

Interactive document retrieval and classification

Ernest Valveny, Oriol Ramos, Joan Mas and Marçal Rossinyol

Abstract In this chapter we describe a system for document retrieval and classification following the interactive-predictive framework. In particular, the system addresses two different scenarios of document analysis: document classification based on visual appearance and logo detection. These two classical problems of document analysis are formulated following the interactive-predictive model, taking the user interaction into account to make easier the process of annotating and labelling the documents. A system implementing this model in a real scenario is presented and analyzed. This system also takes advantage of active learning techniques to speed up the task of labelling the documents.

1 Introduction

Huge amounts of documents are being stored currently as digital images at private and public organizations. However, for these raw digital images to be really useful, they need to be annotated with informative content. Document Image Analysis and Pattern Recognition techniques are at the heart of current solutions to this problem. However, when dealing with difficult unconstrained documents (see figure 1), standard solutions (for instance, commercial OCR products) are simply not usable since, in the vast majority of these documents, elements can by no means be isolated automatically. Given the high error rates involved in post-editing solutions, only semi-automatic or computer-assisted alternatives can be currently foreseen.

In this context, interactive tools emerge as a very appealing alternative to reduce the cost of labelling and annotating documents and, at the same time as a way of obtaining user feedback to improve the model for classification and retrieval. Hence, in this chapter we describe an interactive tool to annotate documents with semantic information, such as the category of the document or the location of relevant elements of the document which are difficult to automatically isolate. This tool follows

Computer Vision Center, Dept. Ciències Computació, Universitat Autònoma de Barcelona



Fig. 1 Examples of unconstrained documents.

an adaptation of the multimodal interactive-predictive approach (see figure 2) using adaptive learning techniques to reduce the human effort required to annotate these document images. The user can validate the initial labelling of the documents and, if necessary, edit this initial labelling by choosing among a set of alternatives. Using active learning methods, the system automatically proposes the optimal set of samples to validate and/or label. The information obtained by validation or edition of the labelling is used to update the model defined to annotate the documents. Once the labelling has been validated, this semantic information can be used for interactive retrieval of the documents stored in the database. This tool is used to annotate and retrieve difficult documents, specifically unstructured documents or documents with heterogeneous contents (containing printed and handwritten text, graphics, symbols, etc) such as administrative or ancient documents (see figure 1).

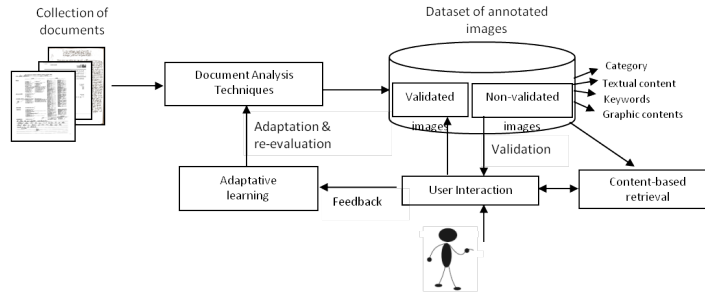


Fig. 2 Adaptation of the interactive-predictive framework in our prototype.

We focus on two particular problems of document classification and retrieval where the use of an interactive tool can be especially appropriate: appearance-based document classification and logo-based retrieval. In both cases, we use state-of-the-art techniques to obtain the initial labelling of the documents. These techniques are described in section 2. Then, in section 3 we adapt them to be used within interactive-multimodal framework. In section 4 we explain the implementation details of the prototype and show its main functionalities and some results obtained with its application to a set of documents. Finally, in section 5 we draw the main conclusion of the work presented.

