2013 12th International Conference on Document Analysis and Recognition

Feature extraction with convolutional neural networks for handwritten word recognition

Théodore Bluche*†, and Hermann Ney †‡, Christopher Kermorvant*
*A2iA SA, Paris, France
†LIMSI CNRS, Spoken Language Processing Group, Orsay, France
‡RWTH Aachen University, Human Language Technology and Pattern Recognition, Aachen, Germany

***Abstract*—In this paper, we show that learning features with convolutional neural networks is better than using hand-crafted features for handwritten word recognition. We consider two kinds of systems: a grapheme based segmentation and a sliding window segmentation. In both cases, the combination of a convolutional neural network with a HMM outperform a state-of-the art HMM system based on explicit feature extraction. The experiments are conducted on the Rimes database. The systems obtained with the two kinds of segmentation are complementary : when they are combined, they outperform the systems in isolation. The system based on grapheme segmentation yields lower recognition rate but is very fast, which is suitable for specific applications such as document classification.**

I. Introduction

Offline handwritten word recognition is the conversion of an image into text. Compared to many computer vision problems, the images of text are simple. We can often binarize them, and the task consists of recognizing the shape of the ink. However, beyond a relatively small vocabulary size, it becomes infeasible to build a word classifier, due to the large number of classes and to data sparsity. The common unit for handwriting recognition is the character. The goal is to recognize a sequence of characters. In practice we obtain word models by concatenation of character models, and we can use a vocabulary and a language model to constrain the search.

The cursive nature of handwriting makes it difficult to segment a word image into character images. Hidden Markov models (HMMs) help to alleviate this problem. A word HMM, although made of character HMMs, can recognize a word as a whole. It performs an implicit segmentation that can be retrieved by looking at the sequence of states in the best path.

The HMMs need a sequence of observation vectors, which is basically a one dimensional signal, while the image can be considered as a two dimensional signal. The main methods to obtain this sequence are the sliding window method and the heuristic over-segmentation method. The sliding window technique consists of scanning a window horizontally over the image, and consider the content of the window at each position as an observation. The over-segmentation techniques define heuristic rules to break the image into parts, which are the observations.

This work was partly achieved as part of the Quaero Program, funded by OSEO, French State agency for innovation and was supported by the French Research Agency under the contract Cognilego ANR 2010-CORD-013.

Rather than using the list of pixel values in the extracted images directly, we compute feature vectors. The simplest approach applies a PCA on the list of pixel values, but it relies on a good preprocessing. Higher level approaches may consist in handcrafting features (e.g. [1]), which takes time, or to let a system learn them directly with auto-encoders [2], or neural networks [3].

Neural networks, and especially deep neural networks, learn intermediate representations of their inputs, that are useful for a subsequent classification task. Their combination with HMMs improved the performance of both speech recognition [4], [5], and handwriting recognition [3], [6], [2].

With their particular structure, including the notion of receptive fields via weights sharing and distortion invariance with pooling operations, convolutional neural networks (ConvNN) [7] handle conveniently two-dimensional structures such as images, and can incorporate many hidden layers - hence many intermediate representations - while keeping the total number of free parameters relatively small. They have been successfully applied to computer vision problems, particularly to isolated character recognition [8], [9], [7] and handwriting recognition [10], [11].

In this, parper, we compare two segmentation strategies - sliding window and heuristic segmentation - on the one hand, and several feature extraction techniques (PCA, handcrafted, and ConvNN) on the other hand. The convolutional neural networks are used with HMMs either in hybrid mode [12], where they both extract features and compute emission probabilities for the HMM, and in tandem [13] mode, where the computed state posteriors are used as features for standard GMM-HMMs.

We evaluate the performance of the proposed systems on the publicly available Rimes database. We show that the systems using a sliding window give superior performance. The tandem combination achieves similar performance as LSTM-RNN, which gave the best results on Rimes database since the ICDAR competition in 2009. Yet the systems using heuristic over-segmentation remain very fast, and when combined with a ConvNN, they achieve a level of performance that may be sufficient for applications such as document classification. Finally, we observe that both kind of systems make different errors, and their combination outperforms each system in isolation.

