Image Based Retrieval and Keyword Spotting in Documents

24

Chew Lim Tan, Xi Zhang, and Linlin Li

Contents

C.L. Tan (✉)
Department of Computer Science, National University of Singapore, Singapore, Singapore
e-mail: tancl@comp.nus.edu.sg

X. Zhang • L. Li
National University of Singapore, Singapore, Singapore
e-mail: xizhang@comp.nus.edu.sg

D. Doermann, K. Tombre (eds.), *Handbook of Document Image Processing and Recognition*, DOI 10.1007/978-0-85729-859-1_27,

Abstract

The attempt to move towards paperless offices has led to the digitization of large quantities of printed documents for storage in image databases. Thanks to advances in computer and network technology, it is possible to generate and transmit huge amount of document images efficiently. An ensuing and pressing issue is then to find ways and means to provide highly reliable and efficient retrieval functionality over these document images from a vast variety of information sources. Optical Character Recognition (OCR) is one powerful tool to achieve retrieval tasks, but nowadays there is a debate over the trade-off between OCR-based and OCR-free retrieval, because of OCR errors and wastage of time to OCR the entire collection into text format. Instead, image-based retrieval using document image similarity measure is a much more economical alternative. Till now, many methods have been proposed to achieve different sub-tasks, all of which contribute to the final retrieval performance. This chapter will present different methods for presenting word images and preprocessing steps before similarity measure or training and testing and discuss different algorithms or models for achieving keyword spotting and document image retrieval.

Keywords

Character shape coding • Document image retrieval • Holistic word representation • Keyword spotting

Introduction

Background and History

Over the years, various ways have been studied to query on document images using physical layout, logical structure information, and other extracted contents such as image features using different levels of abstraction. However, for document images containing predominantly text, the traditional information retrieval (IR) approach using keywords is still often used. For these document images, conventional document image processing techniques can be easily utilized for this purpose. For instance, many document image retrieval systems first convert the document images into their text format using Optical Character Recognition (OCR) techniques (for details about OCR techniques, please see ▸Chap. 10 (Machine-Printed Character Recognition)) and then apply text IR strategies over the converted text documents. Several commercial systems have been developed based on this model, using page segmentation and layout analysis techniques and then applying OCR. These systems include Heinz Electronic Library Interactive Online System (HELIOS) developed by Carnegie Mellon University [1], Excalibur EFS, and PageKeeper from Caere. All these systems convert the document images into their electronic representations to facilitate text retrieval.

Generally the recognition accuracy requirements for document image retrieval are considerably lower than that for document image processing tasks [2].

A document image processing system will analyze different text regions, understand the relationships among them, and then convert them to machine-readable text using OCR. On the other hand, a document image retrieval system asks whether an imaged document contains particular words which are of interest to the user, ignoring other unrelated words. Thus, essentially a document image retrieval system answers "yes" or "no" to the user's query, rather than exact recognition of characters and words as in the case of document information processing. This is sometimes known as keyword spotting with no need for correct and complete character recognition but by directly characterizing image document features at character, word, or even document level. Motivated by this observation, some efforts are concentrating on tolerating OCR recognition errors in document image retrieval.

However, high costs and poor quality of document images often prohibit complete conversion using OCR. Moreover, non-text components in a document image cannot be easily represented in a converted form with sufficient accuracy. Under such circumstances, it may be advantageous to explore techniques for direct characterization and manipulation of image features in order to retrieve document images containing text and other non-text components. So far, efforts made in this area include applications to word spotting, document similarity measurement, document indexing, and summarization. Among all these, one approach is to use particular codes, known as character shape codes [3], to represent characters in a document image instead of a full conversion using OCR. It is virtually a trade-off between computational complexity and recognition accuracy. This method is simple and efficient, but has the problem of ambiguity. Moreover, in order to obtain character codes, correct character segmentation is important which may not be possible in some cases, such as when characters are interconnected or overlapping resulting in segmentation errors. To address this issue, word-level methods are proposed which treat a single word as a basic unit for recognition or keyword spotting, or even segmentation free methods are proposed to avoid any segmentation errors.

In addition to segmentation, there are other important issues in document image retrieval research. Font shape is one such issue. Different fonts have their distinguished, intrinsic patterns. Thus features which are perfect for one kind of font may not be proper for another. Marinai et al. [4] proposes font adaptive word indexing method. However, font is one of the many characteristics the imaged document has, and many other features should be chosen carefully to adapt to different situations, such as different languages. Besides, inflectional languages also present partial word recognition problem. For example, "develop" is the partial word for "development," "developed," "develops," etc. Complete matching will miss out other variants of the query word which should be retrieved in keyword spotting.

While imaged document retrieval has been mainly on printed document images in the past, there have been rising research interests on handwritten document image retrieval particularly in view of the increasing number of handwritten materials being scanned in digital format for safe keeping or for public access. For instance, valuable historical documents are digitized for permanent storage. Bank checks and handwritten receipts are scanned or photographed as records for

accounting purposes. Handwritten medical records are kept in image format for future reference. Handwriting recognition presents even much greater challenges to OCR which often fails miserably due to high variability of writing styles, severe noisy condition, and poor image quality (please see ▸Chap. 12 (Continuous Handwritten Script Recognition) for challenges in handwritten text recognition). Various approaches based on raw and uncorrected OCR'ed text or direct image features have been studied [5] for handwritten document retrieval.

Need for Document Image Retrieval

There is a perception in the industry and even among IT professionals that document image retrieval is only a temporary solution which will be phased out eventually due to technological progresses into the paperless age with online applications and publishing on the internet [6]. Paper-based documents are viewed as legacy documents which will disappear in the foreseeable future. The truth of the matter is that the rapid advancement in printing technology has in fact resulted in rapid proliferation of hard copy documents resulting in many "save trees" slogans by cutting down on printing. The call for the paperless office in the 1980s has in fact been met with even more papers [7]. On the other hand, affordable scanning and digital photography devices enable people to capture document contents as images with ease. Many paper-based documents ended up in image format rather than in their electronic equivalents. From a different perspective, the conversion of paper documents into image format appears to be a never-ending process for many years to come. This is because of the fact that there are still an enormous collection of documents and books in printed form that need to be converted to images. Huge collections of historical documents in libraries and national archives are waiting to be digitized before they deteriorate with time. Massive collection of out-of-print or rare books can only be preserved by turning them into images. Google's effort in digitizing books on a grand scale [8] is only touching the tip of an iceberg.

Yet from another perspective, there are various reasons that many documents will continue to appear in paper form. From a forensic point of view, paper-based documents are still much preferred over their electronic versions. Willful forgeries on papers, however sophisticated they may be, can be easily detected with modern forensic technology. Many legal documents, payment bills, and receipts must be presented in paper format for reasons such as the need for tangible proofs and personal or corporate privacy. For some legal requirements, paper-based documents should be scanned and OCRed, but they should be kept online in image format. The OCR results should be viewed as an "annotation" of the document image and not as the definitive representation. Publishers will continue to print books rather than making them available in electronic format for obvious reasons to prevent unauthorized copying and distribution. Today, in many parts of the world for some cultures and languages, paper-based documents are still prevalent in their societies. While advances in the internet world have led to many standards for semantics of electronic documents such as XML to facilitate search and retrieval,

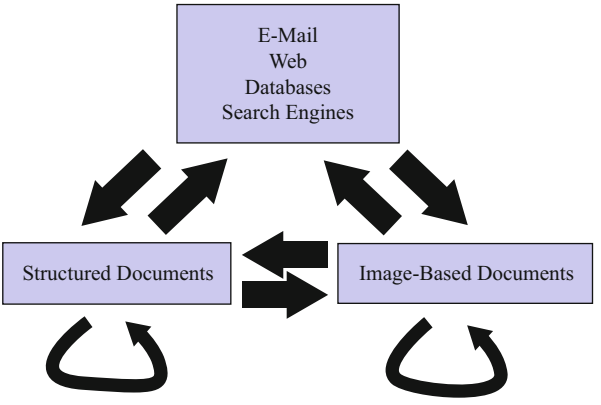


Fig. 24.1 Major role of image-based documents alongside with structured electronic documents in future document retrieval systems (From [6])

they are unable to handle document images. Document image retrieval definitely has a place in the modern world of information retrieval [6]. It is foreseen that in future document retrieval systems, image-based documents will play a major role alongside structured electronic documents as depicted in Fig. 24.1.

A scenario for a particular need for document image retrieval can be seen when there is a huge collection of legacy or historical document images scanned or captured with digital cameras under some very poor conditions. Doing OCR for the entire collection is an overkill as most of them may not be read after all, not to mention the likely poor quality output hampering readability. Instead, document image retrieval using word features to spot keywords and hence to facilitate rapid search for candidate documents in their original image rendition for easy reading will be a more economical and practical solution. A web application is presented in [9] for retrieving scanned legacy documents from a digital library. As shown in Fig. 24.2, all imaged documents are preprocessed to convert them into some code representation. When a user inputs a set of query words, the input words are converted into image features using the same coding scheme to allow speedy matching for document retrieval.

Chapter Overview

In the following sections, methods for document image retrieval and keyword spotting will be introduced in detail. In section "Word Image Representation," different methods for presenting word images in Character Shape Coding, holistic word representation, and handwritten word representation will be first presented. Section "Keyword Spotting'' will next describe how these different word representation schemes are used to achieve keyword spotting in printed and handwritten documents. Section "Document Image Retrieval" will then describe how the methods are used in retrieval of printed and handwritten document images. Finally, section "Conclusion" will conclude this chapter with an outlook for the future.

