Special Object Extraction from Medieval Books Using Superpixels and Bag-of-Features (BoF)

Ying Yang[a], **Holly Rushmeier**[a]
[a]Yale University, Department of Computer Science, 51 Prospect Street, New Haven, USA, 06511

Abstract. We propose a method to extract special objects in images of medieval books, which generally represent, for example, *figures* and *capital letters*. Instead of working on the single pixel level, we consider *superpixels* as the basic classification units for improved time efficiency. More specifically, we classify superpixels into different categories/objects by using a bag-of-features (BoF) approach, where a superpixel category classifier is trained with the local features of the superpixels of the training images. With the trained classifier, we are able to assign the category labels to the superpixels of a historical document image under test. Finally, special objects can easily be identified and extracted after analyzing the categorization results. Experimental results demonstrate that, as compared to the state-of-the-art algorithms, our method provides comparable performance for some historical books, but greatly outperforms them in terms of generality and computational time.

Keywords: Historical documents, superpixel, bag-of-features (BoF), clustering.

1 Introduction

Over the past few years, significant effort has been dedicated to digitizing historical documents, resulting in numerous digital libraries all over the world. As such, there is a pressing need for the computer-aided analysis techniques that are capable of extracting various types of information from massive collections. Information of particular interest to scholars includes, for example, text blocks,[1,2] text lines[3,4] and figures and capital letters.[5]

Similar to the work presented by Yang et al.,[5,6] the focus of this paper is on the extraction of special objects from medieval books. By "special objects", we refer to those objects that are generally colored and shaped distinctively from normal text, such as figures and line fillers as shown in Figs. 3 and 4. In general, special objects in historical documents are used to convey special meaning. For instance, in medieval Books of Hours, the combinations of initials and line fillers are an indication that the page contains a litany,[7] which is a series of invocations for deliverance and

intercession. Scholars, such as those we worked with, are particularly interested in special object extraction so as to assist their studies.

Indeed, extracting special objects can be considered a clustering problem, where the pixels are first clustered into different groups and special objects are then extracted by analyzing the clustering results. Following this idea, Yang et al.[5,6] propose two automatic algorithms. By solving a minimization problem, they compute a content-adaptive K as the number of clusters. The resulting K is subsequently used by machine learning algorithms to perform pixel clustering. As demonstrated by experimental results, the computation of a desirable K, although quite challenging, is very successful when applying these two methods to certain medieval manuscripts. Unfortunately, the two approaches suffer from two obvious shortcomings in terms of *generality* and *computational time*.

- They can produce good results only for medieval manuscripts whose text does not have a highly curved style, like old English and Latin manuscripts. This is because both algorithmic pipelines estimate K based upon an assumption regarding text style. If the assumption does not hold, an unreasonable K will be produced, resulting in unsatisfactory results.

- Both approaches operate at the pixel level, making them time-inefficient when working on high-resolution images. For example, they require approximately 3.5 minutes for a 3128 $\times$ 2274-sized image when running on a PC with an 8 Intel Core i7-5820K CPU 3.30GHz processor and 16GB memory.

An efficient approach to reducing computation is to use superpixels (semantically meaningful image patches) as the analysis units; that is, an image is first decomposed into a set of superpixels and the resulting superpixels are then analyzed. Recent methods have taken this approach. Cohen

et al.[8] exploit both spatial and color features of superpixels to extract drawings from images with only background and noise pixels. However, the method assumes drawings occupy relatively large areas of the page. Chen et al.[9] train an SVM classifier with superpixel labels and features to classify superpixels into four classes. This approach is faster than pixel based approaches,[10] but still has issues with time efficiency (see Table 3) due to the use of high dimensional features. While showing the promise of the superpixel approach, neither of these methods has been extensively evaluated. Just one historical book was used in evaluation for one approach[8] and three books for others.[9,10]

In this paper, we concentrate on the same problem of special object extraction from medieval books as previous methods[5,6] do, while approaching it from a different perspective. Our method successfully addresses their afore-mentioned shortcomings. To deal with the generality problems, we fix $K = 3$ rather than putting a lot of effort to compute an optimal (dynamic) K for each book page under test, by broadly dividing pixels into three categories: *background*, *normal text* and *special object*. For improved time efficiency, we also propose to work at the superpixel level rather than the single pixel level. We tested the proposed method on eight medieval books, which are different in terms of, for example, writing style and texture. We show in Section 4 that our approach is able to extract special objects with both satisfactory accuracy and improved computational cost.

In summary, our contributions in this work include:

- A new algorithm for extracting special objects from historical books, with no dependency on K. Note again that our method has better generality than the approaches,[5,6] in that it is able to handle any books with any text styles, while the previous methods[5,6] can only deal with manuscripts that do not have highly curved text (see Fig. 5).

- Significantly improved time efficiency achieved while having comparable or better results as compared to the state-of-the-art algorithms.[5,6,9,10] For instance, as demonstrated by the experimental results in Table 3, our method is ∼5 times faster than the method,[9] and moreover ∼395-580 times faster than the approach.[10]

- To the best of our knowledge, the first attempt of using both superpixels and their Bags-of-Features (BoF) representations in historical document analysis.

2 Related Work

There are many recent works on historical document analysis, superpixel and BoF, so for conciseness we will only review the most relevant here. Analysis tasks for historical documents include, for example, word matching[11],[12] word segmentation[13],[14] text line extraction[15][16],[17] text block segmentation[1,18] and figure extraction.[19]

Superpixels, which are perceptually meaningful image patches, are becoming increasingly popular for use in computer vision applications. Many previous methods[20–22] in computer vision use superpixels as the underlying representation to speed up processing.

A BoF method represents an image as orderless collections of its quantized local image descriptors. Due to simplicity and performance, BoF methods have been widely used in computer vision tasks, including image classification[23] and video retrieval.[24]

Special Object Extraction. Extracting special objects is commonly considered as a classification problem and hence previous methods generally use the well-established machine learning algorithms, such as SVM (Support Vector Machine), to cluster pixels.

