IMPROVING FEATURES EXTRACTION FOR SUPERVISED INVOICE CLASSIFICATION

Alberto Bartoli
DEEI - University of Trieste
Via Valerio 10, Trieste, Italy
email: bartoli.alberto@units.it

Giorgio Davanzo
DEEI - University of Trieste
Via Valerio 10, Trieste, Italy
email: giorgio.davanzo@deei.units.it

Eric Medvet
DEEI - University of Trieste
Via Valerio 10, Trieste, Italy
email: emedvet@units.it

Enrico Sorio
DEEI - University of Trieste
Via Valerio 10, Trieste, Italy
email: enrico@sorio.net

ABSTRACT
An essential step in the understanding of printed documents is the classification of such documents based on their class, i.e., on the nature of information they contain and their layout. In this work we are concerned with automatic classification of such documents. This task is usually accomplished by extracting a suitable set of low-level features from each document which are then fed to a classifier. The quality of the results depends primarily on the classifier, but they are also heavily influenced by the specific features used. In this work we focus on the feature extraction part and propose a method that characterizes each document based on the spatial density of black pixels and of image edges. We assess our proposal on a real-world dataset composed of 560 invoices belonging to 68 different classes. These documents have been digitalized after their printed counterparts have been handled by a corporate environment, thus they contain a substantial amount of noise—big stamps and handwritten signatures at unfortunate positions and so on. We show that our proposal is accurate, even a with very small learning set.

KEY WORDS
Intelligent data analysis, machine learning and document image classification

1 Introduction

Document classification is a crucial premise for high level document analysis, for instance when extracting and processing information from large volumes of printed documents.

In this work we are concerned with the understanding of printed documents, e.g., commercial invoices, patents, laws, scientific papers and so on. We consider a system capable of extracting automatically a specified set of information items from a document, once the class (roughly corresponding to the layout) of the document is known. For example, the system could extract Date, Amount and Number from an invoice, once the emitter of the invoice is known [1]. Or, it could extract Authors and DOI from a scientific paper once the publisher is known. In this paper we focus on the problem of determining automatically the class of the document to be processed. The details of the underlying document understanding system are orthogonal to this work. We propose a novel feature extraction method that may greatly simplify and improve the preliminary classification step of document understanding in such scenarios.

What makes this scenario difficult to cope with is the combination of: (i) documents of different classes often exhibit strong visual similarity; and (ii) the number of different classes may be large, in the order of tens or thousands, as it usually depends on the requirements of the overall system. The latter effect may greatly increase the former impact and make the scenario more challenging: in practice, different documents correspond to different classes even though the corresponding documents are visually similar. For example, thinks about invoices from different emitters or papers from different journals.

We will hence focus on selecting features strict enough to grant a precise classification among visually similar documents; specifically, we will investigate how to enrich features already proposed in literature—like density of black pixels—while reducing the learning set size to minimal levels. We found that considering image edges, while extracting numerical features, may allow achieving these goals.

2 Related work

A classifying system is usually characterized by the following three key aspects: the features nature (i.e., what each feature means), the features representation (e.g., graphs, numeric vectors of fixed or variable size, etc.) and the classification algorithm itself. We focus our interest on the first and second stages, and aim at improving existing features extraction techniques.

The classification features may be grouped as follows [2]:

- *image features*, which are extracted directly from the image, like the density of black pixels in a given re-

gion [3], or the number of white separation blocks in the segmented image [4] or the gaps between column and rows [5];

- *structural features*, which are obtained from physical or logical layout analysis. In [6] the authors use a fully connected graph of segmented blocks with attributes like size, position of blocks and font size; in [7] features are computed by constructing a layout hierarchy of document components grouping together small elements with a bottom-up approach;

- *textual features*, that may be computed without performing an OCR (like the character shape coding in [8]) or after processing the document through an OCR (a noise resistant approach is presented in [9]).

