An HMM-based Approach for Date Recognition

Marisa Morita[1], Edouard Lethelier[1], Abdenaim El Yacoubi[1],
Flávio Bortolozzi[1] and Robert Sabourin[2]

[1] Pontifícia Universidade Católica do Paraná
Laboratório de Análise e Reconhecimento de Documentos (LARDOC)
Rua Imaculada Conceição 1155, Prado Velho, 80215-901
Curitiba-Pr, BRAZIL
[2] Ecole de Technologie Supérieure
Laboratoire d'Imagerie, de Vision et d'Intelligence Artificielle (LIVIA)
1100, rue Notre Dame Ouest, H3C IK3
Montréal, CANADA

Abstract. This article presents the first results of a system developed to recognize automatically handwritten dates on Brazilian bankchecks. Considering an omni-writer context, we detail here our recognition module dedicated to process the month field. This module is based on the combination of holistic and analytical approaches with a limited lexicon. Both approaches operate with a single explicit segmentation technique that provides a grapheme sequence for the Hidden Markov Models of each recognizer. We show significative improvements in combining both modules to get a satisfactory recognition rate considering the small database we work with. Finally, we present several perspectives of our future work.

1 Introduction

Several works and applications have been carried out in the field of off-line handwriting recognition attempting to reach the human behavior while reading handwritten words. Despite the high computing capabilities of the current computers, these systems are usually unable to accomplish tasks as the human vision. Therefore, the research in this area is still intensive.

In handwriting recognition, most of drawbacks are due to the high variability of the handwriting styles present in an omni-writer context, and also to the size of the vocabulary (lexicon size). Moreover, the off-line approach is more complex than the on-line approach, due to the presence of noise in the image acquisition process and the loss of temporal information such as the writing sequence and the velocity. This information is very helpful in a recognition process.

In order to reduce the complexity of the problem, many research teams make use of contextual information in their systems by adopting a lexicon-based strategy. In applications where the lexicon size is limited, e.g in the context of the legal amount on checks, a holistic approach can be considered to represent each word class with a specific model [2], [5]. However, when the lexicon size is large, e.g in postal applications, most of the current techniques are based on analytical

approaches. In such a case, one model for each character is built. Then, the word models are built by the concatenation of the appropriated character models [4], [5], [6], [8].

In analytical approaches, the words are segmented into graphemes that can represent characters or pseudo-characters. This segmentation strategy is the most used due to the large variability of the handwriting. In this case, the definitive segmentation points are determined in the recognition phase by the graphemes concatenation. Even if holistic approaches do not require any kind of segmentation, analytical approaches have advantages over holistic approaches. Once for each new application, it is not necessary to train the set of words of the associated lexicon. This portability is very useful when the document image contains several fields to be processed by the same word recognizer. Moreover, the training of the characters does not depend on the lexicon size.

Hidden Markov Models (HMMs) have been successfully applied in speech recognition. More recently, they have been used in handwritten word recognition [1], [4], [5], [6], [8]. The HMM word architecture is pretty well adapted to describe a word image as a sequence of observations. Some approaches are based on explicit segmentation where words are split into characters or pseudo-characters to provide a grapheme sequence [1], [5], [8]. In other approaches, this kind of segmentation is carried out implicitly, during the recognition phase through the HMMs [4], [6]. In some holistic approaches, the explicit segmentation can also be used to improve the word-length estimation (for a given word-class) and to permit a better class discrimination while training the word models.

Our work focus on the off-line recognition of handwritten dates on brazilian bankchecks, considering an omni-writer context. The peculiarity of our application lies in the complexity to process the mixed information presented in the date field: words (city, month, separator) and digits (day and year). We show in this article the first results of our work to recognize isolated month words. Although this study deals with a limited lexicon size (12 classes), we try to improve the discrimination between specific classes when they contain a common sub-string such as the classes "Setembro" (September) and "Novembro" (November). To face this problem, we developed a combination strategy involving both holistic and analytical HMM approaches using an explicit segmentation technique. In this work, we have used a laboratory database which contains about 2,000 fields extracted from brazilian bankchecks with a resolution of 300 dpi.

We describe in section 2 the context of dates in brazilian bankchecks. In section 3, we detail our segmentation approach to provide the sequence of observations to the HMMs. In section 4, we describe our classification approaches and we show the experimental results in section 5. Finally, section 6 presents our conclusions and perspectives in our future work.

