A Discriminative Approach to On-Line Handwriting Recognition Using Bi-Character models

S.Prum*, M.Visani*, A. Fischer†, JM.Ogier*
*L3i Laboratory, University of La Rochelle, France
{sophea.prum, muriel.visani, jean-marc.ogier}@univ-lr.fr
†CENPARMI, Concordia University, Montreal, Canada
an_fisch@encs.concordia.ca

***Abstract*—Unconstrained on-line handwriting recognition is typically approached within the framework of generative HMM-based classifiers. In this paper, we introduce a novel discriminative method that relies, in contrast, on explicit grapheme segmentation and SVM-based character recognition. In addition to single character recognition with rejection, bi-characters are recognized in order to refine the recognition hypotheses. In particular, bi-character recognition is able to cope with the problem of shared character parts. Whole word recognition is achieved with an efficient dynamic programming method similar to the Viterbi algorithm. In an experimental evaluation on the Unipen-ICROW-03 database, we demonstrate improvements in recognition accuracy of up to 8% for a lexicon of 20,000 words with the proposed method when compared with an HMM-based baseline system. The computational speed is on par with the baseline system.**

Keywords—on-line handwriting recognition; combining on-line and off-line features; support vector machine; bi-character recognition; dynamic programming

I. Introduction

Research in on-line handwriting recognition (HWR) started during the 1960's and has been re-activated in the 1980's, after a break in the 1970's [15]. In contrast to off-line recognition of handwritten document images, on-line recognition considers temporal handwriting information provided by electronic devices such as PDAs, tablets, and smartphones. Jointly with the on-line signal, many methods also consider the off-line shape of the characters reconstructed from the on-line signal. The HWR system proposed in this paper relies on both the on-line signal and the reconstructed off-line image.

There are two main types of approaches to HWR: global and analytical approaches [12]. Global approaches consider an input word as a whole for recognition. Systems based on this type of approach rely on a large training set containing all the words in the lexicon, and therefore cannot be used with an open lexicon. For these reasons mainly, global approaches seem to be progressively abandoned. In contrast, systems relying on analytical approaches recognize an input word as a sequence of characters. They require a preliminary segmentation step to split the input word into graphemes, *i.e.* characters or subparts of characters. The graphemes are then analyzed by a set of character models covering all the possible characters of the considered language, in order to find the sequence of characters which compose the input word. Analytical approaches are therefore adapted to open lexicon applications. Moreover, no re-training is required when modifying the lexicon. Therefore, these approaches are convenient for large and flexible lexicon applications.

Due to the high variability in writing styles, it is in general not possible to segment cursively written words into characters before recognition. In order to tackle this problem, an implicit segmentation based on hidden Markov models (HMM) and Viterbi recognition is very frequently applied to HWR [10], [13], [4]. In this approach, input words are first oversegmented with a narrow analysis window. The segments are then combined to characters and words during recognition with respect to the HMM. Within this recognition framework, the rich repository of discriminative classifiers such as support vector machines (SVM) can, unfortunately, not be applied directly. Attempts towards discriminative on-line HWR include neural network based recognition [5], [7], [11], often in form of hybrid systems in combination with HMM to address the segmentation problem [5], [11].

In this paper, we propose a discriminative approach to on-line HWR that combines arbitrary discriminative classifiers with Viterbi-like recognition. First, input words are explicitly segmented into graphemes. Then, different grapheme combinations are classified by a single character recognizer with the option to reject combinations that do not match any character of the alphabet. The recognition result is further refined with bi-character classifiers in order to cope with the problem of shared character parts. Afterwards, the recognition hypotheses are efficiently processed by dynamic programming to find an optimal sequence of characters for the input word.

The proposed system can be used together with a large and flexible lexicon, it is adapted to omni scriptwriter applications, without any special requirement concerning the capture devices. An experimental evaluation is performed on the Unipen-ICROW-03 database. Using a kernel SVM for single character recognition and a Logistic Regression classifier for bi-character recognition, we demonstrate that the proposed method outperforms an HMM-based baseline system.

