Picture Extraction from Digitized Historical Manuscripts

Costantino Grana, Daniele Borghesani, Rita Cucchiara
University of Modena and Reggio Emilia
Via Vignolese 905/b
Modena, Italy
{costantino.grana,daniele.borghesani,rita.cucchiara}@unimore.it

ABSTRACT

In this work we propose a system for automatic document segmentation to extract graphical elements from historical manuscripts and then to identify significant pictures from them, removing floral and abstract decorations. The system performs a block based analysis by means of color and texture features. The Gradient Spatial Dependency Matrix, a new texture operator particularly effective for this task, is proposed. The feature vectors are processed by an embedding procedure which allows increased performance in later SVM classification. Results for both feature extraction and embedding based classification are reported, supporting the effectiveness of the proposal.

Categories and Subject Descriptors

I.4 [**Image Processing and Computer Vision**]: Feature Measurement; I.5 [**Pattern Recognition**]: Design Methodology

1. INTRODUCTION

The automatic analysis of the huge amount of paper documents represents a concrete and attractive possibility in terms of data retrieval (text and images) and data presentation. This activity becomes much more important when dealing with artistic or historical documents that cannot be available to the public, due to their value and delicacy. Computer science can fill the gap between people and all these precious libraries: digital versions of the artistic works can be publicly accessible, both locally at the museum owning the original version and remotely. In this manner, users —either experts, tourists or people keen on art— can explore more comprehensively the document, choosing their own personal way to browse and enjoy it.

Italy, in particular, has a huge collection of illuminated manuscripts, but they are not freely accessible to the public. These masterpieces contain thousands of valuable illuminated illustrations: different mythological and real animals, biblical episodes, court life illustrations, and some of them even testify the first attempts in exploring prospective for landscapes. They are normally not evaluated as they are, but only as a part of a global page decoration. Instead they are singularly pictorial masterpieces, but manual segmentation and annotation for all of them is too long to be feasible. The accomplishment of the the same task with an automatic procedure is challenging: the arrangement of these pictures in the pages is heterogeneous, and the framing is not fixed (a lot of different frames, with different boundaries, sometimes highly decorated boundaries too, etc.).

In this work we propose a system for the manuscript layout segmentation and the automatic extraction of valuable pictures from the decorated pages. The application is particularly innovative since for the first time an attempt to distinguish between valuable pictures and decoration is proposed by means of visual cues. As well, we present two novel aspects in the pattern recognition process: a texture feature, specifically aimed at detecting the correlations between the gradient directions and a novel clustering-based embedding process which applied to Support Vector Machines allows to reduce the training requirements both in terms of number of samples and of computational time without impacting on the classification performance.

The paper is structured as follows: in the next section related work is discussed, Section 3 presents the system architecture, while Section 4 describes our proposal for layout segmentation, feature extraction and classification. Finally, results are reported in Section 5.

2. RELATED WORK

Document analysis is one of the most explored fields in image analysis, and a plethora of works has been produced dealing with different aspects of the segmentation of the document. The seminal work of Nagy [22] gives the perfect overview of the techniques proposed until some years ago for text segmentation, OCR and background removal. The most faced problem regarded text, either printed or handwritten, but approaches dealing also with pictures segmentation have been proposed. In [2], Chen et al. provide a general partition of the classification approaches proposed so far. In particular, according to their taxonomy, the page can be classified using image features, physical layout features, logical structure features and eventually textual features. Several works tackle the physical and logical segmentation of the page, exploiting different rules on the page structure, such as geometric constraints over the layout. A different strategy is to compute specific descriptors followed by classification: an example is provided by Diligenti et al. in [3]

CIVR '09, July 8-10, 2009 Santorini, GR

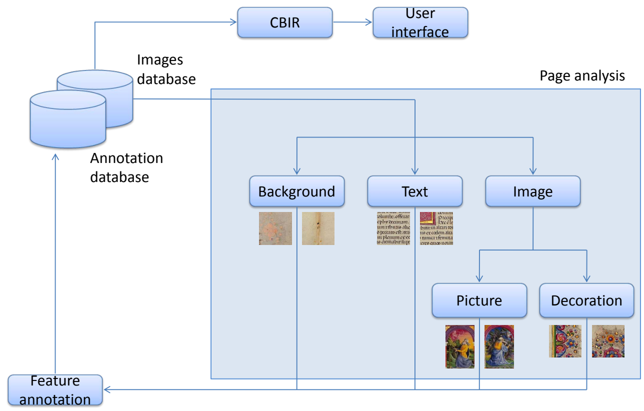

Figure 1: Overall schema of the system

which exploited Hidden Tree Markov models. The majority of these works employs an XY-tree based representation, and graph or template matching approaches in order to perform classification. Our work belongs to Chen's first class, based on image features.