Chen et al.[25] develop a layout structure segmentation algorithm, where each pixel is represented by a feature vector computed based on the coordinates, color and texture information gathered

from the area surrounding the pixel. Aiming at reducing computational time, they also present a superpixel-based algorithm,[9] where an SVM classifier is trained with learned features and their labels. The method, although using superpixels, still does not achieve satisfactory runtime; for instance, 1.4 minutes are required for a 1700×1100-sized image. Again by using superpixels as basic units of segmentation, Cohen et al.[8] use a classifier to separate drawings from background and noise. However, the method works only when the assumption that drawings occupy a large portion of a page is valid, so that it may not work for certain historical books, such as books without large drawings (See Fig. 5).

Grana et al.[19] employ an SVM-based approach to identifying and extracting significant graphical elements from historical manuscripts. Starting from computing an optimal number of clusters, K, Yang et al.[6,26] present two algorithms for automatic extraction of special objects from medieval manuscripts.

Superpixel. Many computer vision algorithms rely on superpixels instead of just on pixels, since superpixels have the major advantages of reducing the number of semantically meaningful entities and enabling feature computation on bigger, more meaningful regions.[27–29] Superpixel computational algorithms can be broadly categorized into two types: *graph-based* and *gradient ascent-based*. For a more exhaustive comparison of state-of-the-art superpixel methods, we refer the reader to a recent survey.[22]

Graph-based methods deem the superpixels as the nodes in a graph, where superpixels are generated by minimizing a cost function defined over the graph. Originally developed by Shi et al.[30] for image segmentation, Normalized Cuts was used by Ren et al.[20] to propose the first superpixel algorithm. Felzenszwalb et al.[31] present another graph-based approach to generating superpixels. The algorithm is based on a predicate that measures the evidence for a boundary

between two regions. Other representative graph-based approaches include work by Moore et al.[32] and Veksler et al.[33]

Gradient ascent-based algorithms produce superpixels by iteratively refining the clustering until some convergence criterion is satisfied. Quick shift[34] is a mode-seeking algorithm that initializes the segmentation using a medoid shift procedure. The superpixels produced by the TurboPixels method[35] not only respect local image boundaries, but also limit under-segmentation through a compactness constraint. Mean Shift[36] and the watershed approach[37] are also gradient ascent-based methods.

Bag-of-Features (BoF). The past decade has witnessed the growing popularity of the BoF-based approaches in computer vision. In general, a BoF image representation follow a three-step procedure: *building a vocabulary*, *assigning "visual words" to features* and *generating a histogram of the "visual words"*.

There are a number of choices involved at each step in the BoF representation. Popular choices for image feature computation include the SIFT descriptor[38] and the SURF feature.[39] To build a visual vocabulary, clustering techniques, such as K-means, are generally used. Common choices for determining the distance between two features are the Manhattan (L_1), Euclidean (L_2), or Mahalanobis distances.[40]

3 The Proposed Algorithm

This section describes in detail the proposed algorithmic pipeline for special object extraction. We consider this a clustering problem, where the categories considered include: *background*, *normal text* and *special object* as mentioned earlier.

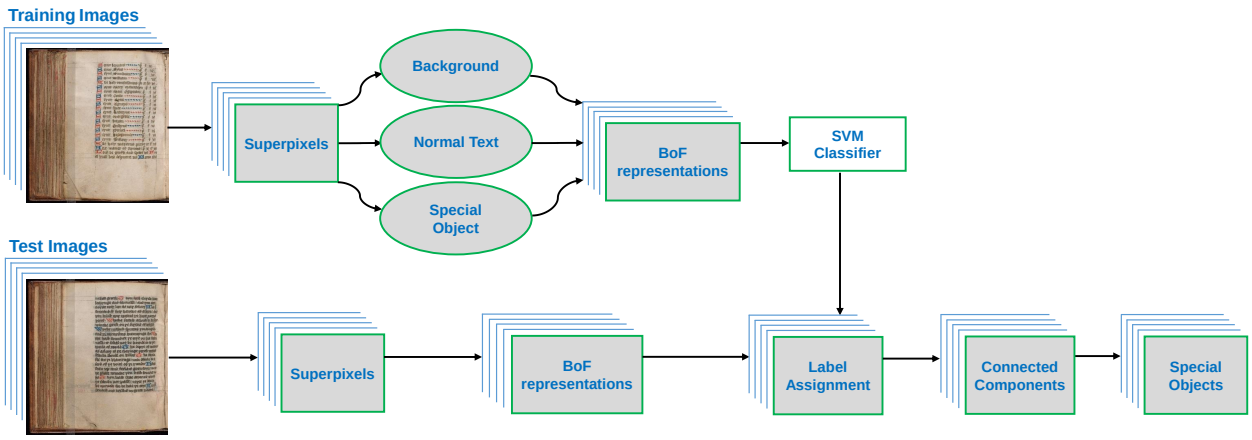

Fig 1 Two-step algorithmic pipeline. In the first step, we build a superpixel categorization classifier by employing a BoF approach. Given the trained classifier, the second step first assigns the labels to the superpixels and then extracts special objects as the connected components of a binary image formed according to the label assignments.

As illustrated in Fig. 1, the proposed pipeline is composed of two steps. The first step is to train a superpixel classifier in a supervised way (supervised learning); by contrast, in the second step, we apply the trained classifier to the superpixels of a test image to determine their categories and then locate the special objects by analyzing the resulting categorization map. The details are described as follows.

3.1 Superpixel Classifier Training

In order to train a supervised superpixel classifier, we need to carry out two tasks beforehand: i) labeling the superpixels of all the images in the training dataset and ii) describing superpixels with BoF representations. In the following, we shall describe the steps required to finish the two tasks for a training image. For each of the remaining training images, exactly same steps should be taken before building the classifier.

Taking an image in the training dataset, we first compute its superpixels. In doing so, various algorithms, such as the methods by Wang et al.[27] and Boix et al.,[28] can be used. We in this paper choose to use the Van den Bergh's method[41] due to its superior performance in terms of both

superpixel generation result and time efficiency (real-time). Fig. 2 shows the superpixels of some example images in our dataset.

Note that, as mentioned before, the use of superpixels can, on the one hand, significantly improve computational speed (see Fig. 6). On the other hand, using superpixels is reasonable from the technical point of view, in that a special object is generally the composition of several superpixels.

