

# Page Classification for Meta-data Extraction from Digital Collections

Francesca Cesarini, Marco Lastrì, Simone Marinai, and Giovanni Soda

Dipartimento di Sistemi e Informatica - Università di Firenze  
Via S.Marta, 3 - 50139 Firenze - Italy Tel: +39 055 4796361.  
{cesarini, lastrì, simone, giovanni}@mcculloch.ing.unifi.it  
<http://mcculloch.ing.unifi.it/~docproc>

**Abstract.** Automatic extraction of meta-data from collections of scanned documents (books and journals) is a useful task in order to increase the accessibility of these digital collections. In order to improve the extraction of meta-data, the classification of the page layout into a set of pre-defined classes can be helpful. In this paper we describe a method for classifying document images on the basis of their physical layout, that is described by means of a hierarchical representation: the Modified X-Y tree. The Modified X-Y tree describes a document by means of a recursive segmentation by alternating horizontal and vertical cuts along either spaces or lines. Each internal node of the tree represents a separator (a space or a line), whereas leaves represent regions in the page or separating lines. The Modified X-Y tree is built starting from a symbolic description of the document, instead of dealing directly with the image. The tree is afterwards encoded into a fixed-size representation that takes into account occurrences of tree-patterns in the tree representing the page. Lastly, this feature vector is fed to an artificial neural network that is trained to classify document images. The system is applied to the classification of documents belonging to Digital Libraries, examples of classes taken into account for a journal are “title page”, “index”, “regular page”. Some tests of the system are made on a data-set of more than 600 pages belonging to a journal of the 19th Century.

## 1 Introduction

Meta-data are “data about data” and generally provide high level information about a set of data. In the field of Digital Libraries, appropriate meta-data allow users to effectively access digital material. When dealing with scanned books and journals three main categories of meta-data can be taken into account: administrative (e.g. the ISBN code of a publication), descriptive (e.g. the number of pages of a book), and structural (e.g. the title of a chapter). Whereas administrative and descriptive meta-data are frequently already available in electronic standard formats, or can be easily extracted from library cards, structural meta-data can be computed from a digital book only after an accurate analysis of the content of the book. In order to automatically extract structural meta-data from a scanned book, document image analysis techniques can be taken into

account. An useful task for the automatic extraction of structural meta-data is page classification, that is appropriate for both extracting page-level meta-data and narrowing the set of pages where to look for some meta-data. Page-level meta-data have a one-to-one correspondence of the meta-data with a physical page. Significant examples are the table of contents page, and pages containing pictures. Page classification can be helpful also for locating meta-data which appear only in some pages, for instance identifying the title page can help to retrieve the title of a book.

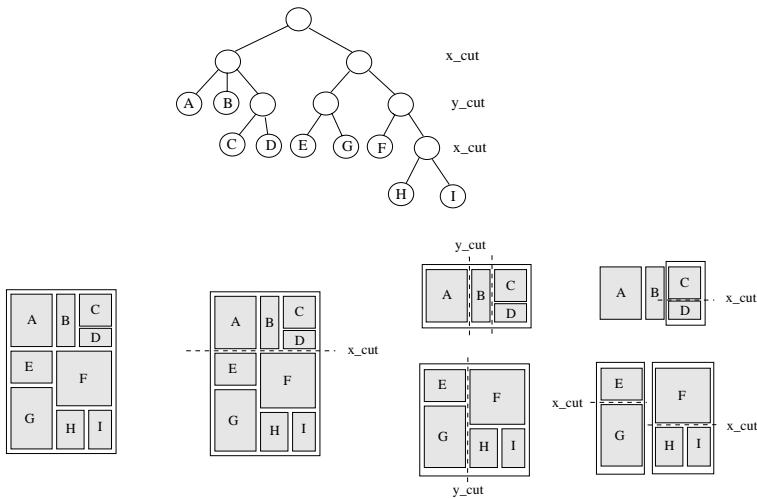
Page classification has been addressed with different objectives and methods. Most work concerned form classification methods that are aimed at selecting an appropriate reading method for each form to be processed [1,2]. Other approaches address the problem of grouping together similar documents in business environments, for instance separating business letters from technical papers [3]. In the last few years the classification of pages in journals and books received more attention [4,5]. An important aspect of page classification are the features that are extracted from the page and used as input to the classifier. *Sub-symbolic* features, like the density of black pixels in a region, are computed directly from the image. *Symbolic* features, for instance the number of horizontal lines, are extracted from a segmentation of the image. *Structural* features (e.g. relationships between objects in the page) can be computed from a hierarchical description of the document. *Textual* features, for instance presence of some keywords, are obtained from the text in the image recognized by an OCR (Optical Character Recognition) program.

