An HMM-MLP Hybrid System to Recognize Handwritten Dates

M. Morita[1-2], L. S. Oliveira[1-2], R. Sabourin[1-3], F. Bortolozzi[3] and C. Y. Suen[2]
[1]Ecole de Technologie Supérieure
[2]Centre for Pattern Recognition and Machine Intelligence
[3]Pontifícia Universidade Católica do Paraná
e-mail: marisa@livia.etsmtl.ca

Abstract - This paper presents an HMM-MLP hybrid system to process complex date images written on Brazilian bank cheques. The system first segments implicitly a date image into sub-fields through the recognition process based on an HMM approach. Afterwards, a recognition and verification strategy is proposed to recognize the three obligatory date sub-fields (day, month and year) using different classifiers. Markovian and neural approaches have been adopted to recognize and verify words and strings of digits respectively. We also introduce the concept of meta-classes of digits, which is used to reduce the lexicon size of the day and year and improve the precision of their segmentation and recognition. Experiments show interesting results on date recognition.

I. INTRODUCTION

Automatic handwriting recognition has been a topic of intensive research during the last decade. The literature contains many studies on the recognition of isolated units of writing such as characters, words or strings of digits. Only recently the recognition of a sentence composed of a sequence of words or different data types has been investigated. Some applications on sentence recognition are reading texts from pages [2], street names from postal address [5] and date processing on cheques [6]. In such applications, usually a sentence is segmented into its constituent parts. In the literature two main different approaches of segmentation can be observed. The former and perhaps the most frequently used method segments a sentence into parts usually based on an analysis of the geometric relationship of adjacent components in an image while the latter uses an implicit segmentation which is obtained through the recognition process.

In this paper we present an HMM-MLP hybrid system to recognize dates written on Brazilian bank cheques that makes use of an implicit segmentation-based strategy. In this application, the date from left to right can consist of the following sub-fields: city name, separator1 (Sep1), day, separator2 (Sep2), month, separator3 (Sep3) and year. Figure 1 details the lexicon of each date sub-field and Figure 2 shows some samples of handwritten dates. In such cases, the grey color represents the obligatory date sub-fields.

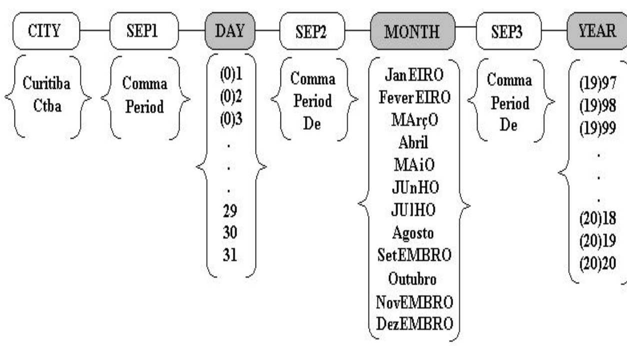

Fig. 1. Lexicon of each date sub-field

The development of an effective date processing system is very challenging. The system must consider different data types such as digits and words written in different styles (uppercase, lowercase and mixed). Although the lexicon size of month words is limited (Figure 1), there are some classes such as "Janeiro" and "Fevereiro" that contain a common substring ("eiro") and can affect the performance of the recognizer. The system must also take into account the variations present in the date field such as 1- or 2-digit day, 2- or 4-digit year, the presence or absence of the city name and separators. Moreover, it must deal with difficult cases of segmentation since there are handwritten dates where the spaces between sub-fields (inter-sub-field) and within a sub-field (intra-sub-field) are similar as shown in Figures 2(b) and 2(c). For example, in Figure 2(b) the intra-sub-field space between "1" and "0" is almost the same as the inter-sub-field spaces between "Curitiba" and "3" or "Fevereiro" and "10". Therefore, it will be very difficult to detect the correct inter-sub-field spaces in this image using a segmentation based on rules.

In order to overcome this kind of problem, our system makes use of the Hidden Markov Models (HMMs) to identify and segment implicitly the date sub-fields and considers multi-hypotheses of segmentation. A recognition and verification strategy has been proposed to recognize the three obligatory date sub-fields (day, month and year). We are using

0-7803-7278-6/02/$10.00 ©2002 IEEE

Fig. 2. Samples of handwritten date images

HMMs and Multi-Layer Perceptron (MLP) neural networks to recognize and verify words and strings of digits respectively. This is justified by the fact that MLPs have been widely used for digit recognition and the literature shows better results using this kind of classifier than HMMs. On the other hand, HMMs have been successfully applied to handwritten word recognition more recently.