2 Peculiarities of our study

There has been a lot of work dedicated to the processing of literal handwritten amounts on checks. Nevertheless, the studies about the date recognition are few, even inexistent for brazilian checks.

The date information consists of the following fields, presented below as they appear from left to right:

- city name (alphabetical) optional;
- **coma** (separator);
- day (numerical);
- "**de**" (separator);
- month (alphabetical);
- "**de**" (separator);
- year (numerical).

The three field separators (in bold) are usually printed on the check as well as the guidelines. Figure 1 shows an example of a date field present in our laboratory database where we do not consider any extraction process of the date from the background of check.

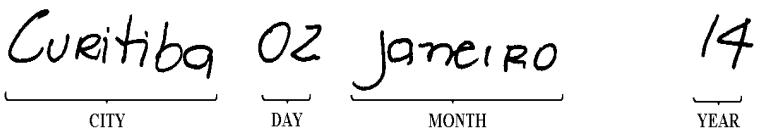

Fig. 1. Example of a date field present in our laboratory database image

Even if the field separators were not pre-printed before collecting our database, we have observed several images where writers put their own separators while filling in the date, such as the comma (,), (;), point (.) or "de". Table 1 contains the whole vocabulary of our laboratory database for the three obligatory parts to be filled in (e.g day, month and year) and its variabilities, such as the 2-digit day (01...09) and 2/4-digit year (1999 or 99).

Table 1. Vocabulary of the date field in our database

Day	Month	Year
(0)1, (0)2,...,	Janeiro, Fevereiro, Março, Abril,	(19)97, (19)98,...,
10, 11,...,	Maio, Junho, Julho, Agosto,	(20)00, (20)01,...,
31	Setembro, Outubro, Novembro, Dezembro	(20)20

3 Isolated word representation

Our current recognition system only deals with isolated words corresponding to the month field. After being manually located on the date image, the month word is explicitly segmented into graphemes, which are then converted into discrete symbols after the feature extraction stage. The resulting sequence of symbols is considered as the input of the HMM model. We employ in this work global features (loops, ascenders and descenders).

3.1 Date segmentation

The goal of this process is to try to locate each relevant field in the date (cf. section 2). This algorithm is based on space detection between connected components (CCs). An average space length is computed from all detected spaces in the median region of the date. We use the horizontal transition histogram to determine the median region. Based on this average length, we compute a threshold to locate every significant space between the entities. Even if all the entities are not correctly segmented (as shown in Figure 2(a) with the city name and the comma), this very simple approach is sufficient to yield a proper estimation of the reference lines from each sub-image of the date.

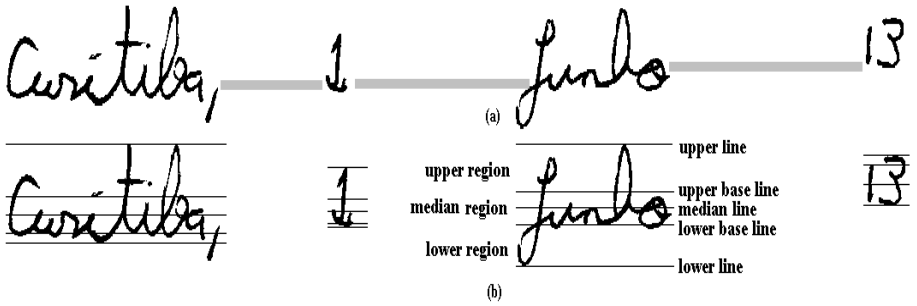

Fig. 2. (a) Significant spaces detection, (b) Reference lines detection

3.2 Reference line detection

Considering a cursive word, we can define reference baselines as the vertical transitions between the upper and the median regions and between the lower and the median regions (see Figure 2(b)). Commonly, the median region contains all the lowercase letters of the word, and the upper and lower regions contain all the letter extensions.

The baseline detection is applied for each segmented sub-image of the date. The upper (lower) baseline is provided by the average height of the upper contour maxima (lower contour minima) previously filtered using the weighted least square technique. The upper and lower lines correspond to the highest point of the upper contour and to the lowest point of the lower contour respectively. The median line is the half distance between the upper and lower baselines.