Texture features based on frequencies and orientations have been used in [15] to extract and compare elements of high semantic level without expressing any hypothesis about the physical or logical structure of the analyzed documents, exploiting a page analysis by blocks. Nicolas et al. in [23] proposed a 2D conditional random field model to perform the same task. Hu et al. [12] use interval encoding features to capture elements of spatial layout, modeled with HMMs. Using grey level images, histogram projection is used in [21] to distinguish text from images, while a more complex approach based on effective thresholding, morphology and connected component analysis has been used in [17]. A multiscale approach has also been proposed in [5] by Fataicha et al. We propose a mixed approach based on both texture and morphology (as [17]) for text and image segmentation, while we define a new method to distinguish between picture and decoration.

All these techniques are the basis of complete systems for the management of digital libraries, which provide tools for semantic annotation, classification and retrieval. For the implementation over a large collection of digital documents, the accuracy of the analysis and the computational effort required are both significant. Until now, the activities on DL of illuminated manuscripts have been accomplished by manual annotation and indexing, but some interesting systems deserve to be mentioned. The AGORA [25] software performs a map of the foreground and the background and consequently propose a user interface to assist in the creation of an XML annotation of the page components. The Madonne system [24] is another French initiative to use document image analysis techniques for the purpose of preserving and exploiting cultural heritage documents.

In [20] Le Bourgeois et al. highlighted some problems with acquisition and compression, then authors gave a brief subdivision of documents classes, and for each of them a proposal of analysis. They distinguished between medieval manuscripts, early printed documents of the Renaissance, authors manuscripts from 18th to 19th and finally administrative documents of the 18th - 20th. In this work, the authors performed color depth reduction, then a layout segmentation that is followed by the main body segmentation using text zones location. The feature analysis step uses some color, shape and geometrical features, and a PCA is performed in order to reduce the dimensionality. Finally the classification stage implements a K-NN approach. Their system has been finalized in the DEBORA project [19], which consists of a complete system specifically designed for the analysis of Renaissance digital libraries. In this paper we are interested in the first class identified by [20], that is composed of illuminated manuscripts.

3. SYSTEM ARCHITECTURE

The approaches for image segmentation and classification presented in this paper have been implemented in an integrated system for document analysis and remote access, including querying and browsing functionalities. The system elements are reported in Fig. 1. Two different databases have been created in order to store images and annotations. The former stores the high resolution digitized manuscripts, while the latter contains both the automatically extracted knowledge and the historical comments added by experts.

The retrieval subsystem shares the canonical structure of CBIR systems. This is the basis for the user interface module, that integrates the visual and keyword-based search engine to propose an innovative browsing experience to the user. The web interface allows to select a manuscript, and for each page the automatic layout segmentation is provided distinguishing between background, text, and images. Different viewing modalities are provided for each part; for example it is possible to immediately get a larger view of pictures, as in Fig. 2.

An offline page analysis module process the stored images and for each of them detects text and images. Then it distinguishes pictures within the decorations and extracts them separately. The details of these steps are fully described in the following sections. After the segmentation, the areas are saved into the annotation database and a feature extraction stage is performed over the picture areas, to allow CBIR

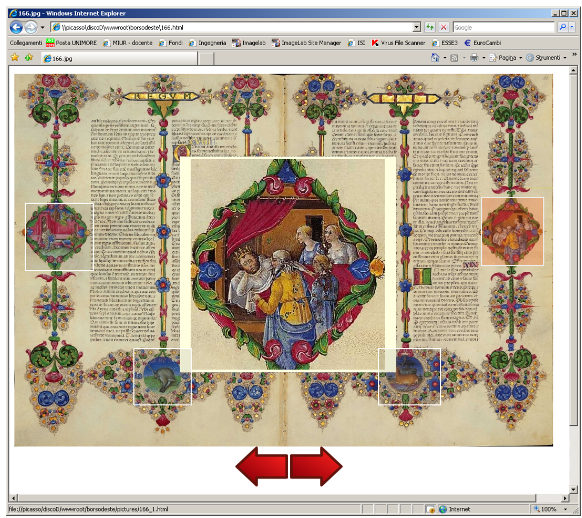

Figure 2: Screenshot of the web interface, showing a zoomed view of a selected picture.

functions such as similarity-based retrieval on their visual appearance.

The text detection module integrated in the page analyzer is based on the approach reported in our previous work [6]. Briefly, we use a two-dimensional autocorrelation matrix, since textual areas have a pronounced horizontal orientation that heavily differs both from background and decoration blocks. Given the autocorrelation matrix, the sum of all the pixels along each direction is computed to form a polar representation of the autocorrelation matrix, called *directional histogram*. This polar distribution is modeled using a mixture of two Von Mises distributions, since the standard Gaussian distributions are inappropriate to model angular datasets. SVMs are then used for learning and classification. The text areas are then also stored in the annotation database, in order to allow the application of OCR functions, or visual keyword spotting [18]. With respect to our previous work [6], here we develop a much more effective approach for picture vs. decoration segmentation, which doesn't relay on the connected components analysis, and also employs texture features.

4. PICTURE EXTRACTION

Miniature illustrations detection begins with a preprocessing stage to distinguish between background, text, and images. The result of the image extraction is a binary mask containing both pictures and decorations. Since morphological or pixel level segmentation are not enough to separate them, a block based analysis is performed and a feature vector is extracted for each block. Finally a SVM is used to classify and separate them. Examples of original digitized pages and the final output are shown in Fig. 5.