2 Basic document classification and retrieval

2.1 Document classification

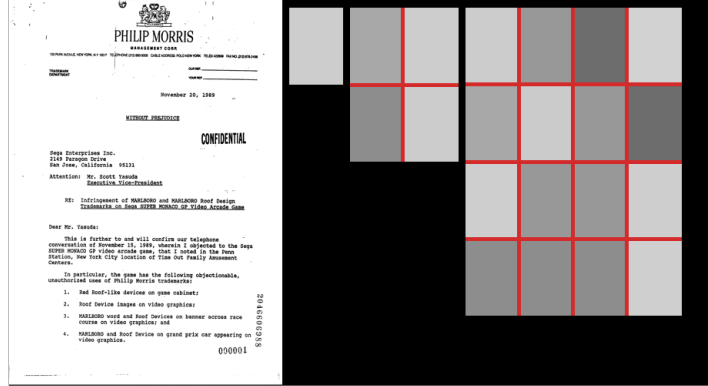


Fig. 3 Example of document image description by means of a pyramid of pixel densities.

Document classification is performed by means of global visual appearance descriptors computed on the whole document image. We have used three different state-of-the-art descriptors that permit to capture document appearance using different techniques. In the following we, first give a brief description of the three descriptors and then, we explain the classification framework.

2.1.1 Visual descriptors

Blurred Shape Model (BSM)

The BSM descriptor was proposed by Escalera et al. in [1] for the recognition of segmented graphical symbols. The BSM descriptor is a zoning-based method [6]. A regular grid of cells with a fixed number of rows and columns is overlaid on the document and then the grid is scanned from left to right and top to bottom where pixel density features are extracted from each cell in order to form the feature vector. In the original formulation of the BSM descriptor, each cell encodes pixel density from the foreground pixels that fall within it but also from the pixels assigned to the neighboring cells. Thus, each pixel contributes to a density measure of its cell, but also to its neighbors in order to achieve robustness to local variations in the document. This contribution is weighted in terms of the distance between the pixel and the centroid of the cell. The output descriptor is a histogram where each component corresponds to the amount of pixels in the context of the cell. Finally, the resulting histogram is $L1$ -normalized.

Pyramid of pixel densities (PYR)

The PYR encodes pixel densities at different scales. It is a hierarchical zoning descriptor introduced in [3]. In order to remove small details and noise from the incoming images, a Gaussian smoothing operator is first used to blur the images before computing the visual descriptor. Each document image is recursively split into rectangular regions forming a pyramid. At each new partition level l , 4^{l-1} dimensions are added to the feature vector. In each of the regions the pixel density computed as the average intensity value is stored in the corresponding position of the feature vector. We can see an example of the first levels of the pyramid in Figure 3. In our experimental setup, we use four levels, yielding to an 85-dimensional visual descriptor. The visual feature vectors are finally $L2$ -normalized.

Runlength descriptor (RLD)

The Runlength descriptor[2] is based on a histogram of the length of the runs of consecutive pixels in the four main directions (horizontal, vertical and diagonals). The length of the runs is quantized in different intervals and thus, every bin of the histograms corresponds to one of these quantization intervals. The histograms in the four directions are concatenated to obtain the final descriptor. In addition, the image can be split into several regions at different levels of resolution (for instance, 2×2 and 4×4). In this case, a histogram for each region is computed and all of them are concatenated to obtain a multi-scale document representation that can capture information about the structure or the layout of the document.

2.1.2 Classifier

The process of classifying a document can be defined in the following way. Let I be a random variable describing the set of visual features (BSM, PYR or RLD) extracted from the document image. Let C be a random variable describing the set of classes. Then, the probability of classifying the document image as belonging to a certain class can be defined as $P(C|I)$. Applying Bayes' rule we get the following:

$$P(C|I) = \frac{P(I|C) \cdot P(C)}{P(I)} \quad (1)$$

where $P(I)$ is assumed to be equal for all images and can be ignored. $P(I|C)$ stands for the probability of the observed visual features given that the document belongs to a certain class. $P(C)$ can be seen as the a priori probability of every class. Initially, this probability is assumed to be equal for all classes. However, in the interactive scenario that will be described in the next section, it will change according to the user interaction.

