Handwritten and Printed Text Separation in Real Document

Abdel Belaïd, Santosh K.C., Vincent Poulain d'Andecy

Handwritten and Printed Text Separation in Real Document

Abdel Belaïd, K.C. Santosh
LORIA - Université de Lorraine
54506 Vandoeuvre-lès- Nancy, France
{abdel.belaid, santosh.kc}@loria.fr

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

Abstract

The aim of the paper is to separate handwritten and printed text from a real document embedded with noise, graphics including annotations. Relying on run-length smoothing algorithm (RLSA), the extracted pseudo-lines and pseudo-words are used as basic blocks for classification. To handle this, a multi-class support vector machine (SVM) with Gaussian kernel performs a first labelling of each pseudo-word including the study of local neighbourhood. It then propagates the context between neighbours so that we can correct possible labelling errors. Considering running time complexity issue, we propose linear complexity methods where we use k-NN with constraint. When using a kd-tree, it is almost linearly proportional to the number of pseudo-words. The performance of our system is close to 90%, even when very small learning dataset are used, where samples are basically composed of complex administrative documents.

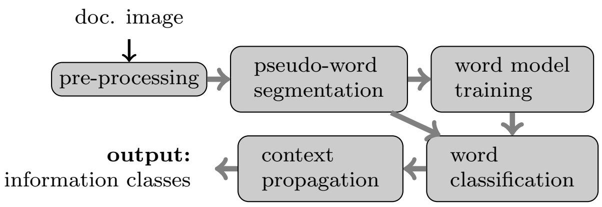

Figure 1. Work-flow showing several consecutive stages, starting from pre-processing to output i.e., $\mathcal{H}\&\mathcal{P}$ text separation.

1 Introduction

Under the purview of document analysis and processing, we are in this paper, motivated to separate handwritten and machine-printed text ($\mathcal{H}\&\mathcal{P}$) so that further processing is feasible such as document information exploitation and retrieval. In other words, such a separation is an important step in the process because it allows retro-conversion to avoid heavy treatments and errors when transcribing the content.

Considering a continuous flow of administrative documents into our system, we face a varieties of document types, content, quality and structure. Fundamentally speaking, documents can be skewed, noisy and sometimes overlapped with graphics i.e., lines and unconstrained annotations. In this context, most of the image samples are required to be properly treated. Without integrating such tools, our system, in this framework, aims to extract the annotations whatever the language: French, German and English used in the document, the content: typed or handwritten, and document structure: structured (e.g. tables), semi-structured (e.g. forms) and structure-free. Although the segmentation topic has been studied since several years [1], different methods have been proposed to solve particular aspects of the separation [2, 3]. Heterogeneous document separation still remains an open problem. Another strong industrial constraint is to reduce running time so that the system can maintain speed. In addition, parameter-free methods are always better since they can generally be applied. In this paper, we are motivated by the work of Kandan et al. [4] where separation has been made into two classes by using descriptors that are insensitive to translation, rotation and scaling. Classifications using SVM and k-NN are first investigated, and a re-classification step is then performed using a Delaunay triangulation. Zheng *et* al. proposed two segmentation approaches and evaluated over noisy documents [5]. The first one is used to determine the most appropriate segmentation where a comparison is made between the segmentation into words, lines and connected components. The latter one deals with word classification by selecting 31 descriptors over a hundred. They also introduce information about class in order to take the noise into account. Fisher classifier is used to label the segmented blocks and Markov field then allows fine classification, considering the contextual information of each word.

The rest of this paper is organised as follows. We start with detailing our proposed approach in Section 2. It mainly includes pre-processing, pseudo-word segmentation, word model training, word classification and pseudo-word grouping. Full experimental results (and of course, analysis) are reported in Section 3. The paper is concluded in Section 4 including a few perspectives.

2 The proposed approach

As illustrated in Fig. 1, our proposed approach consists of several consecutive steps. It includes pre-processing, pseudo-word segmentation, word model training, word classification and context propagation. In what follows, we explain them, one after another.

Preprocessing. The low quality documents require a significant preprocessing. Our pre-processing is composed of the following steps:

1. edge removal by using a rule system based on shape and position of the connected components (CC);
2. noise filtering by using a modified kfill [6];
3. slope detection by using the RAST method [7]; and
4. filtering by using the modified k-fill on the deskewed document.

