Handwritten/printed text separation Using pseudo-lines for contextual re-labeling

Ahmad-Montaser Awal, Abdel Belaïd, Vincent Poulain d'Andecy

Handwritten/printed text separation Using pseudo-lines for contextual re-labeling

Ahmad Montaser Awal, Abdel Belaïd
Université de Lorraine- LORIA
54506 Vandoeuvre-lès-Nancy, France
{ahmad-montaser.awal, abdel.belaid}@loria.fr

Vincent Poulain d'Andecy
ITESOFT Parc d'Andron, Le Séquoia
30470, Aimargues, France
vincent.poulaindandecy@itesoft.com

Abstract—**This paper addresses the problem of machine printed and handwritten text separation in real noisy documents. We have proposed in a previous work a robust separation system relying on a proximity string segmentation algorithm. The extracted pseudo-lines and pseudo-words are used as basic blocks for classification. A multi-class support vector machine (SVM) with Gaussian kernel associates first an appropriate label to each pseudo-word. Then, the local neighborhood of each pseudo-word is studied in order to propagate the context and correct the classification errors. In this work, we first propose to model the separation problem by conditional random fields considering the horizontal neighborhood. As the considered neighborhood is too local to solve certain error cases, we have enhanced this method by using a more global context based on class dominance in the pseudo-line. The method has been evaluated on business documents. It separates handwritten and printed text with better scores (99.1% and 99.2% respectively), contrary to noise which is very random in these documents (90.1%).**

Keywords—document segmentation; pseudo-line and pseudo-word extraction; printed/handwritten/noise separation; contextual analysis; patch classification

I. INTRODUCTION

Documents containing mixed types of text (printed and handwritten) are increasingly present in business and academic environments. They result frequently from annotating printed documents such as bills, administrative forms, birth-certificates, letters, etc. Example of such documents is given in Fig. 1. A major trend of research in this area is to first segment the document into individual basic units, classify them into script categories, and then extend this classification by a neighborhood technique to recover from classification errors. The literature shows a very active research on this problem. A state of the art highly developed is given in [14]. We will retrace the main processing steps: segmentation, classification, and contextual re-labeling.

A. Segmentation

The segmentation step aims at creating stable and regular regions (basic units) that would be labeled into printed, handwritten or noise. The basic units mostly used in the literature are: text lines, words, or characters.

Text lines are the basic units in [2][3]. The document is segmented into text lines using the horizontal projection profiles. This was made possible thanks to the homogeneity of the lines, which is not the case in most of real documents.

In the case of **word level**, connected components (CCs) are grouped in order to approximate words. Obtained blocks are referred to as word blocks, patches, or pseudo-words. In the following, we use the pseudo-word term. A basic approach in [4] uses geometric proximity and size to group CCs into pseudo-words. In [5], CCs are merged based on height regularity and distance. In [6], CCs belonging to the same text line with a distance less than half of the average width of all CCs or with overlapping pixels are merged to form pseudo-words. A morphological closing by a 5x5 structuring element is used in [7]. In [8], an adaptive Run Length Smoothing Algorithm (ARLSA) is first used to find text lines. Afterwards, vertical projection profile of each text line is used to extract pseudo-words. As in the above work, the authors in [9][10] propose an adapted distance thresholds by calculating optimal word size dynamically for each document. Then, they apply a pixel growing algorithm to group CCs into pseudo-words with respect to the approximated word size.

At the **character level**, Kandan et al. [12] work at the CC level since they are easier to extract. Fan et al. [11] work at the character level as they principally work on Chinese documents where characters are easier to extract. They employ an X-Y cut algorithm to extract text lines at first, and then another X-Y cut is applied to extract characters.