In addition to the description of the system, this paper introduces a new strategy to reduce the lexicon size on digit string recognition. Such a scheme makes use of the concept of meta-classes of digits and the number of digits obtained during the segmentation process of the date sub-fields. For this reason, we propose a digit string verifier to validate the information about the number of digits before recognition. Concepts about levels of verification are also presented.

II. DEFINITIONS

A. Meta-Classes of Digits

We have defined 4 meta-classes of digits ($C_{0,1,2,3}$, $C_{1,2}$, $C_{0,9}$ and $C_{0,1,2,9}$) based on the classes of digits present in each position of 1- or 2-digit day and 2- or 4-digit year as shown in Figure 3. This is possible because the lexicon of the day and year is known and limited (Figure 1). While the class of digits C_{0-9} deals with the 10 numerical classes, the meta-classes of digits represented by shaded boxes in Figure 3 cope with specific classes of digits. The objective is to build HMMs based on these meta-classes in order to reduce the lexicon size of the day and year and produce a more precise segmentation. Besides, it can be applied to digit string recognition to improve the recognition results since very often confusions between some classes of digits can be avoided (e.g., 4 and 9, 8 and 0). We can observe the efficiency of this strategy in Section IV.

B. Levels of Verification

Takahashi and Griffin in [7] define three kinds of verification: absolute verification for each class (Is it a “0” ?), one-to-one verification between two categories (Is it a “4” or a “9” ?) and

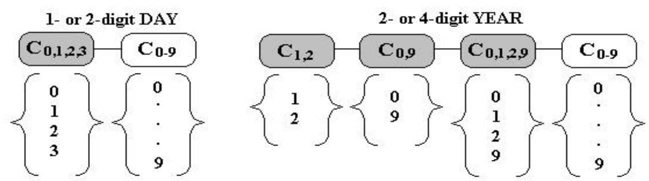

Fig. 3. Classes of digits present in each position of 1- or 2-digit day and 2- or 4-digit year

verification in clustered, visually similar categories (Is it a “0”, “6” or “8” ?). In addition to these definitions, Oliveira et al in [4] introduce the concepts of high-level and low-level verifications. The idea of the high-level verification is to confirm or deny the hypotheses produced by the classifier by recognizing them. On the other hand, the low-level verification does not recognize a hypothesis, but rather determines whether a hypothesis generated by the classifier is valid or not.

Based on these concepts, we propose to use an absolute high-level word verifier and a low-level digit string verifier in order to improve the recognition rate and reliability of the system. The objective of an absolute high-level word verifier is to re-rank the N best hypotheses of word segmentation and recognition using a classifier specialized in the specific problem: words instead of the whole sentence. The word recognizer takes both segmentation and recognition aspects, while the word verifier considers just the recognition aspects. The purpose of a low-level digit string verifier is to confirm whether an image contains one digit or more before its recognition. This information is very important since we have used it in order to identify which classifiers will be applied to digit string recognition. In Sections III-B.1 and III-D present more details about these verifiers and in Section IV we will see the importance of them in the system.

III. DESCRIPTION OF THE SYSTEM

In the following subsections we describe the modules of the system depicted in Figure 4.

0-7803-7278-6/02/$10.00 ©2002 IEEE

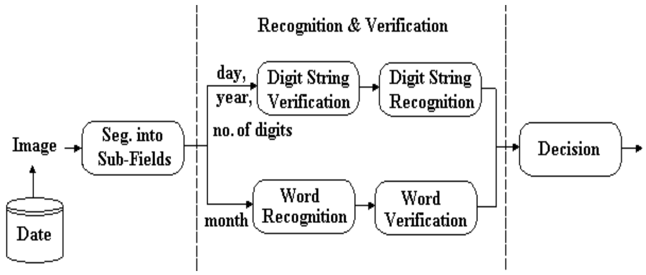

Fig. 4. Block diagram of the date recognition system

A. Segmentation into Sub-Fields

A date image is first segmented into graphemes and then two feature sets are extracted. The segmentation algorithm and the features (global and concavity) are basically the same as that we have presented in [3]. However, here the features differ in the following aspects: both feature sets are combined with the space primitives, the sizes of the concavity feature vector and its codebook. Since the concavity measurements have exhibited a good feature to improve the discrimination of letters and digits as well, we have used them in other modules of the system. They differ in the size of concavity vector and the zoning used.