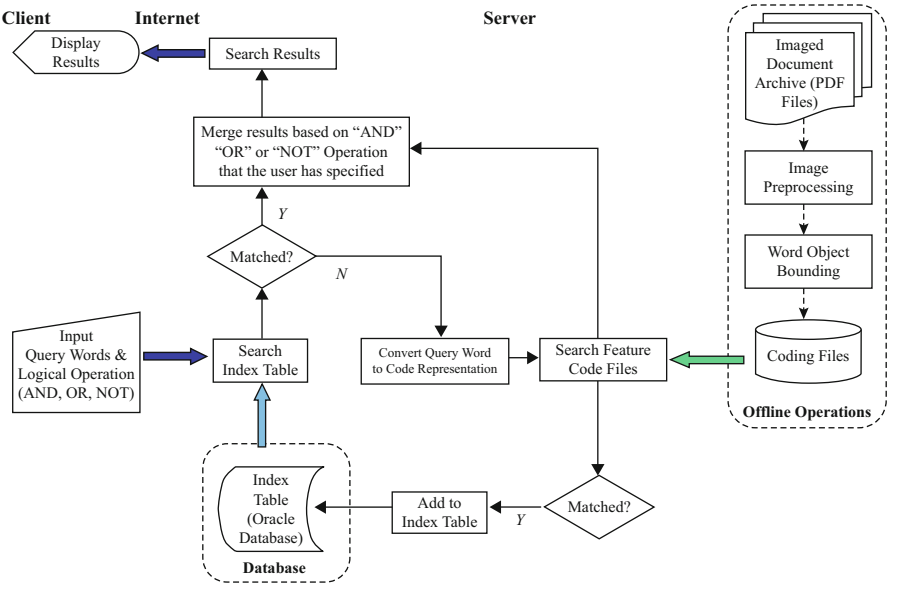


Fig. 24.2 A web-based document image retrieval system (From [9])

Word Image Representation

Character Shape Coding

Due to the language dependency and the execution time of OCR, Character Shape Coding technique (referred to as word shape coding in some literature [10]) has been proposed. Character Shape Coding methods take an individual character as input and map character objects to a smaller symbol set. For example, English characters can be mapped to six different shape codes, by locating the baseline and x-line from the given text line images [3], namely, classifying characters based on ascender, descender, and dot. Figure 24.3 shows the positions of the baseline and x-line of an example text line, and Table 24.1 presents the mappings between shape codes and character images based on the positions of connected components in each character image. For example, the word shape coding representation for the word "left" is "AxAA," which is named as a token, and a sequence of shape codes represents a word image. However, different word images may be mapped to the same token because of the confusions between several characters represented by the same sequence of shape codes. In order to reduce the confusions, characters represented by **x** shape code are further partitioned into two groups, based on the fact that confusions between e and s,r,a,o,n happen most frequently. Shown as Fig. 24.4, considering the middle part of the character image, it can be seen that "c" and "e" have a deep east concavity, but the others have little or none. Based on this

Fig. 24.3 Positions of the baseline and x-line (From [3])

Table 24.1 Mappings of character images to a set of shape codes

Shape code	Character images	Number of components	Component position	Component bottom position
A	A-Zbdfhklt	1	Above x-line	Baseline
x	acemnorsuvwxz	1	x-line	Baseline
g	gpqy	1	x-line	Below baseline
i	i	2	x-line	Baseline
j	j	2	x-line	Below baseline

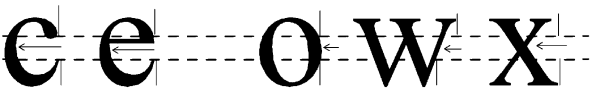


Fig. 24.4 "c" and "e" have a deep east concavity in their middle parts, but others also represented by **x** shape code have little or none (From [3])

observation, "c" and "e" are represented by a new shape code **e** and the rest are represented by **x** as before. Therefore, for a given text file, all generated sequences of shape codes are used for retrieval tasks, with fewer confusions.

Because the encoding (mapping) process is based on a set of simple and universal image features, Character Shape Coding methods are fast computable and language independent to a certain level. Therefore, they are widely employed in document image retrieval applications with speed constraints and language identification for Indo-European languages.

Holistic Word Representation

Holistic word representation treats each word as a whole entity and thus avoids the difficulty of character segmentation. This is different from the character-level processing strategy taken by OCR and Character Shape Coding and is exactly why holistic word representation is robust to degraded image quality. In particular, broken and touching characters are one of the major document image degradation factors, and holistic word representation is naturally immune to them. It even works when some letters are ambiguous. Another important reason why holistic word representation presents an attractive alternative lies in its apparent similarity in the approach to how humans read text. Hull [11] points out the fact that according to psychological experiments, humans do not read text character by character, but, rather, they recognize words or even small groups of nearby words while they are reading. In the same way, features can be extracted at the word level instead of at the character level. Lu and Tan [12] proposed a word image coding technique,

Table 24.2 Values of ADA and the corresponding positions of primitives

Character value of ω	The corresponding position of the primitive
"x"	The primitive is between the x-line and the baseline
"a"	The primitive is between the top boundary and the x-line
"A"	The primitive is between the top boundary and the baseline
"D"	The primitive is between the x-line and the bottom boundary
"Q"	The primitive is between the top-boundary and the bottom boundary

which is a little more complicated compared with Spitz's method introduced in section "Character Shape Coding," but with an impressive advantage of much less ambiguity. For a given word image, it is segmented into several discrete entities from left to right, each of which is named as a primitive. A word image is represented by a sequence of features extracted from all primitives. There are two attributes used to describe a primitive: one is called Line-or-Traversal Attributes (LTA) denoted by σ and the other one is called Ascender-and-Descender Attribute (ADA) denoted by ω. Therefore, the word image with n primitives are represented by $< (\sigma_1, \omega_1)(\sigma_2, \omega_2) \cdots (\sigma_n, \omega_n) >$. For ADA, ω has five character values, i.e., $\omega \in \Omega = \{'x', 'a', 'A', 'D', 'Q'\}$. Each character value represents a characteristic of one primitive, shown in Table 24.2. Different lines in Table 24.2 are illustrated in Fig. 24.5. For LTA, first, the straight stroke lines (SSL) are extracted from the whole word image, and only the vertical and diagonal strokes are under consideration. An example of extracted SSLs are shown in Fig. 24.5a. There are three types of SSLs: vertical stroke, left-down diagonal stroke, and right-down diagonal stroke, evaluated as "l," "w," and "v," respectively. Furthermore, if a primitive has "x" or "D" ADA value, it is necessary to check whether there is a dot on top of the vertical stroke line, and if so, reassign "i" or "j" to the LTA value of the primitive. Next, traversal features are extracted from the remaining image, shown in Fig. 24.5b, by scanning each column. T_N is the traversal number which indicates the number of black-to-white transitions or vice versa in one column. Table 24.3 shows the feature codes for different values of T_N. The combination of the values of SSL features and traversal features are treated as the LTA value of the word image. Consequently, a word image is represented by a series of ADA and LTA values for all primitives, which can be further used to achieve information retrieval or keyword spotting tasks.

Holistic word representation approaches are widely employed in keyword spotting and document image retrieval for degraded document images when traditional OCR or Handwritten Word Recognition (HWR) fails, such as historical handwritten documents, scanned images of envelopes, and ancient (medieval) manuscript archives presenting a major challenge for OCR or HWR.

Handwritten Word Representation

Scanned handwritten documents in image format are becoming common in digital libraries and other image databases. Google and Yahoo have now made handwritten

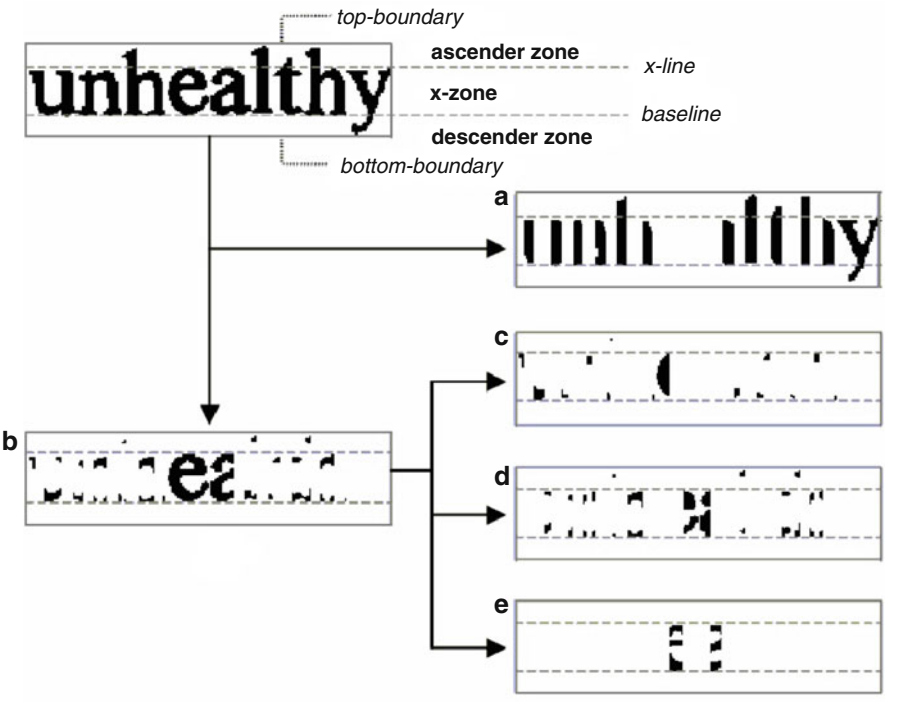


Fig. 24.5 Features of primitives (From [12]). (**a**) Straight stroke line features, (**b**) remaining strokes after removing strokes in (**a**) from the original image, (**c**) traversal $T_N = 2$, (**d**) traversal $T_N = 4$, (**e**) traversal $T_N = 6$

Table 24.3 Feature codes for different values of T_N

The value of T_N	Feature code
0	θ, indicating the space between two consecutive characters
2	κ: the ratio of the number of black pixels to the x-height; $\xi = D_m/D_b$: D_m is the distance from the top-most pixel to the x-line, and D_b is the distance from the bottom-most pixel to the baseline
4,6,8	Similar to $T_N = 2$

documents available for their search engines [13]. Due to severe degradation of such images because of noise contents and imperfect structure, OCR results are usually very poor with handwritten documents. It is therefore impossible to convert handwritten documents in full-text format and store it as such. Research on how to efficiently and correctly index and retrieve these handwritten imaged documents are hence imperative. Tomai et al. [14] discuss how difficult recognizing degraded

historical manuscripts is and that OCR is not a proper tool for historical manuscript recognition. Recent methods are based on keyword spotting for handwritten documents as an alternative and effective way for handwritten document retrieval. While many keyword spotting methods have been developed for printed document images with excellent performances, handwritten imaged documents are quite different from their printed counterparts due to their own particular characteristics. Hence, methods performing well for printed documents do not work well for handwritten documents. For handwritten documents, the following aspects require attention. Firstly, unlike printed documents which generally have predefined structures for different document types, the layout of handwritten documents is unconstrained. Many handwritten texts contain scribbling of characters in disarray and in any direction. Secondly, there is a high variability of writing styles among different writers. Even for the same writer, the writing style may change under different writing conditions. Thirdly, handwritten texts present an even much greater challenge for character or word segmentation as writing tends to exhibit a high degree of continuity between strokes. Antonacopoulos and Downton [15] discuss special problems in dealing with handwritten documents. See also ▸Chap. 12 (Continuous Handwritten Script Recognition) on such problems.

In order to recognize handwritten documents, various methods for preprocessing steps have been proposed, such as text detection [16], line extraction [17], binarization, and noise reduction or removal [18]. In particular, baseline detection and slant removal are even more important which are used to reduce variations of writing styles. The simple method for baseline detection is based on the horizontal projection histogram, and a selected threshold is used to separate the main body part and the ascender or descender. Another supervised machine learning algorithm [19] trains a convolution classifier to classify each local extrema of a word image to locate the baselines. Figure 24.6 shows the definition of labels for each local extrema and how the convolution classifier works. This method can handle more complicated and cursive handwritten words. One of the global methods for slant correction is based on the vertical projection histograms of the shear transformed word images by all possible slant angles [20]. This kind of method tries to correct the slant by removing the slant angles for long vertical strokes in the word image. However, such global slant correction methods suffer from the assumption that all characters or long strokes are slanted by the same angle. To address this problem, local slant correction methods are proposed. After calculating the slant angles for each position of the word image, dynamic programming can be used to decide the optimal slant angle for each position, by propagating the slant angles of long verticals to their neighbors [21]. Figure 24.7 presents the main idea of local slant correction based on dynamic programming.