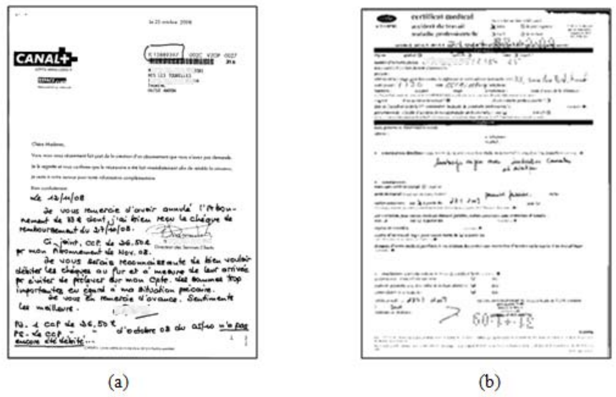

Fig. 1. Example of processed documents: (a) represents a printed letter completed by handwritten textual blocks, (b) is a printed form manually filled out.

2167-6445/14 $31.00 © 2014 IEEE
DOI 10.1109/ICFHR.2014.13

B. Classification

At the text line level, a three-tier tree classifier is employed in [2] to classify text lines into handwritten/printed using three level structural features specific to the Bangala and Devnagari scripts. This classifier allows taking decisions on each features level which avoids calculating all the features systematically. In [3], discriminant analysis is employed on features extracted from the upper/lower profiles of the text lines.

At the pseudo-word level, a HMM is used in [5] to classify pseudo-words based on their projection profile. Da silva et al. [6] extract 11 features for each pseudo-word (structural features, projection profiles, pixel distribution). A rule based classification is then employed to associate the pseudo-word to the handwritten or printed class. Another set of features is extracted in [4]: Gabor filter, crossing count histogram and bi-level co-occurrence. The fisher classifier is used to achieve the classification task. Peng et al. [7] extract a set of features at both pseudo-word and CC levels in addition to Gabor features. A G-means clustering algorithm is used where the number of clusters is estimated from the training dataset. The classification task could be modeled directly by conditional random fields as in [9], using a set of 23 features. In [12], an SVM is employed to classify the extracted CCs using Hu moments invariant features. Characters in [11] are classified by a rule based classifier using CC block spatial features.

In conclusion concerning the best level to choose for script separation, text lines are too global and may contain two types of scripts. Characters and CCs are too small and might be ambiguous. For example, the character '0' written in printed or handwritten can be ambiguous because both shapes are very close. Pseudo-words are shown to be more stable than text lines, characters and CCs; that is why we have chosen to use pseudo-words as basic units in our system.

C. Contextual re-labeling

Labeling errors are inevitable and a post-processing step is often needed to correct such errors. The use of contextual information is introduced to remedy this problem. In this case, the contextual neighborhood of a basic unit participates in the decision on its label. Contextual re-labeling is only justified for CCs, characters or pseudo-words. In the case of text lines, the same label is already given to all the elements of the line.

1) Neighborhood

In [12], the neighborhood of a CC is defined by Delaunay triangulation. All CCs connected to a given CC by the triangles are considered as its contextual neighborhood.

According to Zheng et al. [4] printed, handwritten and noise pseudo-words show different patterns of geometric neighborhood relationship. A pseudo-word, classified as "printed", is associated to two horizontal neighbors. A "noise" pseudo-word exhibits a more random pattern and thus modeled taking its four nearest neighbors. Handwritten pseudo-words are not modeled in order to favor the spread of the printed text label. In [7], a convex hull distance is used to define the nearest 4 pseudo-words regardless their direction. The advantage of the proposed distance measure is that it gives an exponential distribution of neighbors and it measures their spatial similarity. In [9], the neighborhood is defined as follows: four directions are defined for each pseudo-word relative to its gravitational centre with a total of six neighbors (one on the top, one on the bottom, two on the left and two on the right). This gives more importance to the horizontal information.

2) Re-labeling

Once the neighborhood defined, a re-labeling method is needed to update the pseudo-word label in function of its neighbor labels and features. This can be achieved by a simple voting system. In [12], the label of a CC_i is changed based on a majority voting of the neighbors $n_i(CC)$. A similarity condition is added in order to guarantee the homogeneity of the ensemble. The re-labeling can be modeled by Markov Random Fields (MRF). Two functions are involved in this modeling. First, a dependence function measures the confidence of the classification. The other, a similarity function, measures the similarity between the pseudo-word and its neighborhood. In practice, the inference is realized by Gibbs quantification [4] (performance is improved from 96.1 % to 98.1%) or a belief propagation method [7] (performance is improved from 94.2% to 95.5%). The main drawback of MRF models is their complexity; indeed, an enumeration of all labeling configurations is needed to find the optimal one. It has been shown in [9] that discriminative models are more efficient as the optimal labeling configuration can be directly modeled by a Conditional Random Field (CRF). An approximate comparison showed that using CRF model improves system's performance from 98.0% to 99.1%.

