Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 38 (2014) 76 – 83

Procedia
Computer Science

10th Italian Research Conference on Digital Libraries, IRCDL 2014

Illustrations segmentation in digitized documents using local correlation features

Dalia Coppi[a], Costantino Grana[a,*], Rita Cucchiara[a]

[a]*Dipartimento di Ingegneria "Enzo Ferrari", Università degli Studi di Modena e Reggio Emilia, Via Vignolese 905/b, 41125 Modena, Italy*

Abstract

In this paper we propose an approach for Document Layout Analysis based on local correlation features. We identify and extract illustrations in digitized documents by learning the discriminative patterns of textual and pictorial regions. The proposal has been demonstrated to be effective on historical datasets and to outperform the state-of-the-art in presence of challenging documents with a large variety of pictorial elements.

Keywords:
page segmentation; layout analysis; autocorrelation; SVMs

1. Introduction

Digitized documents play an important role in the preservation of historical contents and in their diffusion to the general public. Without digital editions the huge amount of old archives and documents would not be easily accessible. Digitization allows a pervasive diffusion and also the augmentation of masterpieces with multimedia details and appealing contents. Though, the huge amount of historical archives and books make desirable that their annotation and analysis were automatic and require the minimum user's intervention.

Pattern recognition and machine learning offer significant tools for automatically analyzing the content of digitized documents and improving their presentation. If Optical Character Recognition (OCR) methods almost yield completely reliable results, the task of identifying textual regions and separate them from other components of the page is more challenging especially in old documents. State of the art methods for Document Layout Analysis and segmentation have been proposed, and, among them, one of the most important is represented by Tesseract.[2] This OCR engine, sponsored by Google, not only offers a powerful tool for text recognition, but also provides layout analysis and image recognition modules. Despite the satisfactory

*Corresponding author. Tel.: +39-0592056265; fax: +39-0592056129.
Email addresses: dalia.coppi@unimore.it (Dalia Coppi), costantino.grana@unimore.it (Costantino Grana), rita.cucchiara@unimore.it (Rita Cucchiara)

doi:10.1016/j.procs.2014.10.014

results achieved on contemporary documents, the analysis of historical archives is still challenging due to the high variability of possible contents.

We propose a supervised learning approach to segment text and illustration of digitized old documents. The main novelty of our solution is the use of a texture feature based on local correlation, improved with respect to previous approaches thanks to an effective encoding aimed at detecting the repeating patterns of text regions and differentiate them from pictorial elements.

The paper is structured as follows: in Section 2 we briefly discuss the state of the art in Document Layout Analysis (DLA), Section 3 explains the main steps of our proposal, the page segmentation, feature extraction and classification. Section 4 describes in detail the features encoding method we propose. Finally, in Section 5 we report the performance evaluation and compare it against the state of the art, followed by conclusions and future work in Section 6.

2. Related Work

Document layout analysis is an active area of research and a vast number of works have been presented in the literature. The focus of the problem is often the segmentation of text regions and the subsequent Optical Character Recognition (OCR) step of both printed and handwritten text, but approaches dealing also with pictures segmentation have been studied.

Chen *et al.*[3] give a comprehensive survey on document image classification dividing it in three main components: the problem statement, the classifier architecture and the performance evaluation, separately analyzing each component:

- The *problem statement* is related to the set and type of documents to be analyzed.
- The *classifier architecture* is the core of the problem and includes the aspects related to page segmentation, feature representation and classification.
- *Performance evaluation* is also a critical and important component of a document classifier. The challenge is often the variety of documents considered, either fixed layout documents (books or forms) and flexible layout (newspapers) that inevitably produce different sets of classes as possible outputs. To this aim, a system for ground truthing a large amount of documents and a flexible XML schema has been introduced in.[4]

The page segmentation problem can be further decomposed in *Geometrical Layout Analysis* and *Logical Layout Analysis*. The former step is solely based on the geometric characteristics of the document image and aims at finding homogeneous content portions, while the purpose of the latter is to segment the page image into a hierarchy of regions based on the human- perceptible meaning of the content. Regions are therefore assigned a logical label (*e.g.*title, caption, paragraph) and a logical relation among the regions is determined (*e.g.*reading order, inclusion in the same article).