Afterwards, both feature sets are combined through the HMMs that have been used to identify and segment implicitly the date sub-fields. In this case, the HMMs are built at the city, space and character levels. We have chosen an ergotic model with 5 states to represent globally the city names and noise (e.g., Sep1) and a linear topology to model spaces and characters such as letters and digits. The topology of the space models consists of 2 states linked by two transitions that encode a space or no space. We have considered 3 HMMs to model the different kinds of space: inter-sub-field, intra-word and intra-digit spaces. The topologies of the character models consist of 4 or 5 states which were chosen based on the output of our segmentation algorithm. Considering uppercase and lowercase letters, we have 40 HMMs. For the digit case, we have defined 5 HMMs. The M_{0-9} model considers the 10 numerical classes of digits and the other ones are defined based on the meta-classes of digits (e.g., the model $M_{1,2}$ corresponds to the meta-class of digits $C_{1,2}$ and so on) (see Figure 3).

The Baum-Welch algorithm with the Cross-Validation procedure [8] has been used to train our HMMs. Our training mechanism has two steps of training. In the first step, we train only the city model using 980 images of isolated city names. In the second step, besides the date database we have considered the legal amount database, which is composed of isolated words, in order to increase the training and validation sets. In this case, the parameters of the city model are initialized based on the parameters obtained in the previous step. Then, the other models present in the date and word images are trained systematically since the labeling of each training image is available. For example, the date model is formed by the concatenation of sub-field and space models, and each sub-field with the exception of the city model by character and space models. We have used about 1,200 and 8,300 images of dates and words respectively.

Segmentation of the date into sub-fields is delivered by the HMMs as a byproduct of the recognition process using the Viterbi algorithm [8]. The architecture of the date model in the recognition remains basically the same as in the training. However, as the labeling is not used in the recognition, we have considered 8 date models, which vary according to the presence or absence of the optional date sub-fields. Moreover, some modifications were made in the architecture of the day, year, month and separator models since the number of digits present in the day and year as well as the writing style are unknown. The month model consists of an initial state, a final state and twelve models in parallel that represent the 12 word classes. Each word model has two letter models (uppercase and lowercase) in parallel and four intra-word space models linked by four transitions. The same philosophy is applied to build the "de" separator model (Sep2 and Sep3). The day model consists of an initial state, a final state and the 2-digit day model in parallel with the 1-digit day model. The 2-digit day model is formed by the concatenation of the models: $M_{0,1,2,3}$, intra-digit space and M_{0-9}. The 1-digit day model is related to the M_{0-9} model. The year model is built in the same manner.

In order to improve the overall segmentation rate, two hypotheses of segmentation can be considered. In this case, a second hypothesis will be generated only if the probability of the best date model (the 8 possible) that better represents a date image is smaller than a threshold, which was determined through experimentation.

B. Verification and Recognition of Digits

All classifiers related to the verification and recognition of digits are MLPs trained with the gradient descent applied to a sum-of-squares error function [1]. The transfer function employed is the familiar sigmoid function.

In order to monitor the generalization performance during learning and terminate the algorithm when the improvement levels off, we have used the method of cross-validation. Such a method takes into account a validation set, which is not used for learning, to measure the generalization performance of the network. During learning, the performance of the network on the training set will continue to improve, but its performance on the validation set will only improve to a point, where the network starts to overfit the training set, that the learning algorithm is terminated.

All networks have one hidden layer where the units of input and output are fully connected with the units of the hidden layer. The learning rate and the momentum term were set at high values in the beginning to enable the weights quickly fit the long ravines in the weight space, then these parameters were reduced several times according to the number of iterations to make the weights fit the sharp curvatures. The number of hidden units used by each classifier is mentioned in the following subsections.

B.1 Digit String Verification (DSV)

As discussed before, the digit string recognition (DSR) module uses different classifiers depending on the sub-field (day and year) and the number of digits. In order to provide more reliable information about the number of digits to the DSR module we have proposed a low-level digit string verifier to confirm the number of digits obtained through the HMMs. Thus, when such an information differs from the result supplied by the verifier and its output (which we consider as an estimation of the probability *a posteriori*) is greater than a threshold, which was determined through experimentation, we assume the information provided by the verifier, otherwise we keep the information of the HMMs. This verifier is based on the under-segmentation verifier presented in [4] which makes use of concavity features.