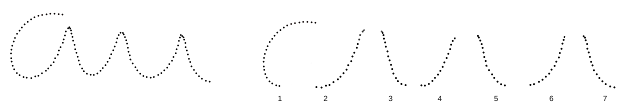
The paper is organized as follows: the proposed method is described in section II, while section III presents the experimental results, and section IV draws some conclusions.

II. Proposed system

Our method relies on explicit grapheme segmentation and SVM-based character recognition. In addition to single character recognition with rejection, bi-characters are considered in order to refine the recognition hypotheses. Finally, word recognition is achieved with an efficient dynamic programming method similar to the Viterbi algorithm.

First, the input signal is normalized and segmented into a set of graphemes. These segmented graphemes are used to create a lattice of L levels (see section II-1) where each node is considered as a candidate character and each pair of neighboring nodes is considered as a candidate bi-character. At the character level, a character recognizer is used to emit recognition hypotheses for each node (see section II-A). At the bi-characters level, we use bi-characters recognizers so as to refine the hypotheses emitted at the single character level (see section II-B). In the last stage, both the outputs of the single character recognizer and the bi-character recognizer are used in a word decoding process providing the W most probable words (see section. II-C).

1) Pre-processing, segmentation and lattice creation: Variation and noise in on-line handwriting signals may be caused by different factors such as caption device, writing speed, writing context, etc. These problems have a great impact on the system. Therefore, pre-processing and normalization methods are required in order to remove noise and standardize the input signal. Four standard pre-processing methods are sequentially applied: size normalizing, interpolating missing points, smoothing and re-sampling.



(a) Input signal (b) Segmented graphemes

Fig. 1. Example of the segmented graphemes.

Once the input signal is normalized, it will be over-segmented into a set of graphemes using the segmentation method proposed in [2]. A grapheme is defined as the set of all the consecutive points of a given stroke located between a local minimum and a local maximum (on the y coordinate), as illustrated in Fig. 1.

Then, these segmented graphemes are used to create a lattice of L levels which represents all the possible concatenations of graphemes (for an example see Fig. 2). A node in the lattice may be a grapheme or a concatenation of graphemes. The value in each node corresponds to the index of the starting and ending grapheme. Each node is considered as a candidate to be a character and is therefore introduced as the input of the single character recognizer (described in the following subsection II-A).

A. Single character recognition system (SCR)

Obviously, the SCR is a crucial system component in the proposed approach. It is used to recognize each node of the lattice. However, some nodes may correspond to an unknown pattern, *i.e.* an intermediate information that does not correspond to a single character. Therefore, the SCR must be able to reject the unknown patterns.

As usual in pattern recognition, our SCR relies on two steps: feature extraction (see section II-A1) and recognition (see section II-A2).

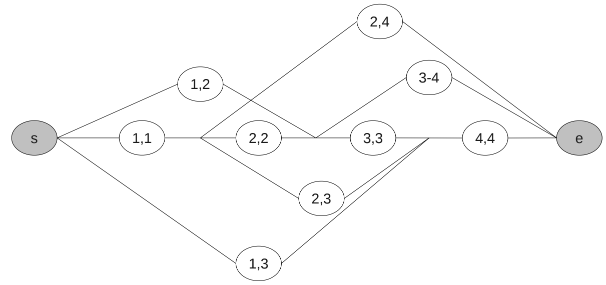


Fig. 2. Example of a 3 level lattice created from 4 segmented graphemes.

1) Feature extraction: In the literature, authors agreed that a single feature extraction method is insufficient to cover the problem of variation in handwriting [9]. Combining different methods clearly allows to improve the effectiveness of the systems. Therefore, both categories of features (on-line and off-line) are used in our proposed system in order to take profit of their complementarity.