The geometrical analysis approaches can be categorized into top-down, bottom- up or mixed segmentation approaches. Top-down methods, such as XY cuts[5,6] or methods that exploits white streams[7,8] or projection profiles[9] are usually fast but tend to fail when dealing with complex layouts. Bottom-up methods are instead more flexible and process the image page from the pixel level and subsequently aggregate into higher level regions but with an higher computational complexity. These approaches are usually based on mathematical morphology, Connected Components (CCs), Voronoi diagrams[10,11,12] or run-length smearing.[13]

Many other methods exist which do not fit exactly into either of these categories: the so called mixed or hybrid approaches try to combine the high speed of the top-down approaches with the robustness of the bottom-up ones. Chen *et al.*[14] proposes a method based on whitespace rectangles extraction and grouping: initially the foreground CCs are extracted and linked into chains according to their horizontal adjacency relationship; whitespace rectangles are then extracted from the gap between horizontally adjacent CCs; progressively CCs and whitespaces are grouped and filtered to form text lines and afterward text blocks. Lazzara *et al.*[15] provides a chain of steps to first recognize text regions and successively non-text elements.

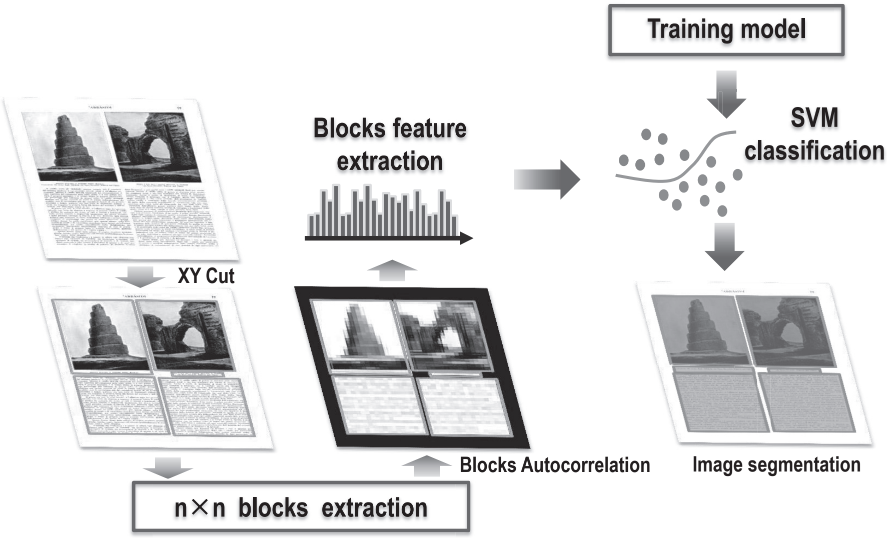


Fig. 1. System overview

Foreground CCs are extracted, then delimiters (such as lines, whitespaces and tab-stop) are detected with object alignment and morphological algorithms. Since text components are usually well aligned, have a uniform size and are close to each other, the authors propose to regroup CCs by looking for their neighbors. Filters can also be applied on a group of CCs to validate the link between two CCs.

Once homogeneous region are extracted, the other important subtask is the classification of the regions into a set of logical predefined classes (*e.g.*text blocks, tables, drawings, photos, etc.). The XY tree representation is a commonly used approach for describing the physical layout of the documents: it is used in[16] with Hidden Tree Markov Models to perform classification, and in[7] with decision trees. Feature vectors composed by a combination of different features are also common. Wang *et al.*[17] propose fixed length feature vectors composed by a combinations of run length encoding, correlation of text lines, spatial and area features.

In[18] a complete layout analysis is performed, dealing with text, picture, drawing, horizontal and vertical lines. The system described in this work could complement such a system, but is limited to only one of the aspects and the features used should be complemented with further image processing features in order to reach that kind of page description.