In this paper we describe a page classification system aimed at splitting pages (belonging to journals or monographs in Digital Libraries) on the basis of the type of page; the input is a structural representation of the page layout. Examples of classes taken into account are *advertisement*, *first page*, and *index*. The structural representation is based on the Modified X-Y tree, a hierarchical description of page layout. The page is classified by using artificial neural networks (multilayer perceptron trained with Back-propagation) working on an appropriate encoding of the Modified X-Y tree corresponding to the page. This page classifier is under development in the domain of the METAe European project<sup>1</sup>. METAe is focused on the semi-automatic extraction of structural meta-data from scanned documents of historical books and journals, in order to make the digital conversion of printed material more reliable in terms of digital preservation. Key components of the project are layout analysis, page classification and specialized OCR for automatic meta-data extraction.

The paper is organized as follows, in Section 2 we describe the structural representation of documents, in Section 3 we analyze the proposed classification method. Experimental results are reported in Section 4, while conclusions are drawn in Section 5.

---

<sup>1</sup> METAe: the Metadata engine. <http://meta-e.uibk.ac.at>



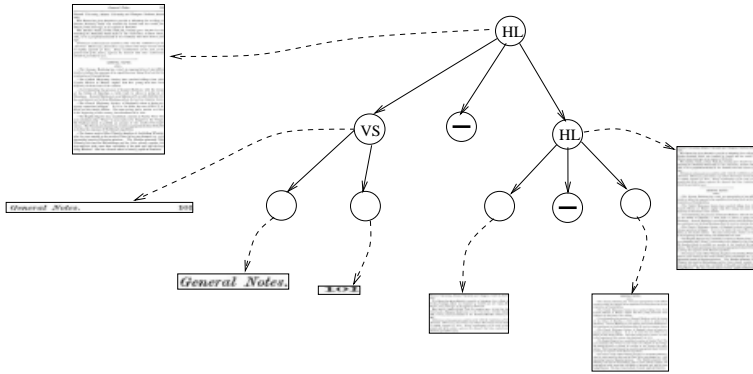
**Fig. 1.** Example of X-Y tree decomposition. In the upper-left part of the image we show the original page. The three images in the lower part describe the position of cuts at different levels of segmentation.

## 2 Document Layout Representation

The structure of the page is represented with a hierarchical representation (the Modified X-Y tree, MXY tree in the following) that is an extension of the classical X-Y tree representation. In this section, we first review the X-Y tree decomposition algorithm, and afterwards describe the MXY tree extension that is designed in order to deal with documents containing lines. Finally the building of the MXY tree starting from a symbolic description of the page is analyzed.

### 2.1 The Modified X-Y Tree

The Modified X-Y tree [6] is an extension of the X-Y tree designed in order to deal with documents containing lines in their layout. The X-Y tree [7] is a top-down data-driven method for page layout analysis. The basic assumption behind the X-Y tree segmentation is the property that elements of the page (columns, paragraphs, figures) are generally laid out in rectangular blocks. Furthermore, the blocks can usually be grouped in such a way that blocks that are adjacent to one another within a group have one dimension in common. The method consists in using thresholded projection profiles in order to split the document into successively smaller rectangular blocks [8]. A projection profile is the histogram of the number of black pixels along parallel lines through the document (see Figure 3 for an example). Depending on the direction of parallel lines the profile can be horizontal or vertical. To reduce the effects of noise, frequently a thresholded projection profile is considered. The blocks are split by alternately making horizontal and vertical “cuts” along white spaces which are found by using the thresholded projection profile. The splitting process is stopped when



**Fig. 2.** The MXY tree of a page. Dotted lines point out to images of regions described in the corresponding nodes. VL (HL) denote Vertical (Horizontal) cutting Line; VS (HS) denote Vertical (Horizontal) cutting Space. Nodes with a line indicate leaves corresponding to line separators.