From the sequence of points of the on-line signal, an artificial image of the word can be reconstructed. The reconstruction is obtained by connecting the points in the sequence based on their times of acquisition (returned by the capture device), and applying a dilatation on the resulting image. This image is considered as the off-line representation. Our method relies on the combination of on-line and off-line features (computed respectively from the on-line signal and the off-line image). We combine both kinds of features for their complementarities. Seven families of off-line statistical and structural features described in [9] are used. Furthermore, in order to enrich the shape description, we add Radon transform which provides a set of projections of the input pattern in different angles, and Zernike moments which are invariant to rotation, translation and scale condition. We also consider on-line features: the number of strokes, the information of starting and ending points of the signal (x, y coordinates, gradients at the start and end points).

Off-line and on-line feature vectors are then concatenated to obtain a single feature vector with a rather large size of 254 dimensions. In order to circumvent the problems related to this large size and for improving the computational efficiency, we use the Sequential Forward Floating Search (SFFS) feature selection method, reducing the size of this vector to F.

For each grapheme or combination of graphemes (*i.e.* for each node in the graph), its feature vector of size F is computed and provided to the single character recognizer.

2) Recognition: Discriminative methods such as Neural Networks (NN), particularly Multilayer Perceptrons (MLP), or kernel Support Vector Machine (SVM), are among the most frequently used in the literature for single character recognition. Both classifiers are able to deal with non linearly separable data and provide very competitive recognition results in many research problems. In the literature, SVM has provided successful results for isolated character/digit recognition systems. In addition, experimental results given in [1] have shown that SVM provides better results than MLP and TDNN for isolated character recognition. For these reasons, we choose to use kernel SVM in our system.

B. Bi-character model

Characters composing cursive handwritten words are naturally more or less connected. Trying to recognize the sequence of characters composing the input signal, the system may face the problem of *character shared part*, that occurs when a given character has a visual appearance very similar to another character (for instance the loop in the character 'd' may look like a 'o'). Let us consider the example given in Fig. 3 by supposing that there are only two best paths: path (1) and path (2). Using only the SCR, path (1) would be recognized as the sequence of characters "ole" (or "oll", or "oil" for instance). Indeed, the shape 1.1 and 1.2 are parts of character "d". If they were observed individually, the visual appearance of the shape 1.1 is very close to character 'o' or 'O'. Hence, it may be recognized by the single character recognizer as character 'o' or 'O' with high probability. *Idem* for the shape 1.2, which may be recognized as character 'i' or 'l' with high probability. In some cases, this problem can be corrected by using a lexicon. But the effectiveness of the system strongly depends on the lexicon. In addition, the lexicon cannot correct all the possible ambiguities, therefore a complex language model is required in order to solve this kind of problems.

Our idea for solving this problem is the following: using bi-characters models, we concatenate the shapes 1.1 and 1.2, to obtain the new shape 1.12. This new shape is further submitted to bi-characters models based on the single character recognizer output. The shape 1.12 will be rejected by these models since its visual appearance is very different from the visual appearance of the hypotheses associated to bi-characters {"ol","oi", "Ol" and "Oi"}. Hence, the probability that the path (1) will be selected is depreciated. As a consequence, the path (2) will be selected since the shape 2.12, which is the concatenation of shapes 2.1 and 2.2, is very close to the visual appearance of "de". Finally, the input signal will be recognized correctly as "de".

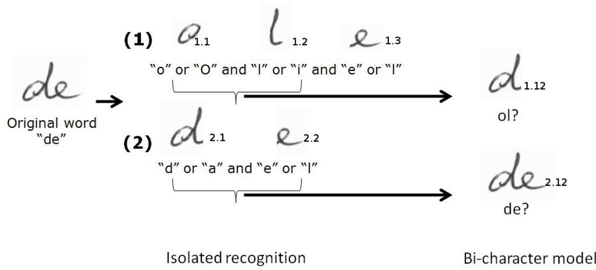


Fig. 3. An example of how the bi-character level can solve the problem of the character shared part.