3.3 Character segmentation of words

The segmentation module provides a sequence of graphemes from a given word. In an analytical approach, the definitive segmentation points are determined in the recognition phase by the grapheme concatenation. In some cases, the segmentation technique could be found also in holistic approaches to improve the word length estimation and to facilitate the primitive detection.

The detection of the segmentation points (SPs) is based on two hypotheses:

- some characters are naturally isolated in the word (letter "n" in Figure 3);
- the local minima of the upper contour of CCs (MPs) corresponds to ligatures between characters.

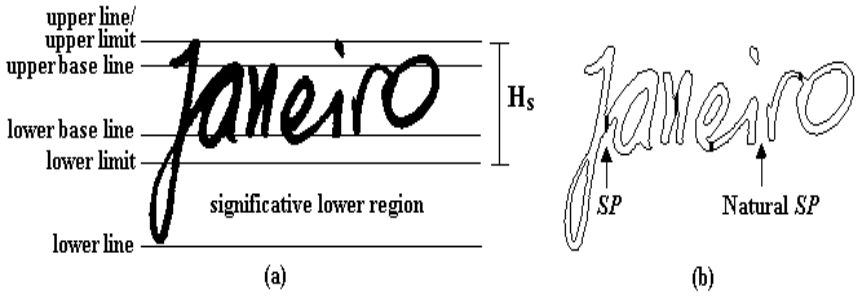

Fig. 3. SPs: (a) delimitation of the segmentation region, (b) types of SPs

The SPs hypotheses are validated if they belong to a determined segmentation height H_s, delimited by the upper and lower limits (Figure 3(a)). These limits represent 40% of the median region height when the lower and upper lines are greater than this value. Thus, in Figure 3(a), the upper limit corresponds to the upper line and the lower limit corresponds to 40% of the median region height. A new area is then determined as the significative lower region to detect some hypothetical descenders.

The generation of SPs results from the validation of the MPs. This validation is based on the detection of the lower contour on the vertical projection of the MP. In particular cases, the SP is shifted from the original MP location. This occurs when:

– the vertical projection crosses an inner loop before reaching the lower contour (Figure 4(a));
– the vertical projection remains tangent with the lower contour (Figure 4(b));
– the width of the vertical projection is not acceptable (Figure 4(c)).

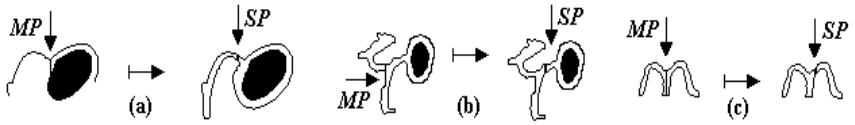

Fig. 4. Cases of shifted *SPs*: (a) with loop, (b) contour tangency, (c) inacceptable *SP* width

For any of these three cases, the algorithm performs a right shift from the *MP* location, following the upper contour, to try to find a new *SP* to validate. A *MP* corresponds to a *SP* if it can be shifted in its right neighborhood at an upper contour point minimizing the vertical width with respect to the lower contour without crossing a loop or a tangent. Otherwise, the algorithm tries the same search in the left neighborhood of the *MP*. For two particular cases, any validated *SP* can be removed from the word image:

– if the contour distance (number of adjacent pixels of contour) between two *SPs* is lower than a threshold T_s, proportional to the stroke thickness;
– if the contour distance between the *SP* and the right or left limit of the *CC* is lower than T_s.

Our segmentation algorithm may produce a correct word segmentation in characters (ideal), an under-segmentation (at least two characters remain linked) or an over-segmentation where a character contains two or three graphemes.

3.4 Primitive extraction

To face the difficult problem of selecting the best primitive set to process the month words, we first decided to implement and test one of the most popular type of primitives that was chosen and validated in several holistic approaches, namely ascenders, descenders and loops.

The primitive extraction algorithm is based on the detection of local maxima from the upper contour and local minima from the lower contour. Depending on the region where each extremity is located, the extension or loop is classified as a big or small primitive.