We were inspired by the system of Kandan et al. [12], except that our method is based on pseudo-word level instead of CC one. In our re-labeling technique, we also use a specific contextual neighborhood definition. In the current work, we first implement a CRF model using discriminant classifiers and local neighborhood. We then propose a more global neighborhood definition based on pseudo-lines.

The rest of the paper is organized as follows. In section II, we present a brief description of our system proposed in [1]. Our novel methods for contextual re-labeling are presented in section III. We finally present our experimentations held on a set of real documents in section IV. The performance of our system is also compared with the literature systems. In section V, we conclude and give some perspectives.

II. Previously proposed system

A. Segmentation

Our system is based on a two level segmentation into pseudo-lines and pseudo-words already operated in [1]. Pseudo-lines extraction is achieved by constructing strings of close CCs. All CCs respecting given horizontal and vertical thresholds are added to the same string, as shown in Fig. 2.a. At the end, each string represents a pseudo-line. Similarly, the same operation is repeated on all the extracted pseudo-lines in order to locate the pseudo-words. The criterion of linking two CCs into the same string is to have a distance less than a threshold '*ws*' calculated dynamically for each pseudo-line, as shown in Fig. 2.b. At the end, each string represents a pseudo-word.

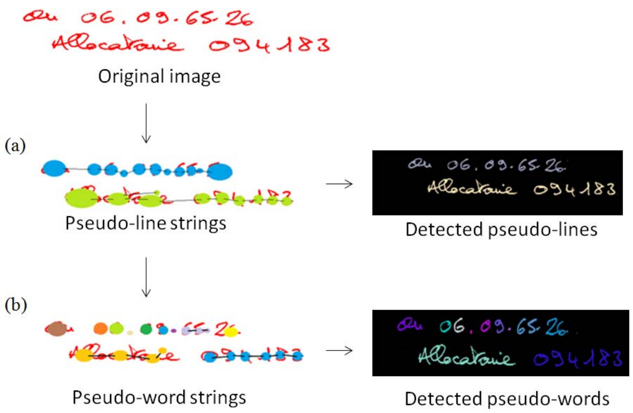

Fig. 2. Pseudo-lines and pseudo-words extraction

B. Classification

A vector composed of 137 features is extracted for each pseudo-word. The used features are inspired from the state of the art [4][6][12]. A multi-class SVM [16] is used to associate each pseudo-word to printed, handwritten or noise classes. To handle classification errors, a regrouping method based on spatial proximity is used to reassemble the pseudo-words.

C. Contextual re-labeling

The contextual re-labeling is operated by three different grouping techniques: *k*-NN, *k*-NN with constraints and confidence propagation.

Grouping by k-NN – It is based on the gathering of the *k* nearest neighbors. If more than 50% of neighbors share the same label, this label is assigned to the central component. The *k* nearest neighbors are taken into account if they are closer than a pre-defined threshold *max_dist* (given manually).

Grouping by k-NN with constraints – An improvement of this algorithm is proposed to avoid small components to interfere with the label updating. Thus, a test is performed before flipping the pseudo-word label to check whether the accumulated number of pixels of the neighbors is significant compared to the number of pixels of the main component.

Grouping by confidence propagation – To avoid random updates, the classifier confidence of the nearest horizontal neighbor is used. The idea is to examine the confidence of the nearest horizontal neighbor of a selected pseudo-word. If the latter is stronger than that of the pseudo-word, the neighborhood class is assigned. A Gaussian function weighs the neighbor confidence by its distance to the pseudo-word. Thus, the nearest the neighbor is, the more impact it has.

