortolozzi[1] and Ching Y. Suen[4]

[1] Pontifícia Universidade Católica do Paraná (PUCPR), Curitiba, Brazil
[2] Universidade Estadual de Ponta Grossa (UEPG), Ponta Grossa, Brazil
[3] École de Technologie Supérieure (ETS), Montreal, Canada
[4] Centre for Pattern Recognition and Machine Intelligence (CENPARMI) Montreal, Canada

Abstract. In this study we evaluate different HMM topologies in terms of recognition of handwritten numeral strings by considering the framework of the Level Building Algorithm (LBA). By including an end-state in a left-to-right HMM structure we observe a significant improvement in the string recognition performance since it provides a better definition of the segmentation cuts by the LBA. In addition, this end-state allows us the use of a two-step training mechanism with the objective of integrating handwriting-specific knowledge into the numeral models to obtain a more accurate representation of numeral strings. The contextual information regarding the interaction between adjacent numerals in strings (spaces, overlapping and touching) is modeled in a pause model built into the numeral HMMs. This has shown to be a promising approach even though it is really dependent on the training database.

Keywords: HMM topology, LBA framework, pause model, handwritten numeral string recognition

1. Introduction

Hidden Markov models have been successfully applied to various pattern recognition environments. Specially to handwriting recognition, in which the HMM approach provides a way of avoiding the prior segmentation of words into characters usually found in OCR systems. This has shown to be a promising strategy, since often word segmentation without the help of a recognizer is difficult or impossible. In this context, the Level Building Algorithm (LBA) [1] is fundamental, since it allows to match models against an observation sequence, without having to first segment the sequence into subsequences that may have been produced by different models.

In this study the LBA is used to recognize handwritten numeral strings using an implicit segmentation-based approach. In this framework we focus on the HMM topology, which is very important in providing a precise matching of the numeral models against the observation sequence representing a numeral string. In addition,

we investigate a way of integrating in the numeral models some contextual information regarding the interaction between adjacent numerals in strings. For this purpose, we use a two-step training mechanism, in which numeral models previously trained on isolated digits are submitted to a string-based training. In the second step of this training mechanism a pause model is built into the numeral models. Cho, Lee and Kim [2] also use pauses in off-line word recognition and show some significant improvement on the recognition accuracy depending on the dictionary size. They use a number of pause models in order to describe categories of character transitions depending on the neighboring characters. In contrast to their approach, Dolfing [3] assumes that the number of ligatures is limited and models all ligatures with the same pause model. In our work we evaluate two strategies: 1) constructing one pause model by numeral class; 2) constructing one pause model representing all numeral classes.

This work is organized into 6 sections. Section 2 describes our system for the recognition of handwritten numeral strings. Section 3 presents the HMM topologies evaluated in this work. The experiments and discussions are summarized, respectively, in Section 4 and 5. Finally, we draw a conclusion in Section 6.

2. System Outline

The system architecture is shown in Figure 1. In the first module, a word slant normalization method has been modified by considering the slant and contour length of each connected component to estimate the slant of handwritten numeral strings. A detailed description of this process is presented in [4].

In the Segmentation-Recognition module, the string recognition is carried out using an implicit segmentation-based method. This module matches numeral HMMs against the string using LBA. To this end, the numeral string is scanned from left-to-right, while local and global features are extracted from each column. The local features are based on transitions from background to foreground pixels and *vice versa*. For each transition, the mean direction and corresponding variance are obtained by means of the statistic estimators defined in [5].