3. Page Layout Segmentation

We approach the page segmentation problem starting from the idea that textual and pictorial regions in a document are characterized by different local patterns: lines of text exhibit a regular structure which can be exploited to successfully differentiate them from illustrations in a classification framework. The method we propose can be decomposed in the steps depicted in Fig. 1. The geometric layout analysis is performed by extracting the main regions from the page using the XY cut segmentation, then each region is divided in small squared blocks of size n, and local correlation features are computed on each block and classified using a Support Vector Machine.

The XY cut is a well known recursive algorithm for top-down page segmentation. The method works by firstly projecting the pixels' values on the vertical and horizontal axis of the image and subsequently by

Algorithm 1 Recursive XY Cut (RXYC)

```
Input: image
Output: list                                          ▷ list of regions on the page

  Binarize(image)
  MorphologicalClosure(image)
  RXYC_Step(image)

  procedure RXYC_Step(ROI)
      Remove white borders
      vProj ← vertical projection (sum of rows values)
      hProj ← horizontal projection (sum of columns values)
      hCut ← first $y$ such as vProj($y$) $<$ Thresh
      if hCut then                                    ▷ Horizontal cut found
          $\overline{ROI}$ ← ROI
          $\overline{ROI}$.h ← hCut
          RXYC_Step($\overline{ROI}$)                 ▷ XYCut on the first sub-image
          $\overline{ROI}$.y ← ROI.y + hCut
          $\overline{ROI}$.h ← ROI.h − hCut
          RXYC_Step($\overline{ROI}$)                 ▷ XYCut on the second sub-image
      else
          vCut ← first $x$ such as hProj($x$) $<$ Thresh
          if vCut then                                ▷ Vertical cut found
              $\overline{ROI}$ ← ROI
              $\overline{ROI}$.w ← vCut
              RXYC_Step($\overline{ROI}$)             ▷ XYCut on the first sub-image
              $\overline{ROI}$.x ← ROI.x + vCut
              $\overline{ROI}$.w ← ROI.h − vCut
              RXYC_Step($\overline{ROI}$)             ▷ XYCut on the second sub-image
          else
              list ← list ∪ ROI
          end if
      end if
  end procedure
```

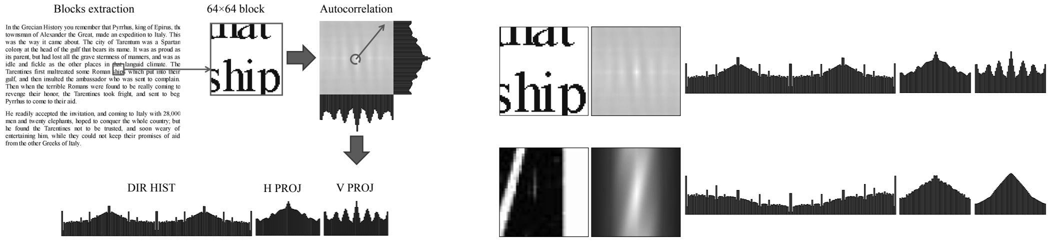
finding a low density region of the projection histograms, *i.e.*by finding the white spaces of the page. In this way the page is recursively segmented in rectangular regions.

We used the recursive XY cut with a preprocessing phase of binarization and morphological closure on the page in order to filter out the white interline spaces. The closure is performed with a squared structuring element of size 41×41 pixels. At each iteration the white borders surrounding the regions are removed before calling the next recursive step to find the internal cuts. Algorithm 1 provides the pseudo code of our approach to recursive XY cuts. By exploiting this algorithm, we obtain a segmentation of the page, usually corresponding to the two columns of text or parts of them, and, if existing, full page images.