III. Novel contextual re-labeling methods

A. Conditional random fields (CRF)

The separation problem can be modeled as the search of best configuration of label field *X* given the observations *Y*. Let *W*={*w*} be the set of all pseudo-words, the CRF model can be written as:

$$P(X=x|Y=y)=\frac{1}{Z}\prod_{w\in W} e^{\sum_k \lambda_k f_k(x,y,w)} \qquad (1)$$

The model is thus defined by the product of exponential linear combination of *k* functions called 'feature functions', where *Z* is a normalization factor. According to [15], these functions can be modeled by discriminant classifiers such as MLP or SVM. Hence, the probability of a pseudo-word *w* is given by:

$$P(X_w|Y_L,Y_C)=\lambda_L f_L+\lambda_C f_C \qquad (2)$$

where: X_w is the label field of the pseudo-word, Y_L represents the local features, Y_C are the contextual features extracted from the pseudo-word neighbors, λ_L and λ_C are weighting parameters, f_L is the local classifier (SVM) and f_C is the contextual classifier (MLP). The combination is done as shown in Fig. 3.

The contextual neighborhood of a pseudo-word is given by its left/right horizontal neighbors based on the *max_dist* threshold. The contextual classifier is trained with the following features: 1) local classification probabilities of the neighbors (the output of the local classifier), 2) structural features extracted from the pseudo-word neighborhood (height ratio, position ratio and density ratio).

The main drawback of the previous algorithms is the use of a prefixed distance threshold *max_dist* which is image resolution dependent. Furthermore, their complexity is of $O(n^2)$ (where *n* is the number of pseudo-words) since we must calculate the distance between each pseudo-word and all the others in the document in order to find close ones.

B. Grouping by pseudo-lines

We propose the use of pseudo-lines in the contextual re-labeling of pseudo-words. A pseudo-line represents a logical relationship between its pseudo-words in addition to the horizontal spatial relationship. Furthermore, a statistical study on the training dataset showed that around 93% of pseudo-lines contain a unique label. Thus, the contextual re-labeling is redefined as follows.

A pseudo-line *l* is composed of *n* pseudo-words belonging to 3 classes (C_1: printed, C_2: handwritten and C_3: noise). We define the dominant class C_D in a pseudo-line as the class with the highest cardinality where *D*={*1,2,3*}. In case of equality of cardinalities, the dominant class is the one with highest average confidence of its pseudo-words. We explore the use of the dominant class by probabilistic and deterministic models.

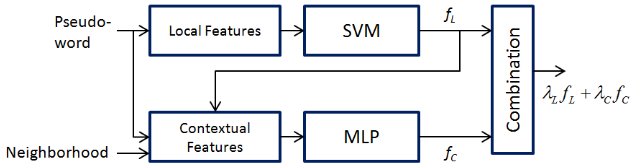

Fig. 3. Classifiers combination in CRF model

1) Probabilistic model

A CRF model is used to estimate the classification confidences of each pseudo-word in a pseudo-line. As mentioned in the section III.A, a contextual classifier is combined with the local one. At the pseudo-line level, the contextual features are calculated with respect to the dominant class. The extracted features are: 1) class cardinalities in the line, 2) structural features measuring the homogeneity of each pseudo-word with the dominant class (height ratio, density ratio, CC count ratio and inter-CC distance variance ratio).

2) Deterministic model

Let f_i be the classification confidence of the current pseudo-word, and h_i its height. The dominant class label is associated to a pseudo-word if it verifies the following condition:

$$(f_i < cf)\ \textbf{\textit{or}}\ (|h_i - H_D| < d) \tag{3}$$

where H_D is the median height of the pseudo-words of the dominant class (the median is used to avoid the effect of small pseudo-words), $0 \leq cf \leq 1$ is the certainty threshold and d determines the regularity degree (corresponding to the pseudo-word height relative to the pseudo-line.). Thus, the pseudo-word label will be changed into the dominant class label in one of the two following cases: 1) the classification confidence is low (1st term of the condition) or 2) the pseudo-word has a similar height as the dominant class in which case the classifier decision is ignored (2nd term). This latter case is inspired from printed text lines where most of the words have a height similar to the height of the pseudo-line reflecting the regularity of the text line. This hypothesis is less present for handwritten text.