Furthermore, methods of representation of handwritten texts have been investigated, such as word segmentation using each word as a single unit for recognition [22] and text line segmentation to avoid word segmentation errors [23]. A sequence of feature vectors extracted from each column in the normalized word or text line images are used to represent handwritten documents. One feature vector is extracted

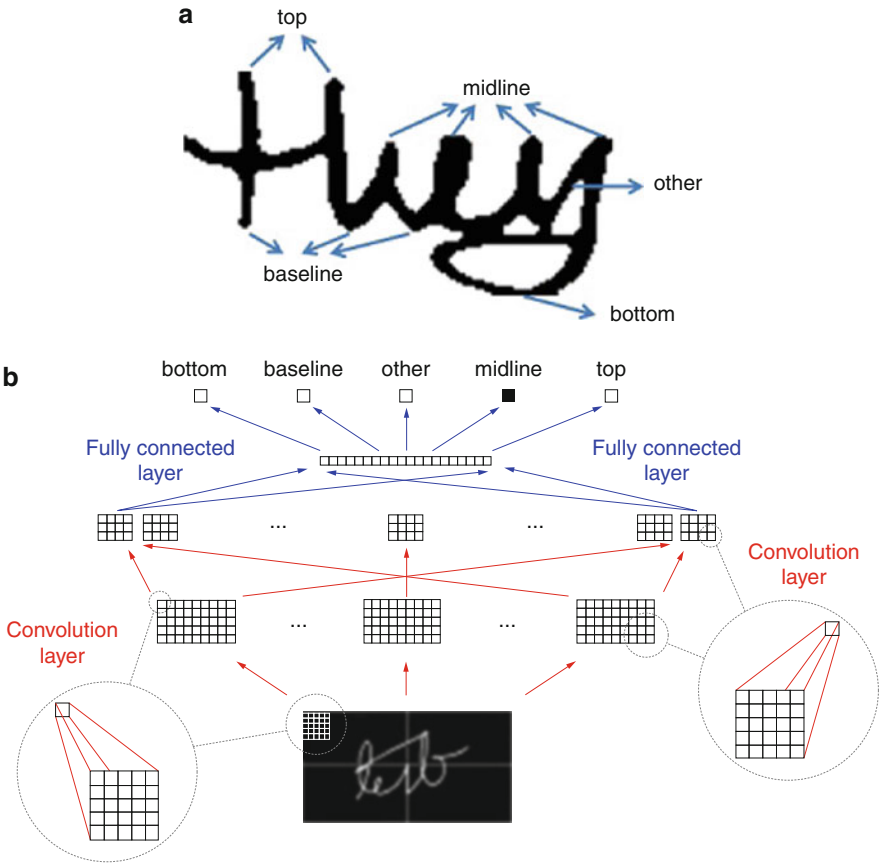


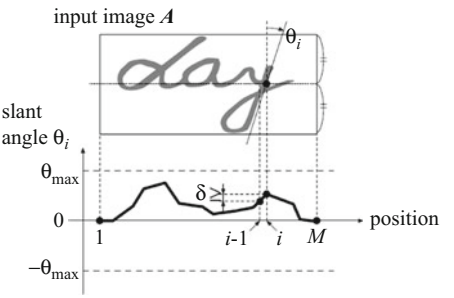
Fig. 24.6 Baseline detection based on supervised machine learning. (**a**) Labels of local extremas. (**b**) The convolution classifier (From [19])

in a sliding column window with width of 1 pixel and moves from left to right along the whole word image. A set of usually used features are listed below:

1. The number of black pixels
2. The center of gravity of the pixels
3. The second order moment of the sliding window
4. The location of the uppermost pixel
5. The location of the lowermost pixel
6. The orientation of the uppermost pixel
7. The orientation of the lowermost pixel
8. The number of black-white transitions
9. The number of black pixels divided by the number of all pixels between the upper- and lowermost pixels

Fig. 24.7 Local slant correction method based on dynamic programming (From [21]). (**a**) Slant angles for each position of the word image. (**b**) Dynamic programming for local slant correction

a



b

1 **for all** $\theta_1 \in \Theta$ **do** $g_1(\theta_1) := s_1(\theta_1)$;
2 **for** $i = 2$ **to** M **do**
3 **for all** $\theta_i \in \Theta$ **do begin**
4 $g_i(\theta_i) := \max_{\theta_{i-1}} [g_{i-1}(\theta_{i-1}) + f_i(\theta_i \mid \theta_{i-1})]$;
5 $b_i(\theta_i) := \theta_{i-1}$ which gives the maximum at Step 4;
6 **end;**
7 $\overline{\theta}_M := \operatorname{argmax}_{\theta_M \in \Theta} g_M(\theta_M)$;
8 **for** $i = M$ **downto** 2 **do** $\overline{\theta}_{i-1} := b_i(\overline{\theta}_i)$;

Fig. 24.8 The first image shows the best links between disconnected CCs, namely, *lines* in between "W," "in," "ches," and "ter." The second image shows the corresponding binary mask, and the last image represents the final contour of the word image (From [24])

Besides, a single closed contour of a word image can also be used to represent handwritten words, even without the need for skew and slant correction [24]. Connected components (CCs) are first extracted from the binary word image, and small groups of pixels are discarded. CCs are ordered based on the horizontal positions of their gravity centers. Next, add the best connecting link between a pair of successive CCs. Only the links pairing the contour points of two CCs, the ends of which are both inside, or within a close distance to the main body part, or far from the main body part, are treated as the valid link. The shortest valid link is the best connecting link between two successive CCs. After all the best links are generated, the final binary mask are created and an ordered contour tracing procedure is applied to the mask to get the closed contour which is used to represent the word image. Figure 24.8 shows the intermediate results of the contour extraction.

Keyword Spotting

Character Shape Coding-Based Keyword Spotting

Character Shape Coding approaches still suffer from broken and touching characters in keyword spotting and thus is difficult to be applied to degraded document images. Marinai et al. [4] proposed a coding scheme, in which character shape codes are not explicitly mapped to their ASCII equivalent; instead, connected components, namely, the character objects (CO), are clustered by an unsupervised approach self-organizing map (SOM). Each character prototype CO used for CO labeling is scaled to a predefined size of grid, and the feature vector of the CO image is represented by density values of each grid item. For example, if the grid has the size of 8×10, the dimension of the feature vector is 80. The SOM is arranged into an $N \times M$ feature map, and each neuron of the SOM, SOM(i, j), indicates the center of one cluster. SOM(i, j) accepts the average of feature vectors of all COs assigned to the corresponding cluster, denoted by Prot(i, j) and other neurons from the map as input, so that Prot(i, j) can be treated as the pattern for a particular cluster. Then, the high-dimension feature vector Prot(i, j) is mapped to 2D space at the end of SOM training, which is refined by the Sammon mapping to denote the position of each neuron in the SOM map, namely, $S(i, j) = (X(i, j), Y(i, j))$. Figure 24.9 shows a trained SOM, each neuron of which represents the CO in the cluster closest to the prototype.

Based on the trained SOM, a query word can be represented as follows:

1. COs, $\{\mathrm{CO}_h(0), \mathrm{CO}_h(1) \cdots \mathrm{CO}_h(n-1)\}$ are extracted from the query word image, W_h, each of which is then labeled with a SOM neuron, SOM(i_k, j_k), whose Prot(i, j) has the smallest Euclidean distance to $\mathrm{CO}_h(k)$.
2. Each CO is divided into a predefined number of vertical slices, and each slice is labeled by the SOM neuron of the largest CO overlapping it.
3. The word image is finally represented by the SOM neurons assigned to groups of slices, called the zoning vector, $\vec{Z}_h$. For example, as shown in Fig. 24.10, the word "France" is first separated into six COs, labeled as $(7, 2), (5, 0), (0, 7), (7, 5), (7, 10), (5, 10)$, respectively, and then each CO is further divided into three or four slices. The final representation of the word image "France" is {(7,2), (7,2), (7,2), (7,2), (5,0), (5,0), (5,0), (5,0), (0,7), (0,7), (0,7), (7,5), (7,5), (7,5), (7,5), (7,10), (7,10), (7,10), (7,10), (5,10), (5,10), (5,10)}.

Thus the similarity between two word images is evaluated as the Euclidean distance between their zoning vectors. Since there is no direct mapping between ASCII characters and shape code strings, a word image has to be generated for any text query to calculate its Character Shape Coding string. This word shape code scheme is suitable for indexing printed documents for which current OCR engines are less accurate. This language independent and unsupervised system beat OCR in Boolean keyword retrieval application in six datasets of different fonts and languages.

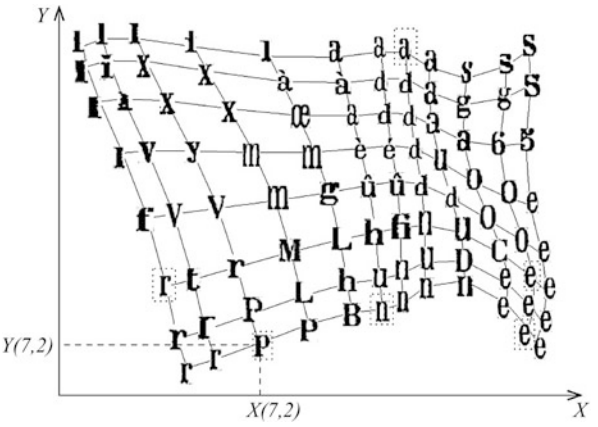


Fig. 24.9 The graph presentation of the trained SOM (From [4])

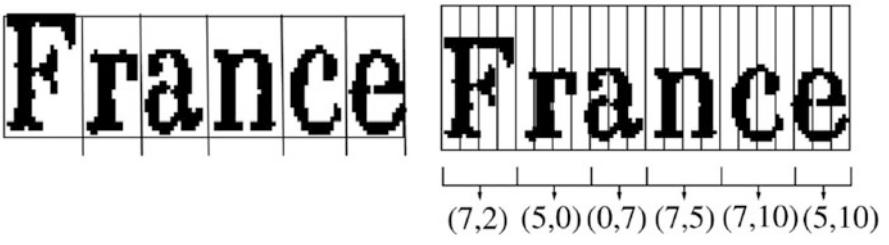


Fig. 24.10 An example of the representation of a word image based on SOM (From [4])

Holistic Word Representation-Based Keyword Spotting

Keyword spotting is to locate the occurrences of keywords in a document image. It is the most fundamental search type and the building block of Boolean document image search which treats any document with the presence of the keyword as relevant.