The remaining of this paper is organized as follows. Section II presents the relation of this work to earlier studies. Section

1520-5363/13 $26.00 © 2013 IEEE
DOI 10.1109/ICDAR.2013.64

III describes our systems. We discuss our experiments and results in Section IV, and conclude with perspectives in Section V.

II. Relation to prior work

Using neural networks with HMMs, to extract features or predict the HMM states is not a new approach in handwriting recognition. For example, Hammerla et al. [2] extract features from a bottleneck autoencoder. They optimize the autoencoder with a regularized non-linear Neighbourhood Component Analysis using the alignments given by the GMM-HMM in an iterative procedure. The results are comparable to those obtained using heuristic features, but not better. In [3], a multi-layer perceptron (MLP) extracts posterior features from consecutive windows of pixels. The authors compare a hybrid and a tandem approach and show that they outperform a GMM-HMM baseline. Similarly, Doestch uses a Long Short-Term Memory Recurrent Neural Network [6] and reports improvements in both hybrid and tandem combination. The tandem approach gives better results than the hybrid combination in these papers.

This work takes advantage of the ability of convolutional neural networks to handle images directly. We combine them with HMMs in the usual way. It is motivated by the significant improvements reported with these approaches in the literature. Like [11], [14], we use ConvNNs with either an explicit or an implicit segmentation approach, but the training procedure, and the association of the ConvNN and the graph is different. We focus on off-line handwriting recognition, and make a comparison with hand-crafted features. We also explore the tandem combination, and report the performance on a publicly available database.

III. System description

A. Pre-processing

We correct the slant in the image with a projection-based method [15], and we enhance the contrast (5% of darkest pixels are mapped to black, 70% lightest to white, and a linear interpolation in between).

For the grapheme segmentation algorithm, and the extraction of hand-crafted grapheme features, the image is binarized with an adaptive thresholding method.

For the systems using a sliding window, we add 20 white pixels on left and right to model the empty context. We also rescale the image to a fixed height of 72px, with different scaling factors applied to three different zones (ascenders, descenders and core region). We ignore this rescaling for the system based on hand-crafted features, which can handle different heights, and works better without rescaling.

B. Grapheme segmentation and feature extraction

The grapheme segmentation [1] is a heuristic over-segmentation algorithm which breaks the ink (the set of black pixels) into several parts. Its aim is to segment the word into units which are either characters or parts of characters. The algorithm first extracts skeletons of connected components, and sets the potential segmentation points. The problem is then represented as a graph, from which graphemes are extracted.

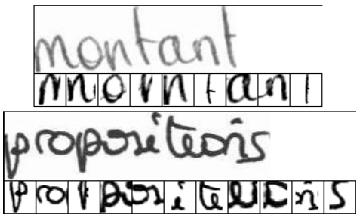

Fig. 1. Grapheme segmentation.

Two examples of grapheme segmentation are illustrated in Figure 1. We can see a good segmentation on top image, but several problems occur in bottom image: the first r and o are segmented as only one grapheme.

The average number of graphemes per character in the training set is 1.38. Only 0.02% of images have more than 3 graphemes/character, and 2% have less graphemes than characters.

From the segmentation, we can compute hand-crafted features [1]. They comprise 74 geometrical (e.g. size of the grapheme, thickness along different directions) and statistical (e.g. pixel densities in different zones) features. Refer to [1] for a complete description of the segmentation algorithm and of the features.

We can also crop the image to the grapheme bounding boxes and extract the pixels values, either applying a PCA with 150 dimensions on the flattened matrix of pixel values, or keeping the two-dimensional structure of the image, which we can give as input of a ConvNN. In order to make all grapheme images the same size, we rescale them to 32x32px in the following way. If the largest dimension is less than 32px, we put the grapheme at the center of a 32x32px blank image (this is the case for accents, or 'i'-dots). Otherwise, we rescale the grapheme image, keeping the aspect ratio, so that the largest dimension is 32px.

C. Sliding window and feature extraction

Again, we extract two kinds of features : hand-crafted and pixels. In hybrid NN/HMM models, several consecutive frames are usually fed to the neural network. Here we just provide it a wider frame, to take advantage of the ability of ConvNNs to handle two dimensional structures. Consecutive overlapping frames would indeed provide the same information several times, which may be a waste of information.