Pseudo-word segmentation. In this section, we create regular and stable areas that will be used to label

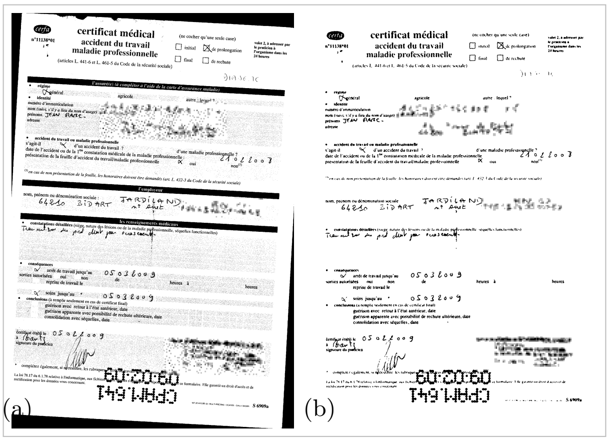

Figure 2. An example showing pre-processing: (a) input sample and (b) its corresponding output.

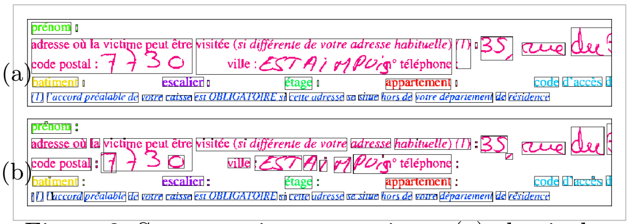

Figure 3. Segmentation comparison: (a) classical RLSA and (b) double RLSA. Extracted pseudo-words are framed and lines are identified by the color of the pseudo-words.

the $\mathcal{H}\&\mathcal{P}$ zones in the document image. To handle this, we use a double RLSA as presented in Algorithm1 i.e., it aims to provide fine word segmentation.

In each one of the extracted lines, smearing is performed first and the distances between the bounding boxes of the adjacent CC are then calculated. This allows to construct a histogram that generally provides an overall shape appearance. It contains two dominant peaks:

1. the first corresponds to the most frequent gap between CC that can be considered as the distance between characters of the same word; and
2. the second peak corresponds to the most frequent gaps between words belonging to the same row.

Note that the first peak can be considered as the distance between the letters in every word and in a similar fashion, the second peak determines the threshold to be used in pseudo-word segmentation. We can therefore apply a second smearing that allows a finer segmentation because handwritten and printed words do not respect similar (usual) distances between the letters and words, and thus we are able to adapt the row content segmentation. Fig. 3 illustrates the comparison between the original and the double RLSA. In this illustration, it is important to notice that words are well segmented in case when double RLSA is used in contrast to text block (that sometimes contains several words within it) from classical RLSA.

Word model training. As said before, we need reliable models to separate $\mathcal{H}\&\mathcal{P}$ information. In order to have these models, we perform two classes of learning from samples by taking words representatives. We

Algorithm 1 Segmentation by double smearing

```
$lines \leftarrow smearing(image)$
for all line $L$ in $lines$ do
    $list\_edistances \leftarrow \emptyset$
    for all CC $c$ in line $L$ do
        $d_{min} \leftarrow distmin(listeccx, c)$
        $list\_edistances \leftarrow add(list\_edistances, d_{min})$
    end for
    $compt \leftarrow bincount(list\_edistance)$
    $histo \leftarrow compt[2::2] + compt[3::2]$
    $i \leftarrow argmax(histo)$
    repeat
        $previous \leftarrow histo[i]$
        $i \leftarrow i + 1$
    until $histo[i] > previous$
    $d_{hs} \leftarrow i + 2$
end for
```

then select several specific descriptors belonging to four different categories:

1. morphological (local properties of pseudo-words such as height, width and pixel number);
2. CC descriptors (11 descriptors as proposed in [5]);
3. pixel repartition (global descriptors like invariant HU moments, variance of the projection profiles [4, 8]; and
4. other local properties such as run length, crossing count and bi-level co-occurrences, as described in [5].