We can notice, from the Algorithm I, that the new method is less complex than the previously proposed one, for two reasons. First, the complexity is linear ($O(n)$). Second, differently from the *max_dist* threshold, the used thresholds cf and d are image resolution independent. In fact, they reflect the re-labeling freedom degree. In the case of the certainty factor $cf=0$, a total confidence is given to the classifier and its decision is always considered true. On the other hand, $cf=1$ indicates that the classifier decision is not considered. For the experimentations performed in this paper, we have set $cf=0.9$. Similarly, a very small value of the regularity factor d indicates that a high regularity is required to re-label a pseudo-word. Thus, only pseudo-words with the exact same height as the dominant class are changed. In contrast, a higher value of d gives more freedom in re-labeling pseudo-words even with heights highly different from the height of the dominant class. The value of d has been set experimentally (d=*10 pixels*).

Four examples of pseudo-line based deterministic model re-labeling are illustrated in Fig. 4. For each pseudo-line, the upper line gives the classification result by SVM where the classification confidence is mentioned above each pseudo-word. In the first two cases (a) and (b), handwritten is the dominant class. In the first pseudo-line (a), the dominant class is successfully associated to the misclassified pseudo-words since their confidence is lower than the certainty factor. In (b), the printed text "*numéro d'immatriculation*" was not affected by the dominant label, since it was recognized with high confidence and in addition its height is less than the dominant height. The height of the misclassified pseudo-word '8' is similar to the dominant height; it is thus associated to the handwritten class. In (c) printed text is the dominant class. However, no updates are applied since the handwritten pseudo-words were classified with a high confidence. Finally, the pseudo-line in the fourth example (d) consists of two pseudo-words with an equality of dominant classes. The word '*singles*' is classified with the higher confidence and thus the printed label is attributed to the dominant class which corrects the label of '*6*' since it has the same height as the dominant class.

C. Segmentation improvements

Main errors are due to the sensibility of the segmentation method to some annotations (or signatures), as shown in Fig. 5.a. This causes the fusion of CCs of many lines into one pseudo-line. In this case, the hypothesis of the dominant class is not appropriate due to the presence of many lines (different script types) in the same pseudo-line.

Initially, CCs are grouped into pseudo-lines only based on a distance threshold. We propose to improve the segmentation step to assure the homogeneity of the grouped CCs. Thus, we prohibit linking two CCs if: 1) they do not have a sufficient horizontal overlapping (less than 30% of the maximum height of both CCs), 2) the area of one of them is big compared to the other CCs. An example of improved segmentation is given in Fig. 5.b. The proposed improvement correctly avoids grouping pseudo-words belonging to different lines. However, the case of pixels overlapping (printed and handwritten share the same CC) is not treated. Finally, a second module is in charge of associating small pseudo-words corresponding to diacritics to the closest non-diacritic pseudo-word.

Algorithm I – Grouping with pseudo-lines

```
Require: $\forall l \in L$   //L denotes the lines of the whole document
for all $l \in L$ do
    $C_D \leftarrow$ find_the_dominantClass(l)
    $H_D \leftarrow$ median_height($C_D$)
    for all $w \in l$ do
        if ($f_i < cf$) or ($|h_i - H_D| < d$) then
            new_label[w] $\leftarrow C_D$
        end if
    end for
end for
```