In practice, we want to use the alignements of the GMM-HMM based on hand-crafted features to bootstrap the training of the neural network, because they are visually better than those obtained with the pixel features. Therefore, we need a one-to-one mapping between frames used to extract hand-crafted features and frames provided to the neural network.

We can achieve this by scanning two sliding windows of different sizes, but with the same shift, on the images (the original size one and the normalized one), as illutrated on Figure 2. We use a width of 9px for the hand-crafted features and the and a size of 39px for the pixels.

The hand-crafted features comprise 26 geometrical and statistical features and 8 directional features based on histogram

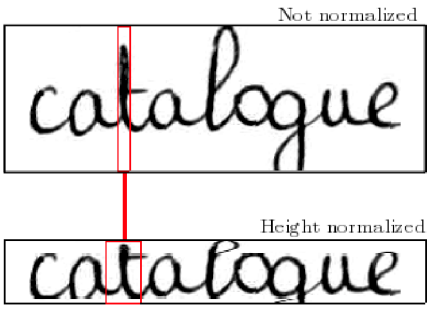

Fig. 2. Sliding window parallel extraction.

of gradients, plus a first-order regression to get 68-dimensional feature vectors [1]. The pixel features are again either obtained with a PCA (30 dimensions kept), or the frames directly, rescaled to 32x32px.

D. HMM modelling

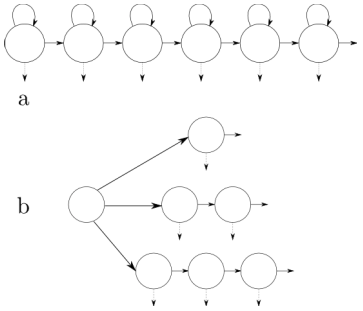

Fig. 3. HMM topology. (a) Sliding window systems, (b) Grapheme systems.

For the sliding window systems, we build six-state models for characters with self-loops and transitions to the next state only, and we add two-state "blank" character models for the beginning and the end of words, corresponding to empty context. For the grapheme systems, we also build HMMs with 6 emitting states, with a special topology, depicted on Figure 3b. For each character, we have three different alternatives, corresponding to the observation of either one, two or three graphemes. The HMMs are built and trained using Kaldi [16].

E. Convolutional neural networks

The architecture of the convolutional neural networks are similar to LeNet5 [8], as shown on Figure 4. They consist of three convolutional layers with 5×5px filters, followed by max-pooling operations. On top of this architecture is a fully connected hidden layer, and an output softmax layer with as many units as there are states in the HMM models (490). We used 32, 64 and 128 feature maps and 700 hidden units for the sliding window system, and 64, 128, 256 feature maps and 1,500 hidden units for the grapheme system. They are trained on a GPU using Alex Krizhevsky's *cuda-convnet* library[1].

[1]http://code.google.com/p/cuda-convnet/

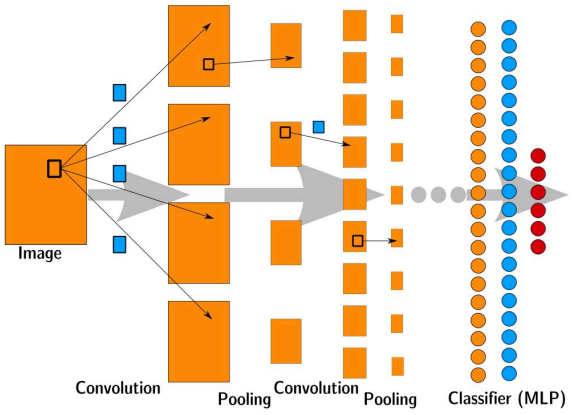

Fig. 4. Convolutional neural network architecture.