With holistic word representation-based keyword spotting, document images are firstly segmented into word images; features of word images are extracted and usually stored as a sequence of feature vectors. A word image synthesized from a set of character samples of various fonts is generated for the query. Feature vectors of the query are then computed. A distance d between the query word and a candidate word is computed, and an occurrence is found when d is below a certain threshold.

Ho et al. [25] proposed a method for word recognition in degraded images of machine-printed postal addresses on envelopes based on word shape analysis. Word images are first partitioned into a 4×10 grid resulting in 40 cells. Local direction contribution is extracted from each cell in the grid and recorded in a feature vector, which stores the normalized number of pixels belonging to four categories of strokes (east-west, northeast-southwest, north-south, and northwest-south) as shown

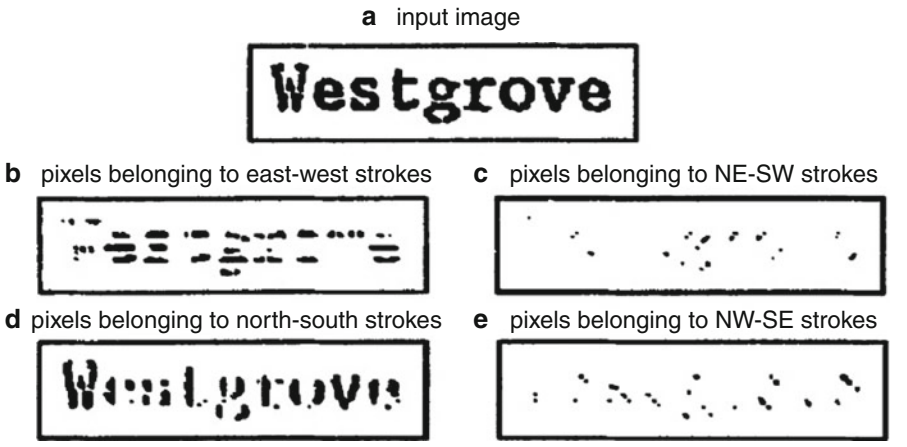


Fig. 24.11 An example of the four categories of strokes. The current run length of each black pixel is calculated in each of the four directions, and the pixel is labeled as one direction in which the run length of the pixel is maximum (From [25])

in Fig. 24.11. By concatenating all feature vectors, each image is represented by a 160-dimensional feature vector. Given an input image, the computed feature vector is compared to a given lexicon. Every word in the lexicon is synthesized into feature vectors for different font samples. Based on the distances calculated by summing over the absolute differences between corresponding feature components, a ranking list is produced. The top number of words in the ranking list are deemed to be most similar to the input image.

For document images where even word-level segmentation is not feasible, Leydier et al. [26] propose a method to detect zones of interest within an image. They explore several word image descriptors and point out that the gradient orientation is the best descriptor. Together with cohesive elastic matching algorithm, this method compares only informative parts of the query with only parts of the documents that may contain information, totally avoiding segmentation. Query images are manually obtained from the archive. In this approach identical words with different forms like singular and plural, "ing" and "ed," might be classified as different classes. In their later work [27], a method to generate query image for text query is proposed by concatenating the glyphs for Alphabets, which are extracted from the target collection. With this method, grammatical variations of the keywords can also be searched at the same time.

Machine-printed words have basically similar forms even with degradation. The handwritten words, however, have much more variance in terms of visual appearance. Therefore, the generation of query image is very essential for the overall performance, especially when the word classes have severe variance, such as a handwritten image collection of multiple words, which will be discussed later.

The keyword spotting paradigm, which is based on matching algorithms, may not be easily scalable to large document collections. It is very time-consuming to calculate the mutual distances between the query template and all word instances. Hence pruning, a fast rejection step in order to eliminate impossible word hypothesis, is often employed using simple word image features, such as the ratio of width to height, before the matching step.

Keyword Spotting for Handwritten Documents

In [28], the authors present a holistic word approach to HWR. The approach treats each word as a single, individual entity and recognizes separated words based on their overall shape characteristics, rather than recognizing individual components in a word, such as using OCR. There are two kinds of HWR, off-line HWR which deals with the recognition tasks after the words have been written down already and online HWR which deals with the recognition tasks while the words are being written and temporal information such as position and speed of writing are collected for later recognition. Off-line HWR will be the focus in the following discussion. Before recognition, individual words are separated, and in order to achieve the segmentation and recognition, different features are extracted directly from the original images. The author presents a host of methods that perform the recognition of separated words or phrases with proper lexicon. The off-line HWR is limited, because it fully depends on the lexicon provided. Features which represent word image characteristics and are used for segmentation and recognition have important influence on performance and should be treated carefully.

A Hidden Markov Model (HMM)-based method which trains subword models can also be used to achieve keyword spotting for handwritten documents [29]. Especially, word segmentation is not needed and the keyword can be arbitrary which is not necessary in the training set. One HMM model is used to represent a character, so that a word or text line can be presented by concatenating a sequence of character HMMs together. Figure 24.12a shows a filter model which can generate any sequence of characters. Denoted by the filter model, it is possible to get a sequence of characters with a probability for a word or text line image. In order to achieve keyword spotting, a keyword model, shown in Fig. 24.12b, is used for decoding and a probability can be computed indicating how likely a text line image contains the keyword. In other words, if the probability is higher than a threshold, it is likely that the text line contains the keyword, otherwise, the keyword probably does not appear in the text line. The proposed method performs better than standard template matching algorithm based on Dynamic Time Warping (DTW) [30] and can adopt different writing styles because binary text line images are normalized prior to feature extraction in order to cope with different writing styles. This method is based on good separation and normalization of each text line.

In order to extract handwritten words, which is essential for final performance, Rodríguez-Serrano et al. [31] developed a supervised system to locate keywords in handwritten document images of multiple writers, with trained word class models,

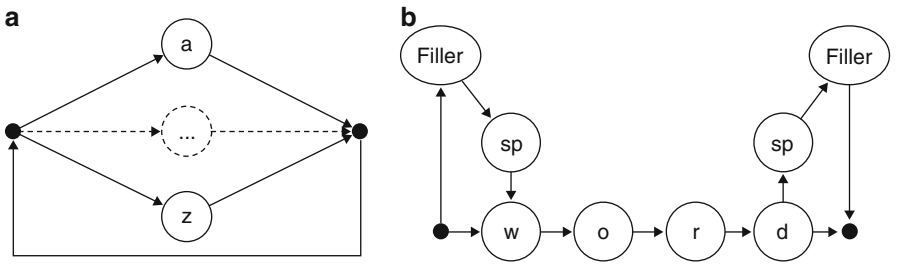


Fig. 24.12 HMM model for keyword spotting (From [29]). (**a**) Filter model. (**b**) Keyword model

instead of character models. For a word image X, the probability to be word W_i is $P(W_i|X)$. From Bayes's rule, the probability will be

$$\log p(W_i|X) = \log p(X|W_i) + \log P(W) - \log p(X_i)$$

With a labeled training set, $p(X|W_i)$ is modeled by HMM and $p(X)$ is modeled using Gaussian Mixture Model (GMM). Note that $\log p(W)$ is a constant for all W_i. In this way, the system need not implicitly generate the query image. Many features are tried to represent the word images, but, the best performance is obtained using local gradient histogram (LGH) features. They also found out that semicontinuous HMMs are superior when labeled training data is scarce as low as one sample per keyword, compared with continuous HMMs. A disadvantage of general HMM-based keyword spotting is that a sufficient large of training data is required. However, this spotting system, shown in Fig. 24.13, can easily be extended to word recognition given appropriate training data.

HMMs have several general drawbacks. First, the probability of the observation is only dependent on the current state, so that useful context information cannot be modeled easily. Secondly, HMMs are always generative and need to model the distribution of the observed variables, but sequence labeling tasks are discriminative. Finally, the number of hidden states for HMM models and the number of GMM models for each state are always decided on experimental efforts, and the estimation of these two parameters is prone to be task dependent. Based on these observations, Artificial Neural Network (ANN) is recently used in handwritten retrieval tasks. Particularly, Recurrent Neural Network (RNN), one kind of ANN, accepts one feature vector at one time and makes full use of all useful context information to estimate the output label. Thus, RNN can be an alternative to HMMs and also overcomes the drawbacks of HMMs. But, traditional RNN suffers one limitation in that it requires pre-segmented data, because a separate output label is required for each position in the input sequence. In order to overcome this limitation, a specially designed output layer Connectionist Temporal Classification (CTC) for RNN is used to directly map the input sequence to the label sequence with a probability, without the need of pre-segmented data.

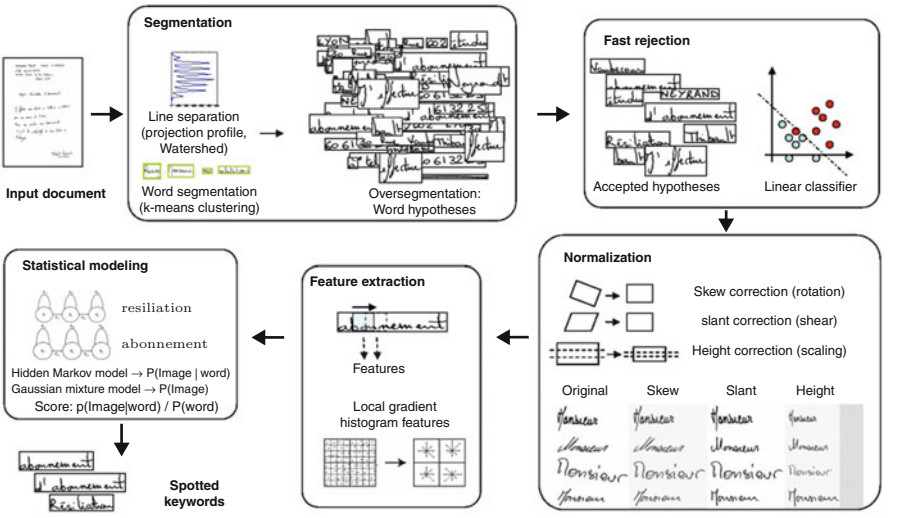


Fig. 24.13 Handwritten keyword spotting system (From [31])