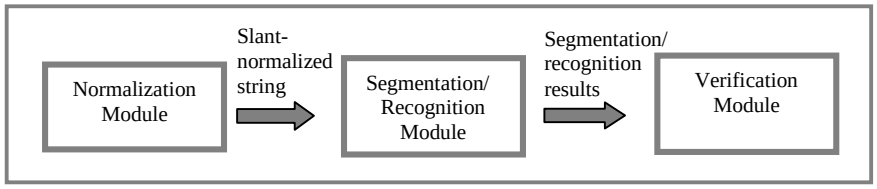

Figure 1. System architecture

These estimators are more suitable for directional observations, since they are based on a circular scale. For instance, given the directional observations $\alpha_1 = 1^o$ and $\alpha_2 = 359^o$, they provide a mean direction ($\overline{\alpha}$) of 0^o instead of 180^o calculated by conventional estimators.

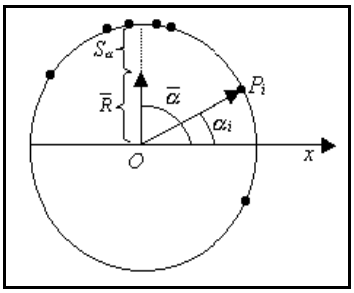

Figure 2. Circular mean direction $\overline{\alpha}$ and variance S_α for a distribution $F(\alpha_i)$

Let $\alpha_1,\ldots,\alpha_i,\ldots\alpha_N$ be a set of directional observations with distribution $F(\alpha_i)$ and size N. Figure 2 shows that α_i represents the angle between the unit vector $\overline{OP_i}$ and the horizontal axis, while P_i is the intersection point between $\overline{OP_i}$ and the unit circle. The cartesian coordinates of P_i are defined as:

$$(cos(\alpha_i),\ sin(\alpha_i)) \tag{5}$$

The circular mean direction $\overline{\alpha}$ of the N directional observations on the unit circle corresponds to the direction of the resulting vector $(\overline{R})$ obtained by the sum of the unit vectors $(\overline{OP_1},\ldots,\overline{OP_i},\ldots,\overline{OP_N})$. The center of gravity $(\overline{C},\overline{S})$ of the N coordinates $(cos(\alpha_i), sin(\alpha_i))$ is defined as:

$$\overline{C} = \frac{1}{N}\sum_{i=1}^{N}\cos(\alpha_i) \tag{6}$$

$$\overline{S} = \frac{1}{N}\sum_{i=1}^{N}\sin(\alpha_i) \tag{7}$$

These coordinates are used to estimate the mean size of $\overline{R}$, as:

$$\overline{R} = \sqrt{(\overline{C}^2 + \overline{S}^2)} \tag{8}$$

Then, the circular mean direction can be obtained by solving one of the following equations:

$$sin(\overline{\alpha}) = \frac{\overline{S}}{\overline{R}} \quad \text{or} \quad cos(\overline{\alpha}) = \frac{\overline{C}}{\overline{R}} \tag{9}$$

Finally, the circular variance of $\overline{\alpha}$ is calculated as:

$$S_\alpha = 1 - \overline{R} \qquad 0 \le S_\alpha \le 1 \tag{10}$$

To estimate $\overline{\alpha}$ and S_α for each transition of a numeral image, we have considered $\{0^0,\ 45^0,\ 90^0,\ 135^0,\ 180^0,\ 225^0,\ 270^0,\ 315^0\}$ as the set of directional

observations, while $F(\alpha_i)$ is computed by counting the number of successive black pixels over the direction α_i from a transition until the encounter of a white pixel. In Figure 3 the transitions in a column of numeral 5 are enumerated from 1 to 6, and the possible directional observations from transitions 3 and 6 are shown.

In addition to this directional information, we have calculated two other local features: a) relative position of each transition [6], taking into account the top of the digit bounding box, and b) whether the transition belongs to the outer or inner contour, which shows the presence of loops in the numeral image. Since for each column we consider 8 possible transitions, at this point our feature vector is composed of 32 features. The global features are based on horizontal projection (HP) of black pixels for each column, and the derivative of HP between adjacent columns. This constitutes a total of 34 features extracted from each column image and normalized between 0-1. A codebook with 128 entries is created using the LBG algorithm [7].