An interesting solution is proposed in [3], where the black pixel density information is extracted and used as an input for a k-NN classifier and for a Multi Layer Perceptron classifier. The results are very promising: yet, differently from our work, the authors used a dataset in which documents of different classes greatly vary in visual aspect, which correspond to a different real-world scenario which is possibly less challenging.

In [10] the authors try to solve a problem similar to the one we are investigating; in this work a k-NN classifier is used on a segmented image, analyzing only the main graphic element of each document (the logo). Since different companies could use the same accounting system, that approach could create need to rely on different classes even in cases in which the information content is located similarly on the page: this could make the proposed approach unsuitable for scenarios in which the number of classes—defined in terms of document understanding—is high.

An interesting view of the state of the art of document image classification is provided by the authors of [2].

3 Our approach

For simplicity, but without loss of generality, we consider invoice documents; we define a document D as a black and white image of an invoice obtained with a scanner, while a class C is a collection of invoices created with the same accounting program—different businesses could issue invoices with the same accounting program. Usually the document image presents several noisy elements such as handwriting notes, stamps, staples and errors produced by the scanner itself.

3.1 Image preprocessing

We noticed that images obtained by scanning real-world documents of the same class could be significantly different due to human errors made during their digitalization. A frequent cause consists in positioning errors on the scanner area, due to non-standard document sizes, cut documents, and so on. We addressed this problem applying an automatic method called *rolling* which aims at aligning each document in the same way, hence obtaining images whose actual content position is substantially constant.

We identify the upper relevant pixel of the image using an edge recognition algorithm (Canny detector, see [11]) applied to a low-resolution version of the image obtained by resizing the original with a $1/6$ scaling factor; we reduce the image in order to remove the noise caused by the scanner and small texts. We consider the first edge as the upper pixel.

To maintain the image size, we remove all the content between the upper pixel and the top border to append it at the end of page. We verified that this method does not remove any relevant content.

3.2 Features extraction

We extract two types of features:

- density of black pixels;

- density of the image edges.

The black pixel density is the percentage of black pixels that cover a given area of the document; therefore, it is a number ranging in $[0, 100]$.

We divide the image in a $n \times n$ grid, and for each cell of the grid we compute the black pixel density; hence, we obtain a feature n^2 length vector $v^b = \{v^b_1, \ldots, v^b_{n^2}\}$.

Then, we reduce the image resolution (see Section 3.1), apply the Canny detector and, on the resulting image consisting only of the edges, we compute the black pixel density, obtaining another features vector of the same length $v^e = \{v^e_1, \ldots, v^e_{n^2}\}$.

These two vectors are finally concatenated to obtain the resulting features vector $v = \{v^b_1, \ldots, v^b_{n^2}, v^e_1, \ldots, v^e_{n^2}\}$.

The basic idea is to improve the identification of regions with the same black pixel density but different content. For example, take two rectangular areas, one filled with tiny text and one containing a logo: they could generate the same number of black pixels, but the edges will differ greatly as well as the corresponding part of the v^e features vector.

Figure 1 summarizes the process here described for a generic invoice.

3.3 Classification

In order to simplify the classification stage, we first perform a Principal Component Analysis (PCA) on the $2n^2$ features extracted as described above. To this end, we apply PCA on a learning set L composed of the vectors v extracted from k documents of each class C. We select the number of features that grants a variance greater than 95% on L. The PCA eigenvectors matrix is then applied on the