Keyword spotting using BLSTM (Bidirectional Long Short-Term Memory) with CTC for off-line handwritten documents is introduced in [32]. Figure 24.14a shows a traditional RNN architecture, and Fig. 24.14b presents a special hidden block LSTM designed to address one limitation of traditional RNN, i.e., the range of context information RNN can accept is very limited in practice, which is called the vanishing gradient problem. The gates in the LSTM block are used to let RNN store and access information over a long period of time. During preprocessing, the document is segmented into individual text lines automatically, and feature vectors are extracted for each text line, which is then sent to the input layer of the BLSTM. There are as many nodes as the number of labels in the CTC output layer, plus one extra "blank" or "no-label" node. y_k^t denotes the probability of label k appearing at time t in the input sequence based on the output of the BLSTM, and $p(\pi|x)$ denotes the probability of observing a path π given the input sequence x with length T, where π is a sequence of labels with the same length T and $p(\pi|x) = \prod_{t=1}^{T} y_{\pi_t}^t$, π_t is the label at time t along π. There are many paths which may be mapped to the same label sequence by the function B, i.e., $B(\pi)$ removes the blanks and repeated labels in π, so that the probability of a label sequence l given x is the sum of probabilities of all possible paths mapped to l: $p(l|x) = \sum_{\pi \in B^{-1}(l)} p(\pi|x)$. The above equation allows the training of the BLSTM with CTC without pre-segmented data. For decoding, it is to find an l with the highest probability given the input x, combining with a dictionary and a language model G, which is formulated as $l^* = \arg\max_l p(l|x, G)$. Associated with a modified CTC Token Passing algorithm in [33] in order to get an approximate solution for l^*, unconstrained handwriting recognition and template-free keyword spotting are achieved. It is not necessary for

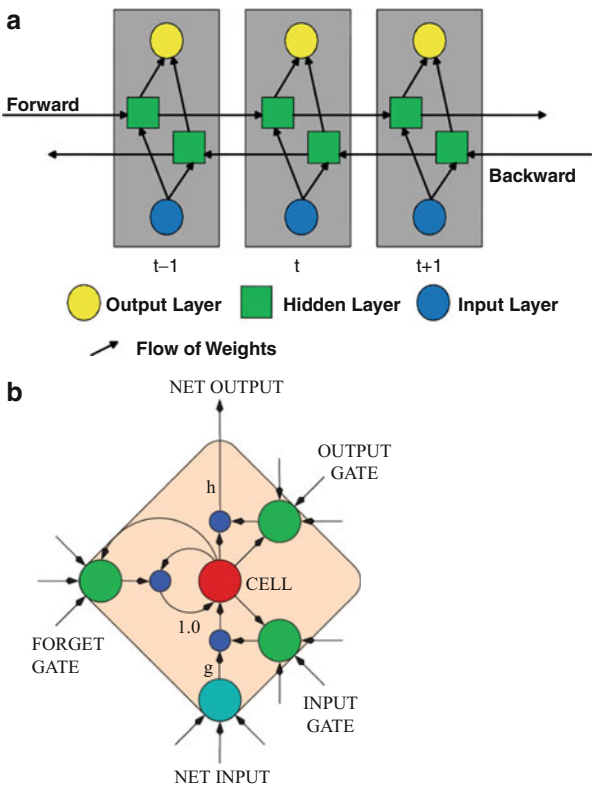


Fig. 24.14 The structure of BLSTM Neural Network. (**a**) Bidirectional RNN with three time frames unfolded. (**b**) An LSTM hidden block with one cell for RNN (From [33])

the query keyword to appear in the training set. Furthermore, the method does not require word segmentation which makes it a very flexible matching paradigm.

Summary of Keyword Spotting Methods

The keyword spotting methods presented in this section are summarized in Table 24.4 as a comparison in terms of their word representation, output, and their pros and cons.

Document Image Retrieval

Retrieval Models

Two statistical information retrieval (IR) models are widely used in relation to imaged document retrieval, namely, vector space (VS) model [34] and probability model [35].

Table 24.4 Keyword spotting methods

Method	Word representation	Output	Pros and cons
Character shape Coding-based method [4]	A word image is divided into character objects each assigned an SOM neuron. Further subdivision of each character object into vertical slices allows representation of a word image as a neuron sequence forming a zoning vector	Similarity between two word images is based on the distance between their zoning vectors	This method is language independent and is able to tolerate degraded documents with broken or touching characters. But results are not satisfactory when documents contain words with large variant and unseen font style
Word shape-based approach [25]	Local stroke direction distribution extracted from each cell of the grid is used to present a word image	A ranking list of a given lexicon, with the top being the most likely word of the input word image	This method treats a word image as a whole and avoids error of character segmentation. It can deal with degraded documents with variable quality but may suffer from word segmentation errors
Zones of interest-based segmentation-free method [26]	The keyword image and the documents to be searched are divided into zones of interest(ZOIs) to allow mapping between ZOIs based on features of gradient orientation	Occurrences of the keyword in documents, ranked by score	This method avoids word segmentation errors. But some words with different forms, such as variant thickness of strokes or character height, may be classified into different classes. The method is also sensitive to fonts
HMM-based method [29]	Word or text line images are presented as a sequence of feature vectors, each extracted from an image column	Probability of a word or text line image containing the keyword and the starting and ending position for each character of the keyword	HMM models can be trained on word or text line images to avoid segmentation errors. However, transcripts of training data should be available and a large amount of training data is needed. HMM models may not make full use of context
RNN-based method [32]	Feature vectors extracted from each column of the input images	Likelihood of a word or text line image containing the keyword and the keyword position in the input image	RNN can incorporate more context information. It can deal with discriminative tasks, i.e., sequence labeling, better than HMM models which are always generative. RNN can also accept word or text line images as input to avoid segmentation errors. RNN however may incur more processing time

The basic assumption of VS model is that a fixed sized term set (lexicon) is used to characterize both documents and queries, and the terms are un-related (orthogonal) to each other. In particular, a lexicon $T = \{t_1, t_2, \ldots, t_\tau\}$ is predefined, and a document D is represented as $\{w_{D,1}, w_{D,2}, \ldots, w_{D,\tau}\}$ where $w_{D,i}$ is the weight of term t_i in this document. The query Q is represented in the same manner. The similarity between Q and D is usually estimated by cosine distance:

$$\text{sim}(Q, D) = \frac{\sum w_{D,i} \times w_{Q,i}}{\sqrt{\sum w_{D,i}^2 \times w_{Q,i}^2}} \tag{24.1}$$

where $w_{D,i}$ is a statistical measure used to evaluate how important a word is to a document in a collection. $w_{D,i}$ could simply be the frequency of t_i in D. However, the most widely used term weighting scheme is *tf.idf* (term frequency and inverse document frequency). *tf* for term t_i is the number of occurrences of t_i in document D over the sum of number of occurrences of all terms in document D, while *idf* for term t_i is obtained by dividing the total number of documents by the number of documents containing the term and then taking the logarithm of that quotient. Other weighting schemes such as [36] are proven to be more efficient in text classification task.

Probability model ranks documents according to their relevance to the query. The ranking function is

$$f(D) = \log \frac{P(r|D)}{P(nr|D)} \tag{24.2}$$

where $P(r|D)$ means the probability of D being relevant to the query, while $P(nr|D)$ means the probability of being not relevant. Based on Bayes' Rules,

$$P(r|D) = \frac{P(D|r)P(r)}{P(D)} \tag{24.3}$$

and

$$P(nr|D) = \frac{P(D|nr)P(nr)}{P(D)} \tag{24.4}$$

Thus

$$f(D) = \log \frac{P(D|r)P(r)}{P(D|nr)P(nr)} \tag{24.5}$$

where these four probabilities could be estimated by a training step: given a training set, $P(r)$ and $P(nr)$ are the probability of relevant and irrelevant documents. With the term irrelevant assumption taken by the VS model, $P(D|r)$ and $P(D|nr)$ can be evaluated based on the presence (or absence) of individual terms t_i in the relevant set or irrelevant set. After simplification,

$$f(D) = \sum_{i=1}^{n} x_i \times \log \frac{P(x_i = 1|r)(1 - P(x_i = 1|r))}{P(x_i = 1|nr)(1 - P(x_i = 1|nr))}$$

where x_i is a binary representation of term t_i. If t_i is present in the collection $x_i = 1$, otherwise $x_i = 0$. $P(x_i = 1|r)$ is the probability of t_i present in relevant documents, while $P(x_i = 1|nr)$ is the probability of t_i present in the irrelevant documents. There are complicated formulae based on non-binary independence assumption, but they are not well tested.

Different ways are employed to evaluate these four probabilities. The most widely used on is proposed by Robertson and Sparck Jones [35]:

$$f(D) = \sum_{i=1}^{\tau} x_i \times \log \frac{r(i)(N - n(i) - R + r(i))}{(n - r(i))(R - r(i))} \tag{24.6}$$

where $r(i)$ is the number of relevant documents containing a query term t_i, N is the number of documents in the collection, $n(i)$ is the number of documents containing t_i, and R is the number of relevant documents.

Normally a threshold is predefined, if the distance between the query and a document is within the threshold, the document is considered as "relevant" by the IR system. Alternative, all top k-ranked documents are considered as "relevant."

Two well-accepted measures for the evaluation of retrieval effectiveness are recall and precision. Recall is the ratio of the number of relevant documents returned to the total number of relevant documents in the collection, and precision is the ratio of relevant documents returned to the total number of documents returned.

Relevance feedback uses users' judgments about the relevance of documents to select appropriate terms to expand the query.

Typical query expansion process is as follows extract terms from relevant documents; weight and rank them; choose top k terms, either add them automatically or display them for user to select; add these new terms to the original query; and perform another search. A term re-ranking function proposed by Robertson and Sparck Jones [35] is as follows:

$$rw_i = \log \frac{(r(i) + 0.5)(N - n(i) - R + r(i) + 0.5)}{(n(i) - r(i) + 0.5)(R - r(i) + 0.5)} \tag{24.7}$$

Character Shape Coding-Based Document Image Retrieval

For the Character Shape Coding-based document image retrieval, document images are firstly converted and stored as symbol strings. A query is translated into the same symbol representation by means of a table lookup. The search can be done at the text level.

In Character Shape Coding methods, each word is mapped uniquely to a corresponding symbol string, but one symbol string may be mapped to several real words because of the reduced symbol set, leading to ambiguity. For example, in Spitz's coding scheme, both "left" and "loft" have identical shape code string

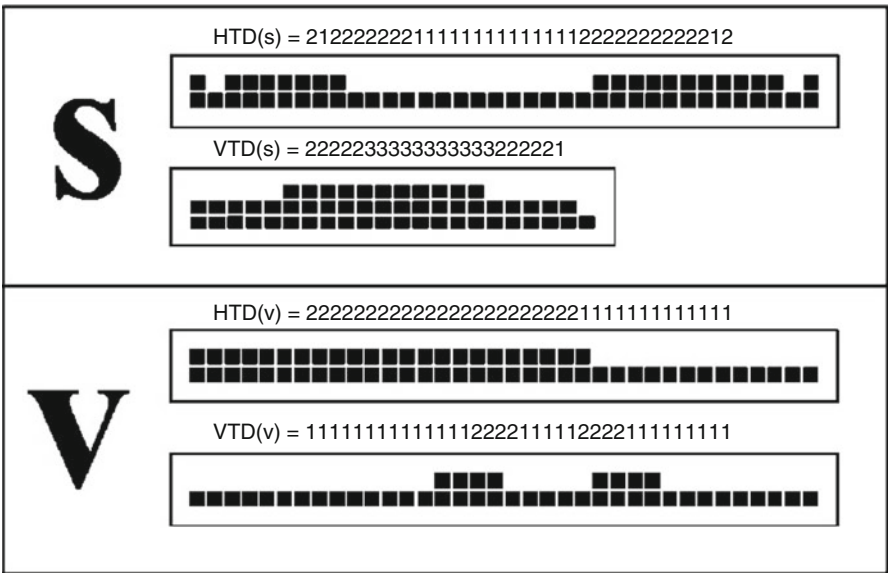


Fig. 24.15 Two examples of character coding (From [37])

"AxAA." The ambiguity level is different from method to method. It is an important retrieval performance indicator.

