IJDAR
DOI 10.1007/s10032-006-0020-2

ORIGINAL ARTICLE

A survey of document image classification: problem statement, classifier architecture and performance evaluation

Nawei Chen · Dorothea Blostein

Received: 1 June 2004 / Accepted: 20 December 2004

Abstract Document image classification is an important step in Office Automation, Digital Libraries, and other document image analysis applications. There is great diversity in document image classifiers: they differ in the problems they solve, in the use of training data to construct class models, and in the choice of document features and classification algorithms. We survey this diverse literature using three components: the problem statement, the classifier architecture, and performance evaluation. This brings to light important issues in designing a document classifier, including the definition of document classes, the choice of document features and feature representation, and the choice of classification algorithm and learning mechanism. We emphasize techniques that classify single-page typeset document images without using OCR results. Developing a general, adaptable, high-performance classifier is challenging due to the great variety of documents, the diverse criteria used to define document classes, and the ambiguity that arises due to ill-defined or fuzzy document classes.

Keywords Document image classification · Document classifiers · Document classification · Document categorization · Document features · Feature representations · Class models · Classification algorithms · Learning mechanisms · Performance evaluation

N. Chen (✉) · D. Blostein
School of Computing, Queen's University,
K7L 3N6, Kingston, ON, Canada
e-mail: chenn@cs.queensu.ca

D. Blostein
e-mail: blostein@cs.queensu.ca

1 Introduction

Document classification is an important task in document processing. It is used in the following contexts:

- Document classification allows the automatic distribution or archiving of documents. For example, after classification of business letters according to sender and message type (such as order, offer, or inquiry), the letters are sent to the appropriate departments for processing [8].
- Document classification improves indexing efficiency in Digital Library construction. For example, classification of documents into table of contents page or title page can narrow the set of pages from which to extract specific meta-data, such as the title or table of contents of a book [12].
- Document classification plays an important role in document image retrieval. For example, consider a document image database containing a large heterogeneous collection of document images. Users have many retrieval demands, such as retrieval of papers from one specific journal, or retrieval of document pages containing tables or graphics. Classification of documents based on visual similarity helps narrow the search and improves retrieval efficiency and accuracy [51].
- Document classification facilitates higher-level document analysis. Due to the complexity of document understanding, most high-level document analysis systems rely on domain-dependent knowledge to obtain high accuracy. Many available information extraction systems are specially designed for a specific type of document, such as forms processing or postal address processing, to achieve high speed

and performance. To process a broad range of documents, it is necessary to classify the documents first, so that a suitable document analysis system for each specific document type can be adopted. The document classifier used in the STRECH system is aimed to work as the front-end for a set of commercial OCR systems [1]. Document classification is used to tune OCR parameters, or to choose an appropriate OCR system for a specific type of document. Classifiers can be used to identify form types for banking applications [41,46]. Subsequently, form data is extracted based on the layout knowledge of that particular form type.

Document classification can be done with or without use of the text content of the document. We use the following terminology, which is not standardized.

Document classification (Also called *document image classification or page classification*). Assign a single-page document image to one of a set of predefined document classes. Classification can be based on various features, such as image-level features, structural or textual features.

Text categorization (Also called *text classification*). Assign a text document to one of a set of predefined document classes. The text document may be a plain text document (e.g. ASCII) or a tagged text document (e.g. HTML/XML). Classification is based on textual features (such as word frequency or word histogram) or on structural information known from tags.

Sebastiani [49] provides a comprehensive survey of text categorization, which is an active research area in information retrieval. The need for text categorization continues to grow, due to the increased availability of text documents, especially on the Internet. More recently, researchers are proposing classification methods that use both textual and structural information [48]. The structural information may be directly available from the tags in a tagged text document. Text categorization techniques can be applied as part of document image classification, using OCR results extracted from the document image. However, OCR errors must be considered.

In this survey, a *document* refers to a single-page typeset document image. The document image may be produced from a scanner, a fax machine or by converting an electronic document into an image format (e.g. TIFF or JPEG). We focus on classification of mostly-text documents, using image-level or structural features, rather than textual features. Mostly-text documents include business letters, forms, newspapers, technical reports, proceedings, and journal papers, etc. These are in contrast to mostly-graphics documents such as engineering drawings, diagrams, and sheet music. Among mostly-text documents, we further focus on classification of documents with significant structure variations within a class, such as business letters, article-pages and newspaper-pages. Forms have rather restricted physical layout. Many papers have been published about form classification (also called form type identification) [10, 23,50,56,59]. We refer to some of this literature, but do not provide an exhaustive survey of form classification.

2 Three components of a document classifier

There is great diversity in document classifiers. Classifiers solve a variety of document classification problems, differ in how they use training data to construct models of document classes, and differ in their choice of document features and recognition algorithms. We survey this diverse literature using three components: the problem statement, the classifier architecture and performance evaluation. These components are illustrated in Fig. 1.

The problem statement for a document classifier defines the problem being solved by the classifier. It consists of two aspects: the document space and the set of document classes. The document space defines the range of input document samples. The training samples and the test samples are drawn from the document space. The set of document classes defines the possible outputs produced by the classifier and is used to label document samples. Most surveyed classifiers use manually defined document classes, with class definitions based on similarity of contents, form, or style. The problem statement is discussed further in Sect. 3.

The classifier architecture includes four aspects: document features and recognition stages, feature representations, class models and classification algorithms, and learning mechanisms. The classifier architecture is discussed further in Sect. 4 with Table 2 presenting an overview of the surveyed classifiers along these four aspects.

Performance evaluation is used to gauge the performance of a classifier, and to permit performance comparisons between classifiers. The diversity among document classifiers makes performance comparisons difficult. Issues in performance evaluation include the need for standard data sets, standardized performance metrics, and the difficulty of separating classifier performance from pre-processor performance. Performance evaluation is discussed further in Sect. 5.

3 The problem statement

The problem statement for a document classifier has two aspects: the document space and the set of doc-