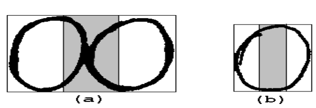

Fig. 5. Zoning used by the verifier

The feature vector we have used is represented by 14 components where 13 components are related to concavity measures and the last one represents the image surface. Since we are dividing the image into three vertical zones, we consider three feature vectors of 14 components each. The objective here is to emphasize the region of the image where the connections between characters occur often (Figure 5). Finally, the overall feature vector is composed of (3×14) 42 components normalized between 0 and 1.

We have used 9,000 samples of isolated and touching digits of NIST database (hsf_$\{0,1,2,3\}$) to train this verifier. The recognition rate on the test set (4,000 images) was about 99.17%. The number of hidden units used was 20.

B.2 Digit String Recognition (DSR)

The number of digits provided by the DSV module is used as information *a priori* on DSR to determine which classifiers will be employed (see Figure 6). The e_{0-9} classifier copes with the 10 numerical classes and the other classifiers $e_{0,1,2,3}$, $e_{0,1,2,9}$, $e_{0,9}$ and $e_{1,2}$ are specialized in the lexicon of the meta-classes of digits $C_{0,1,2,3}$, $C_{0,1,2,9}$, $C_{0,9}$ and $C_{1,2}$ respectively. This strategy aims at reducing the lexicon size on DSR to improve the recognition results.

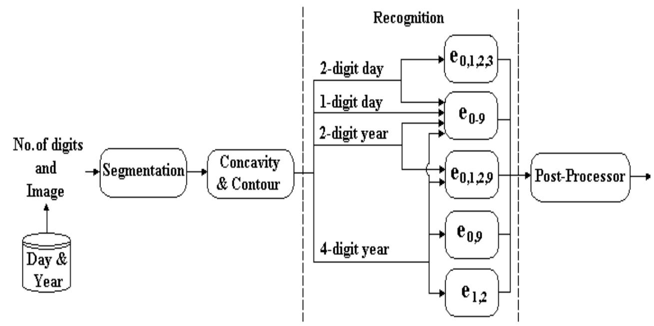

Fig. 6. Block diagram of the DSR module

The segmentation module that we have used is based on the relationship of two complementary sets of structural features, namely, contour & profile and skeletal points. The segmentation hypotheses are generated through a segmentation graph, which is decomposed into linear sub-graphs and represents the segmentation hypotheses.

For each segmentation hypothesis a mixture of concavity & contour is extracted. We have used six concavity feature vectors of 13 components each since we are dividing the image into six zones. In this way, the overall concavity feature vector is composed of (13×6) 78 components normalized between 0 and 1. The contour information is extracted from a histogram of contour directions. For each zone, the contour line segments between neighboring pixels are grouped regarding 8-Freeman directions (Figure 7c). The number of line segments of each orientation is counted (Figure 7b). Therefore, the contour feature vector is composed of (8×6) 48 components normalized between 0 and 1. Finally, the last part of the feature vector is related to the character surface. We simply count the number of black pixels in each zone and normalize these values between 0 and 1. Thus, the final feature vector, which feeds our classifiers, has ($78+48+6$) 132 components.

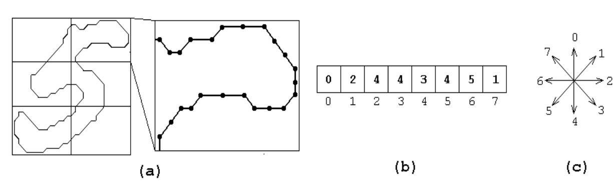

Fig. 7. Contour measurement: (a) Contour image of the upper right corner zone (b) Feature vector and (c) 8-Freeman directions

We have used images of digits extracted from the courtesy amount and date databases to train those classifiers. Table I describes the databases used for training (TR), validation (VL)

0-7803-7278-6/02/$10.00 ©2002 IEEE

and testing (TS), the recognition rates achieved on validation (RR VL) and test (RR TS) sets. The e_{0-9} classifier has 80 hidden units while the other ones have 70.