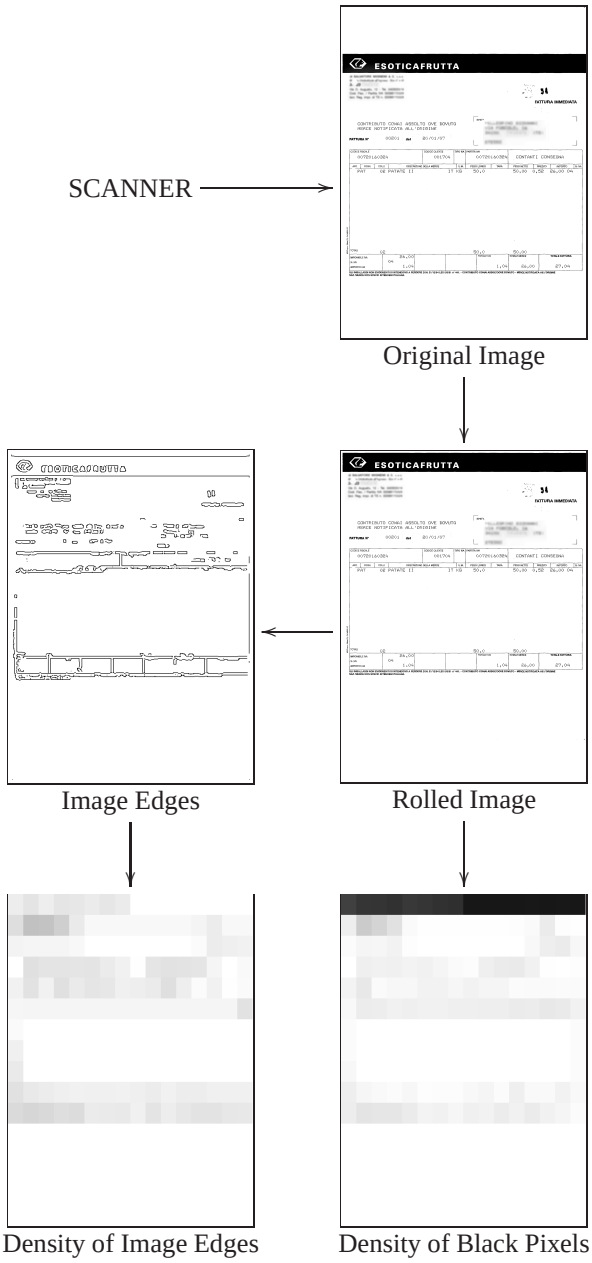

Figure 1. Features extraction work-flow.

features vectors of L, thus obtaining a set of smaller vectors v'.

Then, the actual classification of a given document D is performed considering the vector v'_D obtained by applying the PCA eigenvectors matrix on the features vector v_D extracted from D as described in Section 3.2.

In this work, we consider two classifiers: Support Vector Machines (SVM) and a distance-based classifier. In both cases, we use vectors v of the same learning set L to train the classifier.

For SVM, we used LibSvm [12] and set the kernel to a linear kernel.

For the distance-based classifier, we first compute the centroids of each class C by averaging the coordinates of each v of C present in L; then, an investigated element is assigned to the class whose centroid is closer using a given distance metric. During our experiments we assessed the efficiency of several distances: Manhattan, Euclidean, L^∞ and Mahalanobis. The Euclidean proofed to be the best: all the following results will be based on it.

The distance-based classifier provides an indication of the confidence on the proposed class for a given unknown element, by means of the distance value itself from the corresponding centroid. This can be an advantage over the SVM classifier in some scenario, e.g., when a top N ranking of the most suitable classes should be provided by the classifier rather than a single class.

4 Experiments

In order to assess our approach effectiveness we collected a real-world dataset composed by 562 invoices created by 68 different programs, each emitting program corresponding to exactly one class: the largest class contains 80 invoices, the smallest 2. In all the following experiments, the learning set is composed by randomly chosen documents of this dataset.

Some classes consist of documents whose original paper size is smaller than the A4 paper format: these invoices were scanned as A4 pages and positioned in a variable way with respect to the scanner area, resulting in images whose content position is variable.

We want to assess our approach effectiveness with respect to the size of the learning set k, the grid size n (i.e., the number of blocks considered when computing the black pixels density) and classifier (SVM or distance based).

As a comparison baseline, we will consider the SVM applied to the PCA transformation of v^b—i.e., without the features generated by the image edges; these features are similar to those proposed in [3]. We remark that the fully pyramidal classification method used in the cited paper provides performances on our dataset that are substantially equivalent to the non pyramidal version.

n	Errors	Correct classification rate (%)
8	37	92.51
16	14	97.17
32	44	91.09

Table 1. Classification errors and correct classification rates with different grid sizes.

k	Evaluated elements	Errors	Correct classification rate (%)
1	494	14	97.17
2	436	3	99.31
3	389	2	99.49
4	350	3	99.14
5	321	3	99.07

Table 2. Correct classification rates with different learning set size

4.1 Grid dimension

As a preliminary step we assessed the correct number of elements that should compose the grid used for the black pixel density computation: to this end, we tested different grid sizes ($n = 8$, $n = 16$ and $n = 32$) using the most promising classifier (SVM) and a learning set size $k = 1$.

Table 1 shows the result of this experiment; it can be seen that a grid size of $n = 8$ does not contain enough information to provide a good classification. The best result is obtained at $n = 16$; a larger grid size provides an amount of information that turns out to be too specific for this classification task. Following experiments have been performed with $n = 16$.

4.2 Learning set size

We generate learning sets of different sizes: $k = 1$ (i.e., only one element per class is required to classify other elements of the same class), $k = 2$, $k = 3$, $k = 4$ and $k = 5$. Classes for which our dataset contained less than $k + 1$ documents have been considered in L using all their documents but one.

Results when using the SVM classifier are reported in Table 2. We can see that our method scores a 97.17% success even when the learning set is composed by a single element per cluster; furthermore, we visually inspected all the incorrectly classified documents and noted that on seven there were gross differences from their siblings (e.g., huge stamps that usually do not appear and severe scanning errors).

When we add another element for class to T the correct classification rate jumps to 99.31% and stays stable for all the following values of k; since adding elements to T means to increase the required computation time, $k = 2$ seems an optimal trade-off among computational effort, classification rate and number of required training elements.

4.3 Comparison and further feature reduction

In this section we provide an overall comparison of our method variants. We include also a version of our extraction method which consider only a portion of the document page and hence can be faster. The proposed modification is based on the consideration that a document is usually composed by a header, a footer and a text content in between: the formers usually contains enough information about the visual appearance of the document; in this suite of experiments we verify whether the header and footer content are sufficient to correctly classify a document.

Here we will compare the following methods:

- *baseline*, the baseline algorithm (see Section 4);
- *page-SVM*, our feature extraction method and the SVM classifier;
- *page-distance*, our feature extraction method and the distance-based classifier;
- *header-SVM*, considering only the top 25% of the page, extracting the features as above and classifying with the SVM;
- *header&footer-SVM*, considering the top 25% and bottom 25% of the page, extracting the features as above and classifying with the SVM.

Results are proposed in Figure 2, plotting all the previous methods at different values of k. The *baseline* method scores quite below all the other we proposed, with a correct classification rate of 85.83% when $k = 1$. The *page-distance* scores below the SVM but acceptably well, only nine errors for $k = 2$.

Reducing the number of features as described in the *header-SVM* method pays with respect to the *original* algorithm, but scores sharply below the others. On the other hand, *header&footer-SVM* gives result fully consistent with the *page-SVM* while reducing the computation time (the feature extraction phase itself for *header&footer-SVM* performs about 45% faster than *page-SVM*).

An interesting finding about our proposal is that it performs well even with small training sets ($k \leq 4$): two elements for each class are enough to obtain acceptable performances.

5 Conclusions

In this work we consider the problem of classifying scanned printed documents. We propose a new simple technique for extracting numerical features from such documents which takes into account also the image edges. We

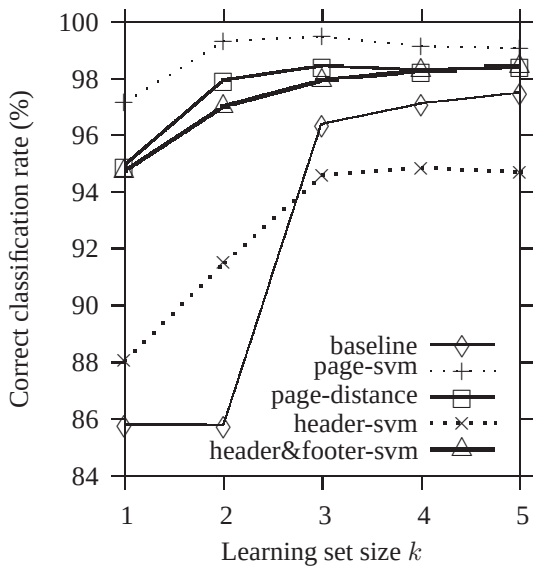

Figure 2. Comparison of the different proposed method variants in terms of correct classification rate, while varying the learning set size k.

test our approach using the SVM classifier and a distance-based classifier.

The classification method here proposed provides very high correct classification rate, even when the learning set have been constructed with only few documents for each class, e.g., 97.17% and 99.31% with a learning set composed of two elements and only one element, respectively. These results have been obtained with a real-world dataset including a substantial amount of noise, as typically occurs when digitalizing printed documents previously handled by a corporate office (e.g., big stamps at unpredictable and undesirable positions, hurriedly scanned documents and so on). The classification rate we obtained scores better than a comparison line from earlier literature. A computationally lightweight flavor of our method provides slightly worse classification rate, which is still better than the baseline. Future work will be devoted by extending further the scope of classification, by allowing automatic detection of new document classes.

References

[1] F. Cesarini, E. Francesconi, M. Gori, and G. Soda, "Analysis and understanding of multi-class invoices," *International Journal on Document Analysis and Recognition*, vol. 6, pp. 102–114, Oct. 2003.

[2] N. Chen and D. Blostein, "A survey of document image classification: problem statement, classifier architecture and performance evaluation," *International Journal on Document Analysis and Recognition*, vol. 10, pp. 1–16, June 2007.

[3] P. Heroux, S. Diana, A. Ribert, and E. Trupin, "Classification method study for automatic form class identification," in *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, vol. 1, pp. 926–928 vol.1, 1998.

[4] J. Hu, R. Kashi, and G. Wilfong, "Document classification using layout analysis," in *Database and Expert Systems Applications, 1999. Proceedings. Tenth International Workshop on*, pp. 556–560, 1999.

[5] C. Shin, D. Doermann, and A. Rosenfeld, "Classification of document pages using structure-based features," *International Journal on Document Analysis and Recognition*, vol. 3, pp. 232–247, May 2001.

[6] J. Liang, D. Doermann, M. Ma, and J. Guo, "Page classification through logical labelling," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 3, pp. 477–480 vol.3, 2002.

[7] C. Nattee and M. Numao, "Geometric method for document understanding and classification using online machine learning," in *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, pp. 602–606, 2001.

[8] A. L. Spitz and A. Maghbouleh, "Text categorization using character shape codes," in *SPIE Symposium on Electronic Imaging Science and Technology*, p. 174–181, 1999.

[9] D. J. Ittner, D. D. Lewis, and D. D. Ahn, "Text categorization of low quality images," in *Symposium on Document Analysis and Information Retrieval*, p. 301–315, 1995.

[10] C. Alippi, F. Pessina, and M. Roveri, "An adaptive system for automatic invoice-documents classification," in *IEEE International Conference on Image Processing, 2005. ICIP 2005*, vol. 2, 2005.

[11] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, p. 679–698, 1986.

[12] C. C. Chang and C. J. Lin, *LIBSVM: a library for support vector machines.* 2001.