



**HAL**  
open science

# Word Spotting and Regular Expression Detection in Handwritten Documents

Yousri Kessentini, Clément Chatelain, Thierry Paquet

► **To cite this version:**

Yousri Kessentini, Clément Chatelain, Thierry Paquet. Word Spotting and Regular Expression Detection in Handwritten Documents. ICDAR, 2013, United States. pp.516-520. hal-00905535

**HAL Id: hal-00905535**

**<https://hal.science/hal-00905535>**

Submitted on 19 Nov 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Word Spotting and Regular Expression Detection in Handwritten Documents

Yousri Kessentini, Clément Chatelain, Thierry Paquet  
Université de Rouen, Laboratoire LITIS EA 4108  
Site du Madrillet, St Etienne du Rouvray, France  
{yousri.kessentini,thierry.paquet}@univ-rouen.fr

**Abstract**—In this paper, we propose a novel system for word spotting and regular expression detection in Handwritten documents. The proposed approach is lexicon-free, i.e., able to spot arbitrary keywords that are not required to be known at the training stage. Furthermore, the proposed system is segmentation-free, i.e., text lines are not required to be segmented into words. The originalities of our approach is twofold. First we propose a new filler model which allows to speed-up the decoding process. Second, we extend the methodology to search for regular expressions. The system has been evaluated on a public handwritten document database used for the 2011 ICDAR handwriting recognition competitions.

## I. INTRODUCTION

The problem of word spotting in handwritten documents has attracted a lot of attention in the community these last years. It consists in detecting any given keyword in document images. This task is important in numerous applications, such as simply querying textual handwritten documents, but also automatic categorization, indexing, information retrieval in handwritten document databases.

Word spotting approaches proposed in the literature fall into the two following categories. Image based methods, also known as "query-by-example", operate through the image representation of the keywords [1], [2], [3], [4], [5]. The recognition based, or "query-by-string" methods, operate with the ascii representation of the keywords [6], [7], [8], [9], [10], [11]. In the first kind of approaches, the input image is represented as a sequence of features and is matched to a set of template keyword images. The performance of this kind of approaches is limited when dealing with a wide variety of unknown writers. On the contrary, recognition based approaches are not limited to a single writer, at the expense of a more complex matching process, derived from conventional handwriting recognition systems. In this context, many works has focused on several variants of Hidden Markov Models (HMMs) to address this intrinsically sequential problem [11], [9], [10], [7]. We categorize the HMM approaches into two main categories: word based and line based. In word based spotting such as [7], the HMM model for each keyword is trained separately so that the lines of text need to be segmented into words. The drawbacks of this strategy is that word segmentation errors are often irreversible, and affect considerably the recognition step. Moreover, as they use a global approach, the system is not able to spot words that are not present in the training set. Using line based spotting approaches circumvents the segmentation problem [11], [9], [10]. In [9], authors present an alpha-numerical information

extraction system in handwritten unconstrained documents. It relies on a global line modeling allowing a dual representation of the relevant and the irrelevant information. The acceptance or rejection is controlled by the variation of an hyper-parameter in the HMM line model. A similar approach is presented in [10], the line model is made of a left and right filler models surrounding the keyword model. The acceptance or rejection is controlled by a text line score based on the likelihood ratio between a keyword text line model and a filler text line model. In [11], authors propose a different rejection method that are not based on filler models. It is based on score normalization between the keyword candidate and non-keywords scores. A reduced lexicon is used to overcome the high computational complexity which results from using all non-keywords. They show that the proposed method outperforms the line based approach presented in [10].

In this work, we propose a line based spotting approach using a new filler model which allow to speed-up the decoding process. We study the extension of the word spotting system to the search for regular expressions. To the best of our knowledge, this problematic has remained unexplored in the literature.

This paper is organized as follows. In Section 2, the proposed keyword spotting model is introduced in detail. Its spotting extension to regular expressions is given in section 3. Section 4 describe the whole spotting system. The experimental evaluation is described in section 5. Conclusion and future works are drawn in the last section.

## II. LINE BASED SPOTTING MODEL

To avoid line segmentation into word, the basic idea is to create one model for an entire line. It should be made of one keyword model and a separate model of the filler (i.e., non-keyword) regions. These two models are joined to form a composite keyword-filler line model that is used to perform a recognition. Figure 1 shows the proposed line model made of a left and right filler models surrounding the keyword model. The keyword model is constrained to contain the exact keyword letter sequence at the beginning, in the middle, or at the end of the text line. To implement this model, we have chosen to use the Hidden Markov Models. As a matter of fact, an HMM based framework offers several advantages due to automatic training of character models on non-segmented lines (embedded training), and the segmentation-free recognition paradigm that fits particularly well to a spotting approach.