We have explored two different ways of obtaining $P(I|C)$. The first one, in the framework of a Bayesian classifier, assumes that this probability follows a normal

distribution with mean and variance computed from the training samples assigned to every class. The second one uses a k -NN classifier to classify the documents and obtain this probability. There are basically two ways of returning the probability of a class using a k -NN classifier, taking into account the classes of the k -th nearest samples to the unknown document: (a) using the relative frequencies of classes and (b) weighting each element by the inverse of its distance to the unknown element and normalizing to ensure a total mass of 1. Both methods asymptotically converges to the true probability when k increase but (b) is more robust and this is the option we have used.

In any of both classification schemes, an accurate estimation of the probability depends on having good labeled datasets. The construction of them are done taking the user feedback after a first unsupervised document classification. The samples validated by the user with the interactive framework described in the next section will be included in the training set and will be used to modify the parameters of the probability distribution used to compute $P(I|C)$.

2.2 Logo detection

Logo detection consists in finding possible locations of a given query logo in the set of documents. In order to spot the position of logos appearing within document images we use a sliding window framework together with the blurred shape model (BSM) descriptor introduced in the previous section, but modified to take into account that we are working with non-segmented images. In the original formulation of the BSM descriptor, pixel density was computed over a regular $n \times n$ grid, assuming that the shapes to compare have been previously segmented. In our case we would like to locate a logo within a cluttered document image. Thus, we reformulate the BSM descriptor by forcing the spatial bins to have a fixed size (100x100 pixels in our experimental setup). Images having different sizes will result in feature vectors of different lengths. By using this reformulation of the BSM descriptor, the chosen size of the buckets will define the level of blurring and subsequently the information reduction for both the logos and the documents.

In order to locate a logo within a document image we use a sliding-window approach computed as a normalized two-dimensional cross-correlation between the BSM description of the model logo and the BSM description of the complete document. We use the two-dimensional cross-correlation proposed in [4] and computed as follows. Let t be the sought template represented by the BSM descriptor of the model logo and $f(x,y)$ the BSM description of the whole document image. The mean values of the template and a particular zone of the document descriptor are formulated as \bar{t} and $\bar{f}_{u,v}$ respectively. The correlation coefficient is then computed as

$$\gamma(u,v) = \frac{\sum_{x,y} [f(x,y) - \bar{f}_{u,v}][t(x-u, y-v) - \bar{t}]}{\left\{ \sum_{x,y} [f(x,y) - \bar{f}_{u,v}]^2 \sum_{x,y} [t(x-u, y-v) - \bar{t}]^2 \right\}^{0.5}} \quad (2)$$