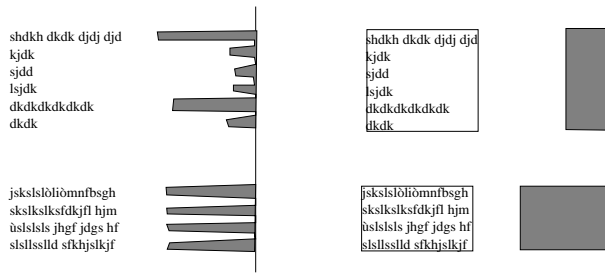
a cutting space (either horizontal or vertical) cannot be found or when the area of the current region is smaller than a pre-defined threshold. The result of such segmentation can be represented in a X-Y tree, where the root is for the whole page, the leaves are for blocks of the page, whereas each level alternately represents the results of horizontal (x\_cut) or vertical (y\_cut) segmentation. Figure 1 contains an example of a page segmented into blocks and the corresponding X-Y tree representation.

Two improvements to this approach have been proposed in literature. The *lossless optimization* proposed in [9] is based on the consideration that it is sufficient to perform the projections only up to the threshold  $T_p$ . In [10], projections profiles are obtained by using bounding boxes of connected components instead of single pixels in order to reduce the computational cost for calculating the projection profile. This method is tightly related to the symbolic extraction of MXY tree that we propose in Section 2.2.

When dealing with documents containing lines, the X-Y tree algorithm can give rise to uneven segmentations for the presence of regions delimited by lines. The MXY tree extends the basic X-Y tree approach by taking into account splitting of regions into sub-parts by means of cuts along horizontal and vertical lines, in addition to the classical cuts along white spaces. Each node of an MXY tree is associated either to a region of the page or to a horizontal or vertical line. In particular, internal nodes can have four labels (corresponding to two cutting directions, and two cutting ways), and leaves can have 4 labels. Figure 2 shows an example of a page with the corresponding MXY-tree.

## 2.2 Symbolic Building of Modified X-Y Tree

When using the X-Y tree (and also the MXY tree) for document segmentation, the purpose is to extract the blocks composing the page, and the algorithm is applied directly to the document image. To this purpose, appropriate algorithms



**Fig. 3.** Two approaches for computing the projection profile of textual regions. Left: the classic method which computes the profile directly from the image. Right: the profile is computed taking into account an uniform contribution for each block.

must be considered for the extraction and analysis of the projection profile, and for the location of separating lines (Section 2.1). However, the MXY tree data structure can be taken into account also for hierarchically representing the layout of the page, and this representation is helpful for understanding the meaning of items in the page, and also for page classification. In order to build an MXY representation of a document already split into its constituents blocks (e.g. provided by a commercial OCR), we developed an algorithm for the symbolic extraction of the MXY tree of a segmented document. Another advantage of the use of this algorithm is the possibility of integrating the algorithm with other approaches (e.g. bottom-up methods) that are less sensitive to the skew of the page, but which provide less structured representations of the page.

The input to the algorithm is a list of rectangular regions (corresponding to the objects in the page), and the list of horizontal and vertical lines. Since the input format is quite simple, various segmentation algorithms can be easily adapted in order to deal with this algorithm. The page classifier that we describe in this paper (Section 3), was integrated with a commercial OCR that is able to locate regions corresponding to text, and regions corresponding to images. Since horizontal and vertical lines are not provided by the OCR package, we look for them in zones of the image not covered by regions found by the OCR. Moreover, in order to locate segmentation points corresponding to horizontal and vertical white spaces, we compute an approximate projection profile (Figure 3). This profile is computed by considering an uniform contribution from each region extracted by the OCR both in the horizontal and in vertical direction. The amount of contribution to the profile depends on the average number of black pixels in each region, and this value can be either computed directly from the image or estimated on the basis of the number of characters in the region. A side effect of this approach is that noise in the image (not included in segmented regions) does not affect the MXY tree building. This approach is similar to the use of connected components for computing profiles [10] described in Section 2.1. The main difference is that in our approach we use whole regions instead of connected components, and the contribution to the projection profile is related to the density of the region.