4.1 Image areas extraction

Since the background is much lighter then the other page elements and at the same time it covers a large area, it can be detected binarizing the image with automatic thresholding, using the classical Otsu algorithm (Fig. 3.b). This technique proved to be sufficiently robust to remove the paper background, since the digitalization process is very accurate, and the chromatic range of the spoiling is limited. Other techniques specifically proposed for printed documents, such as Iterative Global Thresholding [16], are often too aggressive on the thin decoration borders, causing too much shrinking on the detected areas.

The connected components of the image are then labeled (Fig. 3.c) and their area is computed in order to extract only large ones, compatibly with the smallest accepted size for a blob (fixed at the double of the height of a single text line)(Fig. 3.d). The contour of each blob is then followed and then filled. Sometimes this procedure extracts too large regions, for example when the external decoration is completely closed, so a second pass is performed to remove excessive filling. The filled pixels (originally background) are labeled and large filled components are restored. The threshold for this task is empirically fixed to half the size of a text column. A final step cares about the removal of overlapping blobs, in order to let a unique blob to describe a specific region of the image. In most cases, the small object removal erases the text areas together with binarization artifacts. Anyway the text masks extracted from the specific module, are employed and combined to remove any leftover. The final result (Fig. 3.e) is composed only of pictures and decorations, but no distinction can be provided at this stage.

4.2 Block level features

The image areas, as identified by the preprocessing output mask, are analyzed at block level, using a sliding window. The window size has been empirically set depending on the image resolution; in our experiments it was set to 200×200 pixels for images of 3894×2792 pixels. To ensure an effective coverage of the images, the window is moved so to obtain an overlap of 80% of its area between each step. For each block, a set of color and texture features is extracted; in particular we adopt both *RGB Histogram* and *Enhanced HSV Histogram* as color features, and we propose a new texture descriptor named *Gradient Spatial Dependency Matrix (GSDM)*.

RGB Histogram. A basic 3D color histogram on the RGB components of the image is computed. Each component is quantized to 8 values, resulting in a 512-bin histogram. Each bin of the resulting histogram is then normalized so that they add up to one.

Enhanced HSV Histogram. The idea of this feature is to separately account the chromatic and achromatic contribution of pixels. To this aim, 4 bins are added to the standard MPEG-7 HSV histogram, resulting in a 260-bins descriptor that proved to be more robust to bad quality or poorly saturated images [8]. This representation provides an advantage with respect to the standard HSV histogram definition because images have been depicted by hand, so they do not have photographic quality, despite of their high resolution digitalization.

GSDM. This feature is inspired to the well known Haralick's grey level co-occurrence matrix (GLCM) [9], which provides a representation of the spatial distribution of greyscale pixels of the image. Unlike GLCM, we provide this new representation, which accounts for the spatial distribution of gradients within the image.

The original image I is convolved with a Gaussian filter with $\sigma = 1$. The filtered image I_{gauss} is then used to compute the horizontal and the vertical gradients image using central differences.

$$\begin{array}{l} G_x(x,y) = I_{gauss}(x+1,y) - I_{gauss}(x-1,y) \\ G_y(x,y) = I_{gauss}(x,y+1) - I_{gauss}(x,y-1) \end{array} \quad (1)$$

Gradient images are used to compute the module and the direction for each pixel $\mathbf{p}$:

$$M(\mathbf{p}) = \sqrt{G_x(\mathbf{p})^2 + G_y(\mathbf{p})^2} \quad (2)$$

$$D(\mathbf{p}) = \begin{cases} \frac{\pi}{2}, & \text{if } G_x(\mathbf{p}) \neq 0 \\ \left(\tan^{-1}\frac{G_y(\mathbf{p})}{G_x(\mathbf{p})} + \pi\right) \bmod \pi, & \text{otherwise} \end{cases} \quad (3)$$

Finally D is uniformly quantized into Q using 8 levels. Said $L_x = \{1, 2, \ldots, N_x\}$ and $L_y = \{1, 2, \ldots, N_y\}$ the X and Y spatial domains, and $L = L_x \times L_y$ the set of pixel coordinates of the greyscale image I, in order to summarize the relations between the gradients of neighbor pixels, we start defining $S_\delta(i,j)$ as the set of all point couples displaced by vector δ, with gradient directions i and j respectively:

$$S_\delta(i,j) = \{r, s \in L | Q(r) = i, Q(s) = j, r - s = \delta\}. \quad (4)$$

Since we are also interested in the strength of the texture, the magnitude of the gradients is considered in the final