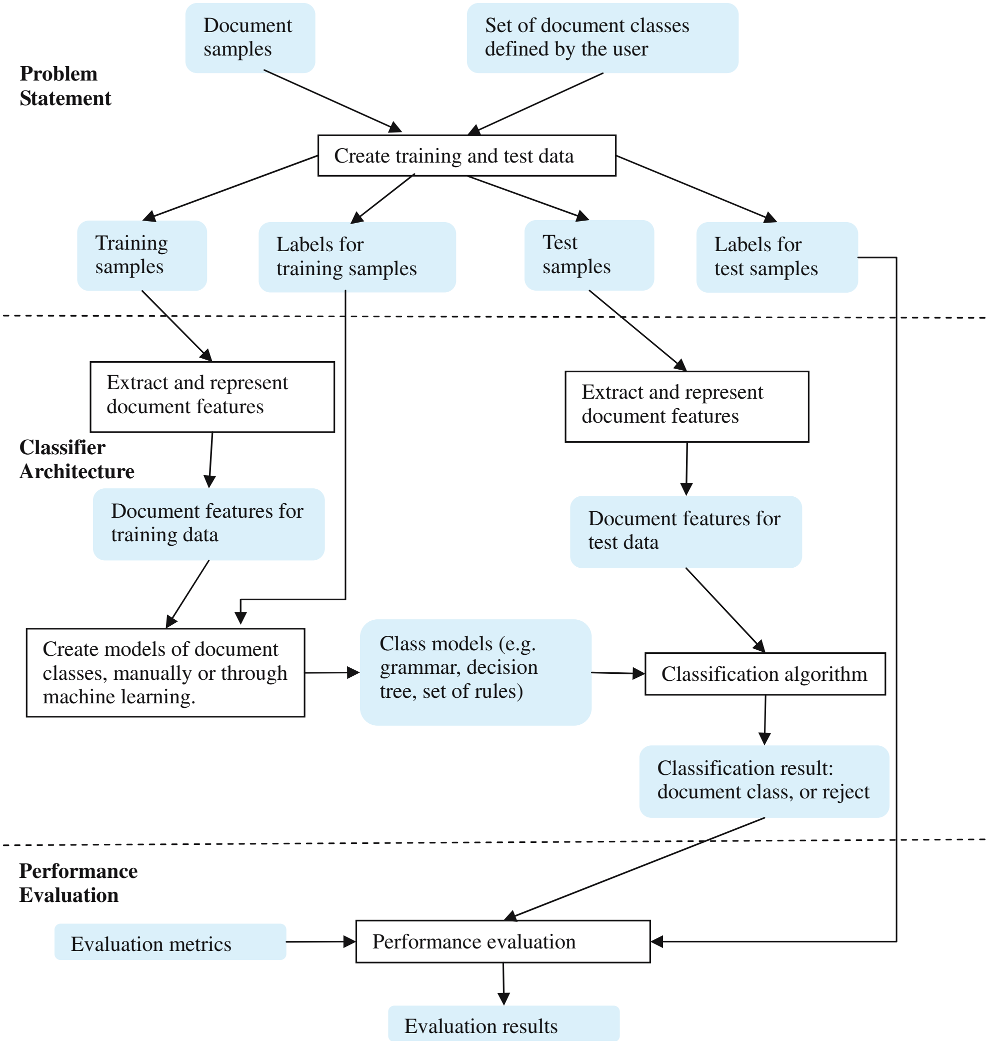

Problem Statement
Document samples
Set of document classes defined by the user
Create training and test data
Training samples
Labels for training samples
Test samples
Labels for test samples
Classifier Architecture
Extract and represent document features
Document features for training data
Extract and represent document features
Document features for test data
Create models of document classes, manually or through machine learning.
Class models (e.g. grammar, decision tree, set of rules)
Classification algorithm
Classification result: document class, or reject
Performance Evaluation
Evaluation metrics
Performance evaluation
Evaluation results

Fig. 1 Three components of a document classifier: the problem statement, the classifier architecture, and performance evaluation. The *rectangular boxes* represent processes. The *shaded regions* represent data. This figure provides a framework for discussing document classifiers. The classifier design process is not shown; this typically involves iteration, with iterative changes to the set of document features, the class models, and the classification algorithms

ument classes. The former defines the range of input documents, and the latter defines the output that the classifier can produce.

3.1 The document space

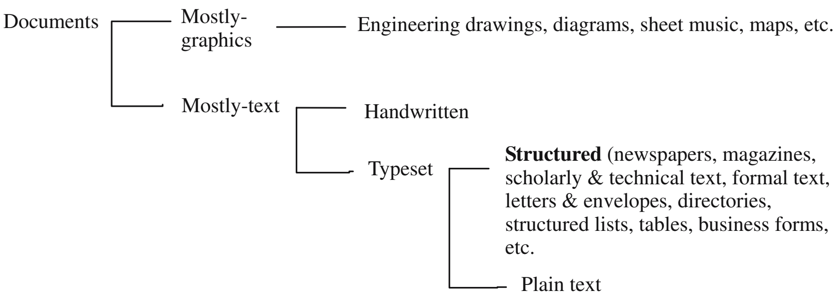
The document space is the set of documents that a classifier is expected to handle. The labeled training samples and test samples are all drawn from this document space. The training samples are assumed to be representative of the defined set of classes. The document space may include documents that should be rejected, because they do not lie within any document class. In this case, the training samples might consist of positive samples only, or they might consist of a mixture of positive and negative samples. Document classifiers with reject options are reported in [12,21,23,33,41,55].

For any classifier, the document space is a subset of the entire set of possible documents (which includes all existing documents, as well as documents that are yet to be created). There is no precise definition of *document*. We use the document taxonomy defined by Nagy [38] as shown in Fig. 2.

Structured documents are mostly-text documents that have identifiable layout characteristics. All classifiers we survey use a document space consisting of structured documents. Text categorization methods, dealing with plain text documents, are surveyed by Sebastiani [49].

Nagy's characterization of documents focuses on document format: mostly-graphics or mostly-text, handwritten or typeset, etc. Another way of characterizing

Fig. 2 Document taxonomy defined by Nagy [38]


Documents
Mostly-graphics
Engineering drawings, diagrams, sheet music, maps, etc.
Mostly-text
Handwritten
Typeset
Structured (newspapers, magazines, scholarly & technical text, formal text, letters & envelopes, directories, structured lists, tables, business forms, etc.
Plain text

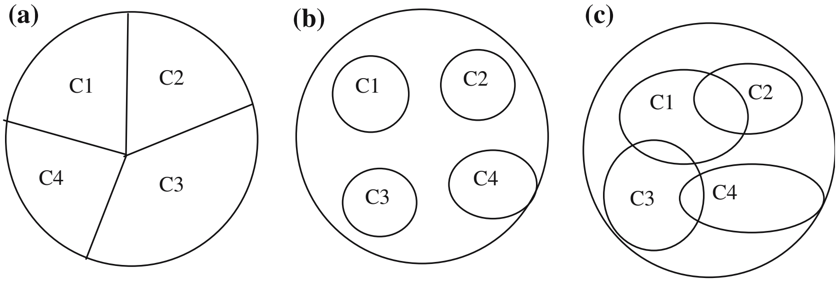


Fig. 3 Three possible partitions of document space. **a** A set of four classes (C1, C2, C3, and C4) uniquely divides the document space. **b** The document space is larger than the union of document classes. The documents that do not belong to any of the document classes should be rejected. **c** There is fuzziness (overlapping) in the partition. A single document may belong to multiple classes

documents is by application domain, such as documents related to income tax or documents from insurance companies. Some of the classifiers we survey use document spaces that are restricted to a single application domain. Others use document spaces that span several application domains. Here is a summary of the document space of selected classifiers characterized by application domains.

- A single domain document space
  Bank documents [1].
  Business letters [4,8,13,15,24].
  Business reports [3].
  Invoices [1,16].
  Business forms [10,23,50,56,59].
  Forms in banking applications [41,46].
  Tax forms [51].
  Documents from insurance companies [62].
  Book page [5].
  Journal pages [12,21,28,34,40,53].
- A multiple-domain document space
  Articles, advertisements, dictionaries, forms, manuals, etc. [19].
  Journal pages, business letters, and magazines [26].
  Bills, tax forms, journals, and mail pieces [33].
  Journal papers, tax forms [51].
  Business letters, memoranda, and documents from other domains [55].
  Business letters, reports, technical papers, magazines, etc. [31,36].

3.2 The set of document classes

The set of document classes defines how the document space is partitioned. The name of a document class is the output produced by the classifier. Several possible partitions of document space are shown in Fig. 3. A set of document classes may uniquely separate the document space (Fig. 3a), with a single class label assigned to a document. If the document space is larger than the union of the document classes (Fig. 3b), the classifier is expected to reject all documents that do not belong to any document class. Fuzziness may exist in the definition of document classes (Fig. 3c), with multiple class labels assigned to a document.

A *document class* (also called *document type* or *document genre)* is defined as a set of documents characterized by similarity of expressions, style, form or contents [3]. This definition states that various criteria can be used for defining document classes. Document classes can be defined based on similarity of contents. For example, consider pages in conference papers, with classes consisting of "pages with experimental results", "pages with conclusions", "pages with description of a method" [62]. Alternatively, document classes can be defined based on similarity of form and style (also called *visual similarity*), such as page layout, use of figures, or choice of fonts [19].