Tan et al. [37] propose a Character Shape Coding scheme for imaged document retrieval by text. The encoding was based on the vertical traverse density (VTD) and horizontal traverse density (HTD) of the character object, shown in Fig. 24.15. HTD is a vector indicating the numbers of line segments as scanning the character horizontally line by line from top to bottom. VTD is another vector obtained from vertical scanning from left to right. After characters are extracted from documents, each character object i can be represented by $(\text{HTD}_i, \text{VTD}_i)$. For two character objects, i and j, their distance D_{ij} is defined as

$$D_{ij} = \text{diff}(\text{HTD}_i, \text{HTD}_j) + \text{diff}(\text{VTD}_i, \text{VTD}_j) \tag{24.8}$$

where $\text{diff}(V_i, V_j)$ is used to calculate the distance between two vectors. Based on the distance between any two character objects, all character objects in one document image can be grouped into a set of classes, each of which is presented by the mean of all objects in the corresponding class, namely, the centroid of the class. Because different document images can have different number of classes, and the centroids of classes may have different dimensions, normalization of the centroid vector is needed. If the original vector is $V = v_1 v_2 \cdots v_n$, the normalized vector is $V' = v'_1 v'_2 \cdots v'_{n_c}$, where the constant n_c is the dimension of the normalized vector, then $v'_i = v_{(i \times n/n_c)}, \forall i \in [1, n_c]$. After all centroids of classes in all documents are normalized to the same dimension n_c, the distance between two classes are

computed, and if two classes have a very close distance, they are merged to the same class. Consequently, equivalent character objects from different documents can be denoted by the same unified class. A sequence of n-grams is obtained by moving a sliding window with n items along the text, one character forward at one time. In order to assign a unique number to every different n-gram and keep track of the frequencies of all distinctive n-grams, a hash table is created and used as a vector to represent a document image. So, the similarity between two document images is calculated as

$$\text{Similarity}(X_m, X_n) = \frac{\sum_{j=1}^{J} x_{mj} x_{nj}}{\sqrt{\sum_{j=1}^{J} x_{mj}^2 \sum_{j=1}^{J} x_{nj}^2}} \tag{24.9}$$

where X_i is the document vector of image i with J dimension, and $X_i = x_{i1} x_{i2} \cdots x_{iJ}$. This retrieval method is language independent and very useful for documents with similar font and resolution, without the use of OCR.

Holistic Word Representation-Based Document Image Retrieval

Touching adjacent characters are difficult to be separated and present challenges to character coding methods. In view of this, some retrieval systems try to apply segmentation free methods to represent the entire word directly. In [12], the word image is expressed as $P =< p_1 p_2 \cdots p_n >=< (\sigma_1, \omega_1)(\sigma_2, \omega_2) \cdots (\sigma_n, \omega_n) >$, which has been introduced in Section "Holistic Word Representation." Due to the low resolution and poor quality, adjacent characters may be connected together, such that some features may be lost between two adjacent characters. Furthermore, noise can add or substitute features in the word image. An inexact feature matching is proposed to address such problems. $V(i, j)$ is the similarity between two prefixes $[a_1, a_2 \cdots a_i]$ and $[b_1, b_2 \cdots b_j]$, so that $V(n, m)$ is defined as the similarity between two primitive strings A with length n and B with length m. Using dynamic programming, $V(i, j)$ is formulated as

$$V(i, j) = \max \begin{cases} 0 \\ V(i-1, j-1) + \epsilon(a_i, b_j) \\ V(i-1, j) + \mu(a_i, -) \\ V(i, j-1) + \nu(-, b_j) \end{cases} \tag{24.10}$$

where $1 \le i \le n, 1 \le j \le m$, and $\forall i, j, V(i, 0) = V(0, j) = 0$. Matching a primitive with a spacing primitive "–" is defined as $\mu(a_k, -) = \nu(-, b_k) = -1$ for $a_k \ne -, b_k \ne -$, and matching a primitive with a spacing primitive "&" is defined as $\mu(a_k, \&) = \nu(\&, b_k) = -1$ for $a_k \ne \&, b_k \ne \&$. The score of matching two primitives is defined as

$$\epsilon(a_i, b_j) = \epsilon((\sigma_i^a, \omega_i^a), (\sigma_j^b, \omega_j^b)) = \varepsilon_1(\sigma_i^a, \sigma_j^b) + \varepsilon_2(\omega_i^a, \omega_j^b) \tag{24.11}$$

where $\epsilon(\&, \&) = 2$, and if the two elements are the same as ε_1 or ε_2, its value is 1, otherwise, -1. Finally, the matching score of two images is formulated as

$$S_1 = \max_{\forall i,j} V(i, j)/V_A^*(n, n) \tag{24.12}$$

where $V_A^*(n, n)$ is the score of matching A with itself. This feature matching algorithm has the ability to match partial word. Given a document image, word objects are first extracted. If the ratio w/h, where w, h is the width and height of the word object, is smaller than a threshold λ, the word image is denoted as a stop word which provides little contribution to the similarity between document images and discarded. All the remaining word images are represented by the feature sequence of primitives. Based on the feature matching method introduced above, words in one document are grouped into different classes, in each of which the similarity score between any two word images is greater than a predefined threshold, δ. The frequencies of all classes are used as the document vector, normalized by dividing the total number of word images in the document image. Consequently, the similarity between document $\overline{Q}$ and $\overline{Y}$ is

$$S(\overline{Q}, \overline{Y}) = \frac{\sum_{k=1}^{K} q_k \times y_k}{\sqrt{\sum_{k=1}^{K} q_k^2 \sum_{k=1}^{K} y_k^2}} \tag{24.13}$$

where q_i and y_i are the document vector of $\overline{Q}$ and $\overline{Y}$ respectively, and K is the dimension of the document vector, which is equal to the number of different classes in the document. Because of the inexact feature matching, the method can tolerate noise and low resolution and overcome the difficulties caused by touching characters. Shown as the experiment results, the proposed information retrieval method is feasible, reliable, and efficient.

Handwritten Document Image Retrieval

Retrieval of handwritten document images remains a challenging problem in view of the poor recognition results from OCR, high variability in writing style, and low quality of many written documents such as historical manuscripts. Retrieval based on OCRed text is understandably very difficult though attempts have been made to clean up the OCR output. Other than that, retrievals are usually based on matching with word features in handwritten documents with query words which are converted using the same feature set. The query may be entered into the retrieval system by the user. It may also come in the form of printed or handwritten image as query. A recent approach for retrieval to historical documents makes use of the corresponding transcript for each document as a means for query.

Rath et al. [38] take advantages of the availability of transcribed pages for historical manuscript images, and the web-based retrieval system shown in Fig. 24.16.

Fig. 24.16 The web-based retrieval system (From [38])

Word images in documents are extracted and holistic shape features are used to represent these word images, in order to avoid character segmentation errors caused by degradation. In [38], each word image is represented in terms of a discrete feature vocabulary (denoted as H) as a kind of "image language." A representation scheme is proposed consisting of 52 feature terms per word image, out of a feature vocabulary of size 494. A learning model is trained to map any given word image to a word from an English vocabulary V with a probability. If the training data is T, containing a set of word images with their transcripts, and each word W_i, the ith word in T, is represented by (h_i, ω_i), where ω_i is the corresponding transcript from V, and h_i is represented by a set of features $(f_{i,1} \cdots f_{i,k})$ from H. So, W_i is denoted by $(\omega_i, f_{i,1} \cdots f_{i,k})$. The probability of the distribution over W_i is formulated:

$$P(W_i = \omega, f_1 \cdots f_k | I_i) = P(\omega | I_i) \prod_{j=1}^{k} P(f_j | I_i) \tag{24.14}$$

$$P(\omega, f_1 \cdots f_k) = E_i[P(W_i = \omega, f_1 \cdots f_k | I_i)] = \frac{1}{|T|} \sum_{i=1}^{|T|} P(\omega | I_i) \prod_{j=1}^{k} P(F_j | I_i) \tag{24.15}$$

with the assumption that elements in H and V are underlying multinomial probability distribution and observed values are i.i.d random samples. So that, based on the trained model, the probability of ω is the transcription of the given testing word image, with feature vector $(f_1 \cdots f_k)$, is

$$P(\omega | f_1 \cdots f_k) = \frac{P(\omega, f_1 \cdots f_k)}{\sum_{v \in V} P(v, f_1 \cdots f_k)} \tag{24.16}$$

Consequently, given a query $Q = q_1 \cdots q_m$ and a document page Pg, the probability of Pg containing the query is $P(Q|Pg) = \prod_{j=1}^{m} P(q_j|Pg)$, where $P(q_j|Pg) = \frac{1}{|Pf|} \sum_{o=1}^{|Pg|} P(q_j | f_{o,1} \cdots f_{o,k})$, $f_{o,1} \cdots f_{o,k}$ are the features for the word image at position o in Pg. This method for handwriting retrieval is called Probabilistic Annotation Model. Direct Retrieval is also applied by $P(f, Q) = \sum_{W \in T} P(W) P(f|W) P(Q|W)$, $f \in H$. These two models can both retrieve unlabeled images with a text query without recognition, because the recognition always provides poor results for historical manuscripts.

A modified Vector Model named Multiple-candidate Vector Model is used to index and retrieve degraded medical forms with very low recognition accuracy [39]. The performance of information retrieval may decrease because of the incorrect recognition of the query word. But if the n top candidates given by the word recognizer are used for a query, the performance can be improved obviously. The count of the occurrences of the query word in one document is very essential for document retrieval, and poor recognition cannot provide valid and reliable counts. However, the count calculated based on the n candidates is more reliable, due to the fact that the true transcript of the query word is very likely in these n candidates. So, given a word recognizer WR, the output of which is denoted as O, and a collection of documents I, the frequency of a term t_i in a document d_j, is formulated as

$$tf_{i,j}^{\text{ocr}} = E\{tf_{i,j}|O\} = E\left\{ \frac{\text{freq}_{i,j}}{\sum_l \text{freq}_{l,j}} | O \right\} = \frac{E\{\text{freq}_{i.j}|O\}}{\sum_l E\{\text{freq}_{l,j}|O\}} \tag{24.17}$$

$$E\{\text{freq}_{i,j}|O\} = \sum_{\omega \in \omega(d_j)} P(\omega = t_i) \tag{24.18}$$

where $\text{freq}_{i,j}$ is the raw frequency of t_i in d_j, l is any term in d_j, and $\omega(d_j)$ is the set of words in d_j. For a word image η of word ω, WR gives a rank list of η, $(\omega_1, \omega_2 \cdots \omega_n)$, so the rank of ω_i is i, and the probability of ω equal to ω_i is denoted

by $P^{(i)}$. When considering the IDF of a term, documents which contain the term with $P^{(i)}$ smaller than a threshold $c_t \times P^{(1)}$ are discarded, so that

$$idf_i^{\text{ocr}} = \log \frac{|\{d_j\}|}{|\{d_j | \exists \eta \in d_j s.t. P^{(\text{rank}_\eta^{(t_i)})} \geq c_t \times P^{(1)}\}|} \tag{24.19}$$

Based on the modified definition of tf and idf values, the modified vector model can retrieve and tolerate OCR errors of degraded medical forms very well.