F. Combination of ConvNN and HMM

The first method to combine neural networks and HMMs is the hybrid combination [12]. The ConvNN is trained to predict the HMM state s corresponding to a given observation x (i.e. $p(s|x)$). The training requires labels for every observations. They are obtained with a bootstrapping procedure, consisting of training a GMM-HMM first, and then recording the forced alignments. The parallel sliding window extraction descibed in Section III-C allows to use the GMM-HMM based on hand-crafted features to assign labels to the images given to the ConvNN. The dataset we obtain from Rimes training set is divided into new training (90%) and validation (10%) sets. Once trained, the ConvNN, like any neural network, can compute state posteriors, which we divide by the state priors to obtain pseudo-likelihoods $p(s|x)/p(s)^{\alpha}$. The state priors are computed from the forced alignments, and α is tuned on the validation set. The pseudo-likelihoods serve as emission probabilities for decoding.

The second method is the tandem scheme [13]. The neural network extract features, used to train a new standard GMM-HMM. In this work, we used the pseudo-likelihoods, computed as explained above, decorrelated and reduced by PCA to 50 dimensions.

G. Language model

We take into account the prior distribution of words using a unigram language model. Given an observation sequence x, we want to retrieve the most likely word w, i.e. to maximize $p(x|w)^{\kappa}p(w)$. κ is the *optical scaling factor*, tuned on the validation set, used to balance the influence of the optical model with respect to the language model.

The different systems we build are summarized on Table I. The prefix (*gpm* or *sw*) represents the segmentation technique, and the suffixes *pix*, *feat*, *hyb*, and *tan* respectively stand for pixel features GMM-HMM, hand-crafted features GMM-HMM, hybrid and tandem combination of ConvNN and HMMs.

Extraction	Name	Features (dim.)	Optical model
Grapheme seg.	***gpmpix***	PCA on pixels (150)	GMM-HMM
	gpmfeat	hand-crafted features (74)	GMM-HMM
	gpmhyb	pixels (32 x 32)	ConvNN-HMM
Sliding win.	***swpix***	PCA on pixels (30)	GMM-HMM
	swfeat	hand-crafted features (68)	GMM-HMM
	swhyb	pixels (32 x 32)	ConvNN-HMM
	swtan	PCA on ConvNN post. (50)	GMM-HMM

TABLE I. WORD RECOGNITION SYSTEMS PRESENTED IN THIS PAPER.

IV. RESULTS

A. Rimes database [17]

We work on the word recognition task proposed for the ICDAR 2009 competition. The training set is composed of 44,197 images with 4,508 unique words and the development set contains 7,542 images with 1,636 unique words. The systems are tested on the ICDAR 2009 evaluation set, made of 7,464 images, with a closed vocabulary composed of either 1,612 words (WR2 setting, 0% OOV) or 5,335 words (WR3 setting, 0% OOV).

B. Result analysis

Most parameters - such as sliding window parameters, HMM topology, prior scaling factor α, optical scaling factor (balancing the contribution of the optical model and the language model), number of Gaussians, PCA dimension - have been optimized on the validation set. The GMM-HMM are trained with the Maximum Likelihood crierion and Viterbi alignments. The ConvNN are trained with stochastic gradient descent, with a learning rate of 0.001 and mini-batches of 128 examples, to optimize the negative log-likelihood criterion.

The results for WR2 and WR3 tasks are presented in Table II. Several observations arise from these results. First, the systems using ConvNNs (*-hyb* and *-tan*) perform better than the systems based on pixels or on hand-crafted features. We record up to 60% relative improvement on WR2 from *swpix* to the tandem system. The best system is *swtan*, to which we add context-dependency (CD) and a few iterations of Maximum Mutual Information (MMI) training. Its performance on WR3 is comparable to those of systems using recurrent neural networks (RNN).

Model	Rimes-WR2	Rimes-WR3
gpmpix	39.0%	41.8%
gpmfeat	28.0%	30.6%
gpmhyb	16.8%	18.9%
swpix	19.8%	21.4%
swfeat	14.6%	16.4%
swhyb	10.0%	11.7%
swtan	8.5%	9.9%
swtan + CD + MMI	7.9%	9.2%
7 RNN + HMM [18]	-	**4.8%**
RNN [19]	**6.8%**	9.0%
Tandem LSTM-HMM [6]	-	9.7%

TABLE II. WORD ERROR RATE ON THE TEST SET FOR THE DIFFERENT SYSTEMS ON THE ICDAR-2009 EVALUATION SET FOR TWO DIFFERENT VOCABULARY SIZES (WR2 AND WR3).