The significant upper and lower regions were split into two sub-regions to improve the discrimination of primitives:

– a small region, corresponding to 40% of the median region height;

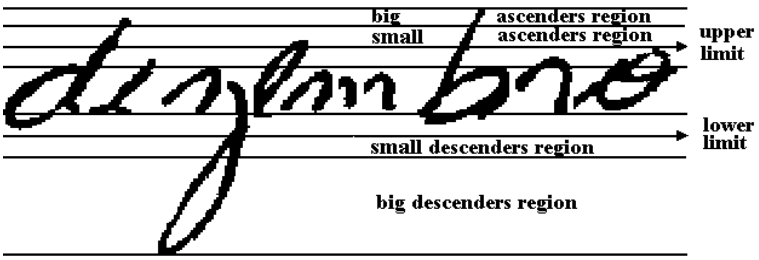

Fig. 5. Regions for ascenders and descenders detection

– a big region, covering the complementarity of the significant region.

To avoid spurious detections (as false ascenders or descenders) in the neighborhood of the median region, we also defined a neutral region between the small and the median regions (upper and lower limits in Figure 5). The gap of the neutral regions is also 40% of the median region height.

Thus, for each grapheme, the detection is processed as follow:

– if a local maximum (minimum) is located in the small upper (lower) region, it corresponds to a small ascender (descender);
– if a local maximum (minimum) is located in the big upper (lower) region, it corresponds to a big ascender (descender).

When belonging to the median region, we classify the loop into two categories (big or small) of primitives depending on its vertical size. When a loop is located in the upper or in the lower region, it is implicitly associated with an ascender or a descender. Our primitive extraction does not provide any specific class taking into account the detection order to be able to discriminate characters as "b"/"d" and "p"/"q". Indeed, the month field does not contain any "p" nor "q" and the positions of characters "b" and "d" in the month words are different.

Finally, to improve the word discrimination in our recognition system, we also use a specific primitive class when a grapheme does not contain any relevant primitive. This class considerably improves the discrimination through the word length estimation. Then, we obtain a 20-symbol alphabet where each symbol has been evaluated on the training set in order to provide a significant consistency to the HMMs.

4 HMM-based isolated word recognition using a limited lexicon

Due to the limited lexicon size, the context of isolated month processing seems to be naturally adapted to the holistic approach. However, in handwriting recognition, the performance of the system strongly depends on the size of the training

set. Since we started working with a small database, we decided to implement both holistic and analytical approaches. We can significantly increase the size of the database from the same data set considering the analytical approach and we can also evaluate the correctness of our primitive set through both recognition systems based on HMMs.

4.1 HMMs topology

For both approaches we are using the left-right model (Bakis) in order to consider the writing arrangement of characters. The observation sequences are emitted on the model transitions in order to take advantage of the explicit segmentation. Figure 6(a) details the character model that we have chosen, according to the definitions met in [8]. This architecture is able to consider the several configurations of the segmentation. Thus, the transition $t_{03} = \Phi$ models character under-segmentation. The transitions t_{01}, t_{12} and t_{23} model character segmentation into 2 or 3 graphemes. This topology permits a better absorption in the homogeneity of the graphemes provided by the segmentation. Considering uppercase and lowercase characters, we have 40 HMMs. Since the month alphabet is reduced to 20 character classes, we do not consider unused characters.

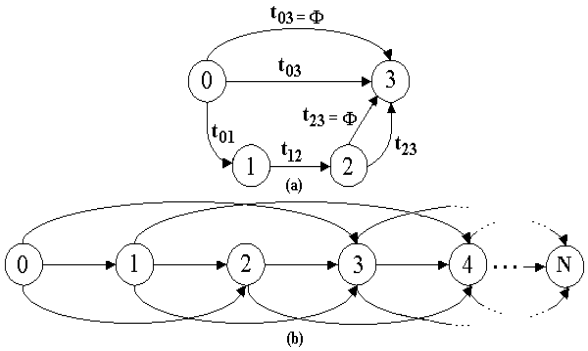

Fig. 6. (a) Character model, (b) Word model with N states

Figure 6(b) shows our word model where 3 or less transitions start from each state in order to consider the several configurations of the word segmentation. In this approach, the transitions do not have straightly the segmented characters but the purpose is to assimilate the over-segmentation and the interaction between characters in a word. Considering uppercase and lowercase month words, we have 24 HMMs. A word is classified as uppercase if it contains at least half number of uppercase characters. The mixed case was not considered in this study, because of the small size of the training database. For a given model, the number of states is determined during the training step.