Another different approach can be found in the work of [5] which aims to deal with imperfect recognition results of OCR due to illegible writing such as those in medical documents which often pose great challenges for retrieval tasks [40]. Figure 24.17 shows the flowchart of the search engine in [5]. Due to large variation in writing styles, low image quality, and large amount of medical words, the authors present three methodologies to solve the retrieval problem based on poor OCR recognition. The first one attempts to correct errors in OCR'ed text by a strategy based on lexicon reduction [41]. All the forms are separated into different categories according to their topics and then generate a lexicon of words which are used to extract phrases from the original recognized text by OCR. The extracted phrases are then used to obtain topic or category distribution for every document in the test set. For later recognition, the reduced lexicon and the corresponding document categories are used. In [42], the information about the document topic is also applied to compute the posterior probability of every term appearing in the document. The second method is based on modified IR model to tolerate the uncorrected OCR'ed text. Instead of computing the frequency of the terms directly from the document in the classic vector model, the modified model calculates the frequency based on the estimation of term count, word segmentation probability, and word recognition likelihood. After noise removal, normalization, and segmentation in preprocessing step, the modified vector model is used for the search engine. The query provided by the user is converted into the same vector format as input into the model and the test documents are ranked based on similarity with the input vector. The third method does not depend on the OCR'ed text, and instead deals with the retrieval tasks by image features, named keyword spotting, locating all the occurrences of the query word. Keyword spotting-based retrieval can be achieved by probabilistic word spotting model or feature-based word spotting model.

Summary of Document Image Retrieval Methods

The document image retrieval methods covered in this section are summarized in the following two tables with comparison of their word representations, means of documents retrieval, and their pros and cons. Table 24.5 summarizes the methods for printed document image retrieval. Table 24.6 summarizes those for handwritten documents.

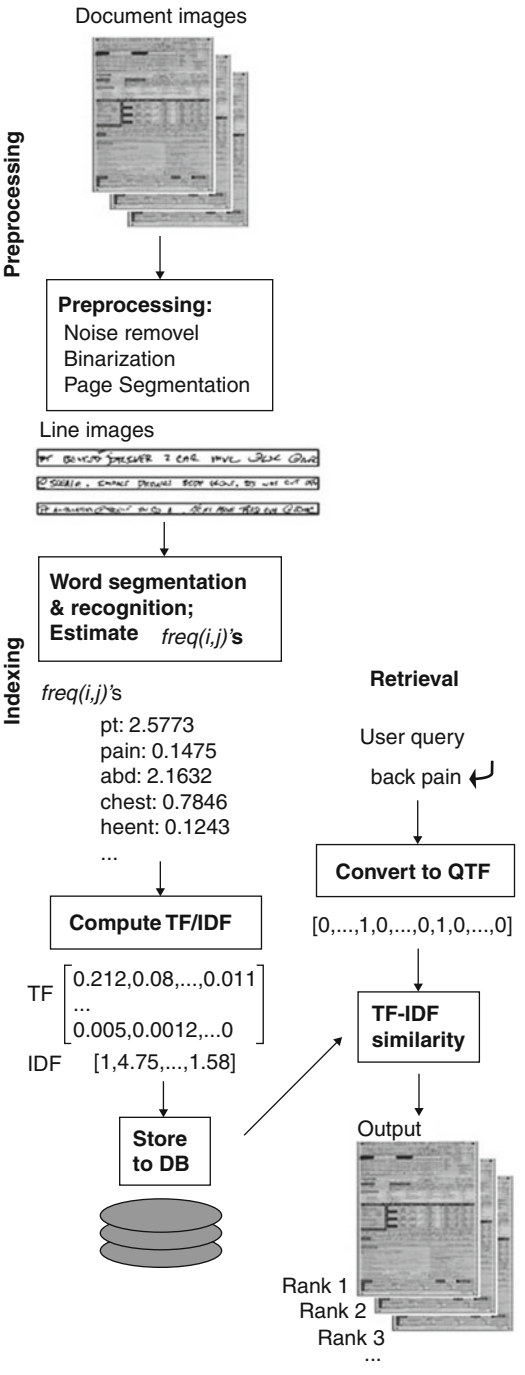


Fig. 24.17 The flowchart of the proposed search engine (From [5])

Table 24.5 Coding methods for printed document image retrieval

Coding method	Word representation	Document retrieval	Pros and cons
Symbol strings [3]	Each word is represented as a symbol string, where each symbol represents the shape of a component character	Documents are represented and indexed by a set of sequences of symbol strings for all their occurring words	This method reduces the size of symbol set, but at the expense of ambiguity due to symbol collision. Document degradation also leads for wrong symbol representation
VTD and HTD features [37]	Each character is represented by two vectors VTD and HTD through vertical and horizontal scanning	Documents are segmented into connected components to identify characters. Each document is then represented as a feature vector constructed from the occurrences of character classes based on VTD and HTD. Similarity between two document images based on feature vectors forms the basis for retrieval	This method is applicable to multilingual, different fonts documents and robust to degraded, low resolution documents
Holistic word coding [12]	Each word is treated as a whole object and represented by a sequence of features extracted directly from the whole word image	Word objects extracted from documents are segmented into primitives from left to right to form primitive strings. Documents are then represented as vectors based on classes of primitive strings of word objects to facilitate retrieval by means of similarity between document vectors	This method treats the word image as a whole and avoids character segmentation problems due to touching of characters. An inexact feature matching scheme is proposed to allow partial word matching

Conclusion

Keeping documents as image format is a more economical and flexible alternative than converting imaged documents into text format by OCR. Furthermore, it is more robust for different variations and degradation. OCR always suffers from poor quality, low resolution, large distortion, nonuniform lighting, complicated layout, and deformations of imaged documents, such as historical documents with much noise and missing strokes, camera-based documents with larger distortions, whiteboard notes with complicated layout and signatures which have very large variations and cannot even be recognized by a normal recognizer. In recent decades, imaged document processing methods are proposed to deal with specific tasks, such

Table 24.6 Handwritten document image retrieval methods

Method	Word representation	Document retrieval	Pros and cons
Probabilistic annotation model [38]	Words are represented by shape features and Fourier coefficients of profile features. To achieve cross-language task, the range of values of each feature is divided into several bins	Given transcripts and feature vectors of all words in the training set, a probabilistic model for retrieval can be trained to estimate the probability of a word transcript and a sequence of feature vector occurring together	This method is text-based retrieval method and can achieve multiple-words query tasks. But it suffers from queries which do not appear in the training set
Statistical classifier [39]	Each word is represented by a set of scores, each associated with a possible text transcript	The query word and all words in documents are converted into N-best lists. Document relevance can be obtained by measuring the metric scores between N-best lists	This method is text-based retrieval and is not dependent on word redundancy to overcome transcription errors. It can also query multiple words. However, transcriptions for the query word and documents must be available
Adapted vector model [5]	Documents and queries are represented by a vector space of term frequency (TF) and inverse document frequency (IDF) for each term in the vocabulary. TFs and IDFs are estimated and refined by use of word segmentation and recognition likelihood	Retrieval is achieved by measuring the similarity between vectors of query and data documents with a rank list	This method can overcome poor results of OCR and achieve queries on both printed or handwritten documents. But, it is still susceptible to OCR errors due to large variations in writing style or low image quality

as identifying different writers based on writing style, which cannot be achieved by OCR, and others try to solve general problems, i.e., keyword spotting and document retrieval, for different languages.

For research in keyword spotting in degraded image collections, holistic word representation is a popular approach. The key point of holistic word representation technique is to find good representation and distance measure that are sensitive to difference among word classes while tolerant to the variation within a word class. This is particularly so for handwritten document retrieval. From the literature, DTW-based comparison is appealing in terms of resilience to variances in handwritten words. HMM and ANN with appropriate training show even more powerful generalization ability to greater variances. Another issue in keyword spotting in

document images is the grammatical variation of the query word. Using multiple queries may be a solution, while interaction with the user for variant forms is another.

Current State of the Art with Applications

With the development of computer technology, huge quantities of imaged documents whose text format versions are not available are stored in digital libraries for public access and permanent preservation. Document image retrieval technique is widely used in digital libraries [43]. For retrieval purpose, three steps are needed: document storage (or indexing), query formulation, and similarity computation with subsequent ranking of the indexed documents with respect to the query. There are two kinds of retrieval techniques available, recognition based and retrieval without recognition, such as keyword spotting. The former approach is based on the conversion into electronic text with the advantages of easy integration into a standard information retrieval framework and lower computational cost of similarity computation and results ranking. But this recognition-based method cannot deal with documents with high-level noise, multilingual content, nonstandard font, or a free-style variable layout. So, the latter kind of methods, which is recognition free, has been proposed to address the above problems. Because of the large scale of document categories, a variety of applications are possible. Based on recognition, it is possible to deal with retrieval from OCR'ed documents, citation analysis, handwriting recognition, layout analysis, and born-digital documents. On the other hand, without recognition, keyword spotting for image retrieval, recognition and retrieval of graphical items, dealing with handwritten documents, and documenting image retrieval based on layout analysis have led to some dedicated uses. Furthermore, even though recognition-free methods are promising, how to integrate them into a standard digital library which mostly adopts to OCR-based techniques is another important issue because of the difficulty in using sophisticated recognition-free matching algorithms [44]. One possible approach is to consider combining recognition-based and recognition-free methods together to improve the performance. Imaged document retrieval, whether recognition-based or recognition-free, is an answer to the current large scale document imaging projects such as the Google Book Search [8] and the million-page complex document image tested [45].