The comparison between hand-crafted features and ConvNN-based methods would be fairer if we added a classifier on top of the features. It has been done in [1] for the grapheme system, where a multi-layer perceptron predicts grapheme classes. The reported results are 20.5% (resp. 27.5%) on WR2 and WR3. Since no language model seems to be used in [1], we evaluated *gpmhyb* without the unigram, and got 20.8% and 25.6%. Thus, if not better, this system is at least comparable with a hybrid model based on hand-crafted features. Note that in [1], the MLP predicts grapheme classes rather than individual HMM states, while our method is simpler because we do not have to worry about grapheme classes, and use directly the HMM states.

We also note that the grapheme systems are globally worse than the systems based on sliding windows. Even the best grapheme model (*gpmhyb*) is not as good as *swfeat*. We argue that we should however not abandon them. First, they produce less observations, and therefore less data to process. It results in a much faster recognition, as shown in Table III. We report the number of observations extracted from the different data sets, and the recognition speed on the validation set. The recognition speed is measured on a single CPU for the whole set, divided by the number of images, and averaged over ten runs. We used *swfeat* on the one hand, and *gpmfeat* on the other hand. We observe that although the number of observations is divided by ten for the grapheme system, the factor we gain in speed is even larger.

	Sliding Window	Grapheme
Num. Observations - Train	3,230,884	308,760
Num. Observations - Valid	551,959	52,606
Num. Observations - Test	539,912	52,125
Decoding time - Valid	719 ms/word	46 ms/word

TABLE III. GRAPHEME VS SLIDING WINDOW

Moreover, we notice that the errors made by the sliding window systems are not the same as those made by the grapheme systems, suggesting that both approaches are complementary. To verify this hypothesis, we tried two simple types of system combination. For each system, we extracted N-best lists ($N = 10$), and normalized the recognition scores in each list so that they lie in the interval (0,1). The first combination (*Sum*) adds up the scores of all considered systems, and returns the word with the highest score. The second combination (*Wei.Sum*) is the same, but the scores are weighted. We chose heuristic weights, which are proportional to the accuracy of the system.

The results of these combinations on Rimes validation set are presented in Table IV. We present the WER of individual systems, and the WERs made by combinations of selected systems. We observe that the best combination uses all systems but *gpmpix*. Moreover, we noticed that the best two combinations of two systems involve *gpmhyb*, and outperform the combination of the best two single systems. The best two combinations of three systems also involve *gpmhyb*.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we discussed two directions for the problem of handwritten word recognition. First, we showed that the combination of convolutional neural networks - which operate directly on the images - with HMMs give state-of-the-art performance, and the results on Rimes database are comparable with the best reported results, which include recurrent neural networks. In particular, hybrid and tandem

Models	Sum	Wei.Sum
gpmpix	41.2%	41.2%
gpmfeat	30.5%	30.5%
gpmhyb	17.7%	17.7%
swfeat	14.7%	14.7%
swhyb	10.9%	10.9%
swtan	**8.7%**	**8.7%**
swtan, swhyb	8.6%	8.5%
swhyb, gpmhyb	8.5%	8.2%
swtan, gpmhyb	7.9%	7.7%
swtan, swhyb, gpmhyb	7.3%	7.3%
swtan, swhyb, swfeat, gpmhyb	**7.1%**	7.1%
swtan, swhyb, swfeat, gpmhyb, gpmfeat	7.2%	**6.9%**

TABLE IV. COMBINATION RESULTS ON RIMES 2009 VALIDATION SET

combinations significantly outperform a standard GMM-HMM based on hand-crafted features. We also showed that a hybrid ConvNN/HMM is better than a GMM-HMM based on hand-crafted features, and comparable to a hybrid MLP/HMM (also based on these features) in the case of explicit over-segmentation into graphemes. We showed that the sliding window and grapheme approaches are complementary: the best combinations of the presented system always include a system based on graphemes.