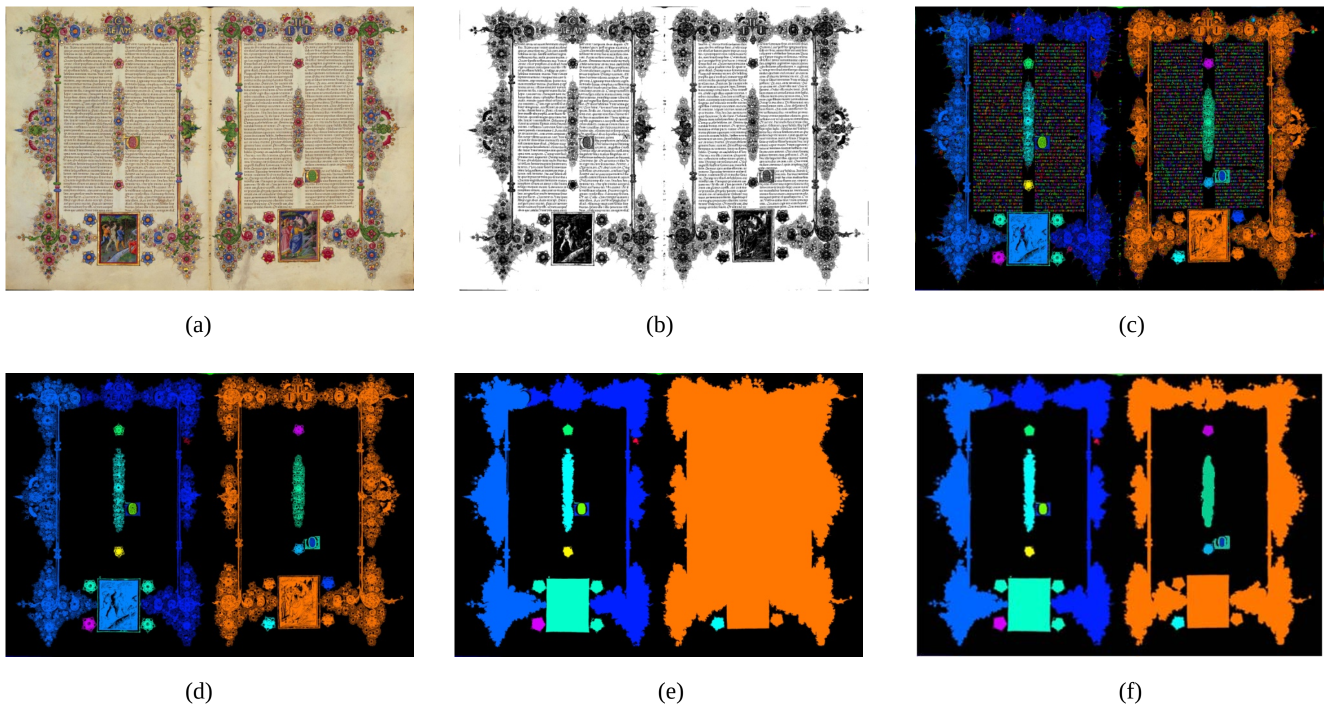

Figure 3: Steps of the preprocessing. The cropped image (a) passes through the Otsu thresholding (b), then a labeling algorithm is performed (c), removing the smallest blobs from the result (d). The filling procedure is performed (e): in this particular case, the need of the excessive filling restoration is highlighted (f).

matrix:

$$P_\delta(i,j) = \sum_{(r,s)\in S_\delta(i,j)} M(r) + M(s) \tag{5}$$

In our setup, δ was taken in the set {(1,-1), (1,0), (1,1), (0,1)}, that contains the 4 main directions $\{45°, 0°, -45°, -90°\}$ at 1 pixel distance. Concluding, the feature used is composed by four square matrices with size 8×8, leading to a 256-dimensional feature vector.

4.3 Segmentation as a Classification Problem

Support Vector Machines are a common technique for data classification. Given a training set of n labeled instances (x_i, y_i) for $i = 1 \dots n$, where $x_i \in \Re^n$ and $y_i \in \{+1, -1\}$, SVM finds out a linear separating hyperplane (defined by the support vectors) with the maximal margin in a higher dimensional space. Several kernel functions have been proposed, with the property of distance in feature space and with the positive semidefinite matrix for all elements.

One remarkable property of SVMs is that their ability to learn can be independent of the dimensionality of the feature space [14]. In this particular application this is not true. In fact, in order to obtain acceptable performance we had to use a large number of training samples and these were directly related to the number of features employed. The use of an RBF kernel, usually providing better performance then linear or polynomial ones, required unacceptable training times with our training set. Neither a reduction of the size of the training set with this kernel is acceptable because of lower classification performances and overfitting (all training samples where selected as support vectors). Indeed, a reduction of the training set size would be particularly useful, in face of the final application scenario, in which the final user could obtain automatic annotation providing fewer manual samples, thus reducing his work.

4.3.1 *Dimensionality reduction with Embedding*

The amount of data and the dimensionality of feature vectors are challenging problems. A typical example is the similarity searching, in which we want to find the most similar results to a given query in a CBIR system. When we work with large datasets, the number of distances evaluations necessary to complete the task could become prohibitive. In order to limit this amount of computations and at the same time to maintain an acceptable quality of the results, an embedding approach can be exploited.