The last system module is based on an isolated digit classifier, which is under construction. This verification module was not used in the experiments described in this paper.

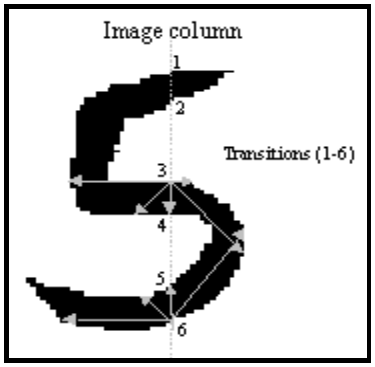

Figure 3. Transitions in a column image of numeral 5, and the directional observations used to estimate the mean direction for transitions 3 and 6

3. Topology of the Numeral HMMs

The topology of the numeral models is defined taking into account the feature extraction method and considering the use of the LBA. The number of states is experimentally defined based on the recognition of isolated numerals. The HMM topology used in the baseline system is shown in Figure 4.

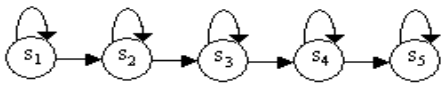

Figure 4. Left-to-right HMM model with 5 states

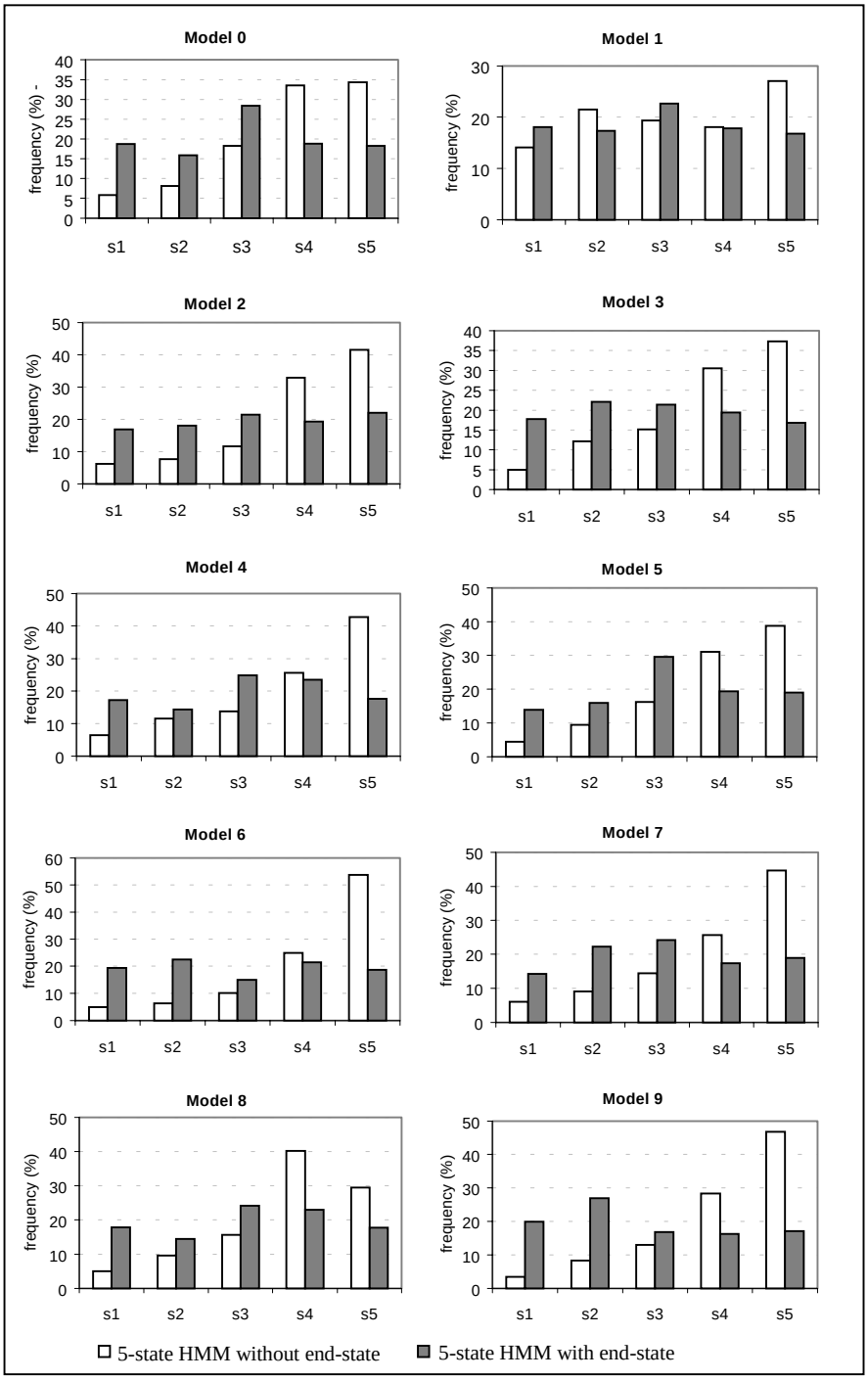

Figure 5. Distributions of observations among the HMM states computed during model training

The same or similar topology can be found in related works. In [8] the authors use it to model character classes to recognize fax printed words. The same structure is used for modeling numeral classes to recognize handwritten numeral strings in [9]. Similar HMM topology, with additional skip transitions, is used for modeling airlines vocabulary in [10]. In all these works the LBA is used as a recognition algorithm.

As we can see the HMM topology used in our baseline system does not present additional states or transitions to allow the concatenation of numeral models, since they are not necessary in the LBA framework. In this system 10 numeral models independently trained on isolated numerals are used to recognize strings, and the LBA is responsible for finding the best sequence of these models for a given numeral string. However, this kind of topology does not allow us to model the interaction between adjacent numerals in strings. Moreover, in the experiments on numeral strings we have observed a significant loss in terms of recognition performance as the string length increases. In order to better understand the behavior of these numeral models, we compute the distribution of observations among the HMM states during the training of them on 50,000 isolated numerals (5,000 samples per class).