The models based on grapheme should not be neglected. They are much faster than their sliding window counterparts, and therefore more suited to industrial applications. However, the grapheme segmentation could be improved, to avoid to issues observed on Figure 1. There is also room for improvement in system combinations. The presented systems are all wrong for only 2.7% of the words, and the correct word does not appear in any of the 10-best lists in less than 0.3% of the examples, and further work could investigate more elaborated combination techniques. In particular, keeping in mind that recognition speed is crucial in industrial applications, a "cascade" combination approach, starting with systems based on graphemes, could be interesting.

ACKNOWLEDGMENT

This work was partly achieved as part of the Quaero Program, funded by OSEO, French State agency for innovation and was supported by the French Research Agency under the contract Cognilego ANR 2010-CORD-013.

REFERENCES

[1] A.-L. Bianne, F. Menasri, R. Al-Hajj, C. Mokbel, C. Kermorvant, and L. Likforman-Sulem, "Dynamic and Contextual Information in HMM modeling for Handwriting Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 2066 – 2080, 2011.

[2] N. Hammerla, T. Plötz, S. Vajda, and G. Fink, "Towards Feature Learning for HMM-based Offline Handwriting Recognition," in *International Workshop on Frontiers in Arabic Handwriting Recognition*, 2010.

[3] P. Dreuw, P. Doetsch, C. Plahl, and H. Ney, "Hierarchical Hybrid MLP/HMM or rather MLP Features for a Discriminatively Trained Gaussian HMM: A Comparison for Offline Handwriting Recognition," in *International Conference on Image Processing*, 2011.

[4] F. Grezl, M. Karafiat, S. Kontar, and J. Cernocky, "Probabilistic and Bottle-Neck Features for LVCSR of Meetings," in *International Conference on Acoustics, Speech and Signal Processing*, vol. 4, 2007, p. 757.

[5] G. Dahl and D. Yu, "Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[6] P. Doetsch, "Optimization of hidden markov models and neural networks," Master's thesis, RWTH Aachen University, Aachen, Germany, Dec. 2011.

[7] Y. Le Cun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," *International Symposium on Circuits and Systems*, pp. 253–256, May 2010.

[8] Y. Le Cun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional Neural Network Committees For Handwritten Character Classification," in *International Conference of Document Analysis and Recognition*, vol. 10, Beijing, 2011, pp. 1135–1139.

[10] Y. Bengio, Y. Le Cun, and D. Henderson, "Globally trained handwritten word recognizer using spatial representation, convolutional neural networks and Hidden Markov Models," in *Neural Information Processing System*, 1994.

[11] Y. Le Cun, L. Bottou, and Y. Bengio, "Reading checks with multilayer graph transformer networks," in *International Conference on Acoustics, Speech, and Signal Processing*, 1997.

[12] H. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach, Chapter 7*, ser. The Kluwer international series in engineering and computer science: VLSI, computer architecture, and digital signal processing. Kluwer Academic Publishers, 1994, vol. 247, no. 4.

[13] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," *International Conference on Acoustics Speech and Signal Processing*, vol. 3, no. 28149, pp. 1635–1638, 2000.

[14] Y. Bengio, Y. Le Cun, C. Nohl, and C. Burges, "LeRec: a NN/HMM hybrid for on-line handwriting recognition." *Neural Computation*, vol. 7, no. 6, pp. 1289–1303, 1995.

[15] R. Buse, Z. Q. Liu, and T. Caelli, "A structural and relational approach to handwritten word recognition." *IEEE transactions on systems, man, and cybernetics*, vol. 27, no. 5, pp. 847–61, Jan. 1997.

[16] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, Dec. 2011.

[17] E. Augustin, M. Carré, E. Grosicki, J.-M. Brodin, E. Geoffrois, and F. Preteux, "RIMES evaluation campaign for handwritten mail processing," in *Workshop on Frontiers in Handwriting Recognition*, no. 1, 2006.

[18] F. Menasri, J. Louradour, A.-L. Bianne-bernard, and C. Kermorvant, "The A2iA French handwriting recognition system at the Rimes-ICDAR2011 competition," in *Document Recognition and Retrieval Conference*, vol. 8297, 2012.

[19] E. Grosicki and H. ElAbed, "ICDAR 2009 Handwriting Recognition Competition," in *International Conference on Document Analysis and Recognition*, 2009.