The filler model is typically an ergodic HMM composed of all character models, which leads to a high computation



Fig. 1. Global line model containing the keyword, and the filler model. The line model is also made of a space model ("SP"), and structural initial and final models.

complexity at the decoding level. Indeed, the complexity of the Viterbi algorithm recognizing a line of length  $T$ , is measured by  $\mathcal{O}(TN^2)$ , where  $N$  is the number of states in the line model. Assuming that  $N = 2 \times N_f + N_K + 2 \times N_{sp}$ , where  $N_f$  is the number of states in the filler,  $N_K$  is the number of states in the keyword model and  $N_{sp}$  is the number of states in the space model, we notice that the number of states in the filler model can affect considerably the computation complexity. In practice, for Latin script, we use about 70 models corresponding to lower and upper letters, digits and some punctuation marks. Given that each model is composed of several states (usually between 3 and 10), it leads to at least several hundred of states. In [11], authors present a clustering algorithm to reduce the number of pre-trained character model in the filler. Such an approach strongly depends on the test dataset and on the list of keywords. In this work we propose a novel filler model consisting of an ergodic HMM composed of 4 models as shown in Figure 2: an uppercase character model, a lowercase character model, a digit model and a space model. Hence the number of states in the filler is reduced significantly  $N_f = 16$ . Unlike [11], these models do not model specific letters and are trained differently than the keyword character models, i.e MU (respectively ML) model is trained in all uppercase (respectively lowercase) character utterances of the training dataset, and MD model is trained in all digit utterances. This can be justified by the fact that the filler model is used to model the non-keywords without explicitly defining them. The performance of both fillers is given in section 5.

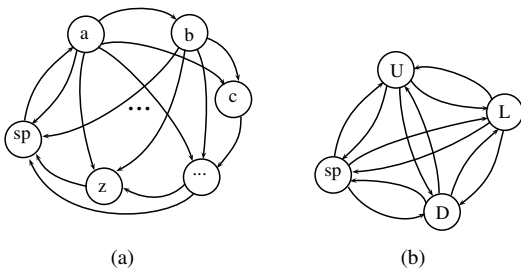


Fig. 2. Filler models (a) character filler HMM (b) proposed filler HMM: (U) uppercase character model, (L) lowercase character model, (D) digit model, (sp) space model

Decoding a text line image represented by an observed feature vector sequence  $X = x_1, \dots, x_N$  using the global line model  $\omega$  is done using the Viterbi algorithm which outputs as a result the most likely letter sequence and the likelihood  $P(X|\omega)$ . Nevertheless, it is well known in a verification tasks like word spotting, that the likelihood is an insufficient measure. Instead, the posterior probability is considered for more confident measure, this is known under the name "score normalization" in the literature [12]. Applying Bayes' rule, we

obtain:

$$P(\omega|X) = \frac{P(X|\omega) \times P(\omega)}{P(X)}$$

Assuming equal priors  $P(\omega)$ , we only take the terms  $\frac{P(X|\omega)}{P(X)}$ . In this work, we present two different methods to evaluate  $P(X)$ : a Filler based strategy and a Vocabulary based strategy.

#### A. Filler based strategy

Generally, in word spotting problems [13], [7],  $P(X)$  can be calculated by decoding the feature vector sequence  $X = x_1, \dots, x_N$  using the filler model  $F$ . It is demonstrated in [10] that this normalized line score corresponds to the normalized keyword score, because the likelihood difference between the global line model and the filler model is zero outside the keyword position.

$$\frac{P(X|\omega)}{P(X)} = \frac{P(X_{s,e}|K)}{P(X_{s,e})}$$

The final text line score is obtained by normalizing the likelihood ratio with the width of the keyword. The detected keyword is accepted if the normalized likelihood score is greater than a certain threshold  $T$ .