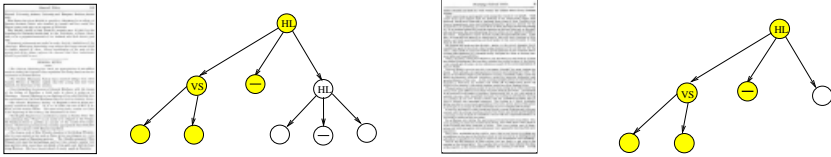
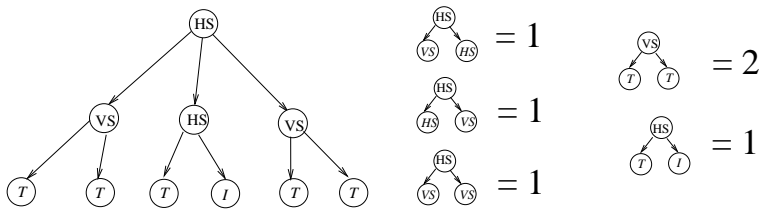


Fig. 4. A common subtree between the MXY trees of two pages of the same class.

### 3 Page Classification

Page classification is performed with a sequence of operations that is aimed at encoding the hierarchical structure of the page into a fixed-size feature vector. MXY trees are coded into a fixed-size representation that takes into account the occurrences of some specific tree-patterns in the tree corresponding to each document image. Lastly, this feature vector is fed to an MLP that is trained to classify document images according to the labels assigned to training data. Most classifiers (e.g. decision trees and neural networks) require a fixed-size feature vector as input. Some approaches have been considered for the mapping of a graph-based representation into a fixed-size vector. One approach (e.g. [11]) is based on the assignment of some pre-defined slots of the vector to each node and edge of the graph. This approach is appropriate when the maximum size of the graph is bounded, and when a robust ordering algorithm of nodes and edges is available. Another method is based on generalized N-grams [12], and the tree structure of logical documents is represented by probabilities of local tree node patterns similar to mono-dimensional N-grams, which are generalized in order to deal with trees. The generalization is obtained by considering “vertical” N-grams (describing ancestor and child relations) in addition to the more usual “horizontal” N-grams (corresponding to sibling relations).

In this paper, we use an encoding method that is used for the classification of trees describing the page layout. The basic idea that is behind this coding is the observation that similar layout structures often have similar sub-trees in the corresponding MXY representation (Figure 4). In real cases, because of noise and content variability, we cannot expect to find exactly the same sub-tree in all the trees of a given class. For instance, a block of text can be sometimes split into two or more sub-parts for other documents. Due to this size variability of the common sub-trees, we describe each tree by counting the occurrences of some tree-patterns composed by three nodes. This approach is somehow similar to generalized N-grams [12]. The main difference with respect to generalized N-grams is that the tree-patterns considered are composed by three nodes connected one to the other by a path in the tree. On the contrary, generalized N-grams include also patterns made by three siblings without taking into account their parent. Trees composed by three nodes can have two basic structures: one composed by a root and two children (referred to as balanced tree-pattern), and one composed by a root, a child, and a child of the second node. Four labels can be assigned to each internal node: HS, VS (for cuts along spaces), HL, VL (for cuts along lines). Each leaf can have four labels: hl (Horizontal line), vl (vertical line), T



**Fig. 5.** A simple MXY tree; in the right part of the figure we show the balanced tree patterns in the tree, with the corresponding occurrences. Non adjacent nodes are considered in the pattern having HS as root and VS as leaves.