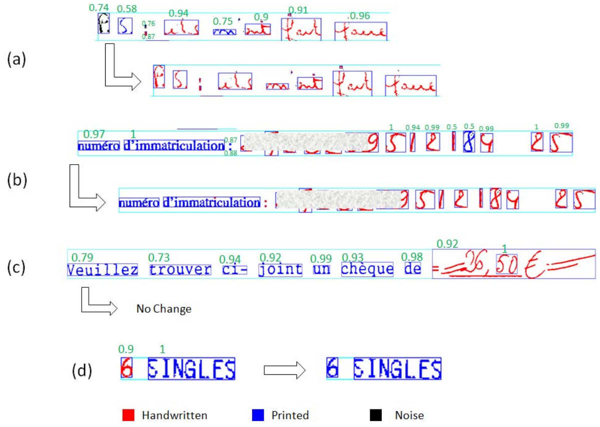

Fig. 4. Examples of pseudo-line deterministic re-labeling

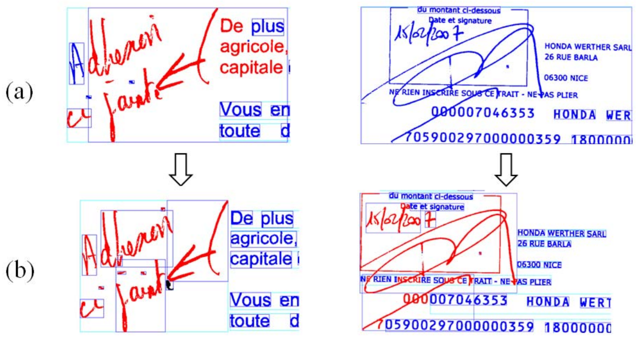

Fig. 5. (a) Multilined pseudo-words, (b) Improved segmentation

IV. Experimentations

A. Dataset

The proposed system is evaluated on a set of business document images obtained from ITESOFT. The system is trained using a set of 107 documents for a total of 32715 pseudo-words. On the other hand, 202 documents are used as a test dataset (77964 pseudo-words). All documents are labeled at the pixel level, which allows a perfect segmentation during the training stage. This also allows the use of the same measure proposed in [13] to evaluate the system. The separation rate at the pixel level is thus the ratio of pixels correctly labeled compared to the ground truth images.

$$\text{Pix\_rate} = \frac{\#\,\text{pixels correctly classified of a given class}}{\#\,\text{total of the class pixels}}$$

However, another evaluation measurement is used in the literature based on the pseudo-words:

$$\text{Pword\_rate} = \frac{\#\,\text{pseudo - words correctly classified}}{\#\,\text{total of pseudo - words}}$$

This measure is only used to evaluate the separation between the printed and handwritten pseudo-words.

B. Evaluation

The evaluation of the new contextual re-labeling methods at the pixel level is given in TABLE I. The proposed re-labeling methods are based on modeling the handwritten and printed text which results in a drop of the noise separation rate. This drop can be explained by the irregularity of the 'noise' pseudo-words (logos, lines, scanning noise…). Hence, noise blocks are associated to handwritten or printed classes. Noise errors caused by small CCs, as in Fig. 6, can be absorbed by the OCR (or ICR) applied on the extracted script. On the other hand, errors due to big CCs, such as logos misclassified as text must be removed by a specific preprocessing step. Experimentations on a subset of the test dataset showed that removing logos improved the performance of the 'noise' class.

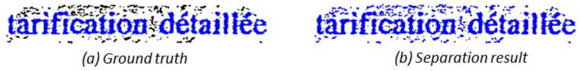

Fig. 6. Example of noise misclassification – (a) the ground truth, (b) the separation result where noise CCs are classified as printed text

TABLE I. System evaluatoin at the pixel level

System	H%	P%	N%
Proposed system without contextual re-labeling [1]	97.7	96.5	**94.3**
k-NN with constraints [1]	95.5	97.5	92.3
Confidence propagation [1]	97.8	96.6	94.0
CRF (method I)	98.5	97.1	94.2
Grouping by pseudo-lines (CRF): Probabilistic (method II)	**98.9**	97.5	93.5
Grouping by pseudo-lines: Determinstic (method III)	98.3	**99.2**	87.9

We can notice that all the three proposed methods outperforms those proposed in [1] for the handwritten class. Indeed, the use of horizontal neighborhood overcomes an important drawback of the *k*-NN method, where a nearby printed text can affect a handwritten pseudo-word and assign it to the wrong class. The use of the new contextual neighborhood based on pseudo-lines allows improving the performance of the CRF model based on the local neighborhood definition (98.5% to 98.9% for the handwritten class and 97.1% to 97.5% for the printed class). On the other hand, the deterministic model allows improving significantly the printed text separation rate to 99.2% with a good rate for handwritten class. In addition, the deterministic model is better in correcting the small pseudo-words (diacritics) which results in a very good pseudo-words rate compared to the two other methods. Furthermore, the introduced improvements to the segmentation method improve the deterministic model rates to 99.1%, 99.2% and 90.1% for the handwritten, printed and noise classes respectively (as shown in TABLE II.).

