Page Similarity and Classification

7

Simone Marinai

Contents

Abstract

Document analysis and recognition techniques address several types of documents ranging from small pieces of information such as forms to larger items such as maps. In most cases, humans are capable of discerning the type of document and therefore its function without *reading* the actual textual content. This is possible because the layout of one document often reflects its type.

S. Marinai
Dipartimento di Ingegneria dell'Informazione, Università degli studi di Firenze, Firenze, Italy
e-mail: simone.marinai@unifi.it

D. Doermann, K. Tombre (eds.), *Handbook of Document Image Processing and Recognition*, DOI 10.1007/978-0-85729-859-1_7,

For instance, invoices are more visually similar to one another than they are to technical papers and vice versa. Two related tasks, page classification and page retrieval, are based on the analysis of the visual similarity between documents and are addressed in this chapter. These tasks are analyzed in this chapter in a unified perspective because they share several technical features and are sometimes adopted in common applications.

Keywords

Block distance • Global features • Graph • Page classification • Page clustering • Page representation • Page retrieval • Region classification tree • Tree distance

Introduction

The first applications of document classification have been considered in the context of office automation, in particular in form processing applications. In this context, form classification methods are aimed at selecting one appropriate interpretation strategy for each incoming form [29, 47]. Other approaches addressed the problem of grouping together similar documents in business environments, for instance, separating business letters from technical papers [18]. One early approach for business document classification was presented in [48] where documents belonging to different classes are identified on the basis of the presence of class-specific landmarks. The rules implemented to identify the landmarks are context-based and rely also on the text recognized by an OCR engine. Since no positional information is modeled, the classification performed in this case is of little interest for layout-based page classification. More recently the classification of pages in journals and books received more attention [25,45], in particular in applications related to Digital Libraries (DL).

When considering the visual appearance of documents, we can notice that in different application scenarios, the pages can be classified taking into account different features of the pages. In some cases, for instance, when dealing with preprinted forms, the categories have a fixed layout and two documents could be split apart just because one ruling line is placed in a different position in the two cases. Some documents, for instance, papers published in one specific journal, are more variable but are still constrained to obey to some explicit (or implicit) rules. In other cases the documents can be distinguished only considering their textual content and the layout similarity is of little interest. As a consequence, also the visual similarity between documents can vary by application. Some examples of different classes considered in the context of Digital Libraries are shown in Fig. 7.1.

One task related to page classification is Web page classification, or Web page categorization, where techniques related to text categorization [43] can be integrated with approaches that analyze the Web page layout [46] and the structure inferred from hyperlinks [42].

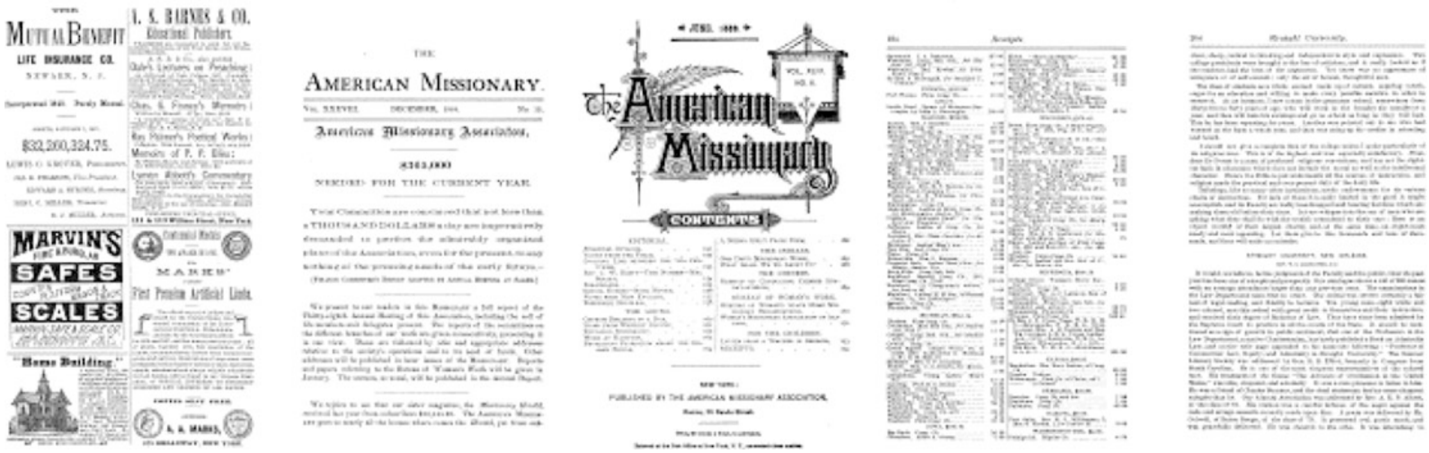

Fig. 7.1 Examples of the five classes considered in [10]. From *left* to *right*: advertisement, first page, index, receipts, regular

Evolution of the Problem

In the late 1990s, thanks to several Digital Library projects, large collections of digitized documents have been made available on Internet or in off-line collections. The first attempts to extend page classification techniques to digitized document in DLs such as books and journals needed to face the intrinsic ambiguity in the definition of classes that may change in different application domains. As a consequence, in some areas page classification techniques evolved in layout-based document image retrieval where pages are sorted on the basis of the similarity with a query page, rather than being explicitly labeled with one class.

Applications

There are various applications in Document Image Analysis and Recognition (DIAR) where the comparison of documents by visual appearance is important. Examples include document genre classification, duplicate document detection, and document image retrieval.

A document genre [6, 13] is a category of documents characterized by similarity of expression, style, form, or content. In this chapter we are more interested on the visual genre that dictates the overall appearance of a document. Genre classification can be used to group together documents that should be processed with document analysis algorithms tuned on specific types of documents so as to improve the overall performance.

Other applications of document classification include the automatic routing of office documents in different work flows (e.g., identifying the type of one filled form) and the automatic archiving of documents, especially in the field of Digital Libraries.

One important issue in page classification is the a priori definition of an exhaustive set of classes that should not change later. This issue is particularly important in DL applications where the set of layout classes may be modified when switching from one collection to another. Moreover, different user needs could require different labels assigned to the same pages. For instance, general public may be more interested on macroscopic page differences, e.g., looking for pages with illustrations. On the opposite, scholars of typography may look for fine-grained differences, e.g., looking for pages with illustrations in specific zones of the page. To address these ambiguities in the definition of classes, recent applications have considered layout comparison techniques in document image retrieval frameworks often applied in the context of Digital Libraries [35]. In DLs the document image retrieval based on layout similarity offers to users a new retrieval strategy that was possible before only by manually browsing documents (by either interacting with physical books-journals or dealing with online images on DLs).

Page classification and retrieval can be considered both for scanned documents and for born-digital documents (see ▸Chap. 23 (Analysis of Documents Born Digital)). One typical application of interest in the case of born-digital documents is the identification of pages containing the table of contents of books that can be useful to allow an easy navigation of book content [37].

Main Difficulties

One significant difficulty of page classification and retrieval is the intrinsic ambiguity of the page similarity that arises when the same set of documents is used in different applications. For instance, pages in a collection of technical papers can be grouped considering the role of the page in each paper (e.g., title page vs. bibliography page) or according to the typographical rules of a given publisher (e.g., IEEE vs. Springer). This is in contrast, for instance, with character recognition where the class of a given symbol is in most cases not ambiguous and independent of the application.

As discussed by Bagdanov and Worring in [6], page classification can be addressed at two levels of detail. Coarse-grained classification is used to distinguish documents with a significant difference of characteristics, for instance, separating technical articles from book pages. At a lower resolution, a fine-grained classification is adopted to identify subclasses of documents that would otherwise be grouped together from a coarse-grained point of view. For instance, we can in this case distinguish pages of articles typeset by IEEE or by Springer.

Chapter Overview

This chapter is organized taking into account the features shared by page classification and page retrieval. In particular, section "Page Representation" addresses the main approaches that have been considered to represent the page layout for either page classification or retrieval. In some cases, the same representation can be used to solve both problems. Several levels can be considered in the page representation. Each representation implicitly defines the possible approaches that can be adopted to compute the distance (or similarity) between pages. Alternative approaches to compare page layouts are described in section "Page Comparison." Section "Region Classification" briefly summarizes some approaches that have been proposed to perform region classification. The latter task aims at assigning a suitable label to zones in the pages that have been previously identified as containing a uniform content. Region classification is described in this chapter, even if it is closely related to more general layout analysis techniques (▶Chap. 5 (Page Segmentation Techniques in Document Analysis)), for two main reasons. First, some approaches for region classification are similar to page classification techniques. Second, page classification and retrieval often takes into account labeled regions that can be obtained as described in section "Region Classification."

The next three sections analyze how page representation and page similarity can be put together in order to implement sections "Page Classification," "Page Retrieval," and "Page Clustering." Some representative systems are summarized in these sections according to the various page representations considered. Due to the large number of different application domains that have been addressed in recent years, it is not possible to provide an effective comparison of the performance achieved by the systems. For page classification we report the classification accuracy of some methods that are discussed in the paper. We also highlight the main techniques used for performance evaluation in page retrieval systems. The section "Conclusions" closes this chapter.

Page Representation

In many application domains, users are generally capable of discerning the type of document and therefore its function without *reading* the actual text content since the layout structure of a document often reflects its type.

The visual appearance of a document is completely determined by the whole set of pixels in its image representation. Even if this description is complete, it is in general too complex to directly process the pixels acquired by the document scanner. Besides computational problems, higher-level representations are needed in order to extract a more abstract page representation that is required to achieve generalization in the classification and retrieval processes. Otherwise, over-fitting can occur, giving rise to the identification of similar pages only when these are near duplicate one of the other. According to Hu et al. [26] low-level page representations based, for instance, on bit maps are easy to compute and to compare, but do not allow for a structural comparison of documents. High-level representations are, however, more sensitive to segmentation errors.