Obviously, labeling the above computed superpixels (labels: *background*, *normal text* and *special object*) can be done manually. However, manual labeling is a time-consuming process and infeasible in certain circumstances, especially when there are lots of superpixels being considered. Consequently, we present a fully automatic method to expedite the labeling process. Given the superpixels, our method starts from computing their color-based features, which are the average values of the R, G and B color components over each superpixel. Then, we construct the clusters (no more than 3 clusters) from the agglomerative hierarchical cluster tree, where the distance metric is cosine similarity. The next step is to use the binarization method[42] to the image being considered, yielding a binary image. Finally, by denoting N_b, N_t and N_s the numbers of the pixels in the background, normal text and special object, respectively, and also by assuming $N_b > N_t > N_s$, we can assign the labels to the superpixels according to the clustering results and binary image.

We observed that the above automatic algorithm cannot always guarantee correct label assignments without making any error, i.e., some superpixels may be labeled incorrectly. For improved labeling accuracy, one possible way is to have human interaction involved. In other words, we can look through the initial assignments and make corrections if necessary.

We now move on to the second task of describing the superpixels with their BoF representations. As such, we take into account the following features: color features (averaged R, G and B),

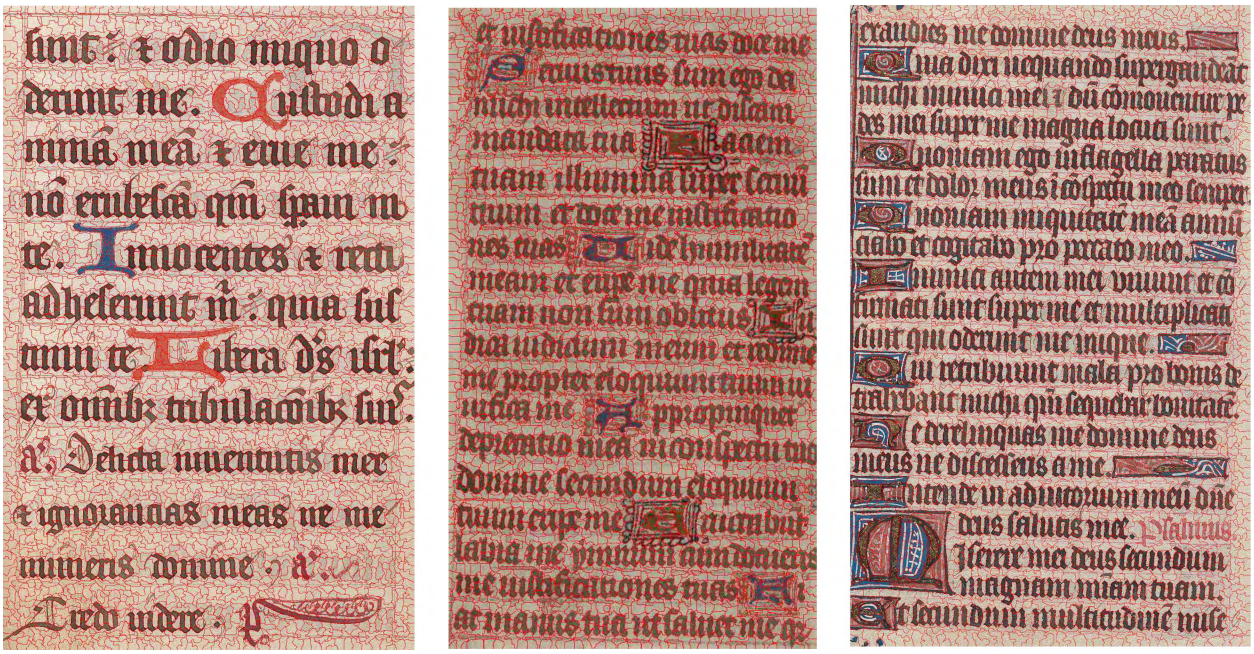

Fig 2 Visualization of superpixels computed using the method[41] for the text blocks of a few example images. Manuscript images courtesy of the Yale University.[43–45]

SURF features extracted from the grayscale image, as well as from the R, G and B maps. Note that the motivation of using these color maps is that the features of normal text and special objects computed from more color channels are expected to be more discriminative than those derived from less channels (since special objects in medieval manuscripts are generally colored distinctively from normal text) and hence that, given more discriminative features, the machine learning algorithm (SVM in the paper) is expected to perform better clustering.

In our implementation, different scales, at which the interest points were detected, are used to achieve multiscale feature extraction. After applying k-means algorithm (k = 500 in the experiments) to the features of all the superpixels with the variance of the features as the feature distance metric, we obtain the so called "dictionary of words" as the centers of these clusters and furthermore compute the BoF representation of a superpixel as the histogram of its features. More specifically, by replacing the superpixel features with their own respective words in the dictionary (the centers of their own associated clusters), the representation is a histogram with the words on

the x-axis and the frequency of each "word" on the y-axis.

The afore-mentioned process is iterated until all of the other training images have been processed, that is, the superpixels of all the training images are labeled with *background*, *normal text* and *special object* and also described with BoF representations. With both the labels and BoF representations, we are able to train a supervised classifier. For efficiency, we employ the SVM (Support Vector Machine) learning algorithm with Gaussian kernel function.

3.2 Superpixel Classifier Testing and Special Object Extraction

Given the trained SVM classifier, we are able to identify special objects. Similar to the training process, the superpixels of a test image are first computed, and the trained classifier is then applied to categorize each superpixel into one of the three categories: *background*, *normal text* and *special object*.

We concentrate on superpixels with the label of "special object". To locate the special objects within the test image, a binary image is first created and initialized with 0 for background and normal text pixels and 1 for special object pixels.

Having the resulting binary image, we extract its connected components and consider as the special objects those that satisfy the following constraint: the sum of the width and height of the bounding box of the connected component is larger than $\beta \cdot H$ ($\beta > 1$). Here, H denotes the text height/leading and we compute it using.[15] Note that the constraint is enforced, because we believe small connected components are created due to incorrect clustering and are not special objects.