We can see the corresponding distributions in Figure 5 as *5-state HMM without end-state*. These unbalanced distributions of observations among the states, associated with the presence of a self-transition with probability value equal to 1.0 in the last state (s_5), have a negative impact on the system segmentation performance. To better explain, let us consider the paths A and B in the LBA trellis in Figure 6, which share the same way until time t=4. Path A reaches the state 5 (s_5) first (at time t=6). From this time path B may not reach the last state even being a promising path. This may happen because the transition probability from state 4 to 5 (a_{45}) (a small value because of the nature of the distributions observed), must compete with the self-transition probability on state 5 (equal to 1.0 since there is no transition going out of this state). Under this condition the numeral recognition at this level may succeed, however without representing the best segmentation path. This non-optimum matching can bring problems to the next levels. This explains why, in the baseline system, the recognition of numeral strings drops drastically as their length increases (see Section 4).

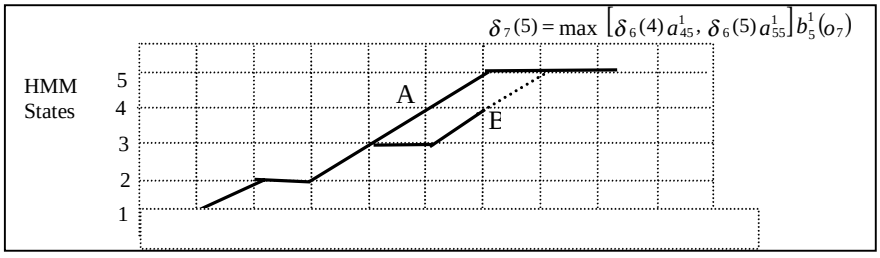

Figure 6. Paths A and B in an LBA level considering model λ_1

To deal with this problem and also adapt the numeral models to a string-based training, we include an end-state in the HMM topology (see Figure 7). The new models show a better distribution of the observations among their states, as we can see in Figure 5 (*5-state HMM with end-st*ate), and avoid a self-transition with probability value equal to 1.0 in the state 5 (s_5). The end-state does not absorb any