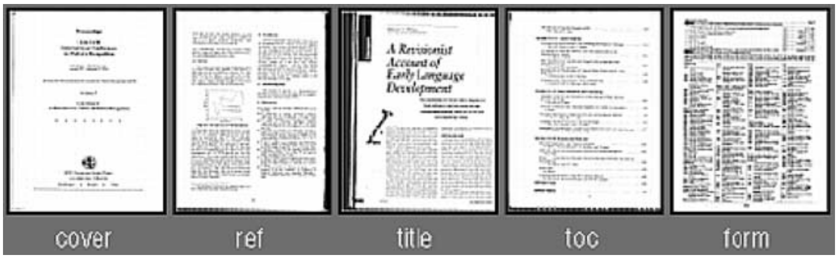


Fig. 4 An example of document classes defined based on visual similarity from [51]: cover, reference, title, table of contents, and form

Table 1 Document classes used in selected classifiers

	Classes based on similarity of contents
Dengel [4,15]	Five classes of business letters based on message types: order, offer, inquiry, confirmation, advertisement
Spitz and Maghbouleh [53]	Seventy-three overlapping classes based on subjects from journal papers (University of Washington CD)
Sako et al. [41,46]	A few hundred form types used in banking applications: money order, utility bills, tax notices, etc.
	Classes based on similarity of form and style (also called *visual similarity*)
Baldi et al. [5]	Seven classes of pages from 19th Century books: with or without caption, two columns with or without images, start of an issue, end-of-section page, section mark page
Diligenti et al. [16]	Nine classes of invoices from different issuing companies
Eglin and Bres [19]	Ten classes defined based on 19 predefined Oulu classes [47], including articles, advertisements, address lists, dictionaries, forms, manuals, mathematical documents
Liang et al. [34]	Title pages from four journal/conference proceedings
Appiani et al. [1]	Nine classes in Test 1: invoices from different suppliers. Four classes of bank documents in Test 2: account notes, cheques, batch headers, and enclosures
Bagdanov and Worring [3]	Ten classes of business reports from trade journals and product brochures
Cesarini et al. [12]	Five classes of journal pages: first pages, index pages, receipts pages, regular pages and advertisement pages
Nattee and Numao [40]	Four classes of journal title pages from ICML, COLT, PAMI, ISMIS
Shin et al. [51]	Five classes in Test 1: covers, references, titles, table of contents, forms. Twenty classes of tax forms in Test 2
Byun and Lee [10]	Seven classes of forms: tax forms, credit card slips, bank forms
Esposito et al. [21]	Three classes of journal title pages from ICML, ISMIS, PAMI
Hu et al. [26]	Five classes: one-column and two-column journal pages, one-column and two-column letters, and magazines
Kochi and Saitoh [31]	Thirty classes: business letters, reports, technical papers, magazines, Japanese articles with character strings aligned vertically, etc.
Wnek [62]	Five hundred classes of documents used by insurance companies
Taylor et al. [55]	Two classes: business letters, memorandums
Lam [33]	Four classes from four different domains: bills, tax forms, IEEE journals, mail pieces

Figure 4 shows an example of document classes defined based on visual similarity. Doermann et al. provide a functional description of a document, which gives insight into defining document classes based on domain-independent functional structures, such as headers, footers, lists, tables, and graphics [17].

Typically, the set of document classes is not given as an explicit input to a document classifier. Instead, a description of the set of classes is provided implicitly, by the labeled training samples. Of course, labeling the training samples requires a definition of document classes. This might be an informal, implicit definition: the document classes are manually defined, and the training samples are manually labeled. Alternatively, document classes can be defined automatically, by clustering unlabeled document samples. Most of the systems we survey use manual definition of the document classes. An exception is Shin et al., who, in addition to defining classes manually, use a self-organizing map to find clusters in unlabeled input data and assign each input document to one of the clusters [51].

Table 1 summarizes the classification problems solved by selected document classifiers. The great diversity of document classes is clearly illustrated.

The set of document classes that are required depend on the goal of the document classification. Document classification is often followed by further document image analysis. The classification allows subsequent processing to be tuned to the document class.

Bagdanov and Worring characterize document classification at two levels of detail, coarse-grained and fine-grained [3]. *A coarse-grained classification* is used to classify documents with a distinct difference of features, such as business letters versus technical articles. A *fine-grained classification* is used to classify documents with similar features, such as business letters from different senders, or journal title pages from various journals.

This completes our discussion of the problem statement for a document classifier. Next, we discuss the classifier architecture.

4 The classifier architecture

We use the following four aspects to characterize classifier architecture: (1) document features and recognition stage, (2) feature representations, (3) class models and classification algorithms, and (4) learning mechanisms. These aspects are interrelated: design decisions made regarding one aspect have influence on design of other aspects. For example, if document features are represented in fixed-length feature vectors, then statistical models and classification algorithms are usually considered. Table 2 provides an overview of the surveyed document classifiers using these four aspects. As seen in Table 2, classification may be performed at different stages of document recognition, with a diverse choice of document features, feature representations, class models and classification algorithms.

We now discuss each of the four aspects in Sects. 4.1–4.4. In the process, we refer to various entries in Table 2.

4.1 Document features and recognition stage

Choice of document features is an important step in classifier design. Table 2 illustrates the great variety of document features used for document classification. Relevant surveys about document features include the following. Commonly used features in OCR are surveyed in [57]. A set of commonly used features for page segmentation and document zone classification are given in [42,58]. Structural features produced in physical and logical layout analysis are surveyed in [22,37,38].

All the features in our surveyed systems are extracted from black and white document images. The gray-scale or color images (e.g. advertisements, magazine articles) are binarized into binary images. Unavoidably, for certain documents, the binarization process removes essential discriminate information. As suggested in the report of the DAS02 working group on document image analysis [52], more research should be devoted to the use of features extracted directly from gray-scale or color images to classify documents.

Before discussing the choice of document features further, we first consider the document recognition stage at which classification is performed.

4.1.1 Document recognition stages

Document classification can be performed at various stages of document processing. The choice of document features is constrained by the document recognition stage at which document classification is performed.

Figure 5 shows a typical sequence of document recognition for mostly-text document images [21]. *Block segmentation and classification* identify rectangular blocks (or zones) enclosing homogeneous content portions, such as text, table, figure, or half-tone image. *Physical layout analysis* (also called *structural layout analysis* or *geometric layout analysis*) extracts layout structure: a hierarchical description of the objects in a document image, based on the geometric arrangements in the image [54]. For example, WISDOM++ uses six levels of layout hierarchy: basic blocks, lines, sets of lines, frame 1, frame 2, and page [21]. *Logical layout analysis* (also called *logical labeling*) extracts logical structure: a hierarchy of logical objects, based on the human-perceptible meaning of the document contents [54]. For example, the logical structure of a journal page is a hierarchy of logical objects, such as title, authors, abstract, and sections [37].

Document classification can be performed at various recognition stages, as shown in Table 2. The choice of recognition stage depends on the goal of document classification and the type of documents.

4.1.2 Choice of document features

We characterize document features using three categories adapted from those discussed in [12]: image features, structural features and textual features. *Image features* are either extracted directly from the image (e.g. the density of black pixels in a region) or extracted from a segmented image (e.g. the number of horizontal lines in a segmented block). Image features extracted at the level of a whole image are called *global* image features; image features extracted from the regions of an image are called *local* image features. *Structural features* (e.g. relationships between objects in the page) are obtained from physical or logical layout analysis. *Textual features* (e.g. presence of keywords) may be computed from OCR output or directly from document images. Some classifiers use only image features, only structural features, or only textual features; others use a combination of features from several groups.