Table 1 Precision (%) and Recall (%) results achieved by our proposed method with a specific classifier using only the images for the same book. The results of our method are compared against those of the state-of-the-art algorithms.[6,26] Note that the *Gower* manuscripts used here do not follow the assumptions made in[26] and.[6] Some example images from the test manuscripts are shown in Fig. 5. "NA" means "not available".

Manuscript Name	# Training Images	# Test Images	TP	FP	FN	Precision [26]/[6]/Our	Recall [26]/[6]/Our
BeineckeMS10	2	132	689	3	0	87.82/97.31/99.57	99.23/94.39/100
BeineckeMS109	2	100	326	26	33	88.55/92.06/92.61	95.80/96.18/90.81
BeineckeMS310	2	105	899	50	25	90.02/95.41/94.73	96.33/95.37/97.29
BeineckeMS360	2	88	850	27	0	98.36/99.60/96.83	89.93/92.98/100
Gower (Cambridge)	2	35	111	19	3	53.02/83.52/85.38	66.95/66.67/97.37
Gower (Yale)	2	20	36	3	4	0/0/92.31	0/0/90.00
Parzival	2	11	29	3	1	NA/NA/90.60	NA/NA/96.67
Saint Gall	2	28	30	2	3	NA/NA/93.75	NA/NA/90.91
# Manuscripts – 8	16	519	2970	133	69	NA/NA/95.71	NA/NA/97.73

4 Experimental Results

In this section, we shall evaluate the proposed method by testing it on eight medieval manuscripts from the Yale Universitys Beinecke Rare Book and Cambridge University Trinity College[46] (available to download at[47]), with the parameter fixed at $\beta = 1.7$ in the experiments (see Section 3.2). The data are very heterogeneous, in terms of layout structure (e.g., text density), conservation (e.g., ageing and ink bleed-through), acquisition resolution and writing style. The algorithm was implemented using C++, MATLAB and the OpenCV library,[48] and tested on a PC with an 8 Intel Core i7-5820K CPU 3.30GHz processor and 16GB memory. The computational time depends heavily on image resolution and the number of superpixels. For a 3128×2274-sized image with approximately 3200 superpixels, our implementation takes about 45 seconds to extract special objects.

The evaluation will be carried out with the performance indicators being the *Precision* and *Recall* values:

$$\begin{cases} \text{Precision} &= \dfrac{\text{TP}}{\text{TP}+\text{FP}} \\ \text{Recall} &= \dfrac{\text{TP}}{\text{TP}+\text{FN}} \end{cases}, \tag{1}$$

where TP, FP and FN indicate true positive, false positive and false negative, respectively. We

Fig 3 Visualization of the special objects extracted using a specific classifier from several example images of pages of BeineckeMS10, BeineckeMS109, BeineckeMS310 and BeineckeMS360 (from left to right). Manuscript images courtesy of the Yale University.[43–45,49] Zoom in to observe details.

obtained the TP, FP and FN values by visually assessing the automatically generated results. More specifically, we drew rectangles over the detected special objects (see Figs. 3 and 4) and then went through all the test images to perceptually verify detection accuracy. As demonstrated by Yang et al.'s work,[6] the Precision and Recall computed with such TP, FP and FN agree well with those when the groudtruth data were available.

4.1 Specific Classifier

To construct a *specific* classifier, the training and test datasets need to be from the same manuscript; that is, we built a classifier for each test manuscript using some of its images. In the experiments, we randomly selected two images for each manuscript as the training images and the remaining as the test images.

Table 1 shows the results when a specific classifier is used. It is easy to see from the table that our method can achieve very satisfactory precision and recall values for all of the test books. The overall precision and recall values reach as high as 96% and 97%, respectively.

To visualize the extracted special objects, we draw rectangles over their own images. Fig. 3 is the visualization of the special objects we extracted from a few example images of four phys-

Table 2 Precision (%) and Recall (%) results achieved by our proposed method with a universal classifier trained using the images for different books. We randomly selected 2 images from each of the 4 test books, forming a training dataset of 8 images. The results of our method are compared against those of the state-of-the-art algorithms.[6,26]

Manuscript Name	# Test Images	TP	FP	FN	Precision [26]/[6]/Our	Recall [26]/[6]/Our
BeineckeMS10	132	689	55	0	87.82/97.31/92.61	99.23/94.39/100
BeineckeMS109	100	347	162	12	88.55/92.06/68.17	95.80/96.18/96.66
BeineckeMS310	105	907	75	17	90.02/95.41/92.36	96.33/95.37/98.16
BeineckeMS360	88	840	24	10	98.36/99.60/97.22	89.93/92.98/98.22
Gower (Cambridge)	35	108	20	6	53.02/83.52/84.38	66.95/66.67/94.74
Gower (Yale)	20	34	0	6	0/0/100	0/0/85.00
Parzival	11	26	5	4	NA/NA/83.87	NA/NA/86.87
Saint Gall	28	28	4	5	NA/NA/87.50	NA/NA/84.84
# Manuscripts – 8	519	2979	345	60	NA/NA/89.62	NA/NA/98.03

ical appearance-distinct medieval manuscripts, further confirming that our proposed method can achieve satisfactory performance.

4.2 Universal Classifier

In addition to the eight specific classifiers, we also constructed a universal classifier, which here was trained with sixteen images (two for each book) randomly selected from the eight test books. The intention is that a universal classifier may be used as a benchmark for measuring the performance of the proposed method on other historical books, from which the data was never used during the process of classifier training.

Table 2 shows the results obtained from using the trained universal classifier. We again achieve high precision and recall values for each of the test books, with the exception of the precision value for the manuscript *BeineckeMS109*. The precision and recall values averaged over the test manuscripts are 89.80% and 98.62%, respectively. The reason for the lower Precision value for *BeineckeMS109* is that as compared to other medieval manuscripts, the *BeineckeMS109* manuscript is of the lowest quality, e.g., lots of stains and a great deal of color degradation, so that the features of special and non-special objects are not that distinguishable. One way to achieve satisfactory

Fig 4 Visualization of the special objects extracted, using a universal classifier, from several example images of pages of BeineckeMS10, BeineckeMS109, BeineckeMS310 and BeineckeMS360 (from left to right). Manuscript images courtesy of the Yale University.[43–45,49] Zoom in to observe details.

results for medieval manuscripts of poor quality is to use relatively more training images from them.