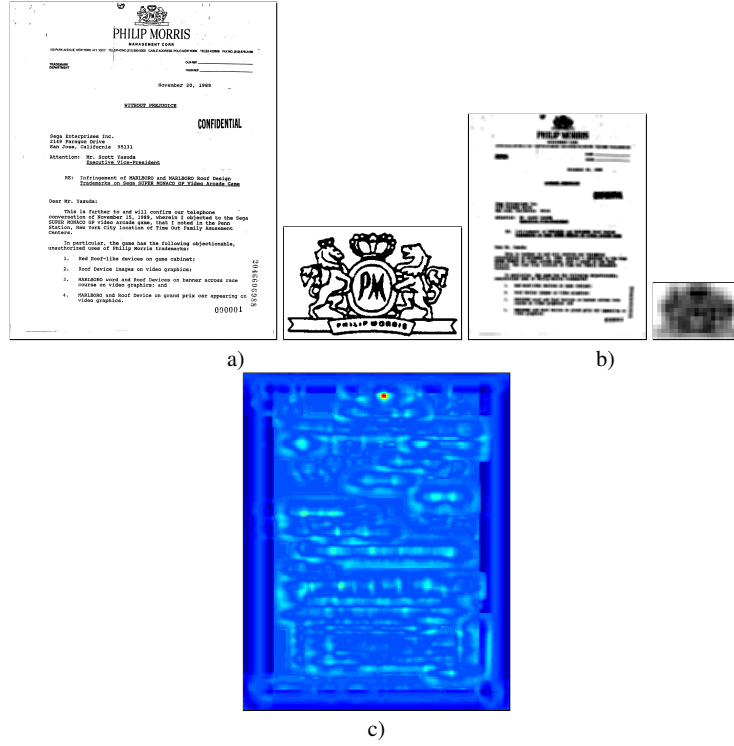


Fig. 4 Logo detection example using a sliding window over BSM descriptors. a) Original document and sought logo, b) BSM descriptors of the document and the logo, c) obtained correlation coefficients by the normalized two-dimensional cross-correlation.

As the result of the cross correlation between the BSM models and the BSM descriptor from the document, a peak should be formed in the correlation coefficient image in the location (u, v) where there is a high probability to find a something similar to the given logo. We can see in Figure 4 an example of the obtained output given a document and a logo to search for.

Using this basic set of techniques, the process of detecting a logo in an image can be defined in the following way. Let I be a random variable describing the set of features (BSM descriptor) extracted at every position of the sliding window over the image. Let X be a random variable standing for a given position of the sliding window (x and y coordinates of the bounding box). Let L be a random variable describing the possible set of logos. Then, the probability of detecting a logo at a certain location of the image, given the set of features extracted from the image can be defined as $P(X, L|I)$. Applying Bayes' rule we get the following:

$$P(X, L|I) = \frac{P(I|L, X) \cdot P(L, X)}{P(I)} \quad (3)$$

where $P(I)$ is assumed to be equal for all images and can be ignored. $P(I|L, X)$ stands for the probability of the observed image features given that we are trying to find a given logo at a certain location of the image. This probability can be modelled by the score of the cross correlation between the model of the logo and the BSM descriptor at every location of the image, as described before. $P(L, X)$ can be seen as the a priori probability of every logo and location. This expression can be further developed in the following way:

$$P(X, L) = P(X|L) \cdot P(L) \quad (4)$$

In a retrieval scenario where we are searching for a specific given logo, $P(L)$, the a priori probability of every logo, can be ignored and therefore, $P(X, L)$ only depends on the probability $P(X|L)$ that stands for the a priori probability of finding a given logo at a certain location of the image. This probability can be estimated from a set of learning documents, just by counting the frequency of appearance of a given logo at every location of the image. User interaction, as introduced in the next section, will permit to modify this a priori probability. The new logo detections validated by the user will be used to update $P(X|L)$ accordingly.

3 Interactive document retrieval

In this section we will explain how the document classification and logo detection tasks described in the previous section can be adapted to take into account the interactive-predictive framework introduced in [5]. In particular, we will consider the history of previous user interaction steps in order to help the system when taking a decision in the current step.

3.1 Interactive document classification

For document classification, user interaction will permit to validate or reject the last hypothesis made by the system concerning the class of the document. Thus, equation 1 must be re-formulated to take this interaction into account. In particular, the class of a document will not only depend on the visual features extracted from the image, I , but also on two new random variables, c' standing for the last hypothesis made by the system concerning the class of the document, and d , corresponding to the decoding of the user interaction, that is, accepting or rejecting the last hypothesis c' . Putting all together, the class of a document will be determined according to the following expression:

$$c = \arg \max_c P(C|I, c', d) \quad (5)$$

This equation can be expanded in a similar way as we did in equation 1 and then, we get that the class c assigned to a document is obtained in through this expression:

$$c = \arg \max_c P(C|I, c', d) = \arg \max_c P(I|c) \cdot P(C|c', d) \quad (6)$$