Table 2 Characterization of classifier architecture according to document features and recognition stage, feature representations, class models and classification algorithms, and learning mechanisms

	Document features	Feature representation	Class model and classification algorithm	Learning mechanism
Classification using image features (without physical layout analysis)				
Shin et al. [51]	Image features such as density, attributes of connected components, column/row gaps, etc.	Fixed-length vectors	Decision tree	Learn a decision tree (manually specify tree splitting and stopping criteria)
Bagdanov and Worring [3]	Various image features including global image features, zone features and text histogram	Fixed-length vectors	A variety of statistical classifiers (such as 1-NN, Nearest Mean, Linear Discriminant, Parzen classifier)	Learn parameters of statistical classifiers
Byun and Lee [10]	Features of lines in a form image	Fixed-length vectors representing difference of coordinate between two neighboring lines	Template matching based on only some areas of the form	Templates constructed automatically; automatically choose matching regions for each template
Hu et al. [26]	Block information of segmented document	Interval encoding using fixed-length vectors	Hidden Markov Model (HMM)	Learn probabilities of HMM (manually define model topology)
Héroux et al. [23]	Image features before block segmentation; Various levels of pixel densities in a form	Fixed-length vectors	K Nearest Neighbor (KNN)	Automatically populate NN space and learn weights for NN distance computation
			Neural Network	Learn weights (manually define network topology)
Shimotsuji and Asano [50]	The location and size of cells in a form	Center points of cells	Point matching using 2D hash table	Automatically construct hash table using one blank sample form per class
Ting and Leung [56]	Features of lines and text in a form	A string representing document features	String matching	Strings constructed automatically using one sample form per class
Classification using physical layout features				
Diligenti et al. [16]	Physical layout and local image features	Modified XY tree	Hidden Tree Markov Model (HTMM)	Learn probabilities of HTMM (manually define HTMM topology)
Baldi et al. [5]	Physical layout features	Modified XY tree	K Nearest Neighbor (KNN). The distance is tree-edit distance	Automatically populate NN space
Bagdanov and Worring [2,3]	Physical layout and the average point size of text, number of text lines in each zone	Attributed graph	First Order Gaussian Graphs	Learn probabilities of edges and vertices in First Order Gaussian Graphs
Cesarini et al. [12]	Physical layout features	Encode MXY tree into a fixed-length vector	Neural Network; MLP (Multi-layer Perceptron)	Learn weights of MLP (manually define MLP topology)
Appiani et al. STRETCH [1]	Physical layout and the average grey level of local regions	Modified XY tree	Document Decision Tree	Learn decision tree from a set of labeled MXY trees
Esposito et al. Wisdom++ [21]	Physical layout features	Using attributes and relations in a first-order language	A set of rules	Inductive rule learning (constrained rule format)
Wnek [62]	Physical layout features	A descriptive language based on representation space schema	A set of rules	Inductive rule learning (constrained rule format)
Héroux et al. [23]	Physical layout features	A tree representing a hierarchy of extracted blocks	Hierarchical tree matching	Learn tree models
Watanabe et al. [59]	Physical layout features	A global structure tree and local structure trees to describe global and local document characteristics	2D Decision Tree	Structure trees built automatically (manually build decision tree)

Table 2 continued

	Document features	Feature representation	Class model and classification algorithm	Learning mechanism
Classification using logical structure features				
Eglin and Bres [19]	Results of functional labeling	Pyramid images describing functional blocks	Linear classifier. Weighted combination of image correlation coefficient	Pyramid images constructed automatically using one representative page per class
Liang et al. [34]	Local image features, physical layout and logical structures	Layout graph	Logical graph matching	Layout graph model learned incrementally
Nattee and Numao [40]	Physical layout and logical structures	Fixed-length vectors	Winnow algorithm	Learned incrementally
Kochi and Saitoh [31]	Physical layout and logical structures	Fixed-length vectors	Template matching	Template constructed automatically (manually define logical structure of a template using a sample document per class)
Lam [33]	Spatial relation, physical and logical structural features	A hierarchy of frames	Knowledge-based approach	Document model automatically built based on manually defined knowledge
Classification using textual features				
Sako et al. [41,46]	Textual features and physical layout	Template based on content and location of keywords	Hierarchical template matching	Template constructed automatically; learn keywords
Spitz and Maghbouleh [53]	Textual features obtained before layout analysis and without OCR	Fixed-length vectors to represent frequency of Word Shape Tokens (WSTs)	Rocchio's algorithm, a technique in text categorization	Learn frequency of WSTs
Dengel OfficeMaid [4,15]	Textual features from OCR results. Layout and font attributes of keywords	A list of word alternatives and a set of rules	Combination of two classifiers, a neural net voting mechanism	Learn font attributes of keywords and extract words and text patterns.
Ittner et al. [28]	Textual features from OCR results	A fixed-length vector representing weights of index terms	Rocchio's algorithm, a technique in text categorization	Learn weights of index terms
Taylor et al. [55]	Textual features from OCR results and segmentation information	A set of rules	Two layer classification. Knowledge-based	Learn frequency of functional blocks (manually define rules to identify functional blocks)
Maderlechner et al. [8,36]	Textual features from OCR results	A list of words and their frequencies	Statistical method based on word relevance	Learn message type of specific words and their frequencies

The classifiers that use only image features are fast since they can be implemented before document layout analysis. But they may be limited to providing coarse classification, since image features alone do not capture characteristic structural information. More elaborate methods are needed to verify the classification result.

Structural features are necessary to classify documents with structural variations within a class. However, there is a risk to using high-level structural features: these rely on the results produced by physical layout analysis, a complex and error-prone process. Some classifiers obtain document layout information from the segmentation results produced by commercial OCR systems [3,12,34].

Most of the surveyed systems use a combination of physical layout features and local image features; this provides a good characterization of structured images. The classification is done before logical labeling, allowing the classification results to be used to tailor logical labeling. For example, Bagdanov and Worring use physical layout features to classify the document, and then adapt the logical labeling phase to the document class [3].

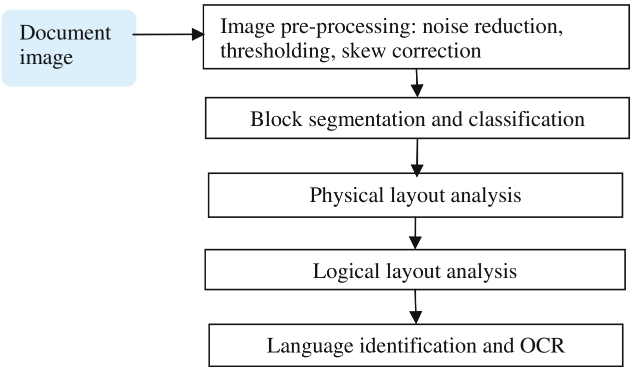

Document image
Image pre-processing: noise reduction, thresholding, skew correction
Block segmentation and classification
Physical layout analysis
Logical layout analysis
Language identification and OCR

Fig. 5 A typical sequence of document recognition for mostly-text document images. Adapted from [21]. Document recognition is not required to follow this order. For example, OCR may be performed before logical layout analysis, with OCR results used to perform logical layout analysis. Also, hypotheses produced in a later stage may be used to revise earlier hypotheses

Document classification using logical structural features is expensive since it needs a domain-specific logical model for each type of document. Early systems use manually-built logical models for each class [33]. The current trend is to learn models automatically from labeled samples [21,34]. However, document labeling is labor intensive, since logical meanings must be assigned to the physical layout objects in each training document.

Classification using textual features is closely related to *text categorization* in Information Retrieval [49]. Purely textual measures, such as frequency and weights of keywords or index terms, can be used on their own, or in combination with image features. Textual features may be extracted from OCR results which may be noisy [13, 28]. Alternatively textual features may be extracted directly from document images [51]. Techniques are being developed for classification based on OCR results from low-quality images. These include *n*-gram-based text categorization to reduce the effect of OCR errors [11] and morphological analysis [30]. The effects of noisy OCR results on classification performance are noticed and considered in the updated OfficeMAID system [15, 24].

4.1.3 Document features used in selected classifiers

We now describe the document features used in selected document classifiers. This elaborates on the summary in Table 2.

Shin et al. [51] measure document image features directly from the unsegmented bitmap image. The document features include density of content area, statistics of features of connected components, column/row gaps and relative point sizes of fonts. These features are measured in four types of windows: cell windows, horizontal strip windows, vertical strip windows and the page window.

Eglin and Bres [19,20] measure spatial positions of segmented blocks, and use the results of functional labeling. Functional labeling is a special case of logical labeling, which doesn't require information dependent on document types. Functional labeling uses texture features of the text blocks, including complexity and visibility.