#### B. Vocabulary based strategy

In this technique, similarly to [11],  $P(X)$  is evaluated using a set of non-keywords. We proceed in two stages: first, the most likely start  $s$  and end  $e$  positions of the keyword are obtained using the Viterbi decoding of the image line feature vector sequence  $X = x_1, \dots, x_N$  with the global line HMM. The second level consists in decoding the detected keyword portion sequence  $x_s, \dots, x_e$  using a lexicon composed of all non-keywords to evaluate  $P(X_{s,e})$ . The non-keywords lexicon can be reduced using the Levenshtein distance to speed-up the decoding.  $P(X_{s,e}|K)$  is also evaluated by a Viterbi decoding of the observation sequence  $x_s, \dots, x_e$  with the keyword HMM. The final score is obtained by normalizing the likelihood ratio  $\frac{P(X_{s,e}|K)}{P(X_{s,e})}$  with the width of the keyword.

### III. EXTENSION TO REGULAR EXPRESSION

Regular expressions represent a way to identify patterns in a text. They can be used to identify a piece of a text for special handling. The search for regular expression in handwritten documents is a very difficult task due to either the variability of the request, or its lack of constraints. Indeed, if we consider the task of identifying instances of some entity, the set of corresponding matches can have variable length. For example, considering the regular expression corresponding to *any words beginning by "con"*, three words "cons, contrat, consideration" are considered as a positive match, made of respectively 4, 7 and 13 letters. When considering the regular expression "five letter words", the set of positives matches is composed of many different words. These difficulties, combined with the high variability of handwriting explain the lack of studies in the handwriting literature dealing with the regular expressions detections.

In order to cope with the variability of string length, we use a HMM to model a sequence of lowercase letters with a

variable length. Note that this HMM is easily derived from the lowercase character model (L), by adding a self transition. Figure 3 shows the global line HMM corresponding to the request *any word beginning by "con"*.

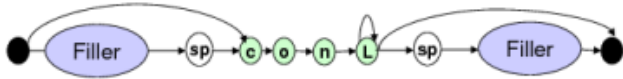


Fig. 3. Global line model corresponding to the query all words beginning by "con"

Similarly, Figure 4 shows the the global line HMM corresponding to the request *any word ending by "ant"*.

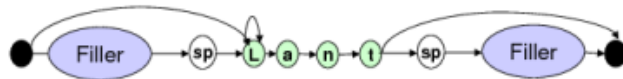


Fig. 4. Global line model corresponding to the query all words ending by "ant"

#### IV. SYSTEM OVERVIEW

We now describe the whole spotting system. Its input is a document image, in which the set of keywords is spotted, while the output is the position hypotheses of each keyword in the document. In the following sections we will provide the details of the processing steps.

##### A. Line segmentation

To evaluate our system, we use the RIMES database used for the 2011 ICDAR handwriting recognition competition [14]. In this database, each document is segmented into lines and the coordinates of the corresponding line polygons are given. As the line segmentation quality is relatively low, we propose an algorithm based on connected-components-analysis in order to remove peripheral noise belonging to other lines. Figure 5 shows the result of this algorithm on a text line image.

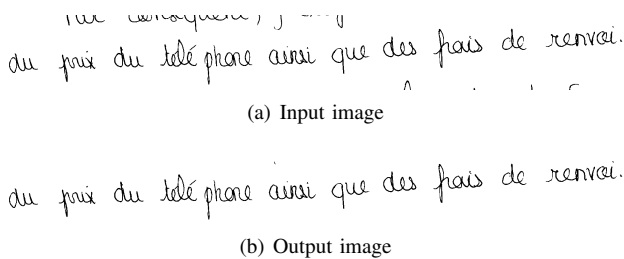


Fig. 5. Line segmentation cleaning

##### B. Preprocessing

Preprocessing is applied to line images in order to eliminate noise and to ease the feature extraction procedure. In an ideal handwriting model, the words are supposed to be written horizontally and with ascenders and descenders aligned along the vertical direction. In real data, such conditions are rarely

respected. We use skew (See Figure 6) and slant (See Figure 7) correction so as to normalize the text line image [15]. A contour smoothing is then applied to eliminate small blobs on the contour.