4.2 Training

The training step is based on the best estimation of the model parameters for a given training set of observation sequences. These parameters are provided by a variant of the *Baum-Welch* algorithm with the *cross-validation* approach [8]. Two sets are used: training and validation sets.

For the analytical approach, each word model is built by the concatenation of the appropriated character models. In this case, the last state of a character model becomes the initial state of the next model, and so on (see Figure 7).

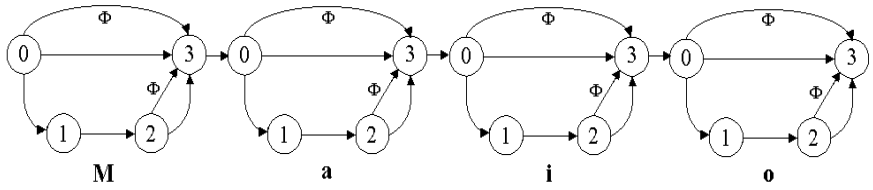

Fig. 7. Training model of class "Maio" (May)

4.3 Recognition

The recognition approaches are based on the maximization of the *a posteriori probability* that a word w has generated an unknown observation sequence O, such as:

$$P(\overline{w}|O) = \max_{w} P(w|O) \tag{1}$$

Applying Bayes rule and after simplification, the recognition decision becomes equivalent to maximizing the joint probability:

$$P(\overline{w}, O) = \max_{w} P(O|w)P(w) \tag{2}$$

where $P(w)$ is the *a priori probability* of the word w (class distribution in the training set). In the current recognition module, we are not considering $P(w)$.

In the analytical approach, the word construction follows the same paradigm defined in the training step. We adopted two character models in parallel (uppercase, lowercase) (see Figure 8) [8], since no information in recognition is available on the handwritten style (uppercase, lowercase) of an unknown word to be recognized. The word models consist of an initial state (I) and a final state (F), and two consecutive character models linked by four transitions: two uppercase characters (UU), two lowercase characters (LL), one uppercase character followed by one lowercase character (UL) and one lowercase character followed by one uppercase character (LU). The probabilities of these transitions are estimated by their

occurrence frequency in the learning set. In the same manner, the probabilities of beginning a word by an uppercase character (0U) or lowercase character (0L) are also estimated. In the holistic approach, we are considering two word models (uppercase and lowercase) for each class. Thus, the estimation of a word probability is given by the combination (sum) of the probabilities associated with the two word models, considering the *a priori probability* of each model.

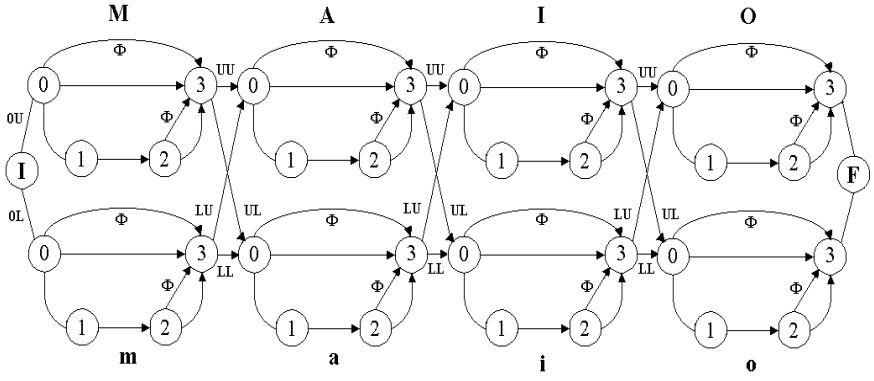

Fig. 8. Recognition model of class "Maio" (May)

5 Experiments and analysis

From our laboratory database which contains 2,000 images, we used 1,188 images for training and 408 images for validation. The distribution of lowercase words is about 80%, against 20% for the uppercase words.

The experiments were carried out on a test set with 404 word images. We used both Viterbi and Forward algorithms to proceed the recognition. However, Table 2 details the best results obtained only with the forward approach. The third and fourth columns detail the recognition rate for each word class ("1" for "Janeiro", etc...), for the holistic (H) and for the analytical (A) approach respectively. The last column (C) gives the result of the combination of both approaches. The last line denotes the average recognition rate for each system with no rejection mode. Considering both approaches, we can observe that their scores are similar. This can probably be explained by the fact that the holistic models are more flexible, since the number of states in this case is not fixed a priori, but optimally derived during the training process. Moreover, the holistic models can assimilate better the interaction between the characters in the same word, due to the strong irregularity of the character segmentation in unconstrained words.