4. Local Correlation Features

In order to distinguish between textual and pictorial regions we used a texture analysis method similar to the one proposed in[19] and.[20] The approach exploits the autocorrelation matrix, an effective feature for finding repeating patterns which is particularly suited in this case since textual textures have a pronounced orientation that heavily differs from that of illustrations. The autocorrelation function is defined as the cross correlation of a signal with itself, and represents a measure of similarity between two signals. Once applied to a grayscale image, it produces a central symmetric matrix, that gives an idea of the degree of regularity of the texture. The methods consists in the subdivision of the original image into square blocks of size n. The formal definition of the autocorrelation of a block is:

$$C(k,l) = \sum_{y=\max(0,l)}^{n-1+\min(0,l)} \sum_{x=\max(0,k)}^{n-1+\min(0,k)} I(x,y) \cdot I(x+k, y+l) \tag{1}$$



In the Grecian History you remember that Pyrrhus, king of Epirus, the townsman of Alexander the Great, made an expedition to Italy. This was the way it came about. The city of Tarentum was a Spartan colony at the head of the gulf that bears its name. It was as proud as its parent, but had lost all the grave sternness of manners, and was as idle and fickle as the other places in that languid climate. The Tarentines first maltreated some Roman ships which put into their gulf, and then insulted the ambassador who was sent to complain. Then when the terrible Romans were found to be really coming to revenge their honor, the Tarentines took fright, and sent to beg Pyrrhus to come to their aid.

He readily accepted the invitation, and coming to Italy with 28,000 men and twenty elephants, hoped to conquer the whole country; but he found the Tarentines not to be trusted, and soon weary of entertaining him, while they could not keep their promises of aid from the other Greeks of Italy.

(a) Descriptors computation.

(b) Feature vectors.

Fig. 2. (a) feature vectors computation from an image block. (b) image showing the autocorrelation on every block of a page and example feature vectors obtained from a text area and from an illustration.

where l and k are defined in $[-n/2, n/2]$. It is important that the size n of the blocks is chosen such that the repeating pattern of the textual blocks is highlighted.

Implementing the autocorrelation following this definition gives an algorithm with a high computational complexity, roughly proportional to n^4. According to the cross-correlation theorem the cross-correlation of two signals is equal to the product of the Fourier Transform of each one, where one of them has been complex conjugated.

$$f \star g \Leftrightarrow F' \cdot G \tag{2}$$

where F and G denote the transformed signals and F' is the complex conjugate of F. Since the autocorrelation is a special case of the cross- correlation, Eq. 2 becomes:

$$f \star f \Leftrightarrow F' \cdot F = |F|^2 \tag{3}$$

and for the Wiener-Khinchin theorem, the Fourier Transform of an autocorrelation function is the power spectrum, or equivalently, the autocorrelation is the inverse Fourier transform of the power spectrum. Following these properties, we efficiently compute the autocorrelation of the blocks only using two steps of the Fast Fourier Transform (FFT) with to a reduction of the complexity from $O(N^4)$ of the naive approach to $O(N \log N)$.

The result of the autocorrelation can be employed to extract an estimate of the relevant directions within the texture. Usually, the autocorrelation matrix is encoded with a *directional histogram*, a polar representation in which each direction is determined by an angle $[0°, 360°)$ and the bin value is given by the sum of the pixels along that direction.

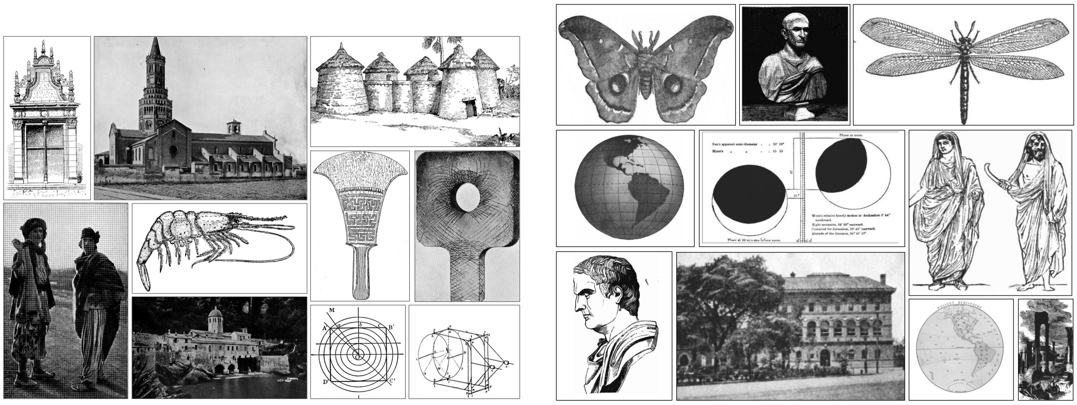
$$w(\theta) = \sum_{r \in (0, bs/2]} C(r\cos\theta, r\sin\theta) \tag{4}$$