Fig. 4 visualizes the special objects extracted from some example images of four historical books. Again, the visualization indicates that our method is successful in extracting special objects.

Unlike other applications which may require lots of training data for satisfactory performance, our case needs only limited training samples, taking into account the tradeoff between performance and manual work. More specifically, the two main reasons are: (i) increasing the number of training images generally requires more manual effort to be dedicated to correcting super-pixel label assignments (see Section 3.1); and (ii) even with limited training images, experimental results show that our method is able to achieve satisfactory results (see Tables 1 and 2). Using more training images will just greatly increase the amount of manual work, without significant performance improvement.

4.3 Algorithm Comparison

In this section, we further evaluate the efficiency of the proposed method by comparing it against four state-of-the-art approaches,[6,9,10,26] with respect to *extraction accuracy*, *generality*, *computa-*

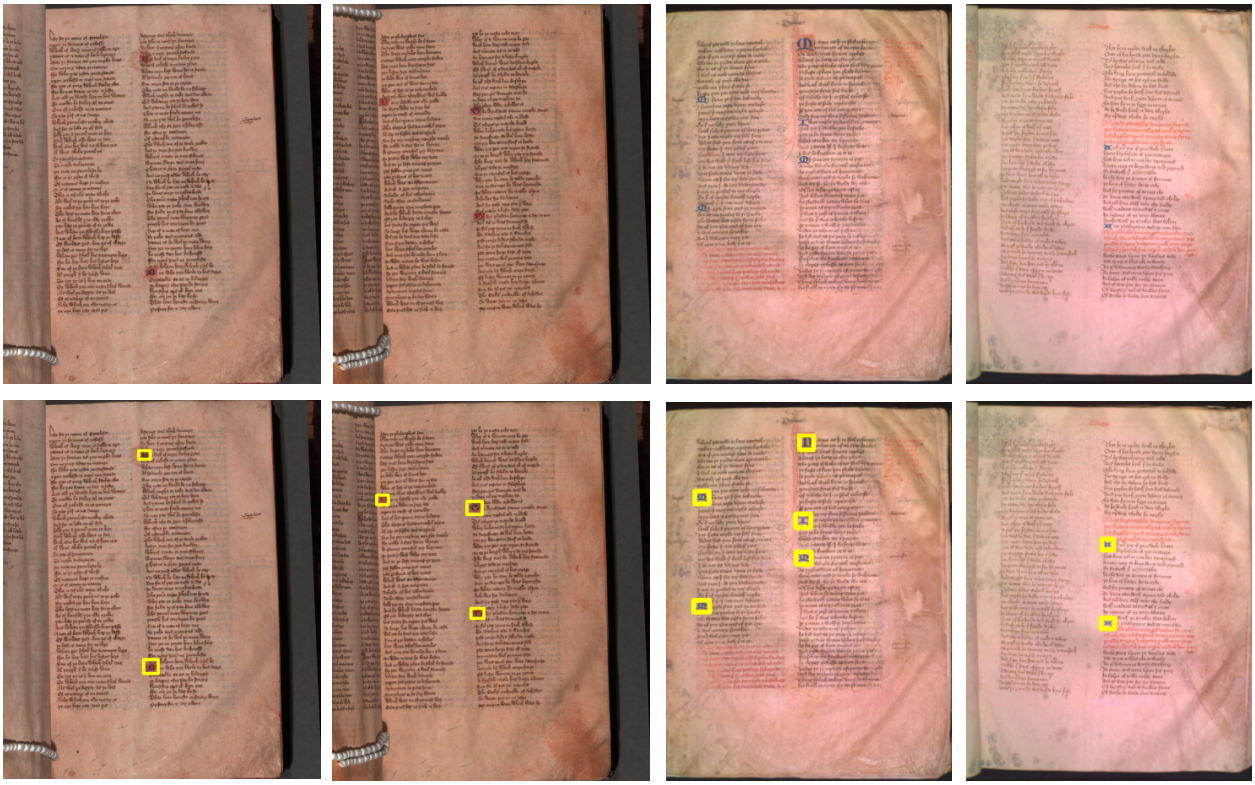

Fig 5 Comparison of results obtained from using[6,26] (top row) and our specific classifier-based method (bottom row) to the Cambridge (left two columns) and Yale Gower (right two columns) manuscripts. Manuscript images courtesy of the Yale University and the University of Cambridge. Zoom in to observe details.

tional time. The comparison experiments were conducted on comparable machines.

A. Comparison against Yang et al.'s algorithms[6,26]

We first carry out comparison between Yang et al.'s algorithms[6,26] and our method using the books that follow the assumption (text shape related assumption) made in.[6,26] Tables 1 and 2 list the comparison results. After comparison, we notice that[6] our method generally obtains higher precision and recall values than,[26] indicating that it is able to extract special objects at a higher accuracy rate than.[6,26]

We now compare the three approaches from the perspective of generality. In other words, performance is compared when applying the three methods to extract special objects from the historical books that do not quite follow the assumption in.[6,26] Table 1 indicates that while other

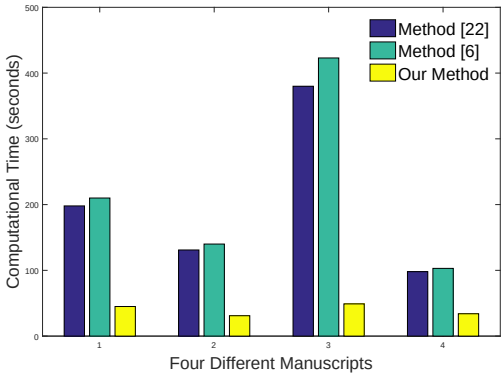

Fig 6 Comparison of the computational times used for extracting the special objects from the books (from left to right): *BeineckeMS10* ($\sim$3128 $\times$ 2274), *BeineckeMS109* ($\sim$2434 $\times$ 1960), *BeineckeMS310* ($\sim$3777 $\times$ 3040) and *BeineckeMS360* ($\sim$2083 $\times$ 1825). Note that the choice of our method (specific classifier-based or universal classifier-based) does not matter, since the runtime is exactly the same for both methods once a classifier is built.

approaches[6,26] fails when applied to analyze the challenging *Gower* manuscripts from Yale and Cambridge, especially the Yale one, our method performs well with desirable Precision (about 87%) and Recall (about 95%) values. From Fig. 6 we can also see that the proposed method is capable of extracting special objects; by contrast, the methods[6,26] both make an incorrect decision that there is no special object existing in these test images (no yellow rectangle in the images of the top row of Fig. 5).