(text region), and I (image). Leaves of tree-patterns can correspond either to a leaf of the MXY tree or to an internal node, consequently internal nodes of a tree pattern can have four values, whereas leaves can have eight values. Taking into account all the combinations of labels, 512 possible tree patterns can be defined. A special care is required by the balanced tree-patterns. Since siblings in the MXY tree can be ordered according to their horizontal or vertical position (depending on the cutting direction described in their parent), the relative position between contiguous blocks is preserved in this description. However, due to noise (or simply variable layouts of the documents), one sub-tree can differ from the reference one only for a node that is inserted between two representative siblings. In order to overcome this problem, when computing the tree-patterns appearing in a MXY tree, we look also for non-adjacent children (Figure 5), and this is another difference with respect to generalized N-grams.

The encoding just described takes into account only discrete attributes in the nodes of the tree. To consider also some information about the size of regions considered in the tree nodes, we added four values in the feature vector, that take into account the size of textual blocks belonging to the same tree-pattern. Textual blocks are labeled as “small” or “big” depending on the ratio of their area with respect to the area of the page. Blocks with area lower than a fixed threshold are labeled as “small”, whereas larger blocks are labeled as “big”. Therefore each tree-pattern containing textual leaves can belong to one of four classes according to the possible combinations of size labels of the leaves. The four features bringing size information are obtained by computing the relative distribution of each of the four combinations in the MXY tree. The addition of these features provides increased classification performance as discussed in the following section.

After extracting the vectorial representation of the MXY tree corresponding to a document, various algorithms can be taken into account for the actual classification. In this paper we addressed the problem with a classical MLP-based classifier (trained with the Backpropagation algorithm), which takes the normalized feature vector as input, whereas outputs describe, with an one-hot coding, the membership of each pattern. One problem with such an approach (that is common to other classification methods) is the large size of the feature vector, since many combinations of node labels can be considered. From a prac-



**Fig. 6.** Examples of the classes considered in our experiments. From left to right: advertisement, first page, index, receipts, regular.

tical point of view, we can easily find out that few tree-patterns can be found in actual documents of a given data set, as we will analyze in the next section.

## 4 Experimental Results

We made a set of experiments in order to evaluate the improvements in classification that can be achieved by considering the information about the relative size of textual blocks, and by using non-adjacent leaves when computing occurrences of balanced tree-patterns. Moreover, we analyzed the results that can be achieved using few patterns in the training set. The experiments are made with a data-set of pages belonging to a historical journal: the *American Missionary*, that is available in the on-line Digital Library *Making of America*<sup>2</sup>. We considered five classes having different layout, and appearing in each issue of the journal. Samples of the 5 classes are shown in Figure 6. Some classes have a very stable layout (e.g. the first page and the index), whereas other classes have a more variable layout (e.g. the advertisement class) and give rise to most errors.

For each experiment the documents are split into two classes: one is used for training, and the other is considered for testing purposes. The training set was furtherly divided into three sub-sets in order to perform a three-folder cross validation that allowed us to find the optimal number of epochs required for MLP training. A simple feature selection step was performed by removing from the feature vectors all the items that never appear in the training set. In this way we used a feature vector containing only 177 elements, instead of the 512 possible combinations of labels assigned to nodes. The classification results obtained with an MLP having 177 inputs, 20 hidden nodes, and 5 outputs are summarized in Table 1. A pattern is rejected when the difference between the highest MLP output and the next one is lower than 0.2 (the outputs are in the range  $[0,1]$ ).

As described in Section 3, in order to take into account the size of textual blocks, we added to the basic features four block size features. First, we selected the most appropriate threshold (that discriminates among “small” and “big”

<sup>2</sup> Document images can be downloaded from the web site of the collection: <http://cdl.library.cornell.edu/moa/>.

**Table 1.** Confusion table of the test set, when using the basic features.

True class	Output class					
	adv	first page	index	receipts	regular	Reject
adv	37	0	1	6	5	2
first page	0	56	0	0	0	3
index	0	0	53	0	0	0
receipts	0	0	0	57	1	1
regular	0	1	0	2	79	0

**Table 2.** Confusion table of the test set, when adding the textual block size features considering a threshold of 28 %.

True class	Output class					
	adv	first page	index	receipts	regular	Reject
adv	36	0	0	6	4	5
first page	0	58	1	0	0	0
index	0	0	52	0	0	1
receipts	1	0	0	56	1	1
regular	0	0	0	1	79	3

blocks), by evaluating the performances with different values of this threshold. From this experiment we selected a threshold value of 28 % as an optimal one (the corresponding confusion table is reported in Table 2). Comparing Table 2 with Table 1 we can see that a lower error rate is achieved when introducing the information about the block area.