In the context of bi-characters models, the number of classes to be recognized is very important compared to the SCR, since there are n^2 possible bi-characters in an alphabet of n characters (which makes for instance $26 * 26 = 676$ bi-characters classes in case of the Roman alphabet). Furthermore, as explained above, the bi-characters models are used for verification (*i.e.* authentication) of the hypotheses provided by the SCR. For these reasons, we choose to build one classifier for each of the n^2 bi-characters, using the one-against-all strategy. The model $B_{c_i c_j}$ of a bi-character class $c_i c_j$ is therefore trained by considering that samples belonging to class $c_i c_j$ are positive samples (*i.e.* positive class) and the samples belonging to all other classes are the negative samples (*i.e.* negative class), where $c_i, c_j \in \{a, b, c, \ldots, z\}^2$. This problem becomes a binary classification problem.

In order to limit the computational cost during the training stage and the storage needs, in our previous publication [14], linear SVM is applied by using LibLinear library [6]. Due to the limitation of this library, the linear SVM returns only a Boolean value, which doesn't provides enough information for the proposed system. Therefore, in this version of bi-character models, we use the Logistic Regression classifier (LR) which is implemented in the LibLinear library and provides recognition probabilities.

C. Word decoding using dynamic programming

For every input handwriting signal, a lattice corresponding to all the possible grapheme concatenations has been created (see section II-1) and enriched with the corresponding single characters and bi-characters probabilities (see sections II-A2 and II-B). This lattice is explored using dynamic programming where, for each word w_i in the dictionary and each lattice (corresponding to a word to recognize), the optimum path is computed together with its probability. Then, for each input word to recognize, the output of the word decoding system is the ordered list of the W most probable words according to dynamic programming.

Let us consider the word to recognize as a sequence of nodes $o_{t,t'}$ in the corresponding lattice, where each node is indexed by its starting grapheme t and ending grapheme t', T is the total number of graphemes in the word and L is the maximum level of the lattice (*i.e.* the maximum number of graphemes in a character). The recognition score $P(T, c_K)$ provided by the model of the word w_i containing a sequence of K characters ($\{c_1, c_2, \ldots, c_K\}$), is computed using equation 1. It is illustrated in Fig. 4, where $T = 4$, $L = 3$ and $w_i = \{i, n\}$.

$$\begin{aligned} P(t, c_k) = \max_{m=1..L} [&P(t-m, c_{k-1}) \\ &b(c_k|o_{t-m+1,t}) \\ a(c_{k-1}, c_k|r(t-m, c_{k-1}), &o_{t-m+1,t)})] \end{aligned} \tag{1}$$

Where

- $t \in \{1, 2, \ldots, T\}$
- $b(c_k/o_{i,j})$: probability that node $o_{i,j}$ is recognized as character c_k. This probability is given by the SCR (see section II-A2).
- $a(c_k, c_l/o_{i,j}, o_{p,q})$: output probability provided by the bi-character model B_{c_k,c_l} for the pair of neighboring nodes $o_{i,j}$ and $o_{p,q}$ (see section II-B).
- $r(t-m, c_{k-1})$ is the retained node found for the sequence of characters $\{c_1, \ldots, c_{k-1}\}$.

Under the following conditions:

- if $k = 1$ and $t \leq L$ then $P(t, c_1) = b(c_1|o_{1,t})$
- if $k = 1$ and $t > L$ then $P(t, 1) = null$, since $o_{1,t}$ cannot be a node (as it would be at a higher level than the maximum level L