The goal is to embed the dataset into a different vector space with a lower dimensionality in such a way that distances in the embedded space approximate distances in the original space. In a more formal way, given a metric space S with a defined distance d, an embedding can be defined as a mapping F from (S, d) into a new vector space $(\mathbb{R}^k, \delta)$ where k is the new dimension and δ is the new distance.

$$\begin{aligned} F &: S \rightarrow \mathbb{R}^k \\ \delta &: \mathbb{R}^k \times \mathbb{R}^k \rightarrow \mathbb{R}^k \end{aligned} \tag{6}$$

Given two object o_1 and o_2, the goal of the embedding approach, as mentioned before, is to assure that the distance $\delta(o_1, o_2)$ is as close as possible to $d(o_1, o_2)$ in the original space. In particular, the embedding assures the contractive property if the distances in the embedding space provides a lower-bound for the corresponding distances in the orig-

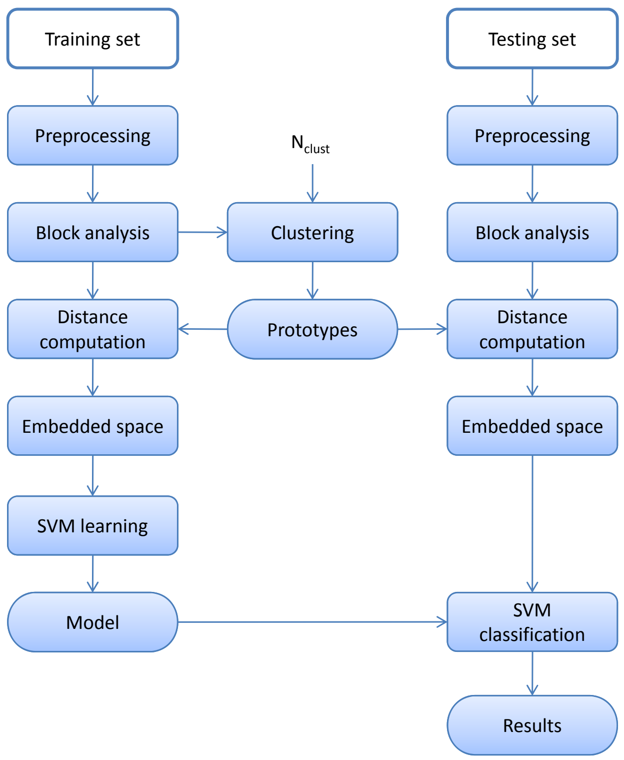

Figure 4: Diagrams that show the way each image is preprocessed and then analyzed in the learning and the classification procedures

inal space. The search in the embedding space should be simpler and faster. Thus, if the provided embedding has the contractive property, we assure that the search preserves the quality of results, because we obtain the same recall as in the original space. There are also different ways to measure the quality, for example distortion, stress, Cluster Preservation Ratio. A more detailed description is provided in [10].

One of the first proposals for embedding is the SparseMap approach [11], based on Lipschitz embeddings [1]. The key idea is to extract information about an unknown object x given the distances $d(x, o_1)$ and $d(x, o_2)$ between x and two arbitrary reference objects o_1 and o_2. This concept is then extended to subsets of object of the original space: these subsets constitute the axes of the embedded space, and the coordinate of any unknown object x in this new space is computed as the distance between x and the closest elements in each subset. SparseMap only includes some heuristics to make the overall computation more affordable (decreasing the number of distances to compute and providing a lower dimension for the embedded space). Another approach is FastMap [4] that implements a dimensionality reduction algorithm (vaguely inspired by PCA), and proved to be contractive only when an Euclidean distance is used. The resulting mapping is a linear transformation that minimize the difference between the corresponding distances in original and embedded space in mean-square terms. The last embedding method we mention is MetricMap [26], which produces a pseudo-Euclidean embedding space, which means that some axes can provide a "negative" contribution to distances between objects. We used a slightly different way to perform embedding, generally derived from the Lipschitz embeddings.

4.3.2 *Complete Link Clustering for Distance Based Embedding*

In order to exploit the distance metric specifically designed for every single feature (or consequently for every group of features, with simple feature fusion approaches), we used Complete Link clustering [13].

Complete Link is a well known hierarchical clustering approach based on the following criterion:

$$x \in C_i \Leftrightarrow d(x, y) > \eta_i, \forall y \in C_j, j \neq i \tag{7}$$

$$\eta_i = \max_{x,y \in C_i} d(x, y) \tag{8}$$

This criterion implies a strong condition on the similarity of elements in a cluster, because each element must be similar to every other in the cluster. In this case, to each cluster C_i is associated a maximum dissimilarity η_i defined by Eq. 8, which measures the maximum dissimilarity between any two elements in the cluster. Any other element outside the cluster must have a dissimilarity greater than η_i from any element in the cluster. Hierarchical clustering methods based on Complete Link generate clusters which satisfy the previous condition. For this clustering method we defined the dissimilarity between two clusters C_i and C_j as

$$d(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y) \tag{9}$$

The algorithm proceeds as follows:

1. Initially we have N clusters $\{x_1\}, \ldots, \{x_N\}$. Let's call E the set of clusters. Each cluster contains a single element x_i.
2. Find the most similar pair of clusters, R and S, according to 9, i.e. find R and S such that $d(S, R) < d(A, B), \forall A, B \in E$.
3. Merge R and S into a new cluster.
4. Repeat from step 2, until the required number of clusters is obtained.