Therefore, one essential aspect of page classification and retrieval systems is the kind of features that are extracted and used to represent the page. Sub-symbolic features, like the density of black pixels in a region, are usually computed directly from the image. Symbolic features, for instance, related to horizontal and vertical ruling lines, are extracted starting from one preliminary segmentation of the page. Likewise, structural features, such as relationships between objects in the page, are computed from a suitable page segmentation. Some techniques take into account textual features, such as the presence of keywords, that can be identified considering the text in the image recognized by an OCR (Optical Character Recognition) engine. The latter techniques are less relevant for this chapter that mostly deals with layout-based page classification and retrieval.

In the rest of this section, we analyze the principal approaches that can be used to describe the page layout. These approaches are also summarized in Table 7.1. The page comparison techniques that are used either in page classification or retrieval are described in the next section.

Global Page Representations

The most straightforward adaptation of statistical pattern classification techniques to page classification and retrieval leads to the development of approaches that

Table 7.1 Main approaches used in page representation for classification and retrieval

Representation		Description	Ref.
Global features	Page features	Image features computed at the page level	[6, 15]
	Zoning: image	Image features computed for each grid zone	[16, 26, 49]
	Zoning: character/symbol	Features computed from characters and accumulated in each grid zone	[16, 45]
	Zoning: region	Features computed from each region and accumulated in each grid zone	[32, 40]
List of blocks	**Lists of text lines**	Features computed for each text line. Page represented by the set of textlines	[27]
	Discriminative blocks	Sub-images discriminate one class from the others	[3]
	Layout blocks	Layout regions are stored in the list	[39]
	Lists of lines	Ruling lines are used to describe forms	[30]
Structural representation	**Graph**	Graph nodes correspond to layout regions; edges describe relationships between nodes	[4, 23]
	Tree	Hierarchical page decompositions (e.g., X-Y trees)	[2, 17, 21]

represent the pages with global image characteristics. These features are in most cases arranged into fixed-size feature vectors that are subsequently used as inputs to classifiers.

Documents containing a large amount of text can be described with features computed from the connected components. On the opposite, form documents (whose processing and classification are also described in ▸Chap. 19 (Recognition of Tables and Forms)) can be represented on the basis of ruling lines that characterize the page layout. In the latter case, due to the fixed page structure, the lines are organized in a structural representation (section "Structural Representations").

When dealing with heterogeneous documents, the features are in most cases extracted with low-level image processing techniques. For instance, morphological operations and texture-based features are often considered.

In [15] a histogram for each of five features extracted from the page to be described is computed. The features are the word height, the character width, the horizontal word spacing, the line spacing, and the line indentation. Each histogram is smoothed with a standard kernel function.

Global page features computed by applying morphological operations on the binarized page image are extracted at multiple resolutions in [6]. The input image is morphologically opened with horizontal (x) and vertical (y) structuring elements of variable sizes. The page is subsequently mapped to one two-dimensional function that considers the area of the black zones in the transformed image obtained

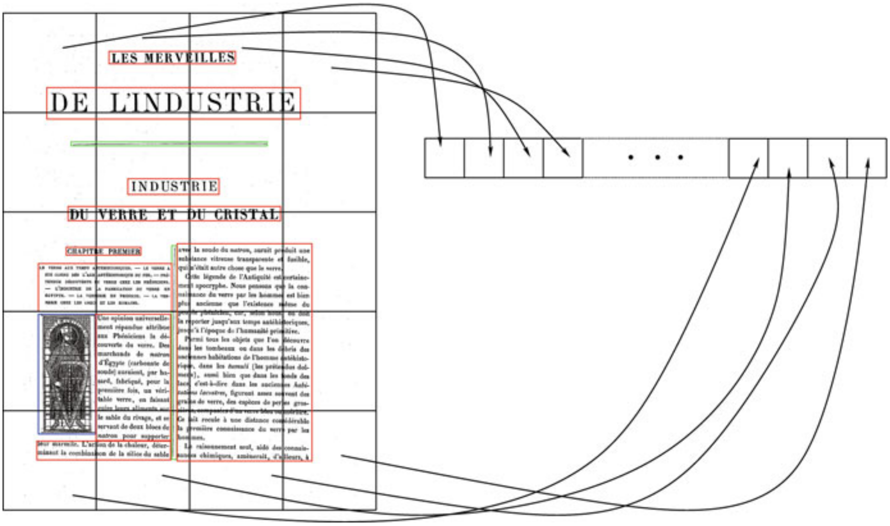

Fig. 7.2 One page with the grid used to compute the zoning vectorial representation

with a combination of x and y values. This distribution is then sampled in the x and y directions obtaining a rectangular size distribution. The size of this high-dimensional feature vector is reduced with Principal Components Analysis (PCA) so as to perform a feature extraction and allow an easy comparison of pages in the reduced space.

One widely used approach to represent the page with a fixed-size feature vector is based on the computation of the desired features in the zones identified by a grid (with sides parallel to the page border) superimposed to the page [26, 49]. This approach is often referred to as zoning. One example is shown in Fig. 7.2 where one 4 × 5 grid is overlapped to page on the left. The features computed for each zone are concatenated row by row in the fixed-size feature vector represented on the right part of the illustration.

One of the first systems proposed to perform page retrieval [16] considers global page features stored in an 80-dimensional feature vector. Some of the features are related to the distribution of *interest points* (computed from high curvature points in the characters' contour), while additional information is obtained by calculating the density of connected components in each cell of one grid overlapping the page.

A zoning-based approach is described in [45] where the features are extracted from one page considering a grid overlapped to the image. Zones in the grid are called windows and the minimum size of windows is defined to assure that each window contains enough objects. Four types of windows are extracted on the basis of the zones identified by the grid: *basic* window (the smallest area defined by the grid), *horizontal* and *vertical strip* windows (a set of basic windows aligned horizontally or vertically), and the *page* window that covers the whole page. Several features are then computed from each window. Most features are extracted from the connected components in the zone (e.g., the median connected component width) or directly from the image (e.g., the foreground-to-background pixel ratio). Specific features are computed for composite window types such as the column and row gaps. The overall description of a page is obtained from the concatenation of the features extracted from each zone, and therefore, standard statistical classifiers (such as decision trees) can be used to identify the page classes.

Another global representation whose features are related to zoning is described in [40] where blocks with uniform content are first extracted from each page. One new abstracted image containing only the block edges is then used to compute the global features. In particular, the horizontal and vertical projection profiles of the edges image are computed and concatenated, obtaining one feature vector of size $w + h$ where w and h are the page width and height, respectively.

To balance low-level and high-level page representations, Hu et al. propose in [26] the *interval encoding* that allows to encode region layout information (i.e., intrinsically structural) into fixed-size vectors that can be easily compared each other using standard vectorial distances such as the Euclidean one. Pages are sampled with a grid of m rows and n columns. Each item in the grid is called bin and is labeled as text or white space according to the prevailing content of the corresponding area.

One different way to obtain a fixed-size representation from a variable number of regions is described in [32] where each document is indexed by considering a set RB of representative blocks. The representative blocks are identified by clustering, with k-means, all the blocks in the collection. The distance between blocks used by the clustering algorithm is based on the overlapping similarity or Manhattan distance between blocks. The features of the representative blocks are obtained by computing the average position of the blocks in each cluster. Each document is afterwards represented by measuring the similarity of each block in the page to each Representative Block and then aggregating all the similarities of page blocks in a feature vector having size $|RB|$. Once this page representation is computed, it is possible to evaluate the page similarity by means of the cosine of the angle between the feature vector of the query page and the feature vector of each indexed page.

In [11] global and structural features (computed from an X-Y tree page decomposition) are merged in the page representation. The global features allow to obtain one representation that is invariant with respect to the book dimensions.

Representations Based on Lists of Blocks

Some of the page representations described in the previous section consider as input the blocks identified in the page. In this section we analyze approaches where the page is explicitly represented with a list of objects (either blocks or ruling lines) rather than summarizing these objects with global numerical features. This type of representation is graphically depicted in Fig. 7.3 where one sample image is shown on the left with colored rectangles corresponding to homogeneous regions. The whole page is represented with one linked list of items that is shown on the right part of the illustration.

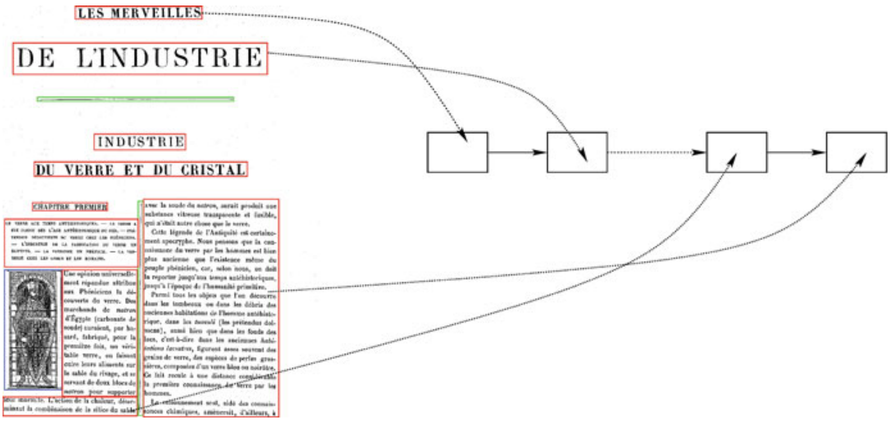

Fig. 7.3 Page representation with a list of blocks

The lists of objects are in most cases matched with analogous lists extracted from reference pages to compute the page similarity. This page representation is more accurate with respect to global page representations, but it is more difficult to handle by comparison algorithms. One of the reasons is the variable size of representations of different but similar pages.

Many algorithms in layout analysis address Manhattan layouts where the page is assumed to be both physically and logically composed by upright rectangles containing uniform content (e.g., text and graphics). ▸Chaps. 5 (Page Segmentation Techniques in Document Analysis) and ▸6 (Analysis of the Logical Layout of Documents) contain extensive descriptions of layout analysis algorithms.

Document images can be described by means of blocks that correspond to text lines. Huang et al. [27] exploit this representation by describing the page layout with the quadrilaterals generated by all pairs of lines. To speedup the retrieval, the quadrilaterals are clustered and cluster centers in the indexed pages are compared with the query one.