Spitz and Maghbouleh [53] use Character Shape Codes for content-based document classification. Character Shape Codes rely on the gross shape and location of character images with respect to their text lines. Alphabetic Character Shape Codes are aggregated into Word Shape Tokens. The Word Shape Tokens are treated like keywords, and the frequency of their occurrences in each document is counted.

4.2 Feature representation

Document features extracted from each sample document in a classifier can be represented in various ways, such as a flat representation (fixed-length vector or string), a structural representation, or a knowledge base. Document features that do not provide structural information are usually represented in fixed-length feature vectors. Features that provide structural information are represented in various formats as summarized in Table 2.

4.2.1 Recommendations for choosing feature representations

Different classes of documents have different characteristics so they require different representation techniques. Diligenti et al. [16] discuss the effects of various formats of feature representation. They claim that a flat representation does not carry robust information about the position and the number of basic constituents of the image, whereas a recursive representation preserves relationships among the image constituents.

Watanabe [60] recommends using certain types of feature representations for each of the five categories of structured documents shown in Table 3. Watanabe also gives the following guideline for the selection of a feature representation: The simpler, the better. If the document can be represented using a list, then use a list because of higher processing efficiency, easier knowledge definition and management. Similarly, a tree representation is better than a graph representation due to its relative simplicity. A rule-based representation is powerful; however, it is complex and the interpretation phase takes longer.

Table 3 Five categories of structured documents and their feature representations [60]

Characteristics of documents	Examples of document classes	Recommended feature representation
Category 1: greatly restricted physical layout. Each item is in a fixed physical position	Forms, cheques	A list or frame
Category 2: physical layout varies, but there is a strong logical layout structure. Items have flexible positions, but relations exist among items	Business cards, letters	A tree representation
Category 3: restricted physical layout, with complex structure. Items may be hierarchical or repeated. Layout structure guided by lines, white space	Tables	Use two binary trees, a global and local structure tree
Category 4: global document structure predefined by physical layout structure, but space allocation for individual items is flexible	Newspaper-pages, article-pages	A rule-based representation
Category 5: standard elements, such as horizontal and vertical axes, axis labels	Bar business graph	A graph or network representation

The choice of a feature representation is also constrained by the kind of class model and classification algorithm that is used.

4.2.2 *Feature representations used in selected classifiers*

An overview of the use of feature representations is given in Table 2. We now describe a few of these representations in detail.

The XY-tree representation is a well-known approach for describing the physical layout of documents [39]. The root of an XY-tree is associated with the whole document image. The document is split into regions that are separated by white spaces. Horizontal and vertical cuts are alternately performed. Each tree node is associated with a document region. A modified XY-tree (MXY tree) is used in some classification systems; a region can be subdivided using either white spaces or lines [1,5,16]. Each node of the MXY tree contains a feature vector describing the region associated with the node. A disadvantage of an XY tree (or MXY tree) representation is that it can be strongly affected by noise and document skew [16].

Graph representations are used in some classification systems. Liang and Doermann represent document layout using a fully connected Attributed Relational Graph [34]. Each node corresponds to a segmented block on a page, and it also corresponds to a logical component. An edge between two nodes represents a spatial relation between the two corresponding blocks in the image. The spatial relation is decomposed into relations between vertical and horizontal block edges. The Attributed Relational Graphs in [2,3] are not fully connected. They model the relations between neighboring text zones only. Each node corresponds to a text zone in the segmented document image. The presence of an edge between two nodes indicates a Voronoi neighbor relation.

Several authors use fixed-length vectors as a feature representation. *Interval encoding* encodes region layout information in fixed-length vectors [26]. The block-segmented image is partitioned into an $m \times n$ grid. Each cell in the grid is distinguished as a text bin or a white space bin. Each row is represented as a fixed length vector, recording how far each text bin is from a white space bin. Cesarini et al. [12] encode an MXY tree into a fixed-length vector. The vector represents the occurrences of tree patterns consisting of three tree nodes.

Various feature representations are used in knowledge-based systems. For example, layout structures are represented in a first-order language, where attributes (e.g. height and length) are used to describe properties of a single layout component, while relations (e.g. contain, on-top) are used to express interrelationship among layout components [21]. Attributes and relations can be both symbolic and numeric.

4.3 Class models and classification algorithms

Class models define the characteristics of the document classes. The class models can take various forms, including grammars, rules, and decision trees; the class models are trained using features extracted from the training samples. They are either manually built by a person or automatically built using machine learning techniques. Class models and classification algorithms are tightly coupled, so we discuss them together. A class model and classification algorithm must allow for noise or uncertainty in the matching process. We begin by reviewing traditional statistical and structural pattern classification techniques that have been applied to document classification.

4.3.1 *Statistical pattern classification techniques*

There are many traditional statistical pattern classification techniques, such as Nearest Neighbor, decision tree, and Neural Network [18,29]. These techniques are relatively mature and there are libraries and classification toolboxes implementing these techniques. Traditional statistical classifiers represent each document instance with a fixed-length feature vector; this makes it difficult to capture much of the layout structure of document images. Therefore, these techniques are less suitable for fine-grained document classification [3].

Decision trees provide semantically intuitive descriptions of how decisions are made, and can have good performance with limited number of training samples [45]. Shin et al. [51] use a decision tree for document classification.

Neural Networks have been successfully used in many pattern recognition applications. A Multi-Layer Perceptron is a type of Neural Network that has advantages concerning decision speed and generalization capacity [23]. Multi-Layer Perceptrons have been used for document classification [12,23].

Eglin and Bres [19] use a linear combination classifier for coarse-grained document classification. The linear function is the weighted sum of correlation coefficients between the input image and the reference image for each class.

A Hidden Markov Model (HMM) is a powerful tool for probabilistic sequence modeling [27]. It is viewed as a particular case of Bayesian networks [6]. An HMM is robust, suitable for handling uncertainties and noise in document image processing [32]. Hu et al. [26] use a top-to-bottom sequential HMM to classify documents. The HMM states correspond to the vertical regions of a document, and the observations are the cluster centers of interval encoding.

4.3.2 *Structural pattern classification techniques*

In this section, we discuss traditional structural classification techniques [43], as well as those extending traditional statistical classification techniques to deal with structural feature representations. These techniques have higher computational complexity than statistical pattern recognition techniques. Also, machine learning techniques for creating class models based on structural representations are not yet standard. Many authors provide their own methods for training class models [1,16,33].

Decision trees can be extended to consider tree-based document representations [1,59]. A Document Decision Tree is used to classify documents [1]. The leaves of a Document Decision Tree contain labeled MXY trees, and the internal nodes contain common sub-trees extracted from MXY trees. A Document Decision Tree is built through the application of insertion, descending, and splitting operations. Splitting decisions are based on sub-tree similarity matching. In related earlier work, a Geometric Tree is automatically created to classify business letters based on physical layout [14].

Baldi et al. [5] use a tree-based K Nearest Neighbor classifier to classify pages, where the distance between pages is computed by means of tree-edit distance. They use an algorithm proposed by Zhang and Shasha to compute the tree-edit distance [64].

Diligenti et al. [16] propose the Hidden Tree Markov Model, an extension to HMM, to classify documents using structural features. A Hidden Tree Markov Model with 11 states is trained for each class. The state transitions are restricted to a left-to-right topology. Based on the view that HMM is a special case of Bayesian networks, the two main algorithms in Hidden Tree Markov Model (inference and parameter estimation) are derived from corresponding algorithms for Bayesian networks.