This algorithm produces a hierarchy of elements partitions with at most N levels and i clusters at level i (the initial level is N). To implement the algorithm a proximity matrix D was used. An $N \times N$ proximity matrix $D = [d(B_i, B_j)]$ contains the dissimilarity between two blocks i and j at element (i, j). At each step, the matrix is updated by deleting rows and columns corresponding to clusters R and S and adding a new row and column corresponding to the newly formed cluster. The values in the new row/column are the maximum of the values in the previous ones. Initial generation of matrix D requires $\frac{1}{2}N(N-1)$ computations of $d(\cdot, \cdot)$.

The final procedure to learn and classify our data blocks, is summarized in Fig. 4: we separately cluster the positive and negative training samples, in order to select the most valuable objects which represent the entire sets. These reference examples become the basis of the new embedded space, and the new coordinates of every element in the dataset are computed as their distances with the reference objects. Now we can apply the regular SVM learning stage, obtaining our classifier. The reference objects can now be used to embed the unknown objects, using the SVM classifier to provide the final output.

Table 1: **Comparison using different feature sets**

	RGB %	eHSV %	GSDM %	all %
Re_{pixels}	82.36	80.35	82.42	83.49
Pr_{pixels}	53.60	57.61	43.32	52.97
Re_{blobs}	84.21	81.50	84.21	85.69
Pr_{blobs}	70.33	74.91	57.27	73.36
Re_{blocks}	68.58	62.85	74.60	75.87
Pr_{blocks}	84.23	87.31	74.23	85.80

Table 2: **Comparison with and without the embedding procedure**

Samples	10 000	1 000	1 000	1 000
Embedding	no	no	yes	yes
Kernel type	Linear	Linear	Linear	RBF
$Re_{blobs}\%$	84.91	84.03	85.75	85.69
$Pr_{blobs}\%$	73.28	69.57	72.46	73.36
$Re_{blocks}\%$	74.44	72.81	74.92	75.87
$Pr_{blocks}\%$	85.90	84.82	85.09	85.80
Support Vectors	1075	802	445	377

5. EXPERIMENTAL RESULTS

In this paper, we used the digitalized pages of the Holy Bible of Borso d'Este, duke of Ferrara (Italy) from 1450 to 1471 A.C., which is considered one of the best Renaissance illuminated manuscript in the world. Tests has been performed among a dataset of 320 high resolution digitalized images (3894x2792), a total amount of 640 pages. These images have been manually annotated, so half of the pages has been used for training and half for testing. Each page of the dataset is an illuminated manuscript composed by a two-column layered text in Gothic font, spaced out with some decorated drop caps. The entire surrounding is highly decorated. Without any prior knowledge, the system aims at extracting the valuable illustrations within the decoration texture (miniature illustrations of scenes, symbols, people and animals), rejecting all the border decorations (ornaments). As mentioned before, this paper is mainly focused on this aspect of the page analysis, so we do not propose our results on text segmentation (see [7] for further information).

Results are reported in terms of recall and precision. The granularity of these results has three levels: pixels, blocks, and blobs. Recall and precision at pixel-level are computed comparing the automatic annotation with the raw number of pixels that have been marked by a human operator as valuable pixels. Since we did not yet implement any refinement on the boundaries of the extracted picture (in order to precisely segment it from the decoration), we already expected quite low precision values. Recall and precision at blocks level correspond to the raw recall and precision values outputted by the SVM: based on the ground truth, we labeled each block within the testing set, choosing a positive annotation if the majority of pixels within the block belongs to a valid picture, and a negative annotation otherwise. Finally recall and precision at blobs level are computed counting how many blobs have a significant overlap with a corresponding blob in the ground truth.

The first tests were conducted on the features. We computed recall and precision values with different sets of features, in order to verify that a higher number of features could effectively contribute to a better classification. Each feature defines its own way to compute the similarity: in particular, RGB and EHSV histograms exploit a histogram intersection approach, while the GSDM feature performs a sum of point-to-point Euclidean distances between the matrices. These values are standardized, and then arithmetic mean is computed to fuse their results. The tests were conducted applying the previously described embedding procedure firstly to the single features, then to their combination.

Table 1 shows that the addition of different features helps improving the classification performance. In particular, simple information about colors in the HSV space proved to be discriminant enough to distinguish the images from the decorations, since decorations have a limited palette and a major amount of background pixels. Texture information help to significantly increase the precision, and a further improvement on recall values is highlighted. Finally the addition of the RGB histogram seems to propose a good compromise between recall and precision: it boost precision values with a minimum loss in recall values.

The subsequent test was focused on the effective usefulness of the embedding. Using the combined feature set, we compared the performance of the system with and without the embedding procedure.