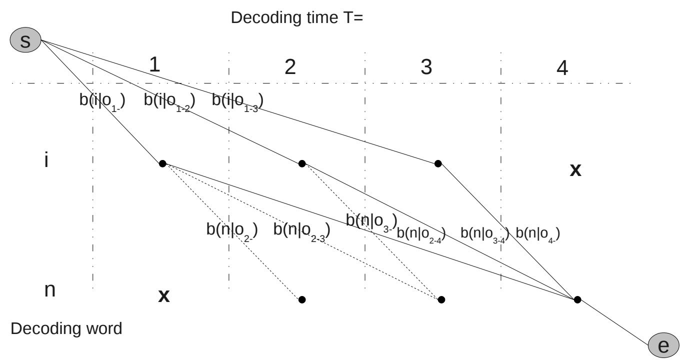


Fig. 4. Illustration of the decoding process of the word $'in'$ with $L = 3$ and $T = 4$ using dynamic programming.

of the lattice). An example of such case is given in Fig. 4 (first cross) starting from top.

- if $t < k$ then $P(t, k) = null$, as computation is impossible (for an example see the second cross in Fig. 4).

Taking the example of Fig. 4, decoding process of word $'in'$ while $t = 4$ and $c_k = n$ is as below:

$$
\begin{aligned}
P(4, i) &= null \\
P(4, n) = \max[&P(1, i) * b(n|o_{2,4}) * a(i, n|r(1, i), o_{2,4}), \\
&P(2, i) * b(n|o_{3,4}) * a(i, n|r(2, i), o_{3,4}), \\
&P(3, i) * b(n|o_{4,4}) * a(i, n|r(3, i), o_{4,4})]
\end{aligned}
$$

We have to mention that, in our preliminary experiments, a flat search strategy is used. A speedup could be achieved with a TRIE representation of the lexicon instead of decoding each word of the lexicon separately.

III. Experimental results

A. Single character recognition system

In a first set of experiments, we have evaluated the SCR system (see section II-A) individually. Three experiments are performed in order to evaluate this system. In these experiments, the Radial Basis Function (RBF) kernel is used in the SVM classifier.

In experiment 1 (Exp.1), the SCR system is trained with isolated characters randomly selected from the IRONOFF and UNIPEN databases [8], [17]. 45 features selected by the feature selection method are used (see section II-A1). In experiment 2 (Exp.2), the SCR system is trained with single characters segmented from handwritten words. This data has been segmented by Ahmad *et al.* [2] using a commercial application. 71 features given by the feature selection method are used (see section II-A1). In experiment 3 (Exp.3), the SCR system is trained with a rejection option by adding an additional garbage class. It uses the same training data and features as the experiment 2. In each experiment, the training set contains 1600 characters per class and the test set contains 400 characters per class, randomly selected from the corresponding databases. The samples of the garbage class are various unknown shapes, segmented manually from handwritten words.

Table I. gives a comparison of the recognition rates of the proposed system (Exp.1) and the system presented in [3]. This comparison was already published in previous work [16]. Our proposed system is competitive in terms of effectiveness and efficiency. The average recognition time per character provided by our proposed system is 3.41ms *vs* $18,51$ms provided by the system presented in [3]. That is, our system is roughly 5 times faster. This speedup can be explained by the fact that, in our system, only 45 features are used while the system presented in [3] uses up to 210 features.

Table II. gives the Top-N recognition rates of the experiments 2 and 3. The Top-1 recognition rates are lower than the Top-1 recognition rate of experiment 1. This may be explained by the fact that the shapes of characters segmented from handwritten words are more heterogeneous than the shapes of isolated characters.

TABLE I. Comparison of recognition rate between our proposed system and the system presented in [3].

	Recognition rate (%)	
Nature of data	Exp.1	Sys. in [3]
Digit	98.7	98.6
Uppercase	95.6	95.1
Lowercase	93.3	93.7

TABLE II. Single character recognition rate: system without and with garbage class.