TABLE I
DESCRIPTION OF THE CLASSIFIERS

Classifier	Classes of Digits	TR	VL	TS	RR VL	RR TS
$e_{0,1,2,3}$	0,1,2 and 3	8,300	1,250	2,500	99.7%	99.4%
e_{0-9}	0-9	14,000	3,000	5,000	99.0%	98.9%
$e_{0,1,2,9}$	0,1,2 and 9	8,300	1,250	2,500	99.7%	99.4%
$e_{0,9}$	0 and 9	3,400	500	1,000	99.9%	99.8%
$e_{1,2}$	1 and 2	4,400	700	1,400	99.8%	99.5%

Since we are dealing with multi-hypotheses of segmentation and recognition the generation of K best hypotheses of a string of digits is carried out by means of a Modified Viterbi, which ensures the calculation of the k best paths of segmentation-recognition graph [4]. Thus, the final probability for a hypothesis of segmentation-recognition is given through the product of the probabilities produced by the classifiers. Afterwards, each hypothesis is submitted to the post-processor module, which verifies whether it belongs to the lexicon of the day or year depending on the sub-field.

C. Word Recognition

The word probabilities for each hypothesis of segmentation are computed through the Forward procedure [8] for the 12 word models. In this case, each word model as well as the feature sets are the same as that we have used in the segmentation into sub-field module.

D. Word Verification

A word image is first segmented into graphemes and then the following features are extracted: global, a mixture of concavity and contour and information about the segmentation points. The segmentation algorithm and the global features are the same as that we have employed in the segmentation into sub-field module.

Since we are dividing a grapheme into two zones, we have two concavity vectors of 9 components each. For each vector, we have introduced 8 more components related to the information about the contour image already described in this paper and they have been used to increase the discrimination between some pairs of letters (e.g., “L” and “N”). Thus, the final feature vector has $(2\times(9+8))$ 34 components. The segmentation features have been used to reduce confusions such as “n” and “l” since they try to reflect the way that the graphemes are linked together. Therefore, the output of the feature extraction is a pair of symbolic descriptions, each consisting of an alternating sequence of grapheme shapes and associated segmentation point symbols.

The two feature sets are combined through the HMMs that have been used to verify the two best hypotheses generated by the word recognizer for each hypothesis of segmentation. The objective of the absolute high-level word verifier is to re-rank the list of hypotheses by multiplying the probabilities generated by the word recognizer and verifier. In this case, the word probabilities are computed by the Forward procedure. We have adopted a similar architecture of the word models used in the word recognition module, but here we are not modeling the spaces. The letter models used to build the word models are based on the topologies of the character models described before, but in this case we are modeling the nature of the segmentation point.

The letter models have been trained through the Baum-Welch algorithm with the Cross-Validation procedure using 9,500 word images extracted from the date and legal amount databases.

E. Decision

A date image is counted as correctly classified if the three obligatory sub-fields are correctly classified.

IV. EXPERIMENTS AND ANALYSIS

The system was capable to identify 95.5% on the test set the best date model (among the 8 possibilities) that better represents a date image (Top1) and 99.7% considering the two best date models (Top2). The test set we have used is composed of 400 date images. Table II details the segmentation rate of each date sub-field and the results when the number of digits is well estimated by the HMMs. The results shown in this Table were evaluated automatically by the system.

Figure 8(a) shows an example where the date sub-fields are missegmented on h_1, but this problem was solved on h_2. While h_1 corresponds to the best date model, h_2 is related to the second one. Figure 8(b) demonstrates a difficult case of segmentation, where the spaces between sub-fields and within sub-fields are very similar. However, our approach succeeded in segmenting the date sub-fields correctly on h_1.

Table III reports the improvements on date recognition using the word and digit string verifiers. This Table also details the results on word and digit string recognition using their respective verifiers.

We can note in Table III an improvement of the recognition rate from 79.8% to 82.0% on date recognition on the test set. In this case, it is very difficult to compare with other sentence recognition engines due to the special application of our work. Regarding the date recognition system, the literature indicates few studies that focus basically on segmentation problems and use different databases.