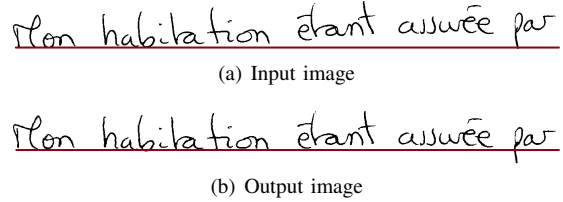


Fig. 6. Deskew correction

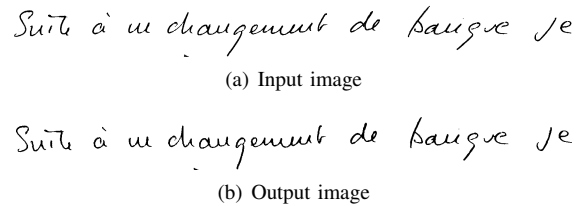


Fig. 7. Deslant correction

##### C. Feature extraction

Each text line image is divided into vertical overlapping windows or frames. The feature set has shown its efficiency in the 2009 ICDAR word recognition competition [16], [17]. Two types of features are considered: (i) contour based features and (ii) density based features. Contour based features are extracted from the upper contour, it is made of of 15 features:

- 8 directional density features.
- 4 structural features providing additional information about structure of the contour like the loops, the turning points, the simple lines, and the end points.
- 3 features indicate the position of the upper contour points in the window.

Density feature set is based on density and concavity features. It is made of 26 features:

- 9 are baseline independent (for instance: black pixels density in the window and in every column of this window, position of the writing pixels)
- 17 are baseline dependent (for instance: black pixel density upon and under baselines, local pixels configurations regarding baselines, ...)

The complete feature set is made of 41 features. More detailed description of features is given in [16].

##### D. Models training

We have considered  $N_c = 71$  characters: 26 lower case letters, 26 capital letters, 10 digits, a space model, accented letters (é, è, ê, à) and punctuations models (.,', -, /). All these models have been trained using an embedded training using labeled text line images. The uppercase (U), lowercase (L), and digit (D) models are trained similarly on the same training

database using respectively all uppercase letters, lowercase letters and digits. The ground truth transcription of each text line is converted into its corresponding upper/lower case digit sequence. The hyperparameters of the HMM have been experimentally determined on a validation dataset. We have  $N_s = 4$  states per character, the width of the frame window has been set to  $N_p = 8$  pixels with an overlapping factor of  $N_r = 6$  pixels.

## V. EXPERIMENTS AND RESULTS

To evaluate the performance of our system, experiments have been conducted on the RIMES database used for the 2011 ICDAR handwriting recognition competitions [14]. The training database is composed of 1.500 documents, the validation and test sets are composed respectively of 100 documents. In order to evaluate the spotting system, we compute recall (R) and precision measures (P). To do this, the number of true positives (TP), false positives (FP), and false negatives (FN) are evaluated for all possible threshold values. From these values, a recall-precision curve is presented by cumulating these values over all keyword queries.

$$R = \frac{TP}{TP + FN} \quad P = \frac{TP}{TP + FP}$$

### A. Word spotting results

We evaluate the proposed word spotting system with the two different fillers, and two score normalization methods, as described in section 2. The vocabulary normalization strategy is evaluated with the complete non-keyword lexicon (1200 words), and with a reduced lexicon of 100 words using Levenshtein distance. Figure 8 shows the results for 25 keywords. The recall-precision curves are given using a global threshold, i.e the threshold value is independent of the keyword. The obtained result show that the normalization strategy based on filler model outperforms the reduced vocabulary based method. The vocabulary normalization strategy gives better result when considering the complete non-keyword lexicon, but with high computational complexity. As shown in Figure 8, the best performance is obtained using the traditional Character filler for decoding and normalization. The proposed filler model appears to be an effective method to speed up the decoding process with a slight effect on the system performance. In practice, this filler model speed up considerably the decoding step, it is 5 times faster than the decoding with traditional filler.

Additional experiments have been conducted using 50 and 100 keywords to investigate the effect of different number of keywords on the system performance. For this experiment, we test only the configuration that have given the best performance. Figure 9 shows the obtained results using the character filler and the filler based normalization. The system performance is affected by the size of the keyword list because the precision of the system decreases as the size of the list increases.

### B. Regular expression results

To evaluate our system for the detection of regular expressions, 15 different queries have been used corresponding to 2 different requests : "finding any word beginning by S"