Classification. To handle pseudo-word classification, we employ a SVM. Although it is initially suggested to separate only the $\mathcal{H}\&\mathcal{P}$ information, we use a multi-class SVM so that an additional class i.e., noise can be taken into account. To handle this, two approaches are basically used: 1) the combination of bi-class SVM and 2) the learning of a unique multi-class SVM (MSVM). MSVM is based on a principle similar to one-vs-all [9] where each class has its own decision function and the class corresponding to the function giving the highest value wins. The difference is that, for a MSVM with Q classes, the Q functions are learnt at the same time with exactly similar constraints. A single optimization problem is solved by using the maximization of the sum of the margins for each class. There are four different methods that differ in terms of application penalty. We use the tool presented by Weston and Watkins [10] where it cumulates the penalty compared to the margins of each class. The implementation is carried out on the *Weka* platform and the SMO classifier with the extension of the problem into three classes by the method one-vs-one as described in Mayoraz et al. [11].

Pseudo-word grouping. This re-grouping method uses spatial proximity to re-group elementary units. For each component, k nearest neighbours are found and the label of the component is compared with the ones in their neighbours. If more than 50% of the neighbours share the same label, this label is assigned to the central component.

Generally speaking, since text is written horizontally, horizontal proximity between components is preferred to be vertical ones. Then, we define the distance as

$$d(e_1, e_2) = \sqrt{(x_1 - x_2)^2 w_x^2 + (y_1 - y_2)^2 w_y^2} \quad (1)$$

Algorithm 2 k-NN grouping with constraints

```
Require: $\forall c \in \mathcal{C}$ ,    old_label$(c) \in (L)$
for all  $c \in \mathcal{C}$ do
    $Neighb \leftarrow k\_nearest\_neighbour(k, c, max\_\mathrm{dist.})$
    $n = card(Neighb)$
    $new\_label[c] \leftarrow old\_label[c]$
    for all  $class \in (L)$ do
        $N_c \leftarrow \{x | x \in Neighb, old\_label[x] = class\}$
        if  $card(N_c) > \frac{n}{2}$ then
            if  $\sum_{x \in N_c} area(x) > \frac{1}{2}(c)$ then,
                $new\_label[c] \leftarrow class$
            end if
           break
        end if
    end for
end for
```

where x_i, y_i are the coordinates of the center of gravity of CC n_i, and $w_{x;y}$ are weights corresponding to each axis. In a similar manner, another distance is computed i.e., the distance is taken from the border of the bounding boxes. Based on the framework, in what follows, we explain three different algorithms i.e., A1:A3.

A1. *Grouping by k-NN.*
It employs a classical k-NN algorithm where parameters k and a threshold i.e., $max_$dist. The k nearest neighbours are taken into account if they are closer than the pre-defined $max_$dist. The distance parameter basically prevents far away neighbours to interfere with the component. In our case,$max_$dist. has been fixed to 1) 300 pixels for distance 1, and 2) 100 pixels for distance 2 with images at 300 dots per inch (dpi). Note that the distance 2 is lower than distance 1, and depends of the relative positioning between the bounding boxes and their sizes. These thresholds however, are image resolution dependent.

A2. *Grouping by the NN with constraints.*
The algorithm can be improved by avoiding big components that are basically be corrupted by small ones (as noise). Before flipping the label of the component, we perform a test to check whether the accumulated pixels of a neighbour contributing the change of label is significant in comparison to the number of pixels of the tested component. For this, in our test, the sum should be at least 50% of the main component. Note that the opposite does not exist. Big components are regrouped with small ones to help gathering main text with small components as commas, apostrophes or accents. Moreover, big components contain more information so they are generally more reliable, and thus the classification is more accurate. An overall idea is presented in Algorithm 2.

A3. *Grouping by confidence voting.*
The classifier confidence helps to maintain the decision. Based on the idea of grouping via nearest neighbours in addition with some specific constraints, we examine the confidence of the nearest neighbour of a selected pseudo-word. If the latter is stronger than that of the pseudo-word, then it takes the neighbourhood class. A Gaussian or polynomial law can weight the neighbour confidence by its distance to the pseudo-word.

3 Experiments

3.1 Dataset and evaluation metric

Dataset. To perform the tests, we have selected 75 documents for learning and a 300 documents for testing. As a reminder, these samples are taken from the real-world industrial problem.