observation and it is useful to concatenate the numeral models during a string-based training. The positive impact of this modification on the HMM topology to the string recognition is shown in Section 4.

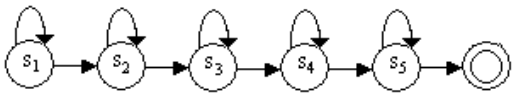

Figure 7. 5-state HMM with an end-state

Based on this new topology, we can pay some attention to the possibility of integrating handwriting-specific knowledge into the model structure to obtain a more accurate representation of numeral strings. We believe that as the knowledge learned from ligatures and spaces between adjacent characters have shown to be very important to increase the word recognition performance, the knowledge about overlapping, touching and spaces between adjacent numerals may play the same role for numeral strings.

Similar to [2,3], we investigate the use of a pause model. The objective is to model inter-digit spaces and local interactions (overlapping, touching) between adjacent numerals in strings. However, our pause model is built-in the numeral models. This strategy allows us to keep the L parameter of the LBA fixed. The pause model is trained on digit-pairs extracted from the NIST database. In this training, for a given digit-pair the corresponding numeral models are concatenated by using the end-state. In fact, the end-state of the first model is replaced with the first state of the second (see Figure 8). The strategy used to train this pause model is presented in the next section.

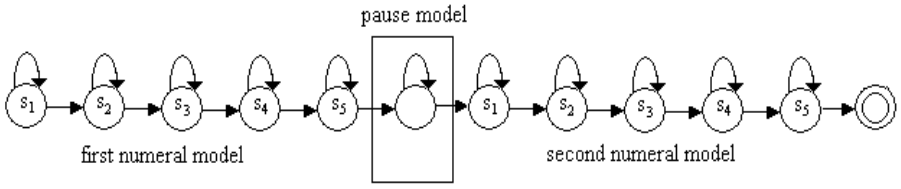

Figure 8. Concatenation of numeral models during string-based training

4. Experiments

The isolated numerals used in these experiments come from the NIST SD19 database. In order to construct 10 numeral models we use 50,000 numeral samples for training, 10,000 for validation and 10,000 for testing. The slant normalization of these numerals is done taking into account contextual information regarding the slant of their original strings, and the feature extraction is performed considering the intra-string size variation. All this process is detailed in [4].

The experiments using numeral strings are based on 12,802 numeral strings extracted from NIST SD19 and distributed into 6 classes: 2_digit (2,370), 3_digit (2385), 4_digit (2,345) and 5_digit (2,316), 6_digit(2,169) and 10_digit(1,217) strings respectively. These strings exhibit different problems, such as touching, overlapping and fragmentation.

During the comparison of the HMM topology with and without the end-state, we did not consider the inter-digit spaces in order to evaluate the LBA in terms of segmentation performance. For each string, features have been extracted considering just foreground pixels (black pixels). There is no symbol to represent inter-digit spaces. This also may give us some idea about the real contribution of the pause model in our system.