Another experiment was performed in order to evaluate the gain that can be achieved when considering tree-patterns generated from non-adjacent siblings. In this experiment we generated feature vectors considering only adjacent siblings. Also in this case the threshold for size selection of blocks was 28 %, and we obtained an error rate of 6.8 % that is higher than the 4.7 % achieved when considering non-adjacent siblings. The last experiment concerns an empirical analysis of the requirements of the proposed method in terms of number of training samples (Table 3). From this experiment we can see that also with few training patterns, the performance are not excessively deteriorated.

## 5 Conclusions

We propose a method for the classification of document images belonging to Digital Libraries, that can be useful for the automatic extraction of structural meta-data. The method is based on a vectorial encoding of the MXY tree representing the document image. Each item in the feature vector describes the occurrences of some tree-patterns in the tree corresponding to the document. After an extensive test on a data-base of more than 600 pages we can conclude that an encoding taking into account non-contiguous siblings (and that uses information on the relative size of textual siblings) is appropriate; moreover with this approach we are able to obtain reasonable performances also when dealing

**Table 3.** Classification error versus number of training samples. Each value corresponds to the average of 10 tests obtained by randomly selecting the corresponding number of training samples. The test set is fixed and is composed by 300 samples different from those taken into account for training.

Error (%)	17.4	10.9	9.2	9.5	7.1	6.5	5.7	5.4	5.3	4.7
Number of training samples	30	60	90	120	150	180	210	240	270	300

with few training samples. Future work is related to the use of other feature selection approaches, and on tests on other kinds of documents. Moreover, other classifiers will be taken into account in place of the MLP-based classifier considered in this paper. We would like to thank Oya Y. Rieger from Cornell University for her help in collecting data taken into account for our experiments.

## References

1. S. L. Taylor, R. Fritzson, and J. Pastor, "Extraction of data from preprinted forms," *Machine Vision and Applications*, vol. 5, no. 5, pp. 211–222, 1992.
2. Y. Ishitani, "Flexible and robust model matching based on association graph for form image understanding," *Pattern Analysis and Applications*, vol. 3, no. 2, pp. 104–119, 2000.
3. A. Dengel and F. Dubiel, "Clustering and classification of document structure - a machine learning approach," in *Proceedings of the Third International Conference on Document Analysis and Recognition*, pp. 587–591, 1995.
4. J. Hu, R. Kashi, and G. Wilfong, "Document image layout comparison and classification," in *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, pp. 285–288, 1999.
5. C. Shin and D. Doermann, "Classification of document page images based on visual similarity of layout structures," in *SPIE 2000*, pp. 182–190, 2000.
6. F. Cesarini, M. Gori, S. Marinai, and G. Soda, "Structured document segmentation and representation by the modified X-Y tree," in *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, pp. 563–566, 1999.
7. G. Nagy and S. Seth, "Hierarchical representation of optically scanned documents," in *Proceedings of the International Conference on Pattern Recognition*, pp. 347–349, 1984.
8. G. Nagy and M. Viswanathan, "Dual representation of segmented technical documents," in *Proceedings of the First International Conference on Document Analysis and Recognition*, pp. 141–151, 1991.
9. T. M. Ha and H. Bunke, "Model-based analysis and understanding of check forms," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 8, no. 5, pp. 1053–1081, 1994.
10. J. Ha, R. Haralick, and I. Phillips, "Recursive X-Y cut using bounding boxes of connected components," in *Proceedings of the Third International Conference on Document Analysis and Recognition*, pp. 952–955, 1995.
11. A. Amin, H. Alsadoun, and S. Fischer, "Hand-printed arabic character recognition system using an artificial network," *Pattern Recognition*, vol. 29, no. 4, pp. 663–675, 1996.
12. R. Brugger, A. Zramdini, and R. Ingold, "Modeling documents for structure recognition using generalized N-grams," in *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, pp. 56–60, 1997.