The performance of the literature systems are compared to our best system (grouping by deterministic pseudo-lines model and using the improved segmentation method) in TABLE II. We must notice that the test datasets are different for each system and thus direct comparison might not be so relevant. Our system achieves a very good pseudo-word classification rate for both handwritten and printed text (97.3% and 99.5% respectively) for a total of 98.7%.

V. Conclusion and Perspectives

In this paper, we have presented an improvement of an existing printed/handwritten text separation system. Examples of result documents are given in Fig. 7. A new post-processing module is proposed to face classification errors. Instead of considering only local neighborhood of a pseudo-word, the whole pseudo-line is considered as the contextual neighborhood. This hypothesis is based on the fact that text lines in documents often contain only one script type. Pseudo-line regularity and classification confidences are considered to determine the dominant class of a given pseudo-line. The proposed re-labeling method reaches very good separation rates compared to our previous method, and to the literature systems.

In the future work, we will study the possibility of logo extraction before proceeding to the text separation task. We will also consider proceeding by a feature selection algorithm in order to reduce the feature vector size and improve the classification accuracy.

TABLE II. SYSTEM EVALUATION AT THE PSEUDO-WORD LEVEL COMPARED TO THE STATE OF THE ART SYSTEMS

System	Description					Pword_rate			Pix_rate			
	Features	**Classifier**	**Neighbors**	**Re-labeling**	**Docs**	**H%**	**P%**	**All%**	**H%**	**P%**	**N%**	**All%**
Kandan et al. [12]	7	SVM	Delaunay triangulation	Majority voting	150	-	-	93.2	-	-	-	-
Peng et al. [7]	12	G-means	4-NN	MRF	82	93.8	95.7	95.5	-	-	-	-
Shetty et al. [9]	23	CRF	6-NN	CRF	27	-	-	-	94.8	98.4	89.8	95.8
Zheng et al. [4]	31	Fisher classifier	Horizontal Left – right	MRF	94	93.0	98.0	97.8	-	-	-	-
Grouping by pseudo-lines: deterministic	137	SVM	Pseudo-line	Dominant class	202	**97.3**	**99.5**	**98.7**	**99.1**	**99.2**	**90.1**	**96.8**

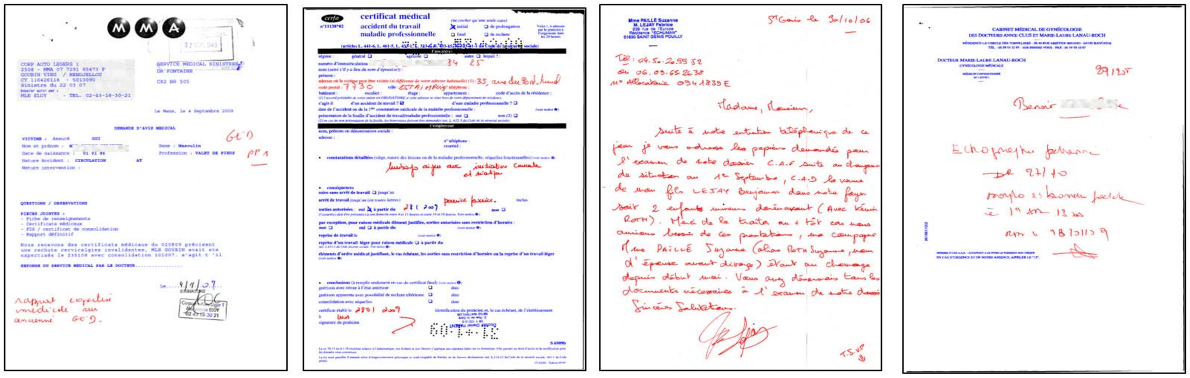

Fig. 7. Example of the printed/handwirtten text separation results

ACKNOWLEDGMENT

The authors would like to thank their collaborator - ITESOFT, for providing the documents database.