Graph matching is a common tool in structural pattern recognition [9]. General graph matching is NP-hard, but various heuristic graph-matching techniques can be used. Graph matching is used in document classification [3,34]. Bagdanov and Worring [2,3] introduce statistical uncertainty into the graph matching. They use First Order Gaussian Graphs to model document classes; these are extensions of First Order Random Graphs proposed by Wong et al. [63]. First Order Gaussian Graphs use continuous Gaussian distributions to model the densities of all random elements in a random graph instead of the discrete densities used by Wong et al. A First Order Gaussian Graph for each class is trained based on hierarchical entropy minimization techniques. Classification is done by computing the probability that an Attributed Relational Graph is an outcome graph of a First Order Gaussian Graph.

4.3.3 *Knowledge-based document classification techniques*

A knowledge-based document classification technique uses a set of rules or a hierarchy of frames encoding expert knowledge on how to classify documents into a given set of classes. This is described as an appealing, natural way to encode document knowledge [3]. The knowledge base can be constructed manually or automatically. Manually built knowledge-based systems only perform what they were programmed to do [33,55]. Significant efforts are required to acquire knowledge from domain experts and to maintain and update the knowledge base. Also it is not easy to adapt the system to a different domain [49]. Recently developed knowledge-based systems learn rules automatically from labeled training samples [21,62]. Rule learning is discussed further in Sect. 4.4.

4.3.4 Template matching

Template matching is used to match an input document with one or more prototypes of each class. This technique is most commonly applied in cases where document images have fixed geometric configurations, such as forms. Matching an input form with each of a few hundred templates is time consuming. Computational cost can be reduced by hierarchical template matching [41,46]. Byun and Lee [10] propose a partial matching method, in which only some areas of the input form are considered. Template matching has also been applied to broad classification tasks, with documents from various application domains such as business letters, reports, and technical papers [31]. The template for each class is defined by one user-provided input document, and the template does not describe the structure variability with the class. Therefore, the template is only suitable for coarse classification.

4.3.5 Combination of multiple classifiers

Multiple classifiers may be combined to improve classification performance [25]. The OfficeMAID system consists of two competing classifiers and a neural net voting mechanism [15,61]. One classifier uses a linear statistical method, based on word and layout information of certain keywords. The other classifier is based on rules, employing linguistic features such as text patterns and morphological information. Experimental results show that the performance of the voting method is higher than that of either of the two single classifiers. Héroux et al. [23] implement three classifiers for form classification: K Nearest Neighbor, Multi-Layer Perceptron and tree matching. K Nearest Neighbor and Multi-Layer Perceptron use image features as input. The tree matching uses structural features based on physical layout. Possible strategies for combining these classifiers include hierarchical combination, and parallel classifier application followed by voting.

4.3.6 Multi-stage classification

A document classifier can perform classification in multiple stages, first classifying documents into a small number of coarse-grained classes, and then refining this classification. Maderlechner et al. [36] implement a two-stage classifier, where the first stage classifies documents as either journal articles or business letters, based on physical layout information. The second stage further classifies business letters into 16 application categories according to content information from OCR. The OfficeMaid system also implements a two-stage classification [15]. The first stage identifies business letters from different senders [14] and the second stage classifies message types [61]. Classification performed in multiple stages requires multiple class models and classification algorithms. Most surveyed systems use single-stage classification.

This concludes our discussion of class models and classification algorithms.

4.4 Learning mechanisms

A learning mechanism provides an automated way for a classifier to construct or tune class models, based on observation of training samples. Hand coding of class models is most feasible in applications that use a small number of document classes, with document features that are easily generalized by a system designer. For example, Taylor et al. [55] manually construct a set of rules to identify functional components in a document and learn the frequency of those components from training data. However, manual creation of entire class models is difficult in applications involving a large number of document classes, especially when users are allowed to define document classes. With a learning mechanism, the classifier can adapt to changing conditions, by updating class models or adding new document classes.

The entire class model may be learned, or aspects of a manually defined class model may be tuned during learning. The last column of Table 2 describes the automated aspects of classifier construction, for each surveyed approach.

Methods for automatically learning traditional statistical models are well developed and there are many software packages available. Shin et al. [50] use OC1, an off-the-shelf decision tree software package to construct decision trees automatically. For some statistical models, training samples are used to tune parameters of the model [5]. Neural Network models typically involve manual specification of network topology; design samples are used to iteratively update the weights [12]. Hidden Markov Models typically involve manual specification of the structure of the model for each class; the probabilities used in each model are learned [26].

For structural models and knowledge-based models, automatic learning is complex, and learning methods are not standardized. In earlier work, class models for a small set of classes are created manually. However, as shown in Table 2, recently developed classifiers exhibit a trend toward increasing automation in the construction of class models. Esposito et al. [21] use Inductive Logic Programming to induce a set of rules from a set of labeled training samples.

There are challenges in automatically learning models from training samples. To generalize class models well, a sufficient number of well labeled training samples are necessary. Wnek [62] mentions that correct and representative labeled samples are crucial for the quality of learning rules. Providing sufficient training data for learning can be expensive. Baldi et al. [5] propose a method to expand the training set: new labeled samples are created by modifying the given labeled samples to simulate distortions occurring in segmentation. The distortions are modeled with tree grammars. The Winnow algorithm [7,35] can be used on-line to incrementally update class models [40]. The on-line nature of this algorithm makes the system more flexible and requires less time in the learning phase.

Class models differ in the amount of retraining needed when document classes change. When new document classes are added or existing document classes are changed, a Neural Network must be retrained from scratch, re-estimating all the weights in the network. In contrast, a Hidden Markov Model requires less training when the set of classes changes. It is not necessary to retrain all the Hidden Markov Models since each class has its own class model. Only the models for new or changed classes are trained on the document samples belonging to those classes. This localized retraining is important since many classifiers deal with a relatively large number of classes, and classes normally vary over time [16].

This concludes our discussion of the classifier architecture, with the four aspects: (1) document features and recognition stage, (2) feature representations, (3) class models and classification algorithms, and (4) learning mechanisms.

5 Performance evaluation

Performance evaluation is a critically important component of a document classifier. It involves challenging issues, including difficulties in defining standard data sets and standardized performance metrics, the difficulty of comparing multiple document classifiers, and the difficulty of separating classifier performance from pre-processor performance.

Performance evaluation includes the metrics for evaluating a single classifier, and the metrics for comparing multiple classifiers. Most of the surveyed classification systems measure the effectiveness of the classifiers, which is the ability to take the right classification decisions. Various performance metrics are used for classification effectiveness evaluation, including accuracy [1, 3], correct rate [34], recognition rate [10], error rate [55], false rate [21], reject rate [12], recall and precision [4,28]. The significance of the reported effective performance is not entirely standard, since some classifiers have reject ability while others do not, and some classifiers output a ranked list of results [1,26,62], while others produce a single result. Standard performance metrics are necessary to evaluate performance.

Document classifiers are often difficult to compare because they are solving different classification problems, drawing documents from different input spaces, and using different sets of classes as possible outputs. For example, it is difficult to compare a classifier that deals with fixed-layout documents (forms or table-forms) to one that classifies documents with variable layouts (newspaper or articles). Another complication is that the number of document classes varies widely. The classifiers use as few as 3 classes [21] to as many as 500 classes [62], and various criteria are used to define these classes. Also many researchers collect their own data sets for training and testing their document classifiers. These data sets are of varying size, ranging from a few dozen [10,26,34], or a few hundred [1,4], to thousands of document instances [62]. The sizes of training set and test set affect the classifier performance [18]. These factors make it very difficult to compare performance of document classifiers. The authors of WISDOM++ lead in the right direction by making data available on line (http://www.di.uniba.it/∼malerba/wisdom++/). Nattee and Numao [40] use the data provided by WISDOM++ and add their own data to test their classification system.