While most methods to identify blocks in the page are class independent, it is possible to identify discriminant areas in the page that can act as signatures for classes. In form classification, Arlandis et al. [3] describe a method based on the identification of discriminant landmarks (the δ-landmarks) that are sub-images suitable to discriminate one class from the others. The set of δ-landmarks of one class contains the sub-images having a significant dissimilarity with respect to the other classes. The class models consist of sets of δ-landmarks and the dissimilarity between sub-images and δ-landmarks is used to classify unknown forms.

One page representation based on a list of blocks is described in [39] where rectangular regions are stored in a Component Block List (CBL) and sorted considering the position of the bottom-left region corner. Page layouts are matched considering the block location and the size attributes. The CBL of one template page is called Template Block List (TBL) and the page classification is achieved by finding the most similar TBL to the CBL of the unknown page. The comparison is

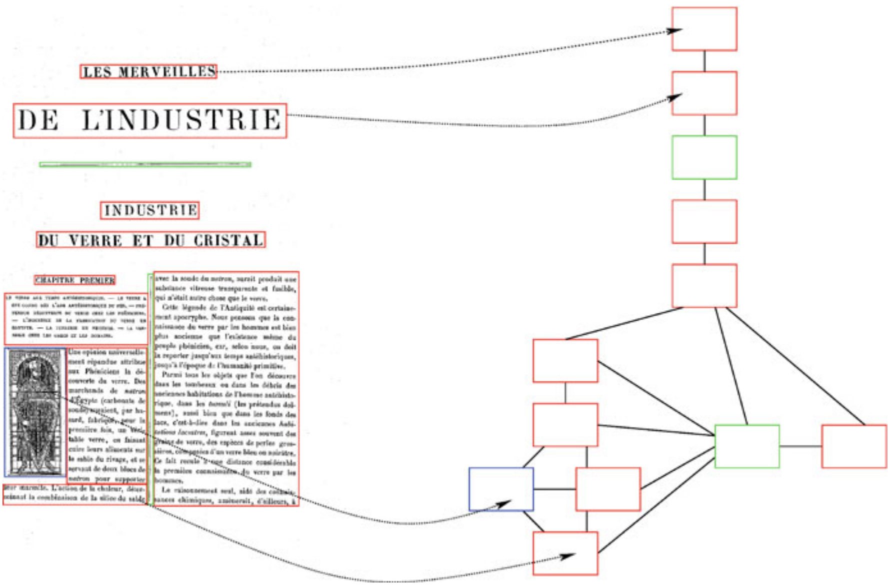

Fig. 7.4 Page representation with an adjacency graph

made with a simple block-matching algorithm that finds for each block in TBL the most similar block in CBL, according to size and position of blocks.

A similar approach can be applied in form classification by considering lists of ruling lines instead of lists of blocks. Jain and Liu [30] propose a form retrieval system where the form class is identified considering one form signature from the lists of horizontal and vertical frame lines. Collinear horizontal and vertical lines implicitly define a grid that contains the form cells. The form signature describes the number of cells contained in each grid element. To deal with missing lines in actual forms (due to noise or design differences among forms of the same class), several signatures are computed for each class considering alternative forms.

Structural Representations

Structural representations of the page layout can be based either on graphs or on trees.

Graph nodes correspond in most cases to text regions that are described with numerical attributes computed from the position and size of the region, from information related to the text size, and from features computed from text lines. Graph edges represent the adjacency relationships between nodes. One example is shown in Fig. 7.4 where the page used in the previous examples is now described with one adjacency graph. Each region is in this case represented by one graph node and the edges link neighboring regions.

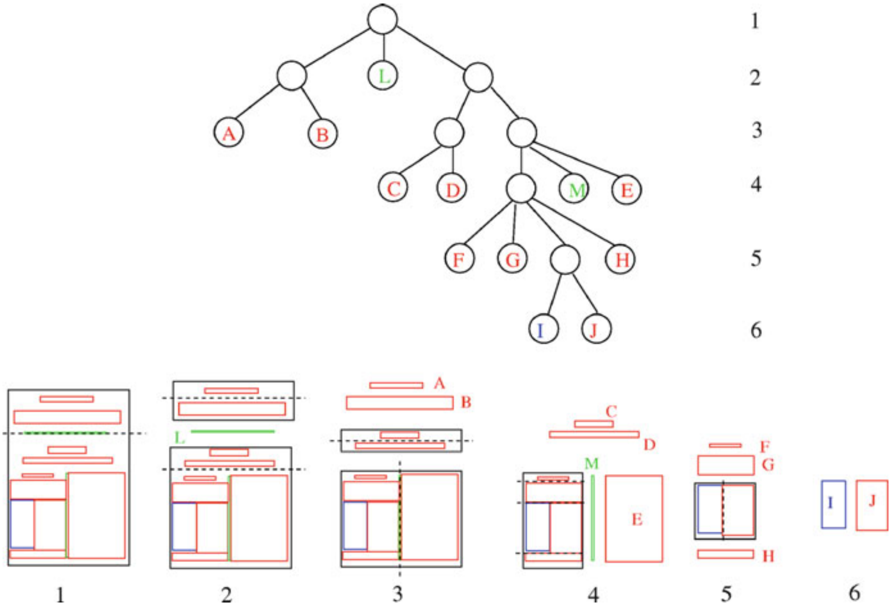

Fig. 7.5 Example of X-Y tree

In [4] *Attributed Relational Graphs* (ARG) are used to represent the page layout by considering the neighboring relationship of zones in the page. Document genres (classes) are represented by First-Order Random Graphs (FORG). An FORG is trained from documents belonging to one class. An unknown page described with an ARG is then classified by identifying the FORG with maximum conditional probability.

A graph-based approach is proposed by Gordo and Valvenyl in [23]. In the cyclic polar page representation, the nodes of the graph correspond to regions in the page and are labeled with the area and the type of region (text or non-text). The edges connect each node with the center of mass of all the regions, and therefore, one complete bipartite graph is built. Edges are labeled with their length and with the angle formed with adjacent edges. This polar graph can be mapped to one sequence of items labeled with node attributes. A cyclic shift of the sequence defines one equivalent representation of the same page.

Taking into account the hierarchical nature of document images, it is natural to adopt tree-based representations of the page layout. Tree-based representations are appropriate also because some segmentation techniques, such as the recursive X-Y tree, are based on a recursive decomposition of the page (▶Chap. 5 (Page Segmentation Techniques in Document Analysis)). The root of an X-Y tree is associated to the whole page and the corresponding region is recursively segmented, splitting the page along horizontal or vertical white spaces. The tree in Fig. 7.5 describes the regions in the page used in the previous examples in an X-Y tree.

The bottom part of the illustration represents the segmentations made at the various tree levels. In this example we show also cuts along horizontal and vertical ruling lines (e.g., the first cut is made along a horizontal line).

In several application domains (in particular, when dealing with forms) ruling lines are an essential element of the page layout. To deal with documents containing ruling lines, one modified X-Y tree (the MXY tree) has been introduced by Cesarini et al. [8] and used in some tree-based page classification systems (e.g., [2, 7, 19, 21]).

The X-Y tree-based approaches to page classification and retrieval take advantage of the limited variability of the tree representation when representing different, but structurally similar, layouts. On the other hand, the segmentation and representation with X-Y tree is difficult in case of noise, skew, and in general when dealing with non-Manhattan layouts.

An early tree-based representation of the page layout has been presented in [17] where a decision tree (the Geometric Tree, *GTree*) is used to model the layout classes by describing a hierarchy of possible object organizations. In analogy with X-Y trees, the pages are described with a recursive segmentation of the page along white spaces. In this representation, each example is one leaf of the *GTree* while intermediate nodes are used to identify cuts shared by different classes. Cuts shared by many documents are described in nodes close to the root, whereas class-specific cuts are contained in nodes closer to the leaves.

To classify one unknown document, the tree is traversed from the root to one leaf that identifies the page class. In each node encountered in this navigation, the input document is compared with the cuts defined in the node's children and the best path is followed.

Another approach that uses decision trees to classify pages represented with X-Y tree is described by Appiani et al. in [2]. Nodes in the Document Decision Tree (DDT) contain MXY trees because cuts along horizontal and vertical ruling lines are required to segment forms and other business documents that are addressed by the system. Both DDT building and traversal are based on sub-tree matching between as described in section "Tree Distance."

The use of one X-Y tree representation for unsupervised page classification is described in [34]. Each node, v, describes a region with one feature vector $f^v = (f_1^v, f_2^v, f_3^v, f_4^v)$, where f_1^v,f_2^v, and f_3^v denote the average character font size, the level in tree, and the X coordinate of the region center. If v is a node on a Y projection profile, f_4^v is the minimum vertical distance between v and its top or bottom sibling on the profile. The tree matching is performed with a tree-edit distance algorithm (section "Tree Distance").

Even if X-Y trees and related data structure have been the prevailing tree representations adopted in document image classification and retrieval, other approaches have been considered as well. For instance, quadtrees are used in [41] to address form classification. Another hierarchical page representation is the L-S tree (Layout Structure Tree) proposed by Wei et al. in [53]. Even if using a different notation, the page decomposition and representation addressed by L-S trees is analogous to X-Y trees.

Page Comparison

In this section we analyze how the distance between page descriptions is used to compare two pages. This distance can be used both to classify the pages and to rank the pages most similar to one query page. The main difficulties when computing the page similarity are due to layout differences induced by low-level segmentation errors and to variations in the design of documents that should be grouped in the same class.

When pages are represented with global features (section "Global Page Representations"), p-norms are natural choices to compute the distance between pages. For instance, [40] adopts the L_1 distance to compare global vectors describing the pages.

One widely adopted strategy to map images in a fixed-size feature vector is the zoning approach described in section "Page Representation." When dealing with these representations, one significant problem is related to the misalignment of pages that are very similar but horizontally and/or vertically displaced. Approaches based on the edit distance can address this problem, but the high computational cost of these techniques limits its actual use on large datasets. The approach proposed in [26] is based on the representation of text regions considering the interval encoding that represents how far is a text bin from one background area. With this representation, it is possible to take care of page translations by using a less expensive L_1 distance between the fixed-size vectorial representations.