Evaluation metric. Our evaluation of $\mathcal{H}\&\mathcal{P}$ separation is performed according to the measure proposed by [12]. All test documents have been perfectly labelled at pixel level, where performance is evaluated in terms of recognition rate.

$$\text{Recognition rate} = \frac{\#\text{ of pixels correctly labelled}}{\#\text{ of pixels used}}. \quad (2)$$

3.2 Results and analysis

Table 1 shows recognition rates for four grouping methods. The k-NN uses $k = 2$. The methods' confidence use respectively f_{gauss}, f_{poly2} and f_{poly4} as weighting functions.

$$f_{gauss}(conf, dist) = conf \times \exp\left(-\frac{10^{-3} * dist^2}{conf^2}\right) \quad (3)$$

$$f_{poly2}(conf, dist) = -5 \cdot 10^{-4}\left(\frac{dist - 1}{conf}\right)^2 + conf \quad (4)$$

$$f_{poly4}(conf, dist) = -10^{-6}\left(\frac{dist - 1}{conf}\right)^4 + conf \quad (5)$$

Based on reported results in Table 1, we observe the following:

1. We note that the classification by k-NN provides better results as expected the recognition rate of double smearing i.e., segmentation without regrouping. In contrast, methods based on confidence degrades performance. This is mainly due to the fact only local vicinity (a single neighbour) is taken into account, that makes misclassification possible.
2. In our study, we have found that handwritten mixes with printed and other cases where grouping changes the isolated handwritten annotations label (e.g., a figure or a symbol). In this situation, we are required more contextual information including the better interpretation, which is beyond the scope of current work.

Table 1. Evaluation of four grouping methods.

Recognition rate	Hand.	Print.	Noise	Average
Double smearing	96.1	98.5	**35.7**	89.48
k-NN	93.4	98.3	27.3	89.54
kNN with constraints	**99.3**	**99.0**	27.9	**90.68**
Gaussian confidence	94.5	97.7	27.2	87.49
Poly confidence2 & 4	93.5	97.7	14.2	86.06

On the whole, for visual understanding, we provide a few examples of $\mathcal{H}\&\mathcal{P}$ text separation in Fig. 4. Furthermore, Fig. 5 shows a comparison between four classifiers: SVM, Tree C4.5 (J48 implementation), REPTree and NN. In this comparison, we have found that

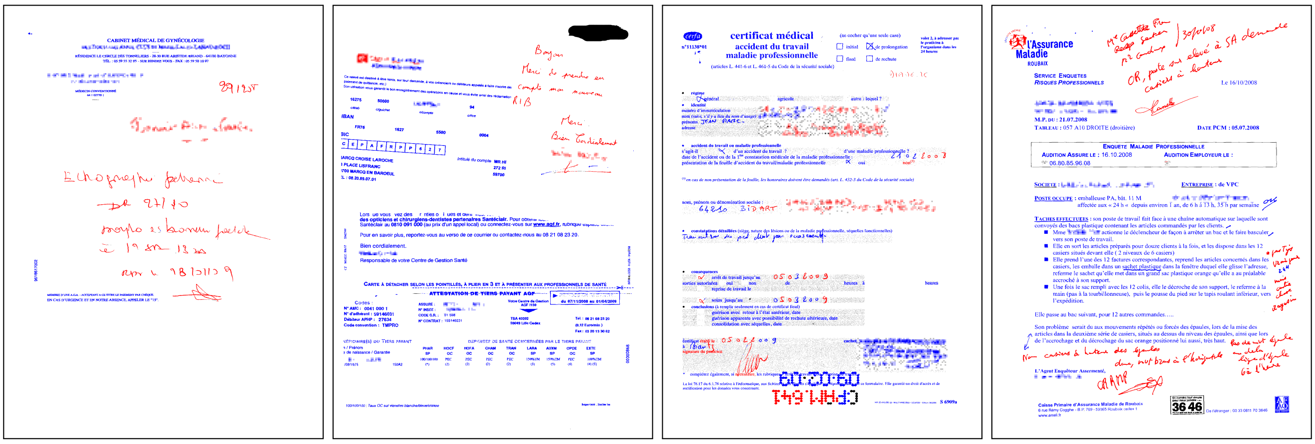

Figure 4. A few examples of $\mathcal{H}\&\mathcal{P}$ text separation, illustrating the robustness of the proposed approach.