	Recognition rate(%)	
Top-N	Exp.2	Exp.3
1	88.23	87.69
3	96.99	96.86
5	98.44	98.42
7	99.02	98.94
10	99.34	99.31

B. Word recognition system

In order to assess the performances of our HWR system and its robustness towards an increase in the size of the lexicon, we perform a series of experiments where the size of the lexicon (containing no accented words) is progressively increased: 5000, 10000 and 20000 words. The words in the lexicon are randomly selected from an English dictionary. The test set contains 3614 writings (in lowercase characters) selected from the Unipen-ICROW-03 database.[1] This database is suitable to evaluate unconstrained on-line handwritten word recognition systems, as it contains free-style writings (handprint, mixed and cursive) written by 72 script-writers of different nationalities (Dutch, Irish, Italian, and mixed) and using different kinds of caption devices.

Two experimental configurations of the proposed HWR system are used. In the configuration 1 (Conf.1), all the outputs provided by SCR are used. According to the experimental results of the SCR (see Table.II), the Top-7 recognition rate is up to $\sim 99\%$ while the recognition rate at Top-8 remains stable. Therefore, in configuration 2 (Conf.2), we consider only the 7 most probable character candidates provided by the SCR. For both configurations, we use a lattice with $L = 7$ levels of graphemes (value corresponding to the maximum number of graphemes contained in a single character). Two variants of the SCR system are considered, one with and one without garbage class.

[1] http://www.ai.rug.nl/∼lambert/unipen/icdar-03-competition/_README

An HMM-based system is used as a baseline for comparison. This system relies on a linear HMM topology. It uses the pre-processing method presented in section II-1. Three families of local on-line features are used: normalized x, y coordinates, sine and cosine of the curvature angle, as well as sine and cosine of the direction angle. These features are often used for on-line HRW systems [11]. The reference system is trained using 72028 writings (in lowercase characters) selected from the UNIPEN database [8]. The number of Gaussian mixtures is fixed to 10 and the number of states per character varies from 9 to 14 depending on the character class. These parameters are optimized on a validation set which contains 800 writings selected from the Unipen-ICROW-03 database. No pruning is used for Viterbi-based recognition.

TABLE III. COMPARISON OF THE TOP-1 RECOGNITION RATES AND COMPUTATIONAL TIMES (CT) OF THE PROPOSED HWR SYSTEM WITH THE HMM-BASED BASELINE SYSTEM.

Lexicon size c	Garbage class	Configuration 1		Configuration 2		HMM based	
		Top1(%)	CT(s)	Top1(%)	CT(s)	Top1(%)	CT(s)
5000	No	68.32	20	68.16	6	65.15	9
	Yes	71.12	20	70.84	6		
10000	No	63.37	34	64.14	7	60.19	22
	Yes	66.91	33	67.02	7		
20000	No	59.16	62	59.57	9	54.51	40
	Yes	62.07	60	62.59	8		

According to the experimental results reported in Table III, three observations can be made:

1) N-best character recognition (configuration 2) greatly improves the efficiency of the system when compared with a full search (configuration 1). For the 20000 word lexicon, a speedup factor of ~ 7.5 is achieved while the recognition accuracy remains stable.
2) Adding rejection in the SCR improves the recognition rates by roughly 3%.
3) Our proposed system is both more efficient and more effective than the HMM-based baseline system. Using lexicons of 5000, 10000 and 20000 words, the recognition rate of the proposed system is respectively $\sim 5.5\%$, $\sim 7\%$, and $\sim 8\%$ better than the recognition rate of the reference system. The improvements are larger for increasing lexicon size.

IV. CONCLUSION AND FUTURE WORK

In this paper, we propose a new discriminative approach to on-line handwriting recognition. The method relies on explicit grapheme segmentation, single character recognition with rejection, bi-character classifiers and an efficient dynamic programming approach to recognize unconstrained cursive handwriting. Computational speed is improved by taking only the N-best character candidates into account.

An experimental evaluation is performed with an RBF SVM for single character recognition and a Logistic Regression classifier for bi-character recognition. On the Unipen-ICROW-03 database we demonstrate that the proposed system outperforms an HMM-based reference system both in terms of accuracy and computational speed. For a lexicon size of 20000 words, we report an improvement in accuracy of 8%.