Other global features are considered in [15] where the pages are represented by histograms describing the distributions of word height, character width, horizontal word spacing, line spacing, and line indentation. The similarity for each feature is computed considering the distance between the distributions by means of the Kullback–Leibler divergence. Two pages are compared by combining with an SVM classifier the five similarity measures together with a text-based similarity.

A related problem is addressed with the cyclic polar page layout representation proposed by Gordo and Valvenyl [23]. In this case two pages are compared considering the distance between the corresponding sequences. It is possible to measure the distance between two sequences by using either the edit distance or the Dynamic Time Warping algorithm provided that one suitable cost is defined for the transformation of one node into another.

Block Distance

In the case of block-based page representations (section "Representations Based on Lists of Blocks"), the upright rectangular blocks identified by layout analysis tools are considered to compute the page similarity. Blocks in one page are matched with blocks in the other pages by minimizing one global page distance. The latter is computed by combining the distances between pairs of blocks.

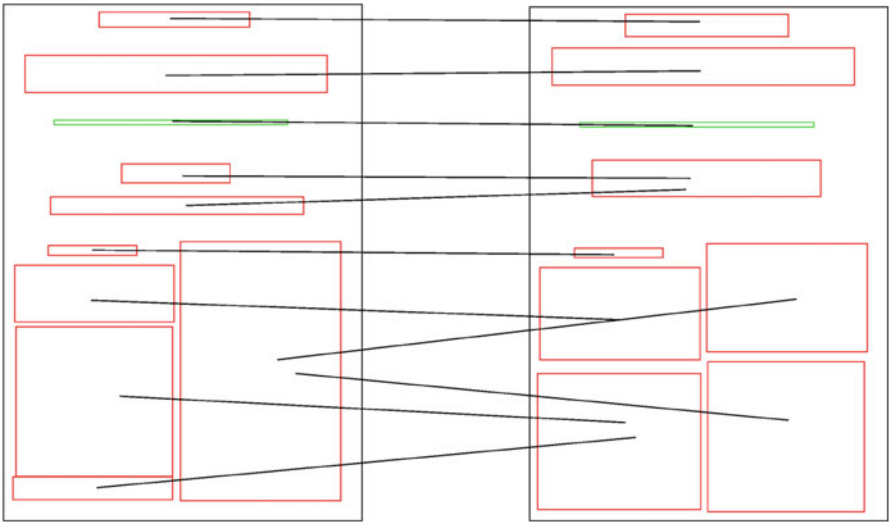

Fig. 7.6 Matching of blocks between two pages

Several methods that can be used to compute block distances are compared in [50]. Block comparison can take into account features that describe the position, size, and area of the blocks. The page comparison is based on an assignment of blocks from the query page to blocks in the reference layout that minimizes the overall distance. The latter is computed by summing up the block distances of the matched blocks. For instance, in Fig. 7.6 we show one hypothetic assignment of blocks in one page with blocks in a second page.

The block distances compared in [50] are the Manhattan distance of corner points, the overlapping area between blocks, and other simple block distances such as the difference in width and in height. Methods adopted to solve this *matching problem* should be tolerant to broken and merged blocks and are expected to be fast to compute. In [50] three alternative matching strategies are compared: the matching based on the assignment problem, the matching obtained by solving the minimum weight edge cover problem, and the use of the Earth Mover's Distance.

The computation of a similarity based on matching can be considered also when documents are represented with lists of lines. In [30] the similarity between one query page and one reference form is computed by searching for the optimal grid pair which results in the minimal overall cell count difference. The grid pairs considered correspond to all the signatures computed for each form. The similarity measure attempts to match the similar parts of two forms as much as possible in terms of the cell count difference. As a consequence, this measure is robust to incorrect line groupings.

Similarly to block matching, pages can be compared by aligning the bin rows that represent the pages [26]. The bin rows are matched using the interval encoding as described in section "Global Page Representations." In the case of classification, this alignment is obtained by using Hidden Markov Models to model different layout classes. HMMs are used because pages are best characterized as sequences of vertical regions.

Tree Distance

When the document layout is represented by means of a tree-based description, such as the X-Y tree, three main approaches can be used in the comparison: flatten the tree structure into one fixed-size representation, use tree-edit distances that take into account the hierarchical structure, or use decision trees where a suitable distance between trees is defined to navigate the decision tree.

One flat representation of X-Y trees is described in [9] where the tree is encoded into a fixed-size representation by counting the occurrences of tree patterns composed by three nodes in a way that is somehow similar to N-gram representation of text strings. The tree patterns are composed of three nodes connected one to the other by a path in the X-Y tree. Trees made by three nodes have two structures: one composed of a root and two children and one composed of a root, a child, and a child of the second node. Considering all the symbolic labels that can be assigned to nodes, a total of 384 possible tree patterns can be defined. In [9] this encoding is used by a feed-forward neural network to classify the pages.

Decision trees (in particular, the Geometric Tree, *GTree*) have been initially used by Dengel [17] to model the layout classes by describing a hierarchy for possible logical object arrangements.

More recently, Appiani et al. in [2] describe the classification of X-Y trees by means of a Document Decision Tree (DDT). DDT nodes contain X-Y trees that are compared with the input tree during classification. The tree comparison is made by looking for the maximum common sub-trees. During one generic step of the classification, the system compares the input X-Y tree to the trees stored in the decision tree nodes and follows the path with the minimum distance until a leaf with one class label is found.

Diligenti et al. [19] propose Hidden Tree Markov Models (HTMM) for modeling probability distributions defined over spaces of X-Y trees. Features are computed from the zone corresponding to each tree node. The generative model is a hidden tree of states which represent the zones. Since the model is a special case of Bayesian networks, both inference and parameter estimation are derived from the corresponding algorithms developed for Bayesian networks. In particular, the inference consists of computing all the conditional probabilities of the hidden states, given the evidence entered into the observation nodes (i.e., the labels of the tree).

One problem in the use of X-Y tree segmentation algorithms is that sometimes these do not produce similar trees starting from similar pages. Supervised classifiers often require the availability of a large enough training set that is expected to model the different trees generated by the segmentation algorithms. In some application domains, these training sets are not easily available and therefore alternative strategies should be explored. In [7] this problem is approached by considering a training set expansion that is based on the manipulation of labeled X-Y trees with a tree grammar. The new training set is built by merging the labeled samples and the artificial ones. The artificial trees are generated with a tree grammar to simulate actual distortions occurring in page segmentation. For instance, one rule splits a text node into two text nodes simulating the presence of a paragraph break. Another rule is used to add one text node below an image node to simulate the presence of a caption below the illustration.

During classification, one unknown tree is classified by comparing it, using one tree-edit distance algorithm, with the trees in the expanded training set. One $k-nn$ classifier is then used to determine the tree class. In analogy with the string edit distance, the tree-edit distance between two trees is the cost of the minimum-cost set of edit operations that are required to transform one tree into the other. In [7] the tree-edit distance algorithm proposed in [54] by Zhang and Shasha is used.

In [34] the pages are represented with X-Y trees and one edit distance based on Wang et al. algorithm [51] is used. The general algorithm is adapted by defining the distances between the X-Y tree nodes in case of substitutions, insertions, and deletions. In particular, the cost for replacing node v with node w (and vice versa) is defined by using a weighted Euclidean distance (the Karl Person distance) that takes into account the variance of each feature.

A related approach is described in [53] where pages are represented with L-S trees that are analogous to X-Y trees. Classes of documents are represented by computing the largest common sub-tree (called Document-Type Tree) in a collection of trees representing the training pages of one class. The class of an unknown page is obtained by finding the Document-Type Tree closest to the L-S tree computed from the page.

Region Classification

Region classification is a task that is of interest both for layout analysis (▶Chap. 5 (Page Segmentation Techniques in Document Analysis)) and for page classification and retrieval. This topic is discussed in this chapter because region classification techniques are similar to those used in page classification and because region classification is a component of some page comparison methods.

In this section we discuss classification methods used to identify the general region content (e.g., text vs. non-text) without considering the actual text in the region. On the opposite, when we aim at identifying the purpose of one text region on the basis of its contents (often recognized by an OCR engine), we deal with functional labeling applications.

The first region classification approaches used global features computed from the region together with linear classifiers designed with hand-tuned parameters [44]. Nowadays, in most cases trainable classifiers are used for region classification. For instance, in [14] texture features are adopted in combination with a decision tree to classify the regions into text or non-text.

Decision trees for region classification are described also in [1] where rectangular regions are found with a variant of the Run Length Smoothing Algorithm (RLSA) (see ▸Chap. 5 (Page Segmentation Techniques in Document Analysis)). The regions are described with features based on their dimensions, the number of black pixels, and the number of black-white transitions along horizontal lines. Other features are based on statistics obtained from the run lengths in the region. The labels assigned to regions are text block, horizontal line, vertical line, picture (i.e., halftone images), and graphics (e.g., line drawings).

A similar region classification approach is described by Wang et al. in [52] where a 25-dimensional feature vector is used as input to 1 decision tree trained to classify regions among 9 classes: large text, small text, math, table, halftone, map/drawing, ruling, logo, and others.

The discrimination of text and non-text regions in handwritten documents is described in [28] where a top-down segmentation is followed by an SVM classifier. The four segmentation algorithms considered are X-Y decomposition, morphological smearing, Voronoi diagram, and white space analysis. Various types of features are used as input to the classifier. The first features are based on run-length histograms of black and white pixels along the horizontal, the vertical, and the two diagonal directions. Another set of features is related to the size distributions of the connected components in the region. The last group of features is based on a two-dimensional histogram of the joint distribution of widths and heights and a histogram of the nearest neighbor distances of the connected components.

Page Classification

Methods for page classification and page retrieval significantly overlap both from the methodological point of view and from the application one. Many papers test the proposed page similarity approaches in both page retrieval and classification contexts. In some cases page retrieval is actually made by exploiting the page classification results, and pages with the same class of the query are shown to the user. In the majority of cases to provide a quantitative evaluation of the system, the performance of page retrieval is measured taking into account a labeling of pages into a finite disjoint set of classes.