Since the autocorrelation matrix has a central symmetry by definition, we consider only the first half of the direction histogram in the range $[0°, 180°)$. ω and r are quantized: the step of ω is set to $1°$ and the step of r is set to 1 pixel. Using this encoding a text block is characterized by peaks around 0° and 180° because of the horizontal dominant direction, conversely an image block is described by a generic multi-modal distribution.

We finally enrich the descriptor concatenating the directional histogram with the projections of the autocorrelation matrix along the vertical and horizontal directions in order to enhance the repeating pattern of the text lines. Fig. 2 provides a visual summary of the feature extraction process and few example results.

5. Experiments

In this section we report the results obtained on the digitized pages of the "Enciclopedia Treccani" (`http://www.treccani.it`). The encyclopedia consists of a set of volumes firstly published between 1925 and 1936. In our evaluation we only considered the first volume which is composed of 1183 pages. The original size of each digitized page is 22110 × 28819 pixels at 1 bpp, and we used a downscaled version with a factor of 0.125 along each dimension and converted to gray -scale so that the new images



(a) Treccani dataset

(b) Gutenberg13 dataset

Fig. 3. Sample images from the two selected datasets.

Table 1. Results in terms of number of TP, FN, FP. The values are the number of images of the three classes.

Dataset	Method	TP	FN	FP
Treccani	***Our Method***	361	5	132
	Tesseract	200	166	16
Gutenberg13	***Our Method***	524	11	62
	Tesseract	376	159	20

are 2763 × 3602 pixels at 8 bpp. The pages are two-column text with a number of illustration per page that may vary from 0 to 10. The main difficulty of this dataset, compared to other document analysis datasets, is the variety of graphical elements that is not limited to photos, but also includes drawings and charts. The overall number of manually annotated illustrations is 1157.

For a more comprehensive analysis we also built the Gutenberg13 dataset. This dataset, available online at `http://www.imagelab.unimore.it`, has been created using a set of publicly available e-books from Project Gutenberg (`http://www.gutenberg.org`). The e-books have been converted to grayscale with a resolution of 300 pixel/inch printing 4 pages for sheet, resulting in a two column text layout with a font size similar to the Treccani dataset. The total number of pages is 268 and each page is 2481 × 3509 pixels. Some example images of the two datasets are provided in Fig. 3(a) and in Fig. 3(b).

In our experiments we set the block size to 64 × 64 pixels, enough to consider a line of text. Recalling that our descriptor is the concatenation of the vertical and horizontal projection of the autocorrelation matrix and its directional histogram, we obtained a 308 dimensional feature vector for each block. We used an SVM with RBF kernel whose C and γ parameters have been estimated using cross validation (the values $C = 4096$ and $\gamma = 0.5$ have been used). The training set consists of 4000 blocks randomly sampled half from text regions and half from image regions.

We compared our proposal with the results given by the layout analysis module of Tesseract (`https://code.google.com/p/tesseract-ocr`), the Optical Character Recognition engine sponsored by Google. Since the image-find function of Tesseract is essentially based on morphological operations, in order to maximize Tesseract performance we threshold the pages to the maximum level of 255, meaning that everything that is not white is considered as black.

In order to evaluate the performance we solved the matching problem between the ground truth and the results obtained by the two methods exploiting the *Hungarian Method*. The nodes of the bipartite graph correspond to the GT bounding boxes and to the automatically extracted bounding boxes, while the edges are weighted using a measure of the overlap between bounding boxes. Given two bounding boxes l and r,

Table 2. Results in terms of % of TP, FN, FP of illustration pixels.