Computational time is the third perspective we are concerned about. For fairness, the data we used are the books that all of the three algorithms can well work with. Fig. 6 compares the computational times required to extract the special objects from the manuscripts *BeineckeMS10*, *BeineckeMS109*, *BeineckeMS310* and *BeineckeMS360*. As Fig. 6 shows, the proposed method improves time efficiency significantly from the original 200 seconds (approximately) to 45 seconds for BeineckeMS10.

B. Comparison against Chen et al.'s algorithms[9,10]

Our algorithm is also compared against Chen et al.'s methods,[9,10] which concentrate on segmentation of historical document images. Their algorithm[10] works at the pixel level; by contrast,

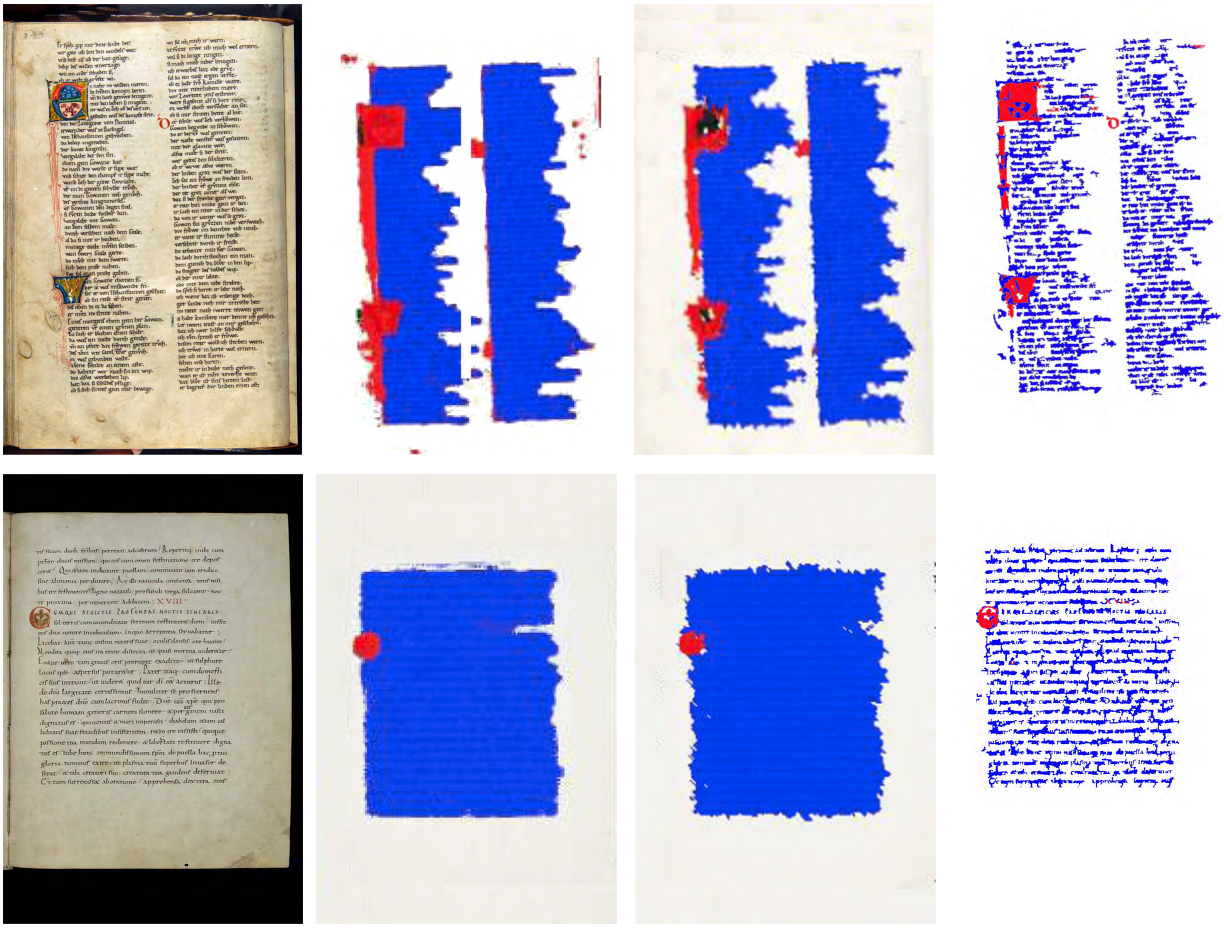

Fig 7 Comparison of segmentation results by Chen et al.'s pixel-level method[10] (second column), their superpixel-level method[9] (third column) and our specific classifier-based method (fourth column) to the two images (first column) from the Parzival[50] and Saint Gall.[51] The colors white, blue and red are used to represent background (and also periphery for Chen et al.'s algorithms), text, and special object pixels, respectively. Zoom in to observe details.

the most recent work[9] is superpixel-based for improved time efficiency. The test data we used for comparison is from the Parzival and Saint Gall datasets.

Fig. 7 illustrates the segmentation results by Chen et al.'s methods and our specific classifier-based method for two images. It is easy to see that our segmentation is able to accurately segment the pixels into three groups (background, text, special object), while Chen et al.'s methods provide rough pixel segmentation (due to the properties of the groundtruth data used for training).

Since the methods[9,10] by Chen et al. work in a way that no assumptions regarding, for example, text shape are required, we believe their algorithms have almost the same generality as ours. We

Table 3 Comparison between Chen et al.'s pixel-level method,[10] their superpixel-level method[9] and our approach on images from (3008×2000-sized) and Saint Gall (4992×3328-sized) in terms of computational time (in minute) required for an image. For fairness, similar to Chen et al., we applied our method to the resolution-reduced images with the scaling factor of 2^{-3} rather than to the original images. Note again that the choice of our method (specific classifier-based or universal classifier-based) does not matter when comparing timing.