Class	HMM without end-state	HMM with end-state	Pause model (1 by class)	Pause model (1 for all classes)
Isolated numerals	91.10	91.73	91.60	91.60
2_digit (2,370)	85.32	87.72	88.23	88.40
3_digit (2,385)	78.19	82.43	83.31	83.61
4_digit (2,345)	71.34	78.17	78.55	78.72
5_digit (2,316)	66.32	75.65	75.35	76.21
2,3,4 and 5_digit (9,416)	**75.32**	**81.00**	**81.40**	**81.77**
6_digit (2,169)	63.85	71.69	71.37	72.01
10_digit (1,217)	44.04	60.64	57.68	61.05
Global (All classes)	**70.43**	**77.51**	**77.45**	**78.15**

Table 1. String recognition results on the test database

For the pause model experiment, we use a two step-training mechanism: 1) 10 numeral models are trained first on isolated digits, 2) the numeral models are submitted to a string-based training using digit pairs (DPs) extracted from the NIST database. The DP database is balanced in terms of number of naturally segmented, overlapping and touching numerals. The NIST series *hsf_0 to hsf_3* were used for providing 15,000 training samples, while *hsf_7* was used for providing 3,500 validation samples.

We use the two-step training mechanism described above to evaluate the following strategies: 1) the use of one pause model for each numeral class; and 2) the use of one pause model representing all numeral classes. In both just the pause model parameters are estimated during the second-step training. The parameters corresponding to the numeral models are kept the same as estimated during the first training step based on isolated numerals. Table 1 resumes all the experiment results, in which a zero-rejection level is used.

5. Discussion

The HMM topology with end-state does not bring a significant improvement to the recognition of isolated numerals (about 0.6%). On the other hand, this brought 7.08% of improvement to the global string recognition rate. This is due the better distribution of the observations between the states, and a better estimation of the self-transition probability in the last HMM-state (s_5). Consequently, the LBA provides a better match of numeral models against the observation sequence. This means a better definition of string segmentation cuts. Maybe, this can also explain the importance given by the authors to space model in [8,9], and the use of durational constraints in [10].

Figure 9 shows an example in which the segmentation cuts at top and bottom were provided respectively by the models with and without end-state. To confirm the improvement on segmentation cuts, we made an error analysis considering the

10_digit strings misrecognized using the models without end-state, which were recognized with the models with end-state. A total of 245 samples were manually checked.

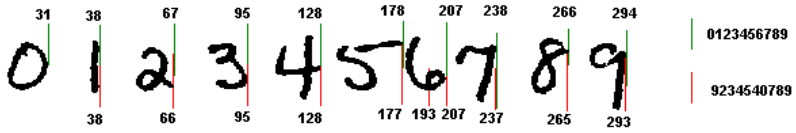

Figure 9: Segmentation points and recognition result produced by the LBA using 5-state HMMs with end-state (top) and without end-state (bottom).

Figure 10(a) shows that 72.2% of these misrecognitions are related to mis-segmentation problems. Moreover, we compute the difference of location of the segmentation points provided by these two HMM-structures in terms of the number of observations. Figure 10(b) shows that the frequencies of location differences equal to 1, 2 and more than 2 observations are respectively 47.3%, 18% and 6.9%.

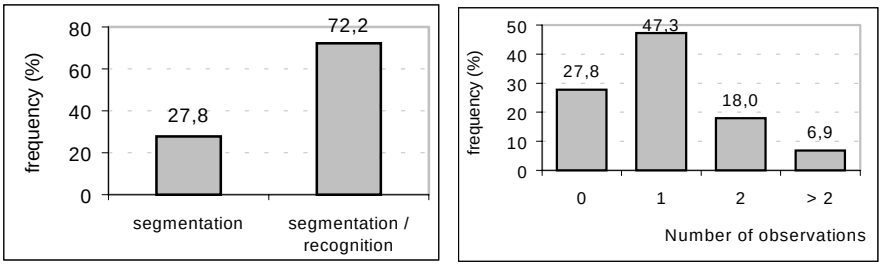

Figure 10: a) Frequency of recognition and recognition/segmentation mistakes; b) Difference between the location of segmentation points considering the number of observations