Dataset	Method	TP pxls (%)	FN pxls (%)	FP pxls (%)
Treccani	***Our Method***	99.57	0.43	4.53
	Tesseract	52.28	47.72	0.39
Gutenberg13	***Our Method***	99.50	0.50	11.50
	Tesseract	83.13	16.87	1.02

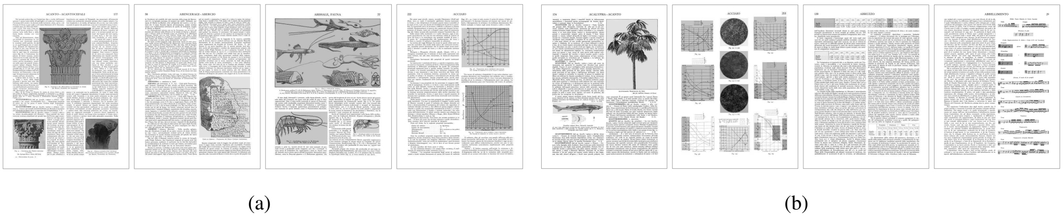


Fig. 4. Illustration segmentation obtained with our method. (a) Photographs, draws and charts correctly segmented. (b) Examples of oversegmented regions and wrong detections.

the weight is computed as:

$$w_{lr} = \frac{l \cap r}{l \cup r} \tag{5}$$

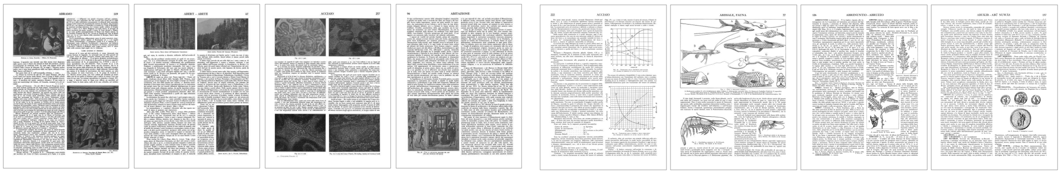
Results are reported in terms of the number of True Positives (TP), False Negatives (FN) and False Positives (FP). Table 1 shows the performance comparison of our method and Tesseract on the Treccani and Gutenberg13 datasets, reporting the number of images of the three classes (TP, FN, FP). Table 2, instead, focuses on the performance in terms of percentage of pixels with respect to the total number of pixels annotated as illustration.

As Table 1 shows, our method outperforms Tesseract on the Treccani dataset. Tesseract has indeed the main drawback of not being able to recognize most of the drawings although it makes a good job in finding photographs. Our proposal, instead, exploiting a supervised classification approach, has the capability of learning the correct classification also of drawings and charts. Table 1 also demonstrates how our method produces a higher number of false positives, but we would like to highlight that despite the high quantity, false positive results correspond to small areas as shown in Table 2. These errors usually correspond to portions of music sheets, tables or drawings as displayed in Fig. 4. Some examples of illustration segmentations obtained with Tesseract, that highlight the difficulty in extracting drawings and charts, are reported in Fig. 5.

The not negligible number of false positives introduced, may rise the question of what are we doing wrong: in fact false positive for pictorial blocks recognition means that some blocks might not be correctly classified as text and the OCR has not been applied, thus affecting the indexing of the document. On the contrary, a false positive for a text block may be less problematic (simply no text is associate to a block). In our experiments, the totality of false positives was on non text areas. In fact, the ground truth annotation has been performed rather selectively, so tables, music scores or lists of symbols were always excluded from the images annotations. Many times, they effectively resemble line drawings, thus they were included in our detection. The solution would be to train a specific detector for music scores, for example, but this would provide an unfair advantage with respect to a generic layout analysis system, such as the one we are comparing against.

6. Conclusion

In this paper we proposed a method to automatic extract illustrations from digitized historical documents. Starting from the assumption that text regions are characterized by a strong horizontal repeating pattern we introduced a descriptor based on the autocorrelation of squared blocks that demonstrated to be effective



(a) (b)

Fig. 5. Illustration segmentation obtained with Tesseract Layout Analysis module. (a) Examples of photographs correctly segmented. (b) Charts and drawings not detected as illustrations.

even with drawing and charts. We compared our method with Tesseract and we showed it is able to detect challenging illustration also when the state-of-the-art fails.