One peculiarity of this context concerns the likelihood of several words, such as:

Table 2. Recognition results with test set

Class n^0	of Images	H (%)	A (%)	C (%)
1	39	82.1	84.6	89.7
2	32	78.1	84.4	87.5
3	36	69.4	69.4	69.4
4	39	87.2	89.7	87.2
5	38	78.9	81.6	81.6
6	30	80.0	80.0	80.0
7	33	78.8	60.6	84.8
8	28	89.3	89.3	89.3
9	31	87.1	87.1	90.3
10	30	86.7	96.7	93.3
11	34	91.2	91.2	88.2
12	34	70.6	67.6	70.6
Total	404	81.4	81.7	84.2

- the terminations in "eiro" for "Janeiro" and "Fevereiro";
- the terminations in "embro" for "Setembro", "Novembro" and "Dezembro";
- almost all characters between "Junho" and "Julho" and between "Maio" and "Março".

Depending on the discrimination abilities of the primitive extractor, all these similarities can seriously affect the performance of the recognition system.

Figures 9(a) and (b) show some examples of correct classification and Figures 9(c) and (d) show some recognition errors which correspond to the main problems of our system. These problems correspond to under-segmentation, high character distortion, lack of training samples, etc.

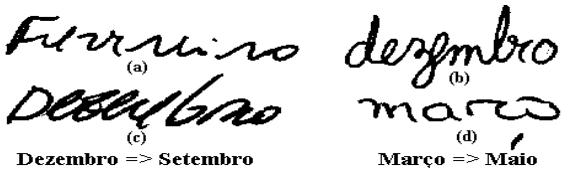

Fig. 9. (a) and (b) Examples of correct classification, (c) and (d) Examples of classification errors

Until now, we are strongly limited to compare our system with other applications dealing with the same context. Fan et al. [3] present one variant of an approach extended by Suen et al. in [7] which deals with dates on canadian checks written in french or in english. These works are more focused on field segmentation problems.

6 Conclusion and perspectives

This article describes the first stage of a system developed to recognize handwritten dates on brazilian checks. The challenge of this study remains in the problematic scheme of processing as a whole several data types such as words and digits in an omni-writer context.

We detail in this article our first work that deals with the recognition of the isolated month words, using HMMs with a limited lexicon. From the same primitive set, we use both holistic and analytical approaches to improve the global recognition rate of the system. Moreover, the analytical approach permits the creation of a character database with a greater number of training examples than in holistic approach. The segmentation module provides a grapheme sequence depending on the ligature location in the cursive word.

The recognition combination provides an averaged rate of 84%. We can consider these first results as satisfactory given the small size of our database and the limitations of the primitive extraction to discriminate uppercase characters.

Our future work will be dedicated to the implementation of a new primitive set in order to improve the discrimination between the several writing styles. Finally, we are studying a new approach to consider the date field as a whole in order to process the day, the month and the year in the same system.

References

[1] M. Y. Chen, A. Kundu, and S. N. Srihari. Variable duration hidden markov model and morphological segmentation for handwritten word recognition. *IEEE Transactions on Image Processing*, 4(12), December 1995.

[2] G. Dimauro, S. Impedovo, G. Pirlo, and A. Salzo. Automatic banckcheck processing: A new engineered system. *International Journal of Pattern Recognition and Artificial Intelligence*, pages 467–503, 1997.

[3] R. Fan, L. Lam, and C. Y. Suen. Processing of date information on cheques. In *Fifth International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pages 207–212, September 1996.

[4] A. M. Gillies. Cursive word recognition using hidden markov models. In *Fifth U.S. Postal Service Advanced Technology Conference*, pages 557–562, 1992.

[5] M. Gilloux, M. Leroux, and J. M. Bertille. Strategies for handwritten words recognition using hidden markov models. pages 299–304, 1993.

[6] M. A. Mohamed and P. Gader. Handwritten word recognition using segmentation-free hidden markov modeling and segmentation-based dynamic programming techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):548–554, May 1996.

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

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