TABLE II

SEGMENTATION RESULTS

	City	Day	Sep2	Month	Sep3	Year	No. of Digits (Day)	No. of Digits (Year)
Top1	95.7%	96.2%	95.5%	99.5%	100%	100%	92.2%	100.0%
Top2	99.7%	99.7%	100%	99.7%	100%	100%	95.7%	100.0%

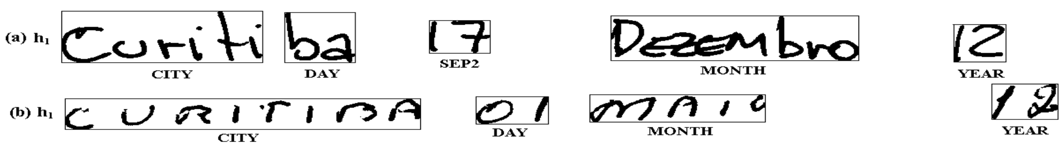

Fig. 8. Examples of (a) missegmented and (b) well-segmented date images

We observed on the validation set that the presence of common sub-strings among some word classes such as “Janeiro” and “Fevereiro” affect the performance on month word recognition. The main problem detected on DSR is when the day is well segmented in one of the two hypotheses of segmentation, but the classifier gives the highest probability to the incorrect hypothesis. In our application, the year segmentation is less complex than the day due to the low frequency of the “de” separator before the year and its location (i.e., the year is the last sub-field present in the date field). This explains why the results on year recognition are higher for 2-digit strings than the results achieved on day recognition for 2-digit strings.

TABLE III

PERFORMANCE OF THE SYSTEM (NV: RESULTS WITHOUT VERIFICATION AND V: RESULTS WITH VERIFICATION

	Date	Month	1-digit Day	2-digit Day	2-digit Year	4-digit Year
NV	79.8%	87.2%	76.1%	93.9%	96.9%	87.5%
V	82.0%	89.0%	90.4%	94.4%	96.9%	87.5%

In order to validate the concept of meta-classes of digits on DSR, we ran two experiments using a subset of hsf_7 series of the NIST SD19 database. This subset contains 986 images of 2-digit strings related to the lexicon of 2-digit day. The former considers the meta-classes of digits employing the classifiers $e_{0,1,2,3}$ and e_{0-9} and the latter makes use of the e_{0-9} classifier, i.e., without the concept of meta-classes. The use of this concept on DSR seems to be a good strategy when the lexicon is known and limited since it enhanced the recognition rate from 97.1% to 99.2%. To train the classifiers $e_{0,1,2,3}$ and e_{0-9} we have used 78,000 and 195,000 images of isolated digits respectively from hsf_$\{0,1,2,3\}$.

V. CONCLUSION

In this paper we presented an HMM-MLP hybrid system to recognize complex date images written on Brazilian bank cheques. The system makes use of the HMMs to segment the date sub-fields and considers different classifiers to recognize the three obligatory sub-fields. This paper also has introduced the concept of meta-classes of digits to reduce the lexicon size of the day and year and improve the precision of their segmentation and recognition. We have shown difficult cases of segmentation in which our HMM-based approach works well and interesting results on date recognition. We also have seen encouraging results on digit string recognition using the concept of meta-classes of digits on the NIST database.

References

[1] C.M.Bishop. *Neural Networks for Pattern Recognition*. Oxford Univ. Press, Oxford - U.K., 1995.

[2] U. Marti and H. Bunke. Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system. IJPRAI, to appear.

[3] M. Morita, A. E. Yacoubi, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Handwritten month word recognition on Brazilian bank cheques. In *Proc. 6th ICDAR*, pages 972–976, Seattle-USA, September 2001.

[4] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. A modular system to recognize numerical amounts on Brazilian bank cheques. In *Proc. 6th ICDAR*, pages 389–394, Seattle-USA, September 2001.

[5] J. Park and V. Govindaraju. Use of adaptive segmentation in handwritten phrase recognition. *Pattern Recognition*, 35:245–252, 2002.

[6] C. Y. Suen, Q. Xu, and L. Lam. Automatic recognition of handwritten data on cheques - fact or fiction ? *Pattern Recognition Letters*, 20(13):1287–1295, November 1999.

[7] H. Takahashi and T.D.Griffin. Recognition enhancement by linear tournament verification. In *Proc. 2nd ICDAR*, pages 585–588, Japan, 1993.

[8] A. E. Yacoubi, R. Sabourin, M. Gilloux, and C. Y. Suen. Off-line handwritten word recognition using hidden markov models. In L. Jain and B. Lazzerini, editors, *Knowledge Techniques in Character Recognition*. CRC Press LLC, April 1999.