Table 2 shows that by using an embedding approach with only 1000 positive samples and 1000 negative samples we can obtain similar performances to those obtained by using ten times more samples. This is a great advantage because it implies that, given a new manuscript to be analyzed, the human operator has to manually annotate only a few pages. This procedure can be also included into a relevance feedback context: using a limited amount of correction on the results proposed with a standardly trained system, in a small amount of time good results can be easily achieved.

We would like to highlight that these results are not post-processed, and that they were obtained using general assumptions on the appearance of blocks, without any prior inference. The set of feature used reflects this approach. We used color features (RGB and HSV histograms) because generally decorations blocks have a different (and quite limited) palette of colors with a lot of background, and we used texture features because generally decorations blocks are quite more regular and repetitive with a lot of symmetry. Some example results are shown in Fig. 5.

6. CONCLUSIONS

This paper described a system for the automatic segmentation of decorations from illuminated manuscripts. Starting from the high resolution replicas of the Bible pages, a preprocess stage focus the processing on the most valuable pixels of the image, then a sliding window analysis extracts low level color and texture features of each block. By the application of the described embedding procedure SVM classification provides good results with less training samples and allows the use of RBF kernels.

7. ACKNOWLEDGMENTS

The authors would like to thank Roberto Vezzani for the invaluable help and discussion.

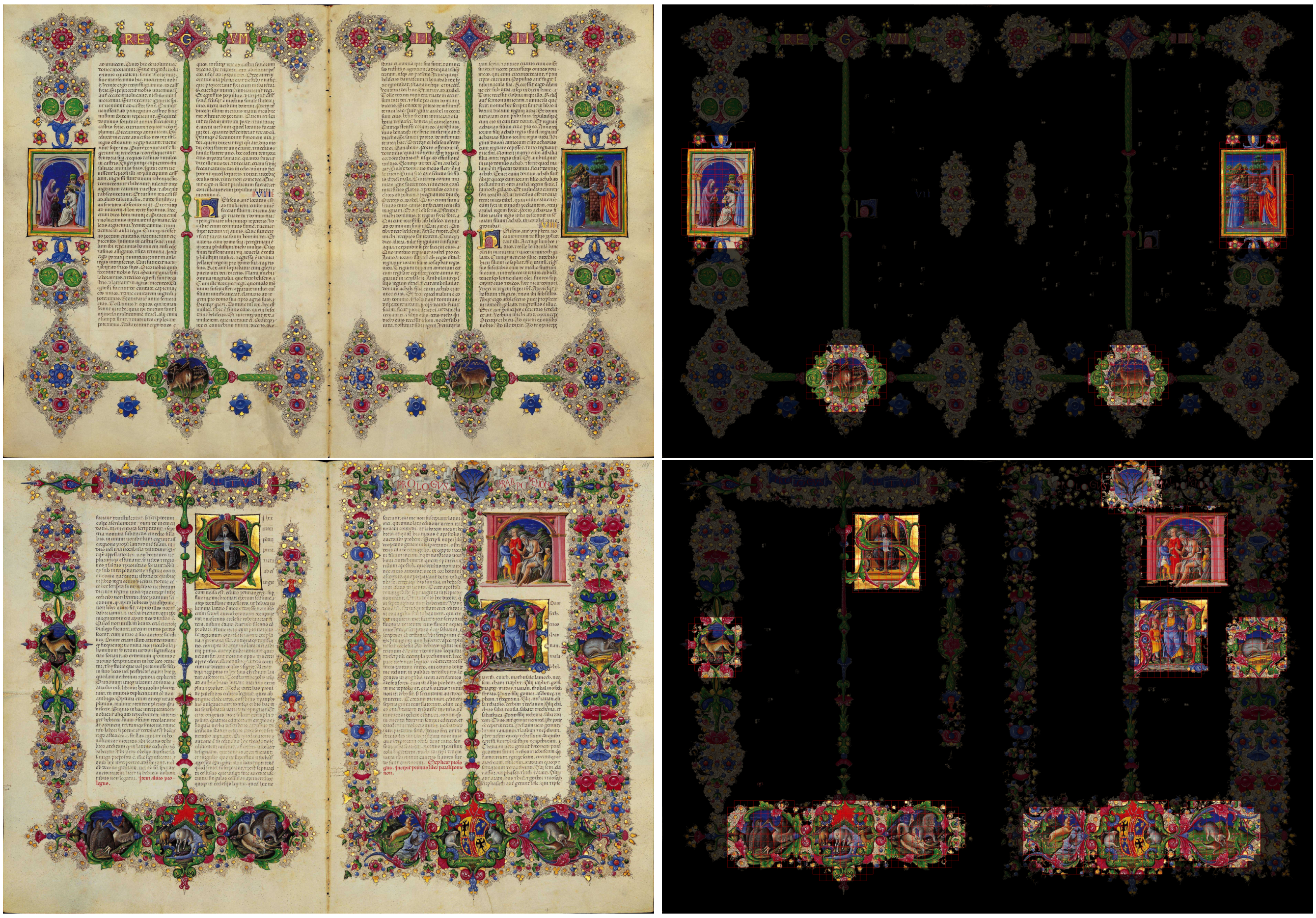

Figure 5: Example of segmentation results

8. REFERENCES

[1] J. Bourgain. On lipschitz embedding of finite metric spaces in Hilbert space. *Israel Journal of Mathematics*, 52(1):46–52, 1985.