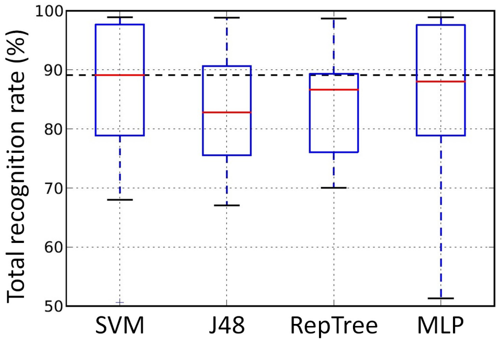

Figure 5. Evaluation of four classifiers

SVM performs the best, by providing marginal difference with NN. This means that MLP can still be applied.

4 Conclusion

In this paper, we have presented an approach to separate handwritten and machine-printed text from a scanned document in addition to the noise. The method is based on a double smearing technique to obtain the pseudo-words. These serve as a basis for classification. For these words, descriptors are extracted where they all have a linear complexity with the number of pixels. Descriptors are then fed into a multi-class SVM with a Gaussian kernel which provides the first label of each pseudo-word. A second analysis is carried out by studying the local vicinity of each pseudo-word that can change label if the neighbours are from another class. This integration allows context to correct several possible errors. In our test, we have found that the method is k-NN with constraints where kd-tree has been used.

Considering our small learning database, the results are fairly encouraging. This will certainly forecast an appropriate commercial application. Based on our reported results, a long-term approach about incremental learning is one of the further issues.

Acknowledgements

The authors would like to thank Didier Grzejszczak and Yves Rangoni for their help and Hervé Locteau for his high level implementation to make the concept commercially useful. The work has been done during their stay with us at LORIA - Université de Lorraine.

References

[1] Kang W.-X., Yang Q.-Q., Liang R.-P., The Comparative Research on Image Segmentation Algorithms, in: *Proceedings of the ECTS*, 2009, pp. 703-707.

[2] S. Chanda, K. Franke, and U. Pal, Structural handwritten and machine print classification for sparse content and arbitrary oriented document fragments, in: *Proceedings of SAC*, 2010, pp. 18-22.

[3] Peng, X., Setlur, S., Govindaraju, V., and Sitaram, R., Handwritten text separation from annotated machine printed documents using markov random fields. *IJDAR*, 16(1): 1-16, 2011.

[4] Kandan R., N.Kumar R., Arvind K. R., Ramakrishnan A. G., A robust two level classification algorithm for text localization in documents, in: *Proceedings of the Advances in visual computing*, 2007, pp. 96-105.

[5] Zheng Y., Li H., Doermann D., The segmentation and identification of handwriting in noisy document images, in: *Proceedings of DAS*, 2002, pp. 95-105.

[6] Chinnasarn K., Rangsanseri Y., Thitimajshima P., Removing Salt-and-Pepper Noise in Text/Graphics Images, *The Asia-Pacific Conference on Circuits and Systems*, 1998, pp. 459-462.

[7] van Beusekom J., Shafait F., Breuel T. M., Combined orientation and skew detection using geometric textline modeling, in: *Proceedings of the ICDAR*, 2010, pp. 79-92.

[8] da Silva L. F., Conci A., Sanchez A., Automatic Discrimination between Printed and Handwritten Text in Documents, in: *Proceedings of the Brazilian Symposium on CGIP*, 2009, pp. 261-267.

[9] V. N. Vapnik. The nature of statistical learning theory. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[10] Weston J., Watkins C., Multi-class Support Vector Machines, *Technical report, Royal Holloway, University of London*, 1998.

[11] Mayoraz E., Alpaydin E., Support Vector Machines for Multi-class Classification, in: *Proceedings of the ANN*, 1999, pp. 833-842.

[12] Shafait F., Keysers D., Breuel T. M., Performance Evaluation and Benchmarking of Six-Page Segmentation Algorithms, *IEEE-PAMI*, 30(6):941-954, 2008.