To compare the performance of two classifiers, a standard data set providing ground-truth information should be used to train and test the classifiers. The University of Washington document image database (UWI, II, and III) is one source of ground truth data for document image analysis and understanding research [44]. UW data is used for text categorization in [28,53]. Spitz and Maghbouleh [53] conclude that UW data is far from optimal for document classification, since it has a small number of documents from a relatively large number of classes. The set of classes defined for UW data by Spitz and Maghbouleh is one of many possible types of class definition for this data set. Finland's MTDB Oulu Document Database defines 19 document classes and provides ground truth information for document recognition [47]. The number of documents per class ranges from less than ten up to several hundred. The documents in this database are diverse, and assigned to predefined document classes, making this database a useful starting point for research into document classification. For example, the Oulu database is used in [19]. Further discussion of standard datasets may be found in the reports of the DASO2 working group [52]. They raise an interesting issue concerning the huge collection of documents in on-line Digital Libraries. How can document classification research make use of these documents? And how will document classification contribute to the construction of Digital Libraries?

It is difficult to separate classifier performance from pre-processor performance. The performance of a classifier depends on the quality of document processing performed prior to classification. For example, classification based on layout-analysis results is affected by the quality of the layout analysis, by the number of split and merged blocks. Similarly, OCR errors affect classification based on textual features. In order to compare classifier performance, it is important to use standardized document processing prior to the classification step. One method of achieving this is through use of a standard document database that includes not only labeled document images, but also includes sample results from intermediate stages of document recognition. Construction of such databases is a difficult and time-consuming task.

6 Conclusions

We summarize the document classification literature along three components: the problem statement, the classifier architecture, and performance evaluation. There are important research opportunities in each of these areas.

The problem statement is characterized in terms of the document space and the set of document classes. We need techniques for more formally specifying document classification problems. Current practice is to define each class via an informal English description and/or via sample documents. Neither gives a complete or precise definition of a document class. The ill-defined nature of the problem statement hampers many aspects of classifier development.

The classifier architecture includes four aspects: document features and recognition stage, feature representations, class models and classification algorithms, and learning mechanisms. We need techniques to better understand the effects of these four aspects. In current classifiers, these four aspects are so closely bound together that it is nearly impossible to evaluate any one of these aspects independently of the others. Our ability to make advances in classifier-construction technology depends on being able to investigate the effects of changing one of these aspects of classifier architecture.

Advances in performance evaluation techniques for document classifiers are needed. Existing standard document databases (University of Washington and Oulu) have been used to test document classifiers. There is need for larger standard databases, with many documents for each document class. These databases should include not only labeled document images, but also intermediate results from document recognition. This would allow document classifiers to be tested under the same conditions, classifying documents based on the same document-recognition results. Currently, it is difficult to separate classifier performance from the performance of preceding document-recognition steps.

Acknowledgements We gratefully acknowledge the financial support provided by the Xerox Foundation, and by NSERC, Canada's Natural Sciences and Engineering Research Council.

References

1. Appiani, E., Cesarini, F., Colla, A.M., Diligenti, M., Gori, M., Marinai, S., Soda, G.: Automatic document classification and indexing in high-volume applications. Int. J. Doc. Anal. Recognit. **4**(2), 69–83 (2001)
2. Bagdanov, A.D., Worring, M.: First order Gaussian graphs for efficient structure classification. Pattern Recognit. **36**(6), 1311–1324 (2003)
3. Bagdanov, A.D., Worring, M.: Fine-grained document genre classification using first order random graphs. In: Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, 10–13 September 2001, pp. 79–90 (2001)
4. Baumann, S., Ali, M., Dengel, A., Jäger, T., Malburg, M., Weigel, A., Wenzel, C.: Message extraction from printed documents – a complete solution. In: Proceedings of the 4th International Conference on Document Analysis and Recognition, Ulm, Germany, 18–20 August 1997, pp. 1055–1059 (1997)
5. Baldi, S., Marinai, S., Soda, G.: Using tree-grammars for training set expansion in page classification. In: Proceedings of the 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, 3–6 August 2003, pp. 829–833 (2003)
6. Bengio, Y., Frasconi, P.: An input output HMM architecture. In: Tesauro, G., Touretzky, D., Leen, T. (eds.) Advances in Neural Information Processing Systems, vol. 7, pp. 427–434. MIT, Cambridge (1995)
7. Blum, A.: On-line algorithms in machine learning. In: Fiat, A., Woeginger, G. (eds.) Online algorithms: the state of the art, vol. 1442, pp. 306–325. Springer, Berlin Heidelberg New York (1998)
8. Brükner, T., Suda, P., Block, H., Maderlechner, G.: In-house mail distribution by automatic address and content interpre-

tation. In: Proceedings of the 5th Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, USA, April 1996, pp. 67–75 (1996)

9. Bunke, H.: Recent developments in graph matching. In: Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain, 3–8 September 2000, vol. 2, pp. 2117–2124 (2000)
10. Byun, Y., Lee, Y.: Form classification using DP matching. In: Proceedings of the 2000 ACM Symposium on Applied Computing, Como, Italy, 19–21 March 2000, pp. 1–4 (2000)
11. Cavnar, W., Trenkle, J.: N-gram-based text categorization. In: Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, USA, 1994, pp. 161–175 (1994)
12. Cesarini, F., Lastri, M., Marinai, S., Soda, G.: Encoding of modified X–Y trees for document classification. In: Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, 10–13 September 2001, pp. 1131–1136 (2001)
13. Dengel, A., Bleisinger, R., Fein, F., Hoch, R., Hönes, F., Malburg, M.: OfficeMAID – a system for office mail analysis, interpretation and delivery. In: Proceedings of International Association for Pattern Recognition Workshop on Document Analysis Systems, Kaiserslautern, Germany, October 1994, pp. 253–275 (1994)
14. Dengel, A., Dubiel, F.: Clustering and classification of document structure – a machine learning approach. In: Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, Canada, 14–15 August 1995, pp. 587–591 (1995)
15. Dengel, A.: Bridging the media gap from the Guthenberg's world to electronic document management systems. In: Proceedings of 1997 IEEE International Conference on Systems, Man, and Cybernetics, Orlando, Florida, USA, October 1997, pp. 3540–3554 (1997)
16. Diligenti, M., Frasconi, P., Gori, M.: Hidden Tree Markov Models for document image classification. IEEE Trans. Pattern Anal. Mach. Intell. **25**(4), 519–523 (2003)
17. Doermann, D., Rivlin, E., Rosenfeld, A.: The function of documents. Int. J. Comput. Vision **16**(11), 799–814 (1998)
18. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. Wiley, New York (2001)
19. Eglin, V., Bres, S.: Document page similarity based on layout visual saliency: application to query by example and document classification. In: Proceedings of the 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, 3–6 August 2003, pp. 1208–1212 (2003)
20. Eglin, V., Bres, S.: Analysis and interpretation of visual saliency for document functional labeling. Int. J. Doc. Anal. Recognit. **7**(1), 28–43 (2004)
21. Esposito, F., Malerba, D., Lisi, F.A.: Machine learning for intelligent processing of printed documents. J. Intell. Inf. Syst. **14**(2–3), 175–198 (2000)
22. Haralick, R.: Document image understanding: geometric and logical layout. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, Seattle, 20–24 June 1994, pp. 385–390 (1994)
23. Héroux, P., Diana, S., Ribert, A., Trupin, E.: Classification method study for automatic form class identification. In: Proceedings of the 14th International Conference on Pattern Recognition, Brisbane, Australia, 16–20 August 1998, pp. 926–929 (1998)
24. Hoch, R.: Using IR techniques for text classification in document analysis. In: Proceedings of the 17th International ACM-SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, July 1994, pp. 31–40 (1994)
25. Ho, T.K.: Multiple classifier combination: lessons and next steps. In: Kandel, A., Bunke, H. (eds.) Hybrid Methods in Pattern Recognition. World Scientific, Singapore, pp. 171–198 (2002)
26. Hu, J., Kashi, R., Wilfong, G.: Document classification using layout analysis. In: Proceedings of the 1st International Workshop on Document Analysis and Understanding for Document Databases, Florence, Italy, September 1999, pp. 556–560 (1999)
27. Huang, X.D., Ariki, Y., Jack, M.A.: Hidden Markov Models for Speech Recognition. Edinburgh University Press, Edinburgh (1990)
28. Ittner, D.J., Lewis, D.D., Ahn, D.D.: Text categorization of low quality images. In: Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, USA, 1995, pp. 301–315 (1995)
29. Jain, A.K., Duin, P.W., Mao, J.: Statistical pattern recognition: a review. IEEE Trans. Pattern Anal. Mach. Intell. **22**(1), 4–37 (2000)
30. Junker, M., Hoch, R.: Evaluating OCR and non-OCR text representation for learning document classifiers. In: Proceedings of the 4th International Conference on Document Analysis and Recognition, Ulm, Germany, 18–20 August 1997, pp. 1060–1066 (1997)
31. Kochi, T., Saitoh, T.: User-defined template for identifying document type and extracting information from documents. In: Proceedings of the 5th International Conference on Document Analysis and Recognition, Bangalore, India, 20–22 September 1999, pp. 127–130 (1999)
32. Kopec, G.E., Chou, P.A.: Document image decoding using Markov source models. IEEE Trans. Pattern Anal. Mach. Intell. **16**(6), 602–617 (1994)
33. Lam, S.: An adaptive approach to document classification and understanding. In: Proceedings of International Association for Pattern Recognition Workshop on Document Analysis Systems, Kaiserslautern, Germany, October 1994, pp. 231–251 (1994)
34. Liang, J., Doermann, D., Ma, M., Guo, J.K.: Page classification through logical labelling. In: Proceedings of the 16th International Conference on Pattern Recognition, Quebec, Canada, 11–15 August 2002, pp. 477–480 (2002)
35. Littlestone, N.: Learning quickly when irrelevant attributes abound: a new linear threshold algorithm. Mach. Learn. **2**(4), 285–318 (1988)
36. Maderlechner, G., Suda, P., Brückner, T.: Classification of documents by form and content. Pattern Recognit. Lett. **18**(11–13), 1225–1231 (1997)
37. Mao, S., Rosenfeld, A., Kanungo, T.: Document structure analysis algorithms: a literature survey. In: Proceedings of Document Recognition and Retrieval X (IS&T/SPIE electronic imaging), Santa Clara, California, USA, 20–24 January 2003, SPIE Proceedings Series **5010**, 197–207 (2003)
38. Nagy, G.: Twenty years of document image analysis in PAMI. IEEE Tran. Pattern Anal. Mach. Intell. **22**(1), 38–62 (2000)
39. Nagy, G., Seth, S.: Hierarchical representation of optically scanned documents. In: Proceedings of the 7th International Conference on Pattern Recognition, Los Alamitos, California, USA, 1984, pp. 347–349 (1984)
40. Nattee, C., Numao, M.: Geometric method for document understanding and classification using on-line machine learning. In: Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, 10–13 September 2001, pp. 602–606 (2001)
41. Ogata, H., Watanabe, S., Imaizumi, A., Yasue, T., Furukawa, N., Sako, H., Fujisawa, H.: Form type identification for banking applications and its implementation issues. In: Proceedings