Thus, the class c assigned to a given document depends on two terms. The first one, $P(I|C)$ the same as in equation 1, only considers the visual appearance of the document and thus, it is constant. The second one, $P(C|c', d)$, accounts for the user interaction and therefore, it will change at each interaction step. It permits to update the a priori probability of each class according to the user feedback d and the previous hypothesis made by the system C' . The user feedback will consist in validating or rejecting the hypothesis C' . We will assume that d takes the value 0 if the previous hypothesis is rejected and 1 if it is validated. Under these assumptions the update of the a priori probability can be expressed in the following way:

$$P(C|c', d) = \left\{ \begin{array}{l} d = 1 \Rightarrow \left\{ \begin{array}{l} 1 \quad c = c' \\ 0 \quad c \neq c' \end{array} \right\} \\ d = 0 \Rightarrow \left\{ \begin{array}{l} 0 \quad c = c' \\ \frac{P(c)}{\sum_{c_i \neq c'} P(c_i)} \quad c \neq c' \end{array} \right\} \end{array} \right\} \quad (7)$$

being $P(c)$ the a priori probability of every class at the previous iteration step. If the previous hypothesis is validated ($d = 1$), then the class corresponding to the last hypothesis c' is assigned probability 1 while all other classes are assigned probability 0. However, if the previous hypothesis is rejected ($d = 0$), the class corresponding to c' is assigned probability 0, while its previous probability value is equally re-distributed among all the remaining classes.

3.2 Interactive logo detection

In a similar way as in document classification, user interaction can be included for logo detection in the form of validation or rejection of the hypothesis that a given logo appears in an image. Then, the probability of finding a logo at a certain location of an image (defined in equation 3) is modified to include a new random variable d accounting for the user feedback. In this way, this probability can be expressed as $P(X, L|I, d)$. User feedback we will consist in validating or rejecting the presence of a given logo in the image. Thus, d will consist of a sequence of pairs (l, y) where l will be any of the possible logos and y will take the value 0 if the logo does not appear in the image and 1 if it appears.

The probability $P(X, L|I, d)$ of finding a logo in the image can be expanded in a similar way as in equations 3 and 4 leading to the following (figure 5):

$$P(X, L, |I, d) = P(I|X, L) \cdot P(X|L) \cdot P(L|d) \quad (8)$$

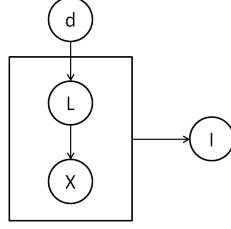


Fig. 5 Graphical model corresponding to $P(X, L|I, d)$.

$P(L|d)$ is the probability of finding a logo in the document giving the history of user feedback d . Initially all logos are assigned the same probability. As user provides feedback this probability is updated in a way that for all logos l for which a pair $(l, 1)$ appears in the sequence d their probability $P(L|d)$ is set to 1, while for all logos for which a pair $(l, 0)$ appears in d , the probability is set to 0.

3.3 Interactive class-based logo detection

We can combine document classification and logo retrieval to propose a new scenario where we can use class information to help in the detection of logos. We can assume that logos appearing in an image will be different depending on the class of the document. Thus, we can relate logos with classes of documents through a new probability distribution, $P(L|C)$, that assigns to every logo a certain probability of appearing in documents of a given class.

Then, the probability of defining a logo in an image (equation 3) can be modified to include this new dependency in the following way:

$$P(X, L|I, C) = P(X|L, C) \cdot P(L|C) \cdot P(C) \quad (9)$$

Up to this point the system is fully automatic: given an image of the document, first, the class of the document is determined and depending on this, the probability of finding each logo in the image can be computed using the previous expression and, therefore, possible location of logos can be retrieved.