[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*, 10(1):1–16, 2007.

[3] M. Diligenti, P. Frasconi, and M. Gori. Hidden Tree Markov Models for Document Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):519–523, 2003.

[4] C. Faloutsos and K. Lin. FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 163–174. ACM, 1995.

[5] Y. Fataicha, M. Cheriet, J. Nie, and C. Suen. Content Analysis in Document Images: A Scale Space Approach. In *Proceedings of the International Conference on Pattern Recognition*, volume 3, pages 335–338. IEEE Computer Society, 2002.

[6] C. Grana, D. Borghesani, S. Calderara, and R. Cucchiara. Şinside the bibleŤ: Segmentation, annotation and retrieval for a new browsing experience. In *ACM International Conference on Multimedia Information Retrieval*, pages 379–386, Vancouver, Canada, Oct. 2008.

[7] C. Grana, D. Borghesani, and R. Cucchiara. Describing Texture Directions with Von Mises Distributions. In *Proceedings of the 19th International Conference on Pattern Recognition*, 2008.

[8] C. Grana, R. Vezzani, and R. Cucchiara. Enhancing HSV Histograms with Achromatic Points Detection for Video Retrieval. In *Proceedings of ACM International Conference on Image and Video Retrieval*, pages 302–308, 2007.

[9] Haralick, R.M. and Shanmugam, K. and Dinstein, I. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, 1973.

[10] G. Hjaltason and H. Samet. Properties of Embedding Methods for Similarity Searching in Metric Spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):530–549, 2003.

[11] G. Hristescu and M. Farach. Cluster-preserving Embedding of Proteins. Technical report, Center for Discrete Mathematics and Theoretical Computer Science, 1999.

[12] J. Hu, R. Kashi, and R. Wilfong. Document

Classification Using Layout Analysis. In *Proceedings of the International Workshop on Database and Expert Systems Applications*, pages 556–560. IEEE Computer Society, 1999.

[13] A. Jain and R. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[14] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning*, pages 137–142. Springer Verlag, 1998.

[15] N. Journet, J. Ramel, R. Mullot, and V. Eglin. Document image characterization using a multiresolution analysis of the texture: application to old documents. *International Journal of Document Analysis and Recognition*, 11(1):9–18, 2008.

[16] E. Kavallieratou. A Binarization Algorithm specialized on Document Images and Photos. In *Proceedings of the 8th International Conference on Document Analysis and Recognition*, pages 463–467. IEEE Computer Society, 2005.

[17] A. Kitamoto, M. Onishi, T. Ikezaki, D. Deuff, E. Meyer, S. Sato, T. Muramatsu, R. Kamida, T. Yamamoto, and K. Ono. Digital Bleaching and Content Extraction for the Digital Archive of Rare Books. In *Proceedings of the International Conference on Document Image Analysis for Libraries*, pages 133–144. IEEE Computer Society, 2006.

[18] T. Konidaris, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis, and S. Perantonis. Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. *International Journal on Document Analysis and Recognition*, 9(2-4):167–177, 2007.

[19] F. Le Bourgeois and H. Emptoz. DEBORA: Digital accEss to BOoks of the RenAissance. *International Journal of Document Analysis and Recognition*, 9(2):193–221, 2007.

[20] F. Le Bourgeois, E. Trinh, B. Allier, V. Eglin, and H. Emptoz. Document Images Analysis Solutions for Digital libraries. In *Proceedings of the International Workshop on Document Image Analysis for Libraries*, pages 2–24. IEEE Computer Society, 2004.

[21] G. Meng, N. Zheng, Y. Song, and Y. Zhang. Document Images Retrieval Based on Multiple Features Combination. In *Proceedings of the International Conference on Document Analysis and Recognition*, volume 1, pages 143–147. IEEE Computer Society, 2007.

[22] G. Nagy. Twenty years of document image analysis in PAMI. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):38–62, 2000.

[23] S. Nicolas, J. Dardenne, T. Paquet, and L. Heutte. Document Image Segmentation Using a 2D Conditional Random Field Model. In *Proceedings of the International Conference on Document Analysis and Recognition*, volume 1, pages 407–411, 2007.

[24] J. Ogier and K. Tombre. Madonne: Document Image Analysis Techniques for Cultural Heritage Documents. In *Digital Cultural Heritage, Proceedings of 1st EVA Conference*, pages 107–114. Oesterreichische Computer Gesellschaft, 2006.

[25] J. Ramel, S. Busson, and M. Demonet. AGORA: the interactive document image analysis tool of the BVH project. In *Proceedings of the International Conference on Document Image Analysis for Libraries*, pages 145–155, 2006.

[26] X. Wang, J. Wang, K. Lin, D. Shasha, B. Shapiro, and K. Zhang. An index structure for data mining and clustering. *Knowledge and Information Systems*, 2:161–184, 2000.