Our proposal represents an useful tool for a future enrichment of historical manuscripts with renovated contents. Starting from the appearance of the extracted images, and eventually exploiting the keywords contained in their captions, is in fact possible to automatically retrieve similar images, for example from the web.

1. Bertini M, Del Bimbo A, Serra G, Torniai C, Cucchiara R, Grana C, Vezzani R. Dynamic pictorially enriched ontologies for digital video libraries. *IEEE Multimedia* 2009;**16**:42–51.
2. Smith R. An Overview of the Tesseract OCR Engine. In: *Int Conf Doc Anal Recogn (ICDAR)* 2007, Washington, DC, USA; pp. 629–633.
3. Chen N, Blostein D. A survey of document image classification: Problem statement, classifier architecture and performance evaluation. *Int J Doc Anal Recogn* 2007;**10**:1–16.
4. Clausner C, Pletschacher S, Antonacopoulos A. Aletheia - an advanced document layout and text ground-truthing system for production environments. In: *Int Conf Doc Anal Recogn (ICDAR)* 2011; pp. 48–52.
5. Ha J, Haralick R, Phillips I. Recursive X-Y Cut using bounding boxes of connected components. In: *Int Conf Doc Anal Recogn (ICDAR)* 1995; pp. 952–955.
6. Cesarini F, Lastri M, Marinai S, Soda G. Encoding of modified x-y trees for document classification. In: *Int Conf Doc Anal Recogn (ICDAR)* 2001; pp. 1131–1136.
7. Appiani E, Cesarini F, Colla A, Diligenti M, Gori M, Marinai S, Soda G. Automatic document classification and indexing in high-volume applications. *Int J Doc Anal Recogn* 2001;**4**:69–83.
8. Pavlidis T, Zhou J. Page segmentation by white streams. In: *Int Conf Doc Anal Recogn (ICDAR)* 1991; pp. 945–953.
9. Esposito F, Malerba D, FA L. Machine learning for intelligent processing of printed documents. *J Intell Inf Syst* 2000;**14**:175–198.
10. Kise K, Sato A, Iwata M. Segmentation of page images using the area Voronoi diagram. *Comput Vis Image Understand* 1998;**70**:370–382.
11. Agrawal M, Doermann D. Voronoi++: A dynamic page segmentation approach based on voronoi and docstrum features. In: *Int Conf Doc Anal Recogn (ICDAR)* 2009; pp. 1011–1015.
12. Winder A, Andersen T, Smith E. Extending page segmentation algorithms for mixed-layout document processing. In: *Int Conf Doc Anal Recogn (ICDAR)* 2011; pp. 1245–1249.
13. Sebastiani F, Ricerche CND. Machine learning in automated text categorization. *ACM Computing Surveys* 2002;**34**:1–47.
14. Chen K, Yin F, Liu C. Hybrid page segmentation with efficient whitespace rectangles extraction and grouping. In: *Int Conf Doc Anal Recogn (ICDAR)* 2013; pp. 958–962.
15. Lazzara G, Levillain R, Geraud T, Jacquelet Y, Marquegnies J, Crepin-Leblond A. The Scribo module of the Olena platform: A free software framework for document image analysis. In: *Int Conf Doc Anal Recogn (ICDAR)* 2011; pp. 252–258.
16. Diligenti M, Frasconi P, Gori M. Hidden tree markov models for document image classification. *IEEE Trans Pattern Anal Mach Intell* 2003;**25**:519–523.
17. Wang Y, Phillips I, Haralick R. Document zone content classification and its performance evaluation. *Pattern Recogn* 2006;**39**:57–73.
18. Altamura O, Esposito F, Malerba D. Transforming paper documents into xml format with wisdom++. *Int J Doc Anal Recogn* 2001;**4**:2–17.
19. Journet N, Ramel J, Mullot R, Eglin V. Document image characterization using a multiresolution analysis of the texture: application to old documents. *Int J Doc Anal Recogn* 2008;**11**:9–18.
20. Grana C, Borghesani D, Cucchiara R. Automatic segmentation of digitalized historical manuscripts. *Multimed Tools Appl* 2011;**55**:483–506.