of Document Recognition and Retrieval X (IS&T/SPIE electronic imaging), Santa Clara, California, 20–24 January 2003, SPIE Proceedings Series **5010**, 208–218 (2003)

42. Okun, O., Doermann, D., Pietikäinen, M.: Page segmentation and zone classification: the state of the art. Technical report, LAMP-TR-036, University of Maryland, College Park (1999)
43. Pavlidis, T.: Structural pattern recognition, 2nd edn. Springer, Berlin Heidelberg New York (1980)
44. Phillips, I.T., Chen, S., Haralick, R.: CD-ROM document database standard. In: Proceedings of the 2nd International Conference on Document Analysis and Recognition, Tsukuba, Japan, 20–22 October 1993, pp. 478–483 (1993)
45. Quinlan, R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers, San Mateo, CA (1993)
46. Sako, H., Seki, M., Furukawa, N., Ikeda, H., Imaizumi, A.: Form reading based on form-type identification and form-data recognition. In: Proceedings of the 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, 3–6 August 2003, pp. 926–930 (2003)
47. Sauvola, J., Kauniskangas, H.: MediaTeam document database (http://www.mediateam.oulu.fi/MTDB/), Oulu University, Finland (1999)
48. Schenker, A., Last, M., Bunke, H., Kandel, A.: Classification of web documents using a graph model. In: Proceedings of the 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, 3–6 August 2003, pp. 240–244 (2003)
49. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surveys **34**(1), 1–47 (2002)
50. Shimotsuji, S., Asano, M.: Form identification based on cell structure. In: Proceedings of the 13th International Conference on Pattern Recognition, Vienna, Austria, August 1996, vol. C, pp. 793–797 (1996)
51. Shin, C., Doermann, D., Rosenfeld, A.: Classification of document pages using structure-based features. Int. J. Doc. Anal. Recognit. **3**(4), 232–247 (2001)
52. Smith, E.B., Monn, D., Veeramachaneni, H., Kise, K., Malizia, A., Todoran, L., El-Nasan, A., Ingold, R.: Reports of the DAS02 working group. Int. J. Doc. Anal. Recognit. **6**(3), 211–217 (2004)
53. Spitz, A.L., Maghbouleh, A.: Text categorization using character shape codes. In: Proceedings of Document Recognition and Retrieval VII (IS&T/SPIE electronic imaging), San Jose, California, 23–28 January 2000, SPIE Proceedings Series **3967**, 174–181 (2000)
54. Tang, Y.Y., Cheriet, M., Liu, J., Said, J.N., Suen, C.Y.: Document analysis and recognition by computers. In: Handbook of Pattern Recognition and Computer Vision, 2nd edn. World Scientific, Singapore, pp. 579–612 (1998)
55. Taylor, S., Lipshutz, M., Nilson, R.: Classification and functional decomposition of business documents. In: Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, Canada, 14–15 August 1995, pp. 563–566 (1995)
56. Ting, A., Leung, M.: Business form classification using strings. In: Proceedings of the 13th International Conference on Pattern Recognition, Vienna, Austria, August 1996, vol. B, pp. 690–694 (1996)
57. Trier, D., Jain, A.K., Taxt, T.: Feature extraction methods for character recognition – a survey. Pattern Recognit. **29**(4), 641–662 (1996)
58. Wang, Y., Phillips, I.T., Haralick, R.: A study on the document zone content classification problem. In: Proceedings of the 5th International Workshop on Document Analysis Systems, Princeton, NJ, USA, 19–21 August 2002, pp. 212–223 (2002)
59. Watanabe, T., Luo, Q., Sugie, N.: Layout recognition of multikinds of table-form documents. IEEE Trans. Pattern Anal. Mach. Intell. **17**(4), 432–445 (1995)
60. Watanabe, T.: A guideline for specifying layout knowledge. In: Proceedings of Document Recognition and Retrieval VI (IS&T/SPIE electronic imaging), San Jose, CA, 27 January 1999, SPIE Proceedings Series **3651**, 162–172 (1999)
61. Wenzel, C., Baumann, S., Jäger, T.: Advances in document classification by voting of competitive approaches. In: Proceedings of International Association for Pattern Recognition Workshop on Document Analysis Systems, Malvern, Pennsylvania, October, 1996, pp. 352–372 (1996)
62. Wnek, J.: Learning to identify hundreds of flex-form documents. In: Proceedings of Document Recognition and Retrieval VI (IS&T/SPIE electronic imaging), San Jose, CA, 27 January 1999, SPIE Proceedings Series **3651**, 173–182 (1999)
63. Wong, A.K.C., Constant, J., You, M.L.: Random graphs. In: Bunke, H., Sanfeliu, A. (eds.) Syntactic and Structural Pattern Recognition: Theory and Applications. World Scientific, Singapore. pp. 197–236 (1990)
64. Zhang, K., Shasha, D.: Simple fast algorithms for the editing distance between trees and related problems. SIAM J. Comput. **18**(6), 1245–1262 (1989)