Going one step further, the interactive-predictive framework can also be used to take into account the user interaction for document classification. In this way, as described in section 3.1, given the initial classification of a document, the user can validate or reject this hypothesis. Then, the system uses this feedback to re-evaluate the classification of the document and, as a result of this evaluation, the probability of each logo is also modified and thus, a new set of possible logo detections is retrieved. All this process can be expressed in terms of probabilities combining equations 6 and 9 in the following way (see figure 6:

$$P(X, L | I_L, I_C, C, C', d) = P(I_L | X, L) \cdot P(X | L, C) \cdot P(L | C) \cdot P(I_C | C) \cdot P(C | C', d) \quad (10)$$

Note that in this expression we distinguish between image features used to find logos, I_L , and image features used for document classification, I_C . The whole process is illustrated in figure 7.

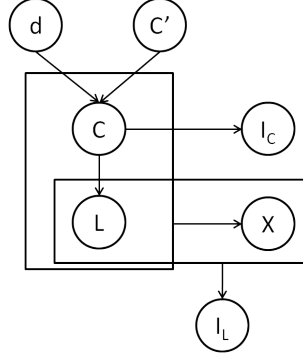


Fig. 6 Graphical model corresponding to the probability of finding a logo using class information and user interaction.

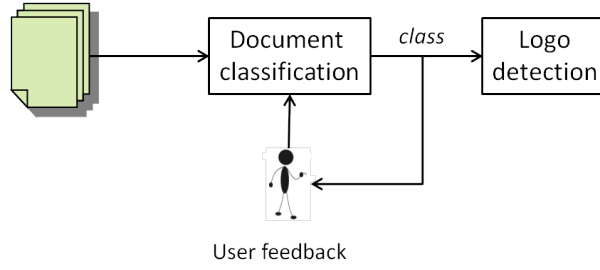


Fig. 7 Interactive process for detecting logos taking into account user interaction for document classification.

4 Prototype

The prototype in its current release includes two document analysis applications: document classification and logo detection, which permits to annotate documents with different kinds of semantic information¹. We have uploaded several collections

¹ available at <http://dag.uab.es/documents> (write miprev as username and password to login in this demo)

of documents and we have also prepared a set of configurations for demonstration purposes.

We have included the prototype functionalities in a web-based application structured in three different layers: (a) a graphical front-end taking care of displaying information and interaction with the user, (b) a back-end where the user sets the application configuration and (c) a set of tools for document analysis and learning.


















































The front-end of the prototype is in charge of the user interaction and displaying results. In the following we will illustrate how it works for the case of document classification. The process for logo detection is very similar, just changing classes of documents by logo detections.

The general view always shows a list of all the classes of documents plus an additional class *Not Assigned* for those documents not classified in any of the previous classes. As a first step, the system performs an unsupervised classification of all the documents using the *K*-means algorithm in order to get a first assignment of documents to classes. The user only has to give a semantic name to the created classes. Then, user interaction is done in a simple way using the mouse and the keyboard. The user selects one of the predefined set of document categories and validates or rejects the documents assigned to it by just clicking on the green ticks and cross red buttons that appear on the right side of a thumbnail image of each document (see figure 8 (a) and (b)). Alternatively, the user can also globally validate or reject all the documents assigned to the class by clicking on the green tick button or the red cross underneath the class label. Additionally, the user can also select the class of documents not assigned to any class because their probability were very low. In this case, the user can directly assign each of them to the correct document class, instead of validating or rejecting them (see figure 8 (c)).


At every interaction step, samples pending of validation are shown to the user sorted using an active learning strategy based on uncertainty sampling. The samples with maximum entropy are selected as the first ones to be validated. After each interaction step, the interactive-predictive framework described in section 3 is used to update all the probability distributions concerning the model. Accordingly, the system modifies the classification hypothesis for all the documents pending of validation or still not assigned to any class.

The back-end view of the prototype controls the basic user functionalities for managing document collections. There, the user is able to create, manage and organize collections of documents. He is able to select the collection or collections to be used at any moment for a given application. In figure 9 we can see the check-list of all available datasets and the first labeled classes. We have selected the public and standard NIST dataset of forms for document classification and the Tobacco logo dataset for logo detection.