REFERENCES

[1] A. Belaïd, K. Santoch and V. Poulain d'Andecy, "Handwritten and Printed Text Separation in Real Document," *Machine Vision Applications,* vol. 2, 2013.

[2] U. Pal and B. B. Chaudhuri, "Machine-printed and hand-written text lines identification," *Pattern Recognition Letters,* vol. 22, pp. 431-441, 2001.

[3] E. Kavallieratou and S. Stamatatos, "Discrimination of Machine-Printed from Handwritten Text Using Simple Structural Characteristics," in *International Conference on Pattern Recognition*, Cambridge, UK., pp. 437-440, 2004.

[4] Y. Zheng, H. Li and D. Doermann, "Machine Printed Text and Handwriting Identification in Noisy Document Images," IEEE Transactions on *Pattern Analysis Machine Intelligence,* vol. 26, pp. 337-353, 2004.

[5] J. K. Guo and M. Y. Ma, "Separating Hadwritten Material from Machine Printed Text Using Hidden Markov Models," in *International Conference on Document Analysis and Recognition*, Seattle, WA, USA, pp. 439-443, 2001.

[6] L. F. da Silva, A. Conci and A. Sanchez, "Automatic Discrimination between Printed and Handwritten Text in Documents," in *Brazilian Symposium on Computer Graphics and Image Processing*, Rio de Janeiro, Brazil, pp. 261-267, 2009.

[7] X. Peng, S. Setlur, V. Govindaraju and R. Sitaram, "Handwritten text separation from annotated machine printed documents using markov random," *International Journal on Document Analysis and Recognition,* vol. 16, pp. 1-16, 2011.

[8] K. Zagoris, L. Pratikakis, A. Antonacopoulos, B. Gatos and N. Papamarkos, "Distinction between handwritten and machine-printed text based on thebag of visual words model," *PatternRecognition,* vol. 47, pp. 1051-1062, 2014.

[9] S. Shetty, H. Srinivasan and S. Srihari, "Segmentation and Labeling of Documents using Conditional Random Fields," in *Proc. SPIE 6500, Document Recognition and Retrieval XIV*, 65000U, 2007.

[10] M. Shirdhonkar and M. B. Kokare, "Discrimination between Printed and Handwritten Text in Documents," *IJCA Special Issue on "Recent Trends in Image Processing and Pattern Recognition,* vol. 3, pp. 131-134, 2010.

[11] K.-C. Fan, L.-S. Wang and Y.-T. Tu, "Classification of machine-printed and handwritten texts using character block layout variance," *Pattern Recognition,* vol. 31, pp. 1275-1284, 1998.

[12] R. Kandan, N. K. Reddy, K. R. Arvind and A. G. Ramakrishnan, "A robust two level classification algorithm for text localization in documents," in *International conference on Advances in visual computing*, Lake Tahoe, NV, USA, pp.96-105, 2007.

[13] F. Shafait, D. Keysers and T. M. Breuel, "Performance Evaluation and Benchmarking of Six-Page Segmentation Algorithms," *Pattern Analysis and Machine Intelligence,* vol. 30, pp. 941-954, 2008.

[14] Ranjeet Srivastva, Aditya Raj, Tushar Patnaik, Bhupendra Kumar, "A Survey on Techniques of Separation of Machine Printed Text and Handwritten Text," *International Journal of Engineering and Advanced Technology*, Volume-2, Issue-3, February 2013.

[15] S. Nicolas, J. Dardenne, T. Paquet et L. Heutte, "Document Image Segmentation Using a 2D Conditional Random Field Model," in *Ninth International Conference on Document Analysis and Recognition*, Curitiba, Brazil, pp. 407-411, 2007.

[16] C.-W. Hsu and C.-J. Lin. "A comparison of methods for multi-class support vector machines", *IEEE Transactions on Neural Networks*, 13, pp. 415-425, 2002.