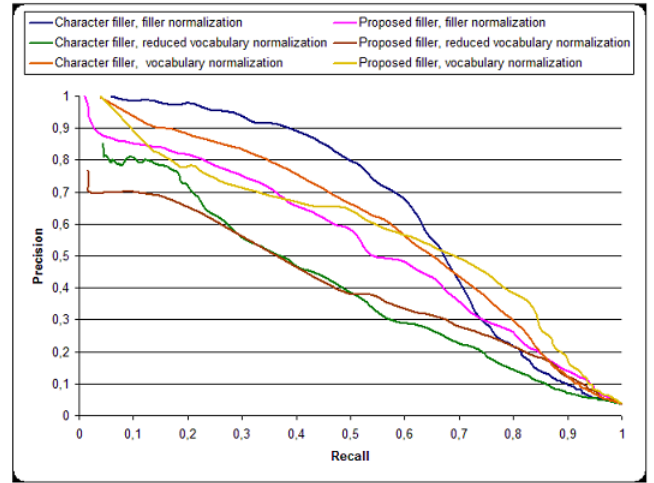


Fig. 8. Word spotting performance for 25 keywords

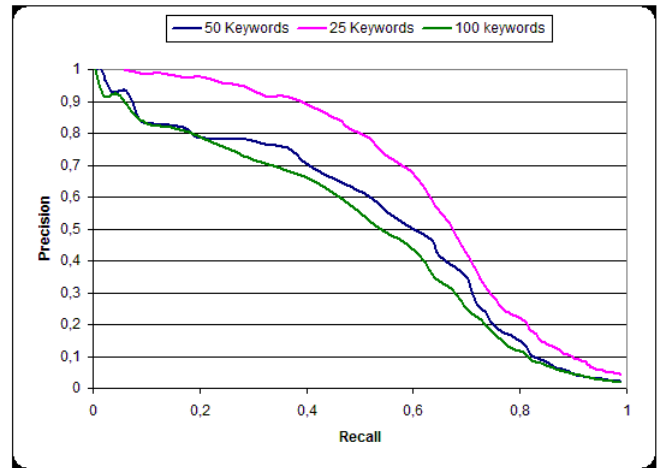


Fig. 9. Word spotting performance with varying numbers of keywords

and "finding any word finishing by S", where S is the query substring. We have selected different substring length which varies from 2 to 5 letters. We evaluate the system with the two different fillers and the filler normalization strategy as shown in Figure 10. The obtained results confirm the difficulty of the task, compared to the word spotting problematic. The obtained break-even-point<sup>1</sup> is equal to 48% which is clearly lower to the word spotting break-even-point for 25 keywords which is equal to 63% as shown in Figure 9. We present in Figure 11 the obtained results of 4 different queries corresponding to the substrings "effe", "pa", "com" and "cha". We notice that the obtained results strongly depend of the substring query. For example, the worst results are for the substring "cha", which can be explain by the presence of many words beginning by "cha" in the vocabulary like (chainement, chang, changement, changer, chaque, charbon, chaussettes).

<sup>1</sup>Given a precision-recall curve, the Precision/Recall Break-Even Point (PRBEP) is the value at which the precision is equal to the recall.