Once the user has selected the datasets and defined the semantic classes partitioning the dataset, the user can also select the configuration of the basic techniques (visual descriptor and base classifier) that will be used by the system, as it can be seen in figure 10. If the user clicks over the *Add configuration* button a new window, with a list of available descriptors and classifiers, appears on the top of the application. By default, each descriptor and classifier works with default parameters that


form 5 [30]		image: 7306		image: 7202
form 6 [5]		Not verified <input checked="" type="checkbox"/> <input type="checkbox"/>		Not verified <input checked="" type="checkbox"/> <input type="checkbox"/>
form 7 [72]				
form 8 [38]				
form 9 [36]				
form 10 [10]				
form 11 [23]				
form 12 [0]				
form 13 [0]				
form 14 [0]				
form 15 [0]				
form 16 [0]				
form 17 [0]				
form 18 [0]				
form 19 [17] 14 to validate		Not verified <input checked="" type="checkbox"/> <input type="checkbox"/>		Not verified <input checked="" type="checkbox"/> <input type="checkbox"/>

(a)



DAG Applications

Document Classification















Document Classification









Config 1

Edit configuration

8 category/ies have no documents, select at least one document to each class to could perform an automatic classification

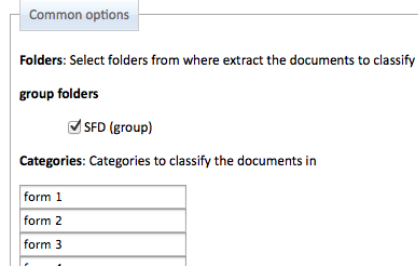
form 1 (55)	 <div>image: 7554</div> 	 <div>image: 7423</div> 
form 2 (10)		
form 3 (18)		
form 4 (2)		
form 5 (30)	 <div>image: 7416</div> 	 <div>image: 7410</div> 
form 6 (5)		
form 7 (72)		
form 8 (38)		
form 9 (36)	 <div>image: 7403</div> 	 <div>image: 7395</div> 
form 10 (10)		
form 11 (23)		
form 12 (0)		

(b)

form 1 (55)			
form 2 (0)			
form 3 (18)			
form 4 (2) 1 to validate		image: 7496 Assign to category <input type="text" value="form 1"/> <input type="button" value="Assign"/>	
form 5 (0)			
form 6 (5) 4 to validate		image: 7181 Assign to category <input type="text" value="form 1"/> <input type="button" value="Assign"/>	
form 7 (0)			
form 8 (0)			
form 9 (0)			
form 10 (10) 9 to validate		image: 7185 Assign to category <input type="text" value="form 1"/> <input type="button" value="Assign"/>	
form 11 (0)			
form 12 (0)		image: 7126 Assign to category <input type="text" value="form 1"/> <input type="button" value="Assign"/>	

(c)

Fig. 8 Front-end of the prototype. On the left side, the list of classes of documents with the number of examples validated (in brackets). In the main frame, (a) a list of documents of one class to be validated. In (b) a list of validated documents for a given class. In (c) the set of documents not assigned to any class.



Common options

Folders: Select folders from where extract the documents to classify

group folders

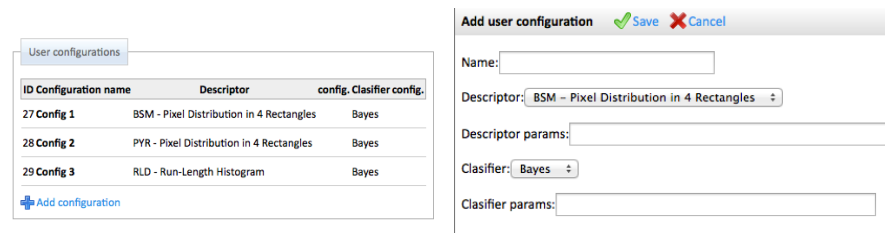
☒ SFD (group)

Categories: Categories to classify the documents in

form 1
form 2
form 3

Fig. 9 Documents configuration menu. The user can check the list of datasets available (SFD in this demo) and define the document semantic classes (labeled as *form 1* to *form 20*).

can be changed by means of *Descriptor params* text box and *Classifier params* text box, respectively.