In the pause model experiments we consider all the spaces (white columns) between adjacent digits in strings. In fact, the pause model is used to absorb all interactions between adjacent numerals including inter-digit spaces, overlapping and touching. The experiment considering one pause model for each numeral class does not show improvements for all numeral string classes. We observe a small loss in terms of recognition rate of numeral strings composed of more than 4 numerals. This is due the lack of training samples of specific digit-pair classes in the database. In fact, the database used for the second-step training is well balanced in terms of natural segmented, overlapping and touching numerals, but it needs also be balanced in terms of isolated numeral classes and digit-pair classes. This experiment also shows that the interaction between adjacent digits varies as the string length. We can see some improvement for all string classes when we use all the database to model just one pause model.

The small improvement obtained by considering the pause model also confirms the nice string segmentation performance provided by the LBA by using the 5-state HMM with end-state. In fact, this shows that even without considering the inter-digit spaces, the models based on this topology can provide a good string segmentation.

6. Conclusions

In this work we have evaluated different HMM topologies on the LBA framework. The inclusion of an end-state in numeral HMM structure allowed us to balance the importance between their states. Under this condition, the LBA finds a more precise match of the numeral models against the observation sequence representing a numeral string. This new HMM structure improves the LBA string segmentation performance. In addition, the end-state provides a way of concatenating the numeral models to evaluate the use of a pause model built-in the numeral models.

The preliminary results on the pause model show us that integrating handwriting-specific knowledge into the model structure to obtain a more accurate representation of numeral strings is a promising approach. However, this approach is strongly dependent on a representative database.

Acknowledgements: The authors wish to thank the Pontifícia Universidade Católica do Paraná (PUC-PR, Brazil), the Universidade Estadual de Ponta Grossa (UEPG, Brazil), the École de Technologie Supérieure (ETS, Canada) and the Centre for Pattern Recognition and Machine Intelligence (CENPARMI, Canada), which have supported this work.

References

1. Rabiner L. and Juang B.H., Fundamentals of Speech Recognition. Prentice Hall, New Jersey, 1993
2. Cho W., Lee S.W., and Kim, J.H. Modeling and Recognition of Cursive Words with Hidden Markov Models, Pattern Recognition 1995; 28:1941-1953
3. Dolfing J.G.A. Handwriting recognition and verification: A hidden Markov approach. PhD. thesis, Eindhoven University of Technology, Netherlands, 1998
4. Britto A. S., Sabourin R., Lethelier E., Bortolozzi F., Suen C.Y. Improvement in handwritten numeral string recognition by slant normalization and contextual information. In: Proceedings of the 7th International Workshop on Frontiers of Handwriting Recognition (7th IWFHR), Netherlands, 2000, pp 323-332
5. Sabourin R., 1990. Une Approache de Type Compréhension de Scène Appliquée au Probleme de la Vérification Automatique de L'Identité par L'Image de la Signature Manuscrite. Thèse de Doctorat, École Polytechnique, Université de Montréal, 1990
6. Mohamed M. and Gader P. Handwritten word recognition using segmentation-free hidden Markov modeling and segmentation-based dynamic programming techniques. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996;18 548-554
7. Linde Y., Buzo A., and Gray R.M. An algorithm for Vector Quantizer Design. IEEE Transactions on Communications, 1980; COM-28: 84-95
8. Elms A.J., Procter S. and Illingworth J. The Advantage of using an HMM-bases Approach for Faxed Word Recognition. International Journal on Document Analysis and Recognition (IJDAR). 1998; 18-36
9. Procter S., Illingworth J. and Elms A.J., The recognition of handwritten digit strings of unknown length using hidden Markov models, In: Proceedings of the Fourteenth International Conference on Pattern Recognition, 1998, pp 1515-1517
10. Rabiner L. and Levinson L. A speak-independent, syntax-directed, connected word recognition system based on hidden Markov models and Level Building. Transactions on Acoustics, Speech, and Signal Processing, 1985;Vol. ASSP-33, Nº. 3.