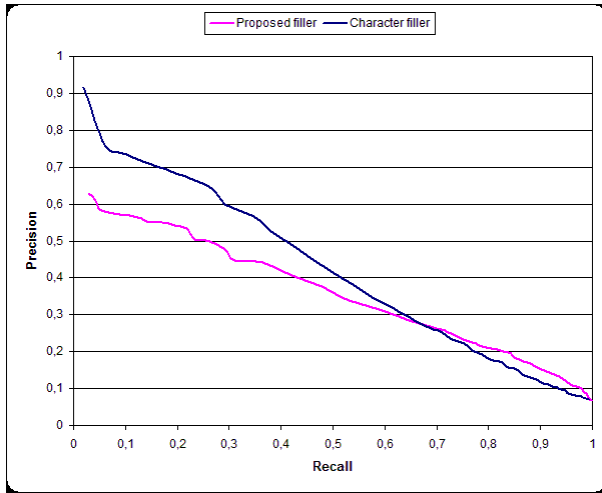


Fig. 10. Recall-precision results for Regular expression detection

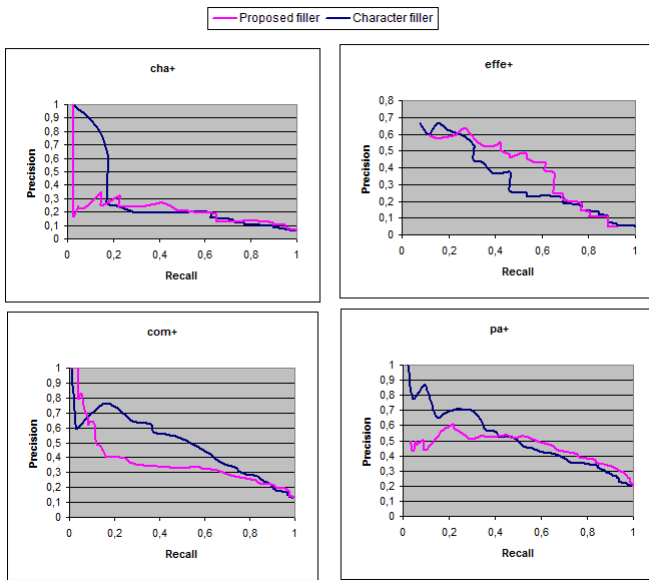


Fig. 11. Regular expression result examples

## VI. CONCLUSION

In this paper, we have presented an original spotting system in unconstrained handwritten documents. It relies on a global line modeling based on HMMs, without the need for word or character segmentation. The originality of the method consists in proposing a new filler model which allows to speed-up the decoding process. In addition, we propose an extension to the search for regular expression. To the best of our knowledge, this problematic has remained unexplored in the literature. Future works will include the improvement of the regular expression detection system by enriching our models to detect digits and lower/upper letters. It is envisaged to proceed in two stages: a first system provides concise and flexible means to localize patterns, and a second level to recognize the detected utterance with more sophisticated classifiers.

## REFERENCES

- [1] T. Rath and R. Manmatha, "Features for word spotting in historical manuscripts," *ICDAR*, pp. 218–222, 2003.
- [2] H. Cao and V. Govindaraju, "Template-free word spotting in low-quality manuscripts," *ICDAR*, pp. 392–396, February 2007.
- [3] T. Adamek, N. E. Connor, and A. F. Smeaton, "Word matching using single closed contours for indexing handwritten historical documents," *IJDAR*, vol. 9, no. 2, pp. 153–165, 2007.
- [4] M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós, "Browsing heterogeneous document collections by a segmentation-free word spotting method," *ICDAR*, pp. 63–67, 2011.
- [5] J. A. Rodríguez-Serrano, F. Perronnin, and J. Lladós, "A similarity measure between vector sequences with application to handwritten word image retrieval," *CVPR*, pp. 1722–1729, August 2009.
- [6] C. Choisy, "Dynamic handwritten keyword spotting based on the nshp-hmm," *Proceedings of the Ninth ICDAR*, vol. 1, pp. 242–246, 2007.
- [7] J. A. Rodríguez-Serrano and F. Perronnin, "Handwritten word-spotting using hidden markov models and universal vocabularies," *Pattern Recognition*, pp. 2106–2116, February 2009.
- [8] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A novel word spotting method based on recurrent neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 211–224, 2012.
- [9] S. Thomas, C. Chatelain, L. Heutte, and T. Paquet, "An information extraction model for unconstrained handwritten documents," *ICPR, Istanbul, Turkey*, pp. 1–4, 2010.
- [10] A. Fischer, A. Keller, V. Frinken, and H. Bunke, "Lexicon-free handwritten word spotting using character hmms," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 934–942, 2012.
- [11] S. Weshah, G. Kumar, and V. Govindaraju, "Script independent word spotting in offline handwritten documents based on hidden markov models," *In proceeding of: The 13th International Conference on Frontiers in Handwriting Recognition, (ICFHR 2012)*, 2012.
- [12] A. Brakensiek, J. Rottland, and G. Rigoll, "Confidence measures for an address reading system," *In 7th Int. Conf. on Document Analysis and Recognition*, 2003, pp. 294–298.
- [13] R. Rose and D. Paul, "A hidden markov model based keyword recognition system," in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, apr 1990, pp. 129–132 vol.1.
- [14] E. Grosicki and H. El-Abed, "Icdar 2011 - french handwriting recognition competition," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, sept. 2011, pp. 1459–1463.
- [15] F. Kimura, S. Tsuruoka, Y. Miyake, and M. Shridhar, "A lexicon directed algorithm for recognition of unconstrained handwritten words," *IEICE Trans. on Information & Syst.*, vol. E77-D, no. 7, pp. 785–793, 1994.
- [16] Y. Kessentini, T. Paquet, and A. Benhamadou, "Off-line handwritten word recognition using multi-stream hidden markov models," *Pattern Recognition Letters*, vol. 31, pp. 60–70, 2010.
- [17] Y. Kessentini, T. Paquet, and A.-M. Benhamadou, "Multi-script handwriting recognition with n-streams low level features," in *19th International Conference on Pattern Recognition ICPR 2008.*, dec. 2008, pp. 1–4.