Manuscript Name	Method[10]	Method[9]	Our Method
Parzival	101	0.91	0.19
Saint Gall	159.07	0.58	0.44

show timing comparisons in Table 3. Chen et al. report using similar hardware (Intel Core i7-3770 3.4GHz and 16GB memory), but we have had to scale the orignal images with the factor of 2^{-3} that they used to reduce the original image size. As for computational time, Table 3 indicates that the proposed method is the best and that the pixel-level based method[10] is the slowest—it takes 101 minutes to segment a 376×250-sized image. Although the superpixel-level based approach[10] greatly reduces the runtime from the original 101 minutes to 0.91 minutes, our method can make further improvement by reducing computational time to 0.19 minutes, which is approximately 532 and 5 times, respectively, faster than Chen et al.'s methods[10] and.[9]

The comparisons clearly show that the proposed method outperforms our previous approaches[26] and[6] since it is able to achieve comparable performance as,[6,26] while having improved generality and computational time. The comparisons also show that our approach outperforms Chen et al.'s algorithms[9,10] in terms of runtime while achieving similar extraction accuracy and generality.

5 Conclusion

We present an algorithm for automatically extracting special objects from medieval books. Starting from labeling the superpixels of the training images and also computing their BoF representations, our approach constructs a superpixel categorization classifier. The trained classifier is employed during the test stage to assist in identifying and finally localizing the superpixels that correspond

to special objects.

We extensively tested our approach on eight distinct historical books. As demonstrated by the results, the proposed method achieves very satisfactory performance—an overall precision and recall of up to 99% and 97%, respectively, and greatly improved time efficiency. As compared to our previous methods,[6,26] our algorithm has better performance in terms of both generality and computational time. This is attributed to the removal of dependency on the text-style related assumption in,[6,26] as well as the use of superpixels. Furthermore, comparison results demonstrate that the proposed method achieves significantly better runtime than Chen et al.'s recent work,[9,10] without sacrificing any accuracy and generality.

References

1 F. Shafait, D. Keysers, and T. M. Breuel, "Performance evaluation and benchmarking of six-page segmentation algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**(6), 941–954 (2008).

2 A. Asi, R. Cohen, K. Kedem, J. El-Sana, and I. Dinstein, "A coarse-to-fine approach for layout analysis of ancient manuscripts," in *14th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 140–145, IEEE (2014).

3 I. Bar-Yosef, N. Hagbi, K. Kedem, and I. Dinstein, "Line segmentation for degraded handwritten historical documents," in *10th International Conference on Document Analysis and Recognition*, 1161–1165, IEEE (2009).

4 A. Garz, A. Fischer, H. Bunke, and R. Ingold, "A binarization-free clustering approach to segment curved text lines in historical manuscripts," in *12th International Conference on Document Analysis and Recognition (ICDAR)*, 1290–1294, IEEE (2013).

5 Y. Yang, R. Pintus, E. Gobbetti, and H. Rushmeier, "Automated color clustering for medieval manuscript analysis," in *2015 Digital Heritage*, **2**, 101–104 (2015).

6 Y. Yang, R. Pintus, E. Gobbetti, and H. Rushmeier, "Automatic single page-based algorithms for medieval manuscript analysis," Tech. Rep. YALEU/DCS/TR-1525, Department of Computer Science, Yale University, New Haven, CT (2016).

7 H. R. C. T. U. of Texas at Austin, *Inside a Book of Hours* (accessed June 30, 2016).

8 R. Cohen, A. Asi, K. Kedem, J. El-Sana, and I. Dinstein, "Robust text and drawing segmentation algorithm for historical documents," in *Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing*, *HIP '13*, 110–117, ACM, (New York, NY, USA) (2013).

9 K. Chen, C.-L. Liu, M. Seuret, M. Liwicki, J. Hennebert, and R. Ingold, "Page segmentation for historical document images based on superpixel classification with unsupervised feature learning," in *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, 299–304, IEEE (2016).

10 K. Chen, M. Seuret, M. Liwicki, J. Hennebert, and R. Ingold, "Page segmentation of historical document images with convolutional autoencoders," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, 1011–1015, IEEE (2015).

11 T. M. Rath and R. Manmatha, "Word image matching using dynamic time warping," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, **2**, II–521 – II–527, IEEE (2003).

12 J. A. Rodríguez-Serrano and F. Perronnin, "Handwritten word-spotting using hidden markov models and universal vocabularies," *Pattern Recognition* **42**(9), 2106–2116 (2009).

13 R. Manmatha and J. L. Rothfeder, "A scale space approach for automatically segmenting words from historical handwritten documents," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(8), 1212–1225 (2005).

14 G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, "Text line and word segmentation of handwritten documents," *Pattern Recognition* **42**(12), 3169–3183 (2009).

15 R. Pintus, Y. Yang, and H. Rushmeier, "ATHENA: Automatic text height extraction for the analysis of text lines in old handwritten manuscripts," *ACM Journal on Computing and Cultural Heritage* **8**, 1:1–1:25 (2015).

16 G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, "Text line detection in handwritten documents," *Pattern Recognition* **41**(12), 3758–3772 (2008).

17 R. Saabni, A. Asi, and J. El-Sana, "Text line extraction for historical document images," *Pattern Recognition Letters* **35**, 23–33 (2014).

18 M. Mehri, M. Mhiri, P. Héroux, P. Gomez-Krämer, M. A. Mahjoub, and R. Mullot, "Performance evaluation and benchmarking of six texture-based feature sets for segmenting historical documents," in *International Conference on Pattern Recognition*, 2885–2890 (2014).

19 C. Grana, D. Borghesani, and R. Cucchiara, "Picture extraction from digitized historical manuscripts," in *Proceedings of the ACM International Conference on Image and Video Retrieval*, 22, ACM (2009).

20 X. Ren and J. Malik, "Learning a classification model for segmentation," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 10–17, IEEE (2003).

21 B. Fulkerson, A. Vedaldi, S. Soatto, *et al.*, "Class segmentation and object localization with superpixel neighborhoods.," in *ICCV*, **9**, 670–677, Citeseer (2009).