Many forms or document files, such as business contracts and legal agreements, contain signatures. Serving as individual identification and document authentication, signatures present convincing evidence and provide an important form of indexing for effective document image retrieval in a broad range of applications [46]. The authors address two fundamental problems for automatic document image search and retrieval using signatures. The first one is detection and segmentation including effectively segmenting the signatures from the background and choosing an appropriate meaningful presentations of signatures for later analysis. Detecting and segmenting signatures which present themselves as free-form objects is a

challenging task [47]. The authors use 2D contour fragments for each detected signature motivated by structural variations of off-line handwriting signatures. This kind of 2D point features provide several advantages, because the preserved topology and temporal order under structural variations or clustered background are not needed compared to other compact geometrical entities used in shape representation. The second problem pertains to the use of matching for determining whether two given objects belong to the same class. Proper representations, matching algorithm, and similarity measure method should be investigated so that retrieval results can be invariant to large intra-class variability and robust under interclass similarity. The authors use two kinds of nonrigid shape matching algorithms: one is based on shape context representation [48], and the other formulates shape matching as an optimization problem that preserves local neighborhood structure [49]. Like the problem that [46] aims to solve, with large repository of business documents, retrieving documents signed by the person provided by the user, in other words, using a query signature image to retrieval from databases, is an interesting task [50]. In a similar vein, indexing and retrieval of Logo [51] or architectural symbols [52] are also state of the art applications of document image retrieval technology.

A new emerging application, though not really on paper-based media, has found practical use of document image retrieval techniques. This involves image capture of whiteboard writing to serve as records of meetings and notes taking for classroom lessons. However, whiteboard writing recognition presents far greater challenges than paper-based handwriting recognition. Writing on the whiteboard tends to be highly disorganized and haphazard. The writing style is free and extremely variable due to the writing posture while standing and the use of different types of marker pens. The writing contains not only text but also a host of different kinds of drawing comprising lines, shapes, and symbols. This requires research into separating different parts of the whiteboard contents into individual functional units, such as texts and graphs, without any prior knowledge of text lines, fonts, character sizes, etc. [53] recognizes handwritten sketches and [54] deals with mind maps [55]. Liwicki and Bunke [56] propose a method for online whiteboard note processing using an off-line HMM-based recognizer. To do so, it uses a special pen-tracking hardware, known as eBeam system which is restricted and suffers from obvious flaw of preventing natural interaction. The data is collected online to get idealized text lines. In the preprocessing step, the authors apply online and off-line preprocessing operations to solve two problems of the recorded online data: noisy points and gaps within strokes. And then, online data is transformed into off-line format as input for the basic off-line recognizer which is an HMM-based cursive handwriting recognizer described in [57]. Most whiteboard image processing works rely simply on camera capture to provide only off-line data. For instance, [58] works on camera-based whiteboard images under real-world conditions obtaining only low quality images with distortion. In this work, text lines are detected and ink contrast is enhanced. Using a sliding window, the normalized extracted text lines are subdivided into a sequence of overlapping stripes which will be presented as semicontinuous HMMs. The HMM-based writing model is then integrated with a

bigram language model to improve the performance. In another application, [59] uses two cameras and a pen capture tool on the whiteboard to recognize Japanese characters based on some character matching.

While whiteboard images have been mostly captured with video or digital cameras mounted in front or above the whiteboard, a new generation of electronic whiteboards (also known as interactive whiteboard) has made it possible to directly capture whiteboard contents using built-in imaging device. This facilitates instant generation of whiteboard images which are free of occlusion by the writer and with considerably better image quality. Thus electronic whiteboard opens exciting applications of document image retrieval to index and recall past imaged records and notes of meetings.

Future Outlook

Looking ahead, efficiently and correctly storing, indexing and retrieving a large amount of imaged documents is an imperative task. This is especially so considering Google and Yahoo are working on making millions of imaged documents accessible through the internet with their search engines. In such situations, rapid and satisfactory responses are quite important. So, large collection recognition with high computation speed, recall, and precision rate is an important issue. And, querying on multiple words, even very long words, effectively should also be researched into. Until now, many methods have been tried to solve recognition problems for languages such as English and Chinese with reasonably good performance, but there are many other languages in which a number of valuable documents are written and which are not yet widely explored. While other language character or word features should be studied, language and script independent models should be researched into also. Such models will be needed for multilingual documents involving different languages or scripts on one single document page.

Today, the pervasive use of digital cameras and mobile phone cameras in the society has led to the creation of many more document images or scene images with text captured under different conditions. While image resolution is often not an issue with advances in camera technology, many such images suffer from unfavorable effects for imaged document retrieval and keyword spotting. These include uneven lighting, curved, or warped document surfaces such as those taken from open pages of thick books, perspective distortion, and complex background in scene text capture. Archiving and storing of such document images on public web sites or other repositories entail similar tasks like indexing, text content-based retrieval, and searching for query words. Exciting challenges are lying ahead for which new paradigms and models will continue to emerge in the years to come.

Cross-References

▸Continuous Handwritten Script Recognition
▸Handprinted Character and Word Recognition

▸Language, Script, and Font Recognition
▸Machine-Printed Character Recognition
▸Page Similarity and Classification
▸Text Segmentation for Document Recognition

References

1. Galloway EA, Gabrielle VM (1998) The heinz electronic library interactive on-line system: an update. Public-Access Comput Syst Rev 9(1):1–12
2. Taghva K, Borsack J, Condit A, Erva S (1994) The effects of noisy data on text retrieval. J Am Soc Inf Sci 45(1):50–58
3. Spitz AL (1995) Using character shape codes for word spotting in document images. In: Dori D, Bruckstein A (eds) Shape, structure and pattern recognition. World Scientific, Singapore, pp 382–389
4. Marinai S, Marino E, Soda G (2006) Font adaptive word indexing of modern printed documents. IEEE Trans Pattern Anal Mach Intell 28(8):1187–1199
5. Cao H, Govindaraju V, Bhardwaj A (2011) Unconstrained handwritten document retrieval. Int J Doc Anal Recognit 14:145–157
6. Breuel TM (2005) The future of document imaging in the era of electronic documents. In: Proceedings of the international workshop on document analysis, IWDA'05, Kolkata. Allied Publishers, pp 275–296
7. Sellen AJ, Harper RHR (2003) The myth of the paperless office. MIT, Cambridge/London
8. Vincent L (2007) Google book search: document understanding on a massive scale. In: Proceedings of the international conference on document analysis and recognition, Curitiba, vol 2. IEEE, pp 819–823
9. Zhang L, Tan CL (2005) A word image coding technique and its applications in information retrieval from imaged documents. In: Proceedings of the international workshop on document analysis, IWDA'05, Kolkata. Allied Publishers, pp 69–92
10. Lu S, Li L, Tan CL (2008) Document image retrieval through word shape coding. IEEE Trans Pattern Anal Mach Intell 130(11):1913–1918
11. Hull JJ (1986) Hypothesis generation in a computational model for visual word recognition. IEEE Expert 1(3):63–70
12. Lu Y, Tan CL (2004) Information retrieval in document image databases. IEEE Trans Knowl Data Eng 16(11):1398–1410
13. Levy S (2004) Google's two revolutions. Newsweek, December 27:2004
14. Tomai CI, Zhang B, Govindaraju V (2002) Transcript mapping for historic handwritten document images. In: Proceedings of the eighth international workshop on frontiers in handwriting recognition, 2002, Niagara-on-the-Lake. IEEE, pp 413–418
15. Antonacopoulos A, Downton AC (2007) Special issue on the analysis of historical documents. Int J Doc Anal Recognit 9(2):75–77
16. Indermuhle E, Bunke H, Shafait F, Breuel T (2010) Text versus non-text distinction in online handwritten documents. In: Proceedings of the 2010 ACM symposium on applied computing, Sierre. ACM, pp 3–7
17. Liwicki M, Indermuhle E, Bunke H (2007) On-line handwritten text line detection using dynamic programming. In: Ninth international conference on document analysis and recognition, ICDAR 2007, Curitiba, vol 1. IEEE, pp 447–451
18. Zimmermann M, Bunke H (2002) Automatic segmentation of the iam off-line handwritten english text database. In: 16th international conference on pattern recognition, Quebec, vol 4, pp 35–39
19. Simard PY, Steinkraus D, Agrawala M (2005) Ink normalization and beautification. In: Proceedings of the eighth international conference on document analysis and recognition 2005, Seoul. IEEE, pp 1182–1187

20. Vinciarelli A, Luettin J (2001) A new normalization technique for cursive handwritten words. Pattern Recognit Lett 22(9):1043–1050
21. Uchida S, Taira E, Sakoe H (2001) Nonuniform slant correction using dynamic programming. In: Proceedings of the sixth international conference on document analysis and recognition, 2001, Seattle. IEEE, pp 434–438
22. Manmatha R, Han C, EM Riseman, Croft WB (1996) Indexing handwriting using word matching. In: Proceedings of the first ACM international conference on digital libraries, Bethesda. ACM, pp 151–159
23. Likforman-Sulem L, Zahour A, Taconet B (2007) Text line segmentation of historical documents: a survey. Int J Doc Anal Recognit 9(2):123–138
24. Adamek T, O'Connor NE, Smeaton AF (2007) Word matching using single closed contours for indexing handwritten historical documents. Int J Doc Anal Recognit 9(2):153–165
25. Ho TK, Hull JJ, Srihari SN (1992) A word shape analysis approach to lexicon based word recognition. Pattern Recognit Lett 13(11):821–826
26. Leydier Y, Lebourgeois F, Emptoz H (2007) Text search for medieval manuscript images. Pattern Recognit 40(12):3552–3567
27. Leydier Y, Ouji A, Lebourgeois F, Emptoz H (2009) Towards an omnilingual word retrieval system for ancient manuscripts. Pattern Recognit 42(9):2089–2105
28. Madhvanath S, Govindaraju V (2001) The role of holistic paradigms in handwritten word recognition. IEEE Trans Pattern Anal Mach Intell 23(2):149–164
29. Fischer A, Keller A, Frinken V, Bunke H (2010) Hmm-based word spotting in handwritten documents using subword models. In: 2010 international conference on pattern recognition, Istanbul. IEEE, pp 3416–3419
30. Myers CS, Habiner LR (1981) A comparative study of several dynamic time-warping algorithms for connected-word. Bell Syst Tech J 60(7):1389–1409
31. Rodríguez-Serrano JA, Perronnin F (2009) Handwritten word-spotting using hidden markov models and universal vocabularies. Pattern Recognit 42(9):2106–2116
32. Frinken V, Fischer A, Bunke H (2010) A novel word spotting algorithm using bidirectional long short-term memory neural networks. In: Schwenker F, El Gayar N (eds) Artificial neural networks in pattern recognition. Springer, Berlin/Heidelberg, pp 185–196
33. Graves A, Liwicki M, Fernández S, Bertolami R, Bunke H, Schmidhuber J (2009) A novel connectionist system for unconstrained handwriting recognition. IEEE Trans Pattern Anal Mach Intell 31:855–868
34. Salton G, Wong A, Yang CS (1975) A vector space model for automatic indexing. Commun ACM 18(11):613–620
35. Robertson SE, Sparck Jones K (1976) Relevance weighting of search terms. J Am Soc Inf Sci 27(3):129–146
36. Lan M, Tan CL, Low HB (2006) Proposing a new term weighting scheme for text categorization. In: Proceedings of the 21st national conference on artificial intelligence, Boston
37. Tan CL, Huