Open World Classification of Printed Invoices

Enrico Sorio
DEEI - University of Trieste
enrico.sorio@gmail.com

Alberto Bartoli
DEEI - University of Trieste
bartoli.alberto@units.it

Giorgio Davanzo
DEEI - University of Trieste
giorgio.davanzo@gmail.com

Eric Medvet
DEEI - University of Trieste
emedvet@units.it

ABSTRACT

A key step in the understanding of printed documents is their classification based on the nature of information they contain and their layout. In this work we consider a dynamic scenario in which document classes are not known a priori and new classes can appear at any time. This open world setting is both realistic and highly challenging. We use an SVM-based classifier based only on image-level features and use a nearest-neighbor approach for detecting new classes. We assess our proposal on a real-world dataset composed of 562 invoices belonging to 68 different classes. These documents were digitalized after being handled by a corporate environment, thus they are quite noisy—e.g., big stamps and handwritten signatures at unfortunate positions and alike. The experimental results are highly promising.

Categories and Subject Descriptors

I.7.5 [**Document and Text Processing**]: Document Capture—*Document analysis, Scanning*

General Terms

Experimentation, Performance, Algorithms

Keywords

Document Image Classification, Machine Learning, SVM, Nearest-neighbor

1. INTRODUCTION

Classification of documents based on the nature of information they contain and their layout is a crucial premise for a variety of document analysis tasks. We are concerned with automated extraction of information from printed documents, e.g., commercial invoices, patents, laws, scientific papers and so on. Several approaches have been proposed for extracting automatically a specified set of information items from a document, once the class (roughly corresponding to the layout) of the document is known [10, 3, 4]. For example, the system could extract Date, Amount and Number from an invoice, once the emitter of the invoice is known. Or, it could extract Authors and DOI from a scientific paper once the publisher is known. In this paper we focus on the preliminary classification step of the problem, i.e., associating a document with a class. We focus on invoice documents, without loss of generality.

Classification problems can be subdivided in two categories [11]. *Closed world* classification aims at classifying documents according to a statically defined set of classes. All classes are known in advance and each document submitted to the system is certainly associated with one of these classes. *Open world* classification deals with scenarios in which the set of classes is not known in advance. A document may be associated with one of the classes already known to the system, but it may also be associated with a class never seen before. In the latter case the system must be able to detect the novelty and define a new class accordingly. Open world classification is more general and allows encompassing a broader range of practical problems than closed world classification, but it is also much more challenging.

In this paper we propose an open world classifier and assess its performance on a dataset composed of hundreds of invoices documents, associated with tens of classes, obtained from a real world corpus of paper invoices that were previously handled in a corporate environment—and thus contain a substantial amount of noise like stamps, staples and alike. We propose an hybrid approach. We determine whether a document represents a new class with a nearest-neighbor approach based on euclidean distance, and choose amongst existing classes with a classifier based on Support Vector Machines. We retune both modules when new classes are detected, by choosing a very small number of samples in the retraining set. We represent documents based only on their image-level features [2]. Our proposal may be used in a fully unsupervised or in a partly supervised way. In the latter case the system may require some feedback from human operators. These different working modes allow addressing different trade offs between classification accuracy and amount of human involvement available or desirable. Our experimental analysis includes the unsupervised mode and two different flavors of supervision. The results appear highly promising.

We represent documents in terms of density of black pixels and image edges. Our proposal is a generalization of a closed world classifier that we proposed earlier [2]. Other features proposed for closed world document classification include using a fully connected graph of segmented blocks

DocEng2010, September 21–24, 2010, Manchester, United Kingdom.

with attributes like block position and font size [9]; a set of quadrilaterals for every pair of text lines [8]; white separation blocks in the segmented image [7]; neural networks for classification based on natural language have been proposed in [5]

Regarding the open-world scenario, the problem presented in [1] is similar to the one we are investigating. In this work a nearest-neighbor classifier is used on a segmented image, analyzing only the main graphic element of each document (usually the logo). The dataset consists of 600 invoices belonging to 30 different classes. The composition of the training set and testing set in terms of documents and classes is not provided. The approach achieves a 78% correct classification rate. An open-world classifier based on graph of words is proposed in [6]. The dataset consists of a training set of 324 invoices and a much smaller testing set composed of 169 documents. There are only 8 classes, whose distribution across the training set and testing set is not known. The reported classification rate is 99%.

Comparing our results to those in these works is not very meaningful, because the datasets are different and the results may depend heavily on such factors as, e.g., quality of the images and separation of the classes. We only remark that we have attempted to build a particularly challenging and realistic scenario. As pointed out earlier, we used paper documents that were previously handled in a corporate environment and are quite noisy. Our dataset is composed of 562 invoices belonging to 68 different classes, each containing up to 22 documents. We use a training set of only 7 classes totalizing only 14 documents, and a testing set including all the remaining 548 documents. We repeated each experiment 20 times by randomly selecting these sets.

2. OUR APPROACH

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 with the same graphical appearance and layout. In practice, D may contain several noisy elements such as handwritten notes, stamps, staples and errors produced by the scanner itself (e.g., black speckles, positioning errors, skewed images). Our system will associate any given document D with either an existing class or, in case the class of D has never been observed before, with a newly created class.

We extract two types of features from each document D: (i) density of black pixels and (ii) density of image edges. We divide the image in a 16×16 grid, and for each cell of the grid we compute the black pixel density. Then, we reduce the image resolution and apply an edge detector. We repeat the previous procedure on the resulting image consisting only of the edges, i.e., we compute the black pixel density on a 16×16 grid. We concatenate the two resulting vectors to obtain the features vector v of length $16 \times 16 \times 2 = 512$. For further details please refer to [2].

Vector v is input to a module called *novelty detector*, which determines whether the document belongs to a class already known and, if not, creates a new class. The novelty detector may be configured to require varying amount of feedback from an operator and may also work in a fully automatic way, as clarified below. When the novelty detector does not require any human feedback, v is passed to a *classifier* based on Support Vector Machines. Novelty detector and classifier group documents with a similar appearance in *clusters*. Each class is associated with one or more clusters, while each cluster is associated with exactly one class.

2.1 Novelty Detector

Initially, we require a starting set of n classes, each composed of a single cluster and each associated with up to k documents. We define the centroid of each cluster as the average of all its document feature vectors. For each cluster: (i) we maintain i euclidean distances between the cluster centroid and the corresponding i documents in that cluster that have been processed by the system (when there are less than i documents in the cluster, we use all the distances available); (ii) we compute mean μ and standard deviation σ of these distances; and (iii) we select two threshold values $t = \alpha \cdot \sigma + \mu$ and $t_t = \epsilon \cdot t$ where $\alpha > 0$ and $\epsilon > 1$ are two parameters.

When a document D is submitted we compute its distance from all the cluster centroids and pick the minimum one d. The output of the novelty detector depends on the comparison between d, t and t_t as follows:

New $(d > t_t)$ the system creates a new cluster S' containing only D and prompts the operator to confirm the creation of a new class; the operator can either (i) Approve Creation: the system associates S' with a newly created class C'. Or, (ii) Reject Creation: in this case the operator instructs the system that D has to be associated with an existing class C; the system then associates S' to C.

MaybeNew $(t < d \leq t_t)$ the system prompts the operator to confirm the creation of the new class; the operator can either (i) Approve Creation: the system creates a new cluster S' containing only D and associates S' with a newly created class C' (like in the New case). Or, (ii) Reject Creation: in this case the operator instructs the system that D has to be associated with an existing class C; the system then associates D with the closest cluster that exists already in C.

NotNew $(d \leq t)$ the system does not require any feedback from the operator and D is classified automatically by the Classifier, as explained below.

That is, a MaybeNew outcome identifies a borderline document. Once D has been classified, if its cluster contains i documents or less then the cluster centroid is recomputed and the corresponding cluster parameters are recomputed as well, by executing steps (i)-(iii) above again.

2.1.1 Operation modes

The novelty detector may actually work in three different modes, depending on the amount of user interaction required:

Manual The procedure described above.

Automated The system assumes that the operator will always approve the suggested creation of a new class. Hence, the system will never prompt the operator.

SemiAutomated The system prompts the operator only when the outcome is uncertain (MaybeNew). When the outcome is New, the system assumes that the operator will always approve the suggested creation of a new class.

2.2 Classifier

The classifier is based on Support Vector Machines (linear kernel) and associates a document D with a cluster S—hence with the class C associated with S. Note, though, that the classifier is invoked only when the outcome of the novelty detector is NotNew. Initially, the classifier is trained with the same information known to the novelty detector: n classes, each composed of a single cluster and each associated with up to k documents. Prior to classification, we perform a dimensionality reduction based on Principal Component Analysis (PCA). We select the set of features that grants a proportion of variance greater than 95% on the training set.

Whenever a new document is classified in a cluster that contains i documents or less, the classifier is retrained and the set of features is updated, as follows. A retraining set is constructed by taking all the documents so far assigned to that cluster (please note that at most i documents will be used per cluster). A PCA is executed on all documents of the retraining set, for selecting the new set of features. The classifier is trained based on the new set of features and the retraining set.

3. EXPERIMENTS AND RESULTS

Our dataset is composed of 562 invoices belonging to 68 different classes. These documents were digitalized after being handled by a corporate environment, thus they are quite noisy with handwritten signatures, stamps etc.; during the digitalization the pages were positioned in a variable way with respect to the scanner area, resulting in images whose content position is variable. Each class may contain up to 22 documents. We remark that the distribution of documents in clusters is not known in advance, as clusters are built at runtime depending on the output of the novelty detector as well as the order in which documents are submitted.

The dataset is divided into a *training set* composed of $n = 7$ classes having up to $k = 2$ documents each, and a *testing set* containing all the remaining 548 documents. We set the number of documents per cluster in the training set to $i = 3$. We performed each test 20 times, randomly selecting the training set and the order followed in submitting the remaining documents. All the performance indexes have been averaged on the twenty runs.

We assessed the performances in terms of:

Classification error rate Number of documents assigned to a wrong class over the number of evaluated documents. Classification errors can be caused by (i) document of a not-yet-existing class associated to an existing class *(NewInExisting)* (ii) document of an existing class associated to the wrong class *(ExistingInWrongExisting)* (iii) document of an existing class associated to a new class. *(ExistingInNew)*

Suggestion rate Number of user prompts required over the number of evaluated documents. In the SemiAutomated mode the optimum is 0, while in the Manual mode the optimum is the ratio of the classes to be discovered over the evaluated documents ($\cong 10\%$)—of course, this index does not apply to the Automated mode.

3.1 Classification performance

In this section we will assess how varying α affects the classification error rate. We experimented on α values ranging from 1.5 to 4; the results are depicted in Figure 1.

Operation mode	New InExisting	Existing InWrong Exisiting	Existing InNew
Manual	2.6	0.9	0.0
SemiAutomated	4.7	0.6	4.6
Automated	8.1	4.7	6.1

Table 1: Classification error rates for each operation mode (%)

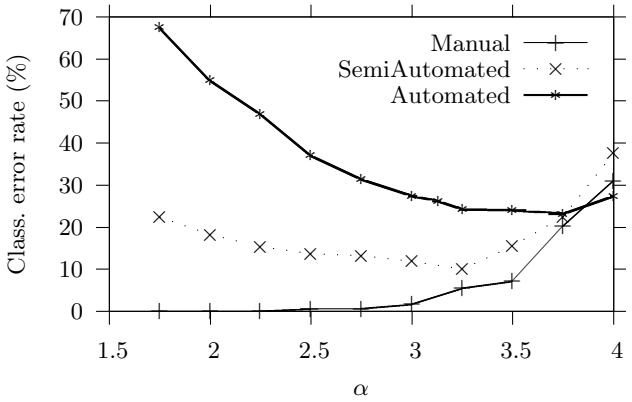

Figure 1: Comparison of the different proposed method variants in terms of classification error rate, while varying the threshold parameter α.

We experimented with several combinations of ϵ values and report in this section only the best results due to space constraints ($\epsilon = 1.3$ for the SemiAutomated mode and $\epsilon = 1.2$ for the Manual mode).

As expected, the Manual mode achieves the best results with an error rate lower than the other modes for almost all the values of α and scoring less than 2% when $\alpha \leq 3$. The Automated mode achieves a classification rate close to 77% ($\alpha = 3.75$). The SemiAutomated mode is more robust toward suboptimal choices of $\alpha = 3.25$ and achieves a classification rate close to 90%. Interestingly, the optimal calibrations in terms of α for the three modes are different.

Table 1 shows the rate of classification errors for each operation mode; it is interesting to note that for both the Automated and the SemiAutomated operation modes the errors are mainly imputable to the novelty detector, since more than 75% and 90% respectively of the classification errors are due to undetected new classes or documents of existing class marked as new. In Manual mode the errors are distributed more equally between the novelty detector (80%) and the classifier (20%).

3.2 Classification error rate vs Suggestion rate

In this section we investigate the trade-off between classification rate and amount of feedback required from operators.

Figure 2 plots the classification error rate versus the suggestion rate for both the Manual and SemiAutomated modes. The number of points is higher than those in the previous section, because here we plot all the α-ϵ combinations that we have tested. Dashed lines denote the Pareto front that can be constructed from these experiments.

The plot for the Manual mode includes a vertical line cor-

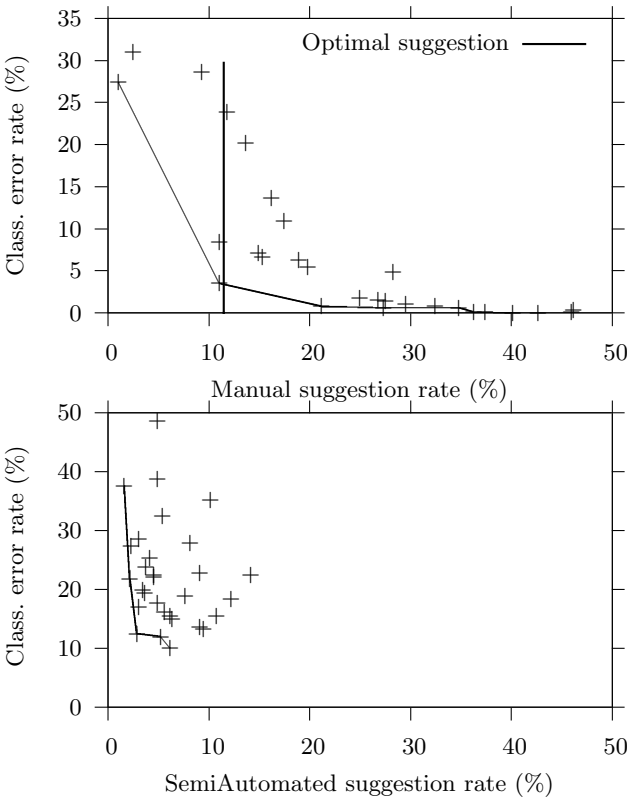

Figure 2: Suggestion rate vs classification error rate for the Manual and SemiAutomated modality; the thin line plots the Pareto front.

responding to the optimal suggestion rate for this mode. As pointed out earlier, this optimal rate is around 10% ($\frac{68-n}{548-n*k}$) since it is expected that the system will identify (and hence prompt the user) on all the new classes. When the suggestion rate is below that value (points at the left of the line), the system detected less new class than it should have had.

As expected, the Manual suggestion rate plot confirms that there is trade-off between quality of classification and amount of operator involvement: a low classification error rate leads to an elevate number of suggestions and vice versa. By looking at the Pareto front, the best solution is the one that obtains a low error rate (below 4%) while keeping the rate of user prompts at the acceptable rate of roughly 11%; this solution is obtained with $\alpha = 3.25$ and $\epsilon = 1.1$. This trade-off may be observed also in the plot for the SemiAutomated mode as well as in the comparison between the two plots: the best classification performance in SemiAutomated mode (error rate 10%) is obtained with a user suggestion rate around 6% ($\alpha = 3.25, \epsilon = 1.3$); in Manual mode, the corresponding indexes for the best solution are, as pointed out earlier, 4% and 11% respectively ($\alpha = 3.25, \epsilon = 1.1$).

In summary, our approach scores as well as the systems existing in literature [1] when operating in a fully unsupervised scenario despite being tested in a more challenging experimental setting. Moreover, we showed that our SemiAutomated operation mode leads to a high effectiveness while keeping the need for operator interventions at a low level.

4. REFERENCES

[1] 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*, volume 2, 2005.

[2] A. Bartoli, G. Davanzo, E. Medvet, and E. Sorio. Improving features extraction for supervised invoice classification. In *Proceedings of the 10th IASTED International Conference*, volume 674, page 401, 2010.

[3] Y. Belaid and A. Belaid. Morphological tagging approach in document analysis of invoices. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1 - Volume 01*, pages 469–472. IEEE Computer Society, 2004.

[4] F. Cesarini, E. Francesconi, M. Gori, and G. Soda. Analysis and understanding of multi-class invoices. *International Journal on Document Analysis and Recognition*, 6(2):102–114, Oct. 2003.

[5] J. Farkas. Neural networks and document classification. In *Electrical and Computer Engineering, 1993. Canadian Conference on*, pages 1 –4 vol.1, 14-17 1993.

[6] H. Hamza, Y. Belaid, A. Belaid, and B. Chaudhuri. Incremental classification of invoice documents. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1 –4, 8-11 2008.

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

[8] M. Huang. Multi-Class document layout classification using random chopping. 2007.

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

[10] E. Medvet, A. Bartoli, and G. Davanzo. A new probabilistic approach for information extraction from printed documents. *Submitted for publication*, 2010.

[11] J. Schrmann. *Pattern Classification*. John Wiley and Sons, 1996.