User configurations

ID	Configuration name	Descriptor	config. Classifier config.
27	Config 1	BSM - Pixel Distribution in 4 Rectangles	Bayes
28	Config 2	PYR - Pixel Distribution in 4 Rectangles	Bayes
29	Config 3	RLD - Run-Length Histogram	Bayes

+ Add configuration

Add user configuration ✓ Save ✗ Cancel

Name:

Descriptor: BSM - Pixel Distribution in 4 Rectangles

Descriptor params:

Classifier: Bayes

Classifier params:

Fig. 10 User configurations menu. At each configuration the user can choose the type of descriptor (BSM, PYR, RLD) and the classifier.

4.1 Experiments

In order to evaluate the performance of the proposed approach we have simulated a series of user interaction steps in the task of document classification. We have used a subset of 3200 images from the NIST dataset of forms which is composed of 20 different classes. We have divided the set of instances in two blocks: training and test. The training set is composed of 200 images randomly selected, not equally distributed among the 20 classes. The test set is composed of the remaining 3000 images. We have used a configuration composed of the BSM descriptor and the k -NN classifier. We have followed this protocol: first, one instance of each class is selected to train the classifier. Then, at each new step, all the images in the training set that are not classified yet are separated in clusters using the K -means algorithm. For each cluster we select the most uncertain sample in terms of entropy and we re-train the model adding this new sample from every cluster according to the model

explained in section 3.1. At each step the classification accuracy is determined. Results are shown in table 1. The first column shows the number of labeled samples used at every step to train the classifier. The second column shows the classification accuracy after each step. It can be seen that the accuracy improves at every step. It is worth noting that in this dataset the state-of-the-art in classification is very close to 100%. Thus, we start with already high accuracy rates and very fast (with a few samples and interaction steps) we converge to rates very close to the state-of-the-art.

N of instances	Accuracy
27	99.1
36	99.2
47	99.0
56	99.4
66	99.5
75	99.6
84	99.5
94	99.6
106	99.6

Table 1 Classification accuracy after several steps of user interaction

5 Conclusions

In this chapter we have shown the adaptation of classical document analysis problems, such as document classification and logo retrieval to the interactive predictive model that permits to take advantage of the user interaction to improve retrieval results and re-train the models. We have seen how, in this scenario, both problems, document classification and logo retrieval can be easily related so that logo retrieval can take advantage of the class information obtained. We have also shown a practical implementation of this framework and some results that confirm that the user interaction can speed up the training process. However, a more exhaustive evaluation should be conducted in the future to establish the real power of combining both tasks under the same framework.

References

1. S. Escalera, A. Fornés, O. Pujol, P. Radeva, G. Sánchez, and J. Lladós. Blurred shape model for binary and grey-level symbol recognition. *Pattern Recognition Letters*, 30(15):1424–1433, 2009.
2. Albert Gordo. *Document Image Representation, Classification and Retrieval in Large-Scale Domains*. PhD thesis, Universitat Autònoma de Barcelona, 2012.

3. P. Héroux, S. Diana, A. Ribert, and E. Trupin. Classification method study for automatic form class identification. In *Proceedings of the Fourteenth International Conference on Pattern Recognition*, pages 926–928, 1998.
4. J.P. Lewis. Fast normalized cross-correlation. In *Vision Interface*, volume 10, pages 120–123, 1995.
5. A. Toselli, E. Vidal, and F. Casacuberta. *Multimodal Interactive Pattern Recognition and Applications*. Springer, 2011.
6. D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, 2004.