Future work includes the investigation of other discriminative classifiers within the proposed framework, the recognition of complete text lines instead of single words, and a broader comparison with different HWR reference systems.

ACKNOWLEDGMENTS

This work has been supported by RecoNomad project founded by OSEO and the Swiss National Science Foundation fellowship project PBBEP2_141453.

REFERENCES

[1] A. Ahmad, Bin Khalid Marzuki, and Christian Viard-Gaudin. Comparison of support vector machine and neural network in character level discriminant training for online word recognition. In *UNITEN Students Conference on Research and Development , SCORED*, page 8 pages, Malaisie, 2004.

[2] A.R. Ahmad, V.G. Christian, and M. Khalid. Lexicon-based word recognition using support vector machine and hidden markov model. In *ICDAR*, pages 161–165, 2009.

[3] Viard-Gaudin E.Poisson A.R. Ahmad, M.Khalia. on-line handwriting recognition system using support vector machine. In *Analog and digital techniques in electrical engineering. Conference*, Chiang Mai , Thailande, 2004.

[4] A.-L. Bianne-Bernard, F. Menasri, R.A.-H. Mohamad, C. Mokbel, C. Kermorvant, and L. Likforman-Sulem. Dynamic and contextual information in hmm modeling for handwritten word recognition. *PAMI, IEEE Transactions on*, 33(10):2066 –2080, oct. 2011.

[5] S. Espana-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez. Improving offline handwritten text recognition with hybrid hmm/ann models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(4):767–779, April 2011.

[6] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, June 2008.

[7] Alex Graves, Marcus Liwicki, Santiago Fernandez, Roman Bertolami, Horst Bunke, and Jurgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on PAMI*, 31:855–868, 2009.

[8] Isabelle Guyon, Lambert Schomaker, Rejean Plamondon, Mark Liberman, and Stan Janet. UNIPEN project of on-line data exchange and recognizer benchmarks. In *International Conference on Pattern Recognition*, 1994.

[9] L. Heutte, T. Paquet, J.V. Moreau, Y. Lecourtier, and C. Olivier. A structural/statistical feature based vector for handwritten character recognition. *Pattern Recognition Letters*, 19(7):629–641, 1998.

[10] Jianying Hu, Sok Gek Lim, and Michael K. Brown. Writer independent on-line handwriting recognition using an hmm approach. *Pattern Recognition*, 33(1):133 – 147, 2000.

[11] S. Jaeger, S. Manke, J. Reichert, and A. Waibel. Online handwriting recognition: the npen++ recognizer. *International Journal on Document Analysis and Recognition*, 3:169–180, 2001. 10.1007/PL00013559.

[12] A. L. Koerich, R. Sabourin, and C. Y. Suen. Large vocabulary off-line handwriting recognition: A survey. *Pattern Anal. Appl.*, 6(2):97–121, 2003.

[13] M. Liwicki and H. Bunke. Hmm-based on-line recognition of handwritten whiteboard notes. In *International Workshop on Frontiers in Handwriting Recognition*, 2006.

[14] Sophea Prum, Muriel Visani, and Jean-Marc Ogier. International conference on frontiers in handwriting recognition, icfhr 2010, kolkata, india, 16-18 november 2010. In *ICFHR*. IEEE Computer Society, 2010.

[15] C.C. Tappert, C.Y. Suen, and T. Wakahara. State of the art in on-line handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12, 1990.

[16] De Cao Tran, Patrick Franco, and Jean-Marc Ogier. Accented handwritten character recognition using svm - application to french. In *ICFHR*, pages 65–71. IEEE Computer Society, 2010.

[17] Christian Viard-Gaudin, Pierre Michel Lallican, Philippe Binter, and Stefan Knerr. The ireste on/off (ironoff) dual handwriting database. In *Proceedings of the Fifth ICDAR*, ICDAR '99, pages 455–, Washington, DC, USA, 1999. IEEE Computer Society.