22 R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(11), 2274–2282 (2012).

23 E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in *Computer Vision–ECCV 2006*, 490–503, Springer (2006).

24 Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *Proceedings of the 6th ACM international conference on Image and video retrieval*, 494–501, ACM (2007).

25 K. Chen, H. Wei, J. Hennebert, R. Ingold, and M. Liwicki, "Page segmentation for historical handwritten document images using color and texture features," in *International Conference on Frontiers in Handwriting Recognition*, (2014).

26 Y. Yang, R. Pintus, E. Gobbetti, and H. Rushmeier, "Automated color clustering for medieval manuscript analysis," in *Digital Heritage*, IEEE (2015).

27 S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 1323–1330, IEEE (2011).

28 X. Boix, J. M. Gonfaus, J. Van de Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzàlez, "Harmony potentials," *International journal of computer vision* **96**(1), 83–102 (2012).

29 M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool, "SEEDS: Superpixels extracted via energy-driven sampling," in *Computer Vision–ECCV 2012*, 13–26, Springer (2012).

30 J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(8), 888–905 (2000).

31 P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision* **59**(2), 167–181 (2004).

32 A. P. Moore, J. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8, IEEE (2008).

33 O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in *ECCV 2010*, 211–224, Springer (2010).

34 A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *Computer vision–ECCV 2008*, 705–718, Springer (2008).

35 A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31**(12), 2290–2297 (2009).

36 D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(5), 603–619 (2002).

37 L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (6), 583–598 (1991).

38 D. G. Lowe, "Object recognition from local scale-invariant features," in *7th IEEE international conference on Computer vision*, **2**, 1150–1157, Ieee (1999).

39 H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Computer vision–ECCV 2006*, 404–417, Springer (2006).

40 S. O'Hara and B. A. Draper, "Introduction to the bag of features paradigm for image classification and retrieval," *arXiv preprint arXiv:1101.3354* (2011).

41 M. Van den Bergh, X. Boix, G. Roig, and L. Van Gool, "SEEDS: Superpixels extracted via energy-driven sampling," *International Journal of Computer Vision* **111**(3), 298–314 (2015).

42 C. Wolf, J.-M. Jolion, and F. Chassaing, "Text localization, enhancement and binarization in multimedia documents," in *Proceedings of16th International Conference on Pattern Recognition*, **2**, 1037–1040, IEEE (2002).

43 BeineckeMS10, "Beinecke rare book and manuscript library - Yale University."

44 BeineckeMS109, "Beinecke rare book and manuscript library - Yale University."

45 BeineckeMS310, "Beinecke rare book and manuscript library - Yale University."

46 Beinecke, "Beinecke rare book and manuscript library," (2014).

47 Beinecke, "Database - download scripts - http://hdl.handle.net/10079/cz8w9v8," (2014).

48 OpenCV, "OpenCV - open source computer vision library." http://opencv.org/ (2013).

49 BeineckeMS360, "Beinecke rare book and manuscript library - Yale University."

50 A. Fischer, A. Keller, V. Frinken, and H. Bunke, "Lexicon-free handwritten word spotting using character hmms," *Pattern Recognition Letters* **33**(7), 934–942 (2012).

51 A. Fischer, V. Frinken, A. Fornés, and H. Bunke, "Transcription alignment of latin manuscripts using hidden markov models," in *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*, 29–36, ACM (2011).

Ying Yang joined the Computer Graphics Group at Yale as a Postdoctoral Associate in March 2013. He received the B.E. degree in Information Security in 2006 and M.E. degree in Computer Science and Technology in 2009 from the School of Computer and Communication, Hunan University, China. From September 2009 to February 2013, he pursued his PhD at the School of

Engineering and Computing Sciences, Durham University UK. His research interests include digital watermarking/steganography, steganalysis, document and 3D shape analysis, and applications of computer graphics in cultural heritage.

Holly Rushmeier is a professor of Computer Science at Yale University. She received the BS, MS and PhD degrees in Mechanical Engineering from Cornell University in 1977, 1986 and 1988 respectively. Between receiving the PhD and arriving at Yale she held positions at Georgia Tech, NIST and IBM Watson research. Her current research interests include acquiring and modeling material appearance, applications of human perception to realistic rendering and applications of computer graphics in cultural heritage. She is a fellow of the Eurographics Association, an ACM Distinguished Engineer and the recipient of the 2013 ACM SIGGRAPH Computer Graphics Achievement Award.

List of Figures

3 Visualization of the special objects extracted using a specific classifier from several example images of pages of BeineckeMS10, BeineckeMS109, BeineckeMS310 and BeineckeMS360 (from left to right). Manuscript images courtesy of the Yale University.[43–45,49] Zoom in to observe details.

4 Visualization of the special objects extracted, using a universal classifier, from several example images of pages of BeineckeMS10, BeineckeMS109, BeineckeMS310 and BeineckeMS360 (from left to right). Manuscript images courtesy of the Yale University.[43–45,49] Zoom in to observe details.

5 Comparison of results obtained from using[6,26] (top row) and our specific classifier-based method (bottom row) to the Cambridge (left two columns) and Yale Gower (right two columns) manuscripts. Manuscript images courtesy of the Yale University and the University of Cambridge. Zoom in to observe details.

6 Comparison of the computational times used for extracting the special objects from the books (from left to right): *BeineckeMS10* ($\sim 3128 \times 2274$), *BeineckeMS109* ($\sim 2434 \times 1960$), *BeineckeMS310* ($\sim 3777 \times 3040$) and *BeineckeMS360* ($\sim 2083 \times 1825$). Note that the choice of our method (specific classifier-based or universal classifier-based) does not matter, since the runtime is exactly the same for both methods once a classifier is built.

List of Tables

3 Comparison between Chen et al.'s pixel-level method,[10] their superpixel-level method[9] and our approach on images from (3008×2000-sized) and Saint Gall (4992×3328-sized) in terms of computational time (in minute) required for an image. For fairness, similar to Chen et al., we applied our method to the resolution-reduced images with the scaling factor of 2^{-3} rather than to the original images. Note again that the choice of our method (specific classifier-based or universal classifier-based) does not matter when comparing timing.