In this and in the next section, we summarize some significant approaches that have been proposed to combine page representation and page comparison in the context of page classification and retrieval. The criterion that we adopted for organizing the approaches in the two sections is based on the type of measure presented by the authors. In page classification we consider methods where the performance is measured with the classification precision (percentage of patterns in the test set that are correctly classified). In page retrieval we include methods where all the pages in the dataset are compared with the query and are then sorted according to some similarity measure.

The overall objective of page classification is to assign one label to each document image. In Fig. 7.7 we graphically depict the organization of the main

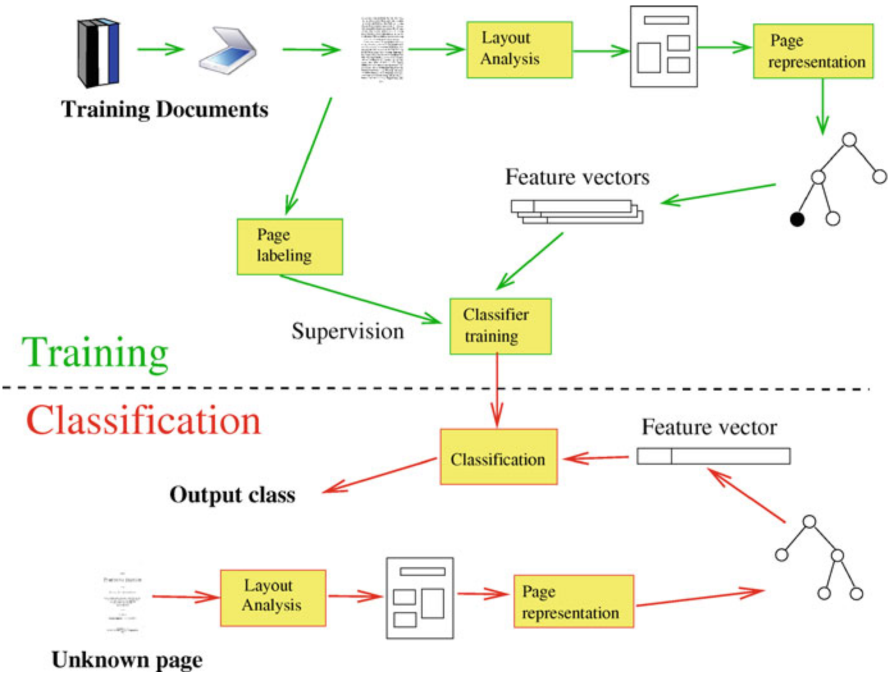

Fig. 7.7 General approach for page classification

modules considered in most page classification systems. In the illustration the page representation is described with a tree, but other representations can be considered as well. The page labeling box requires a human annotation of the class that should be assigned to each page in the training set that is used to train the supervised classifier. The feature vectors encode the page representation in a format that is handled by the classifier both during the training and when classifying an unknown page as described in the bottom part of the illustration.

Form classification and automatic document organization are traditional applications in the office domain. Form classification is aimed at selecting an appropriate reading strategy for each form to be processed and often takes into account the presence of ruling lines in the page layout [29, 33, 47]. In other cases the goal is to group office documents, for instance, setting apart business letters from technical papers [18, 48].

More recently, page classification has been adopted in Digital Library applications where it can be used to discover page-level metadata (e.g., identifying the table of contents page) and to locate pages where to look for specific metadata (e.g., the title of a book can be found in the cover page). In this case the classification addresses scanned pages in journals and books [25, 31, 45].

Whereas labels for region classification are quite standard (e.g., we can look for text, graphics, and line drawing regions), a broad range of categories can be considered in page classification. For instance, one document can be identified as *technical paper* or *commercial letter*; the publisher of one technical paper can be *IEEE* or *ACM*; the layout can be based on two columns or one column; and the type of page can be *title page*, *regular page*, or *index page*.

In [45] relevance-defined data are identified from human annotation of pages in the UW-I dataset in 12 visually different classes. Interestingly, many pages are often labeled with multiple classes, and this is a clear demonstration of the intrinsic ambiguity of page classification in some application domains. Users also provided scores of degree of similarity between the images and the classes, and this information was used to define the training set for the page classification task.

One important issue that must be addressed by page classification systems is the need to deal with generic classes that include heterogeneous pages. Pages in these classes should be grouped together from the application point of view but can have significantly different layouts. This organization of page labels can be dictated by application constraints it is not easy to be addressed by page classifiers that should otherwise finely distinguish between more homogeneous classes. For instance, in [2] 1 class contains more than 20 different subtypes of forms. Forms are addressed also in [3] where 1 class contains 205 pages not belonging to the training classes. In the classification of the first page of journal pages described in [5], one class actually contains three variations of the same logical class.

The page representation and distance computation (section "Page Comparison") are combined to build a complete page classification system. In the rest of this section, we summarize some representative systems whose approaches for page representation and matching have been previously described in this chapter. To provide a synthetic view of the alternative approaches to page classification, we compare the main features of some representative page classification methods in Table 7.2.

Appiani et al. [2] combine a page representation based on MXY trees with one decision tree classification algorithm. The experiments are made on two datasets. The first dataset contains 753 single-page invoices split into 9 classes. In the training set, 20 pages per class are used. The classification accuracy is 97.8 %. The second experiment works with bank account notes that comprise four classes: batch header, check, account note, and enclosures. The last class actually contains more than 20 different types of forms. The training is made with 67 documents and the test with 541 documents. Considering the three main classes (batch header, check, account note), the classification accuracy is about 99 %.

The integration of MXY tree page representation with Hidden Tree Markov Models is described in [19]. In this paper the tests are made with 889 commercial invoices issued by 9 different companies that define the class. The number of samples per class is not balanced and is comprised between 46 and 191. With 800 training examples, the accuracy is 99.28 %.

Even if described as a form retrieval system, the technique described in [30] classifies the pages on the basis of the distance (see section "Representations Based on Lists of Blocks") with respect to forms in the database. The experiments are made on a dataset containing 100 different types of forms including both blank forms

Table 7.2 Main features of some representative page classification methods. Data: type of documents addressed; Representation: page representation adopted; Classification: method adopted for page comparison/classification; NC: number of classes; Ref: cited paper

Data	Representation	Classification	NC	Ref.
Journal/letter/ magazine	Zoning	L_1 distance	5	[26]
Journal (UW-I)/tax forms	Zoning at character level	Decision tree	12–20	[45]
Tax forms	Zoning at character level	SOM clustering	12–20	[45]
Synthetic forms	Global features	L_1 distance	1,000	[40]
Article first pages	List of blocks	Optimal block matching	9–161	[50]
Hand-filled forms	Discriminative blocks	Block distance	7	[3]
Tax forms	List of ruling lines	Optimal line matching	100	[30]
Account note/invoice	X-Y tree	Decision tree	3–9	[2]
Article pages	X-Y tree encoding	Artificial neural networks	5	[10]
Article pages	X-Y tree encoding	SOM clustering	5	[36]
Article first pages	X-Y tree	Tree-edit distance + K-medoids clustering	25–150	[34]

and filled-in forms. The experiments performed with 200 random queries provide a classification accuracy of 95 %.

Form classification is addressed in [3] considering a class model consisting of a set of δ-landmarks. One document is assigned to one class if its sub-images match a significant number of the δ-landmarks that define the class. Each class model is built from one reference image. In total, there are seven forms in the reference set that include pages with very similar layouts differing in a few text areas or words like field names and page numbers. Two experiments are reported. The first one is made with 753 actual form pages belonging to the reference set. The second experiment is made with 205 document images, mostly forms, not belonging to any of the 7 reference pages.

Form classification by means of global features matched with p-norms is described in [40]. The proposed approach is tested on a large dataset of prototype forms. Test pages are generated by simulating various image deformations caused by filled-in document contents, noise, and block segmentation errors. For each experiment 20,000 test images are generated. The classification accuracy is measured with a variable number of classes and decreases from 99.77 % with 50 classes to 99.25 % with 1,000 classes.

In [45] a decision tree classifier and Self-Organizing Maps are combined to classify documents in five main classes: cover, reference, title, table of contents, and form. Three main experiments have been performed to evaluate the use of decision tree and Self-Organizing Map classifiers on relevance-defined, user-defined, and explicit-instance classes.

With relevance-defined classes the decision tree classifier has an accuracy of 83 %. In user-defined classes pages from a collection of office documents are grouped in four classes: cover, title, table of contents, and references. This collection has been integrated with forms from the NIST Special Database 2 Structured Forms Reference obtaining 261 images split into 5 classes. In this case the decision tree classifier has an accuracy of nearly 90 %. In the explicit-instance classes experiment pages came from the previous NIST Special Database 2 that contains 5,590 pages split in 20 types of taxform pages. The training set was based on 2,000 random images and other 2,000 were used in the test set. On these data the decision tree classifier has a classification accuracy of 99.70 %. When using the SOM classifier (in a semi-supervised approach), a classification accuracy of 96.85 % is reported.

Cesarini et al. [10] describe a page classification system aimed at splitting documents (belonging to journals or monographs in Digital Libraries) on the basis of the type of page. The five classes are advertisement, first page, index, receipts, and regular. The structural representation is based on the Modified X-Y tree. The page is classified by using artificial neural networks working on a fixed-size encoding of the page MXY tree. The test set is fixed with 300 pages while the size of the training set varies from 30 to 300 pages.

Pages from five technical journals belonging to Digital Libraries are considered in the experiments described by Bagdanov and Worring [5]. In the classification task, the first page is considered to identify the corresponding journal from 857 articles. The classification task is complicated by the fact that one class contains three variations of first pages with one, two, and three text columns, respectively. The layout is represented with attributed graphs, which naturally leads to the use of First-Order Gaussian Graph (FOGG) as a classifier of document genre. In the experiments a variable number of training pages are considered.

The use of tree-edit distance for classifying book pages represented with MXY trees is discussed in [7]. The classification is made with $k - nn$ where nearest training pages are computed by means of the tree-edit distance. The peculiarity of the approach is the use of training set expansion to improve classification performance. The experiments are made on two volumes of one nineteenth-century Encyclopedia containing 1,300 pages split into 7 classes (e.g., page with illustration, first page of a section, text on two columns) not evenly distributed.

A tree-based representation and subsequent classification by means of a tree comparison are presented in [53] where pages are described with L-S trees (similar to X-Y trees). The system is tested on a dataset composed of 40 pages for each of the 8 classes (letter, memo, call for papers, first page of articles in 5 different journals). Fifteen pages per class are used for training. The remaining 200 pages are used to test the classification that apart from some rejected pages (8 % of letters, 8 % of memos, and 20 % of calls-for-papers) correctly classifies all the documents.

Beusekom et al. in [50] compare several block distances and matching algorithms to classify pages in the MARG (Medical Article Records Ground-truth) database [22] that contains 815 first pages of scanned medical journals, labeled by layout type and journal. The page layout is represented by lists of blocks. The best results are reported when using the overlapping area as block distance and the matching method based on the solutions for the minimum weight edge cover problem. The classification is performed with a 1-nn classifier. The error rate computed with a leave-one-out approach is 7.4 % when looking for the page type (9 classes) and 31.2 % when looking for the journal type (161 classes).

Global features computed from page zoning are described in [16]. The similarity of indexed pages with respect to the query page (either hand-drawn or selected by the user with a graphical interface) is computed by means of the Euclidean distance between 80-dimensional feature vectors. The experiments are performed on images belonging to seven classes (journal, business letter, brochure, handwritten, newspaper, catalog, magazine). A leave-one-out nearest neighbor classifier was considered in the experiment that involved 939 documents. For each test page, one of the three nearest documents is in the right class with a recognition rate of the 97 %.

The use of interval encoding features (section "Global Page Representations") for page classification is proposed in [26] where each class is described with a Hidden Markov Model that represents the page as a sequence of vertical regions, each having some horizontal layout features (number of columns and widths and position of each column) and a variable height. The experiments are performed on 91 documents belonging to 5 classes (1-col-journal, 2-col-journal, 1-col-letter, 2-col-letter, magazine). The accuracy reported is comprised between 64 % (class 2-col-letter) and 100 % (class 2-col-journal).

Page Retrieval

In some application domains it is difficult, or even impossible, to use the page classification paradigm. The problems arise when the classes are not defined in a standardized way, and therefore, the labels assigned to pages can be ambiguous or subjective. For instance, when dealing with books in Digital Libraries, some pages (e.g., Fig. 7.8) could be labeled as *two-column text* or *illustration* or *section start* or any combination of the above.

One possible solution to this problem relies on page clustering described in section "Page Clustering." Alternatively, in some domains users can feel appropriate to use a query by example search and ask to the system to look for pages similar to one sample page. Document image retrieval using layout similarity offers to users a way to retrieve relevant pages that was possible before only by manually browsing documents, by either interacting with physical works or dealing with online images on DLs [35].

In Fig. 7.9 we describe the organization of the most common modules considered in page retrieval. There are several similarities with the page classification overall architecture depicted in Fig. 7.7. In particular the page encoding in the corresponding feature vector is often analogous to the encoding made in page classification. The main differences from the user point of view are the lack of an explicit page labeling that is otherwise required for page classification training and the different response of the system when consulting the dataset. In contrast with

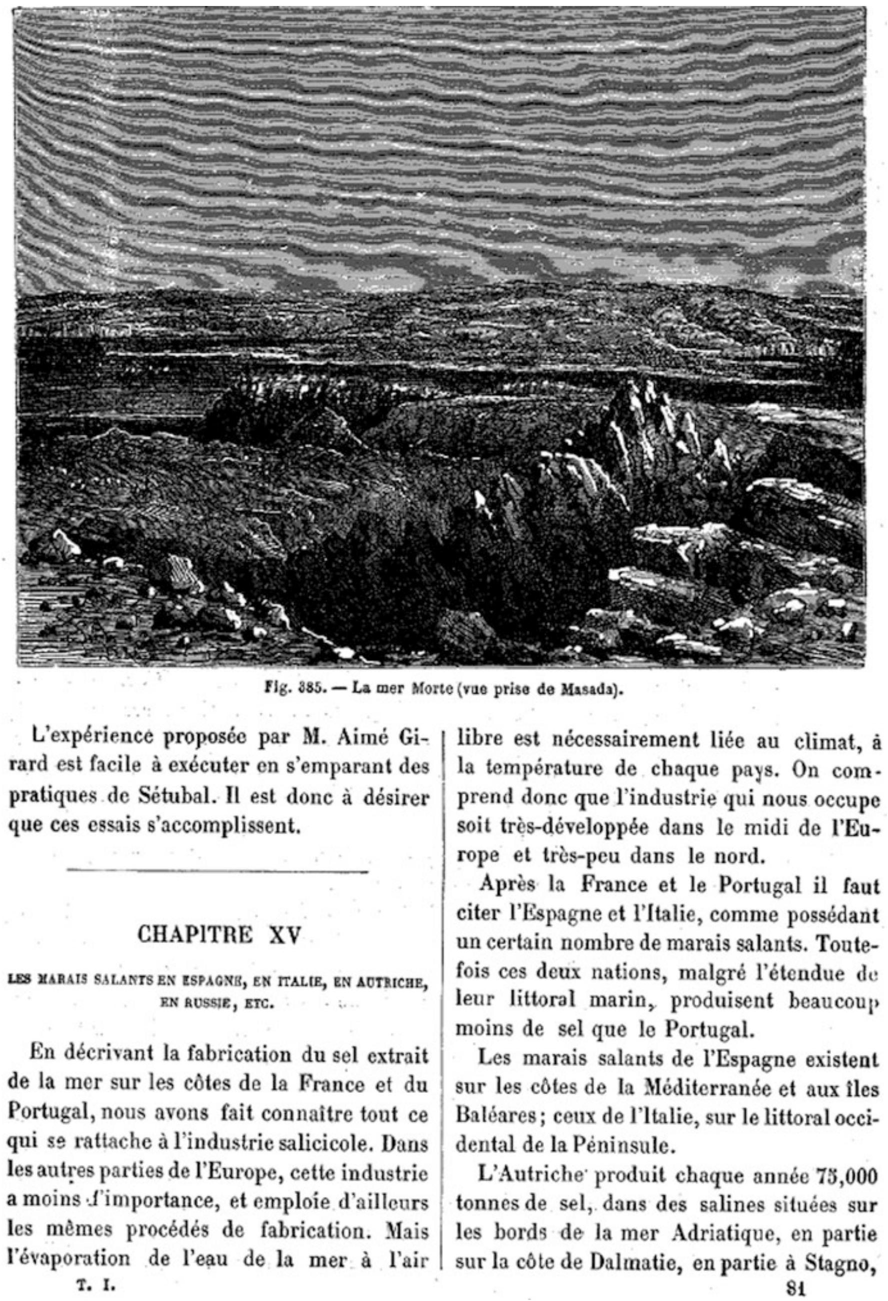

Fig. 385. — La mer Morte (vue prise de Masada).

L'expérience proposée par M. Aimé Girard est facile à exécuter en s'emparant des pratiques de Sétubal. Il est donc à désirer que ces essais s'accomplissent.

CHAPITRE XV

LES MARAIS SALANTS EN ESPAGNE, EN ITALIE, EN AUTRICHE, EN RUSSIE, ETC.

En décrivant la fabrication du sel extrait de la mer sur les côtes de la France et du Portugal, nous avons fait connaître tout ce qui se rattache à l'industrie salicicole. Dans les autres parties de l'Europe, cette industrie a moins d'importance, et emploie d'ailleurs les mêmes procédés de fabrication. Mais l'évaporation de l'eau de la mer à l'air libre est nécessairement liée au climat, à la température de chaque pays. On comprend donc que l'industrie qui nous occupe soit très-développée dans le midi de l'Europe et très-peu dans le nord.

Après la France et le Portugal il faut citer l'Espagne et l'Italie, comme possédant un certain nombre de marais salants. Toutefois ces deux nations, malgré l'étendue de leur littoral marin, produisent beaucoup moins de sel que le Portugal.

Les marais salants de l'Espagne existent sur les côtes de la Méditerranée et aux îles Baléares; ceux de l'Italie, sur le littoral occidental de la Péninsule.

L'Autriche produit chaque année 75,000 tonnes de sel, dans des salines situées sur les bords de la mer Adriatique, en partie sur la côte de Dalmatie, en partie à Stagno,

T. I. 81

Fig. 7.8 Page with non-unique classes

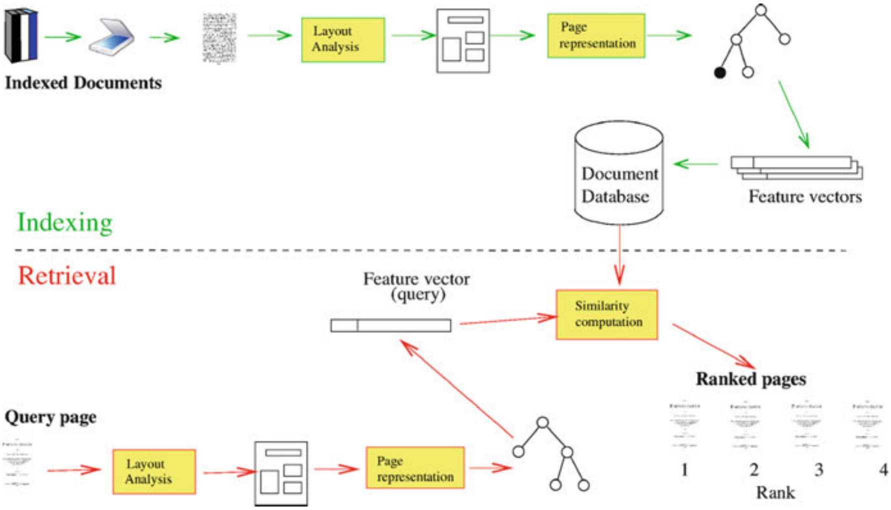

Fig. 7.9 General approach for page retrieval

Table 7.3 Main features of some representative page retrieval methods. Data: type of documents addressed; Pages: number of page in the dataset; NC: number of classes; Measure: metric used to analyze the retrieval performance; Ref: cited paper

Data	Pages	NC	Measure	Ref.
Mixed documents	743	8	Receiver Operating Characteristic (ROC) curves	[24]
Article first page	537	4	Precision-recall plots	[6]
Historical journal and book	1,221	15	Top-N precision	[11]
Forms	–	120	Top-N precision	[21]
Mixed documents	2,550	18	Mean average precision	[27]
Born-digital documents	4,563	40	Rank error	[32]

page classification, the system provides to the user a ranked list of indexed pages rather than assigning a class to the query page.

To provide a synthetic view of the alternative methods, we compare some features of page retrieval systems in Table 7.3. The table also compares the various metrics used to analyze the retrieval performance. In the case of page retrieval, the approaches presented in the literature differ not only for the application domain addressed but also for the measure used to evaluate the proposed techniques.

Page similarity is computed in [24] by combining the probabilities of the unknown document d to belong to the class c_i. This is obtained by first computing the probability that each indexed document belongs to each class and then storing these probabilities in a vector $d =< d_1, d_2, \ldots, d_N >$. The probability is computed by considering the output of one SVM classifier trained for each class. An analogous vector is computed for the query document q. The similarity of the two documents is subsequently computed by considering the inner product between q and d.

The experiments are performed on a subset of the Girona Archives database (a collection of documents related to people passing the Spanish-French border from 1940 up to 1976) that contains 743 images divided in 8 different classes. The results are measured with Receiver Operating Characteristic (ROC) curves.

To adopt rectangular granulometries for page retrieval, Bagdanov and Worring [6] use each document image as a query and then rank the remaining documents according to the Euclidean distance between the size distributions of documents. Precision and recall are used to compare the performance of the different systems in this case.

The integration of global and structural features is proposed in [11]. Global features are based on the *book bounding box* while structural features are based on occurrences of tree patterns. The similarity score is computed by combining the similarity computed on the first set of features using the Euclidean distance with the similarity computed for the second group of values using the inner product. The experiments are performed on the Making of America (MOA) dataset (several issues of one nineteenth-century journal with a total of 608 pages) and on the Gallica one (613 pages from one book). The results are evaluated considering the Top-N precision (the precision on the first N pages returned for each query). The best Top-10 precision for the whole database (that includes 15 classes) is 86.6 %.

Forms are represented by MXY trees and matched with a tree-edit distance in [21]. The tests are made on several datasets containing from 10 to 120 form types. In the case of 120 forms, the original samples are expanded by generating similar but different forms with geometrical modifications. In the experiments the precision when considering the top-N results is computed with N varying from 2 to 25. This value varies from 25 to 100 % on the basis of the size of the results set.

In the retrieval system discussed in [27] instead of performing a complete document analysis, the text lines are detected and the page layout is described by means of the quadrilaterals generated by all pairs of text lines. The experiments are made on 2,555 documents split into 18 classes (e.g., text on one, two, or three columns). The database contains diverse documents including forms, academic papers, and handwritten pages in English and Arabic. To measure the performance two values are computed: the Mean Average Precision (MAP) for 100 documents and the Mean Average Normalized Rank (MANR). The MAP at 100 evaluates the ranking for the 100 top-ranked documents. ANR is defined as

$$ANR = \frac{1}{N \cdot N_w} \sum_{i=1}^{N_w} \left(R_i - \frac{N_w + 1}{2} \right),$$

where N is the number of documents in the set, N_w is the number of wanted documents in the set, and R_i is the rank of each wanted document in the set. ANR has a value of 0 when the wanted documents are all been sorted on top. The MAP measure on the previous dataset is comprised between 0.7 and 0.9.

In [32] documents are ranked by considering a page similarity computed with the cosine of the angle between two vectors representing the similarity of blocks in the page with respect to the set of Representative Blocks (RB). The experiments are

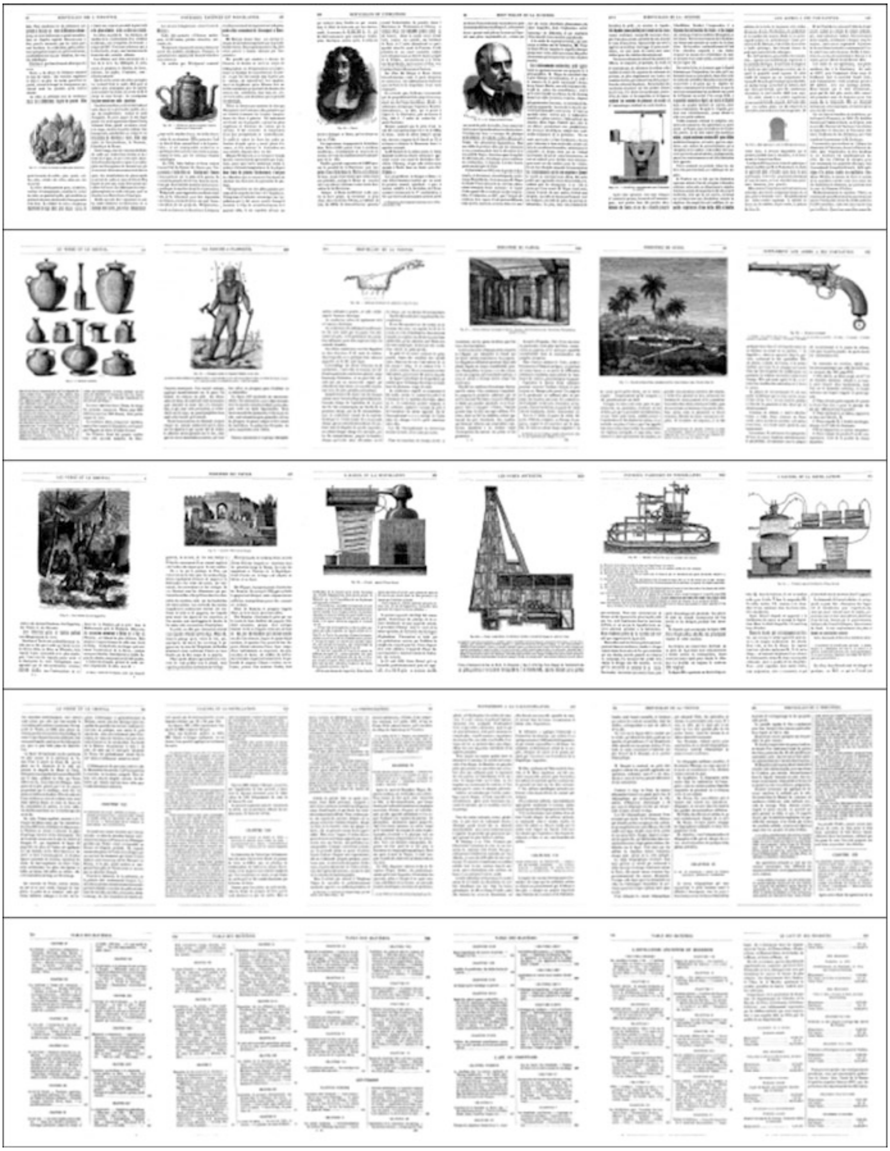

Fig. 7.10 Examples of page clusters

made on a collection of 845 technical documents accounting for 4,563 pages. The test is made considering 40 random pages as queries and then visually measuring the effectiveness of the ranking of the first 15 pages in the answer set. The computed rank error is defined as $\sum_i \frac{|\mathrm{ref}(D_i)-\mathrm{rank}(D_i)|}{15}$, where $\mathrm{ref}(D)$ and $\mathrm{rank}(D)$ are the position of document D in the reference ranking (manually annotated) and function ranking, respectively. The best results reported in the paper provide on average 3.28 positions in the function ranking away from the reference.

Page Clustering

One intermediate task between page classification and retrieval is page clustering. In this case the pages in a collection are not labeled by users and therefore it is not possible to build a supervised classifier. On the opposite, pages are clustered in an unsupervised way so as to facilitate page retrieval. One example of clustering (computed with the method described in [36]) is shown in Fig. 7.10 where each line contains pages grouped in the same cluster.

For instance, in [34] pages are represented by X-Y trees and one tree-edit distance algorithm is used together with the K-medoids clustering to group similar pages. The K-medoids algorithm is used rather than the classical K-means since it only needs the pairwise distance between pairs of objects (trees in this case).

Another clustering approach is proposed by Marinai et al. [36] and uses Self-Organizing Maps (SOM) to cluster pages represented by X-Y trees. The trees are encoded into a fixed-size representation by computing the occurrences of tree pattern as described in section "Tree Distance."

The fixed-size representation allows to compute the average of the patterns in clusters that is required by the SOM and the K-means clustering algorithms.

Self-Organizing Maps are used together with a zoning-based page representation in [45] to identify similar forms in the NIST Special Database 2 that contains 20 different form classes. Self-Organizing Maps are used to relax the need for labeled training samples. The SOM is used to first find clusters in the input data and then to identify each unknown form with one of the clusters.

Conclusions

In this chapter we analyzed and compared various techniques for page classification and retrieval. The most important topic in both tasks is the choice of the approach used to represent the page layout. This representation is tightly related with the technique considered to compare the pages that is used by the page classification or retrieval algorithms. In particular, while pixel-based page representations are more appropriate when a fine-grained differentiation of the pages is required (used, for instance, in form classification), structural representations, e.g., based on trees or graphs, are more suitable when the generalization of the classifier/retrieval system is essential (for instance, in Digital Library applications).

Until recently in document image analysis, the page has been considered as the most common input to processing systems. One page can be easily converted into an image file both in the case of physical documents (where the conversion is made with digitization devices, such as scanners) and in the case of born-digital documents (where the image is generated by means of one suitable rendering software). In the last few years with the advent of large-scale digitization projects and the availability of more powerful computational resources, the processing of whole books is becoming more and more common. In this case page classification can be used as a component of book processing systems.

On the other hand, when considering born-digital documents, the panorama of file formats is rapidly evolving in the last few years. Document distribution is nowadays in the large majority of cases delegated to the PDF format that is essentially a page-oriented format whose main purpose is the faithful representation of the page layout on a broad range of visualization and printing devices. However, the advent of e-book readers and tablet devices is pushing on the stage new formats, such as the e-pub format for e-books, that are essentially based on HTML and are intrinsically reflowable. With e-book readers the page is dynamically reflown when reading the book and therefore it is more difficult to conceptually define the concept of *page*.

Cross-References

►Analysis of the Logical Layout of Documents
►Image Based Retrieval and Keyword Spotting in Documents
►Logo and Trademark Recognition
►Page Segmentation Techniques in Document Analysis
►Recognition of Tables and Forms

References

1. Altamura O, Esposito F, Malerba D (2001) Transforming paper documents into XML format with WISDOM++. Int J Doc Anal Recognit 4(1):2–17
2. Appiani E, Cesarini F, Colla AM, Diligenti M, Gori M, Marinai S, Soda G (2001) Automatic document classification and indexing in high-volume applications. Int J Doc Anal Recognit 4(2):69–83
3. Arlandis J, Perez-Cortes J-C, Ungria E (2009) Identification of very similar filled-in forms with a reject option. In: Proceedings of the ICDAR, Barcelona, pp 246–250
4. Bagdanov AD, Worring M (2001) Fine-grained document genre classification using first order random graphs. In: Proceedings of the ICDAR, Seattle, pp 79–83
5. Bagdanov AD, Worring M (2003) First order Gaussian graphs for efficient structure classification. Pattern Recognit 36(3):1311–1324
6. Bagdanov AD, Worring M (2003) Multi-scale document description using rectangular granulometries. Int J Doc Anal Recognit 6:181–191
7. Baldi S, Marinai S, Soda G (2003) Using tree-grammars for training set expansion in page classification. In: Proceedings of the ICDAR, Edinburgh, pp 829–833
8. Cesarini F, Gori M, Marinai S, Soda G (1999) Structured document segmentation and representation by the modified X-Y tree. In: ICDAR, Bangalore, pp 563–566
9. Cesarini F, Lastri M, Marinai S, Soda G (2001) Encoding of modified X-Y trees for document classification. In: Proceedings of the ICDAR, Seattle, pp 1131–1136
10. Cesarini F, Lastri M, Marinai S, Soda G (2001) Page classification for meta-data extraction from digital collections. In: Mayr HC et al (eds) Database and expert systems applications. LNCS 2113. Springer, Berlin/New York, pp 82–91

11. Cesarini F, Marinai S, Soda G (2002) Retrieval by layout similarity of documents represented with MXY trees. In: Lopresti D, Hu J, Kashi R (eds) International workshop on document analysis systems, Princeton. LNCS 2423. Springer, pp 353–364
12. Chen N, Blostein D (2007) A survey of document image classification: problem statement, classifier architecture and performance evaluation. Int J Doc Anal Recognit 10(1):1–16
13. Chen F, Girgensohn A, Cooper M, Lu Y, Filby G (2012) Genre identification for office document search and browsing. Int J Doc Anal Recognit 15:167–182. doi:10.1007/s10032-011-0163-7
14. Chetverikov D, Liang J, Komuves J, Haralick RM (1996) Zone classification using texture features. In: International conference on pattern recognition, Vienna, pp 676–680
15. Collins-Thompson K, Nickolov R (2002) A clustering-based algorithm for automatic document separation. In: Proceedings of the SIGIR workshop on information retrieval and OCR, Tampere
16. Cullen JF, Hull JJ, Hart PE (1997) Document image database retrieval and browsing using texture analysis. In: Proceedings of the ICDAR, Ulm, pp 718–721
17. Dengel A (1993) Initial learning of document structure. In: Proceedings of the ICDAR, Tsukuba, pp 86–90
18. Dengel A, Dubiel F (1995) Clustering and classification of document structure-a machine learning approach. In: Proceedings of the ICDAR, Montreal, pp 587–591
19. Diligenti M, Frasconi P, Gori M (2003) Hidden Tree Markov models for document image classification. IEEE Trans Pattern Anal Mach Intell 25(4):519–523
20. Doermann D (1998) The indexing and retrieval of document images: a survey. Comput Vis Image Underst 70(3):287–298
21. Duygulu P, Atalay V (2002) A hierarchical representation of form documents for identification and retrieval. Int J Doc Anal Recognit 5(1):17–27
22. Ford G, Thoma GR (2003) Ground truth data for document image analysis. In: Proceedings of the symposium on document image understanding and technology, Greenbelt, pp 199–205
23. Gordo A, Valveny E (2009) A rotation invariant page layout descriptor for document classification and retrieval. In: Proceedings of the ICDAR, Barcelona, pp 481–485
24. Gordo A, Gibert J, Valveny E, Rusiñol M (2010) A kernel-based approach to document retrieval. In: International workshop on document analysis systems, Boston, pp 377–384
25. Hu J, Kashi R, Wilfong G (1999) Document image layout comparison and classification. In: Proceedings of the ICDAR, Bangalore, pp 285–288
26. Hu J, Kashi R, Wilfong G (2000) Comparison and classification of documents based on layout similarity. Inf Retr 2:227–243
27. Huang M, DeMenthon D, Doermann D, Golebiowski L (2005) Document ranking by layout relevance. In: Proceedings of the ICDAR, Seoul, pp 362–366
28. Indermuhle E, Bunke H, Shafait F, Breuel T (2010) Text versus non-text distinction in online handwritten documents. In: SAC, Sierre, pp 3–7
29. Ishitani Y (2000) Flexible and robust model matching based on association graph for form image understanding. Pattern Anal Appl 3(2):104–119
30. Jain AK, Liu J (2000) Image-based form document retrieval. Pattern Recognit 33:503–513
31. Kochi T, Saitoh T (1999) User-defined template for identifying document type and extracting information from documents. In: ICDAR, Bangalore, pp 127–130
32. Lecerf L, Chidlovskii B (2010) Scalable indexing for layout based document retrieval and ranking. ACM Symposium on Applied Computing, Sierre, pp 28–32
33. Lin JY, Lee C-W, Chen Z (1996) Identification of business forms using relationships between adjacency frames. MVA 9(2):56–64
34. Mao S, Nie L, Thoma GR (2005) Unsupervised style classification of document page images. IEEE International Conference on Image Processing, Genoa, pp 510–513
35. Marinai S (2006) A survey of document image retrieval in digital libraries. In: 9th colloque international francophone sur l'Ecrit et le document, Fribourg, pp 193–198
36. Marinai S, Marino E, Soda G (2006) Tree clustering for layout-based document image retrieval. In: Proceedings of the international workshop on document image analysis for libraries 2006, Lyon, pp 243–253

37. Marinai S, Marino E, Soda G (2010) Table of contents recognition for converting PDF documents in e-book formats. In: Proceedings of the 10th ACM symposium on document engineering (DocEng'10), Manchester. New York, pp 73–76
38. Marinai S, Miotti B, Soda G (2011) Digital libraries and document image retrieval techniques: a survey. In: Biba M, Xhafa F (eds) Learning structure and schemas from documents. Volume 375 of studies in computational intelligence. Springer, Berlin/Heidelberg, pp 181–204
39. Peng H, Long F, Chi Z, Siu W-C (2001) Document image template matching based on component block list. PRL 22:1033–1042
40. Peng H, Long F, Chi Z (2003) Document image recognition based on template matching of component block projections. IEEE Trans Pattern Anal Mach Intell 25(9):1188–1192
41. Perea I, Lṕez D (2004) Syntactic modeling and recognition of document image. In: SSPR&SPR, Lisbon, pp 416–424
42. Qi X, Davison BD (2009) Web page classification: features and algorithms. ACM Comput Surv 41:12:1–12:31
43. Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv 34:1–47
44. Shih FY, Chen SS (1996) Adaptive document block segmentation and classification. IEEE Trans SMC 26(5):797–802
45. Shin C, Doermann DS, Rosenfeld A (2001) Classification of document pages using structure-based features. Int J Doc Anal Recognit 3(4):232–247
46. Takama Y, Mitsuhashi N (2005) Visual similarity comparison for web page retrieval. In: IEEE/WIC/ACM international conference on web intelligence (WI 2005), Compiegne, pp 301–304
47. Taylor SL, Fritzson R, Pastor JA (1992) Extraction of data from preprinted forms. MVA 5(5):211–222
48. Taylor SL, Lipshutz M, Nilson RW (1995) Classification and functional decomposition of business documents. In: ICDAR 95, Montreal, pp 563–566
49. Tzacheva A, El-Sonbaty Y, El-Kwae EA (2002) Document image matching using a maximal grid approach. Document Recognition and Retrieval IX, San Jose, pp 121–128
50. van Beusekom J, Keysers D, Shafait F, Breuel TM (2006) Distance measures for layout-based document image retrieval. In: Proceedings of the international workshop on document image analysis for libraries 2006, Lyon, pp 232–242
51. Wang JT-L, Zhang K, Jeong K, Shasha D (1994) A system for approximate tree matching. IEEE Trans Knowl Data Eng 6(4):559–571
52. Wang Y, Phillips IT, Haralick RM (2006) Document zone content classification and its performance evaluation. Pattern Recognit 39:57–73
53. Wei C-S, Liu Q, Wang JT-L, Ng PA (1997) Knowledge discovering for document classification using tree matching in TEXPROS. Inf Sci 100(1–4):255–310
54. Zhang K, Shasha D (1989) Simple fast algorithms for the editing distance between trees and related problems. SIAM J Comput 18(6):1245–1262

Further Reading

Document image classification has been extensively surveyed by Chen and Blostein in [12] where one comprehensive comparison of the different applications and techniques adopted for page classification is provided. On the side of document image retrieval, one classical survey on the topic including both text-retrieval and layout-based approaches is [20]. One recent survey related to applications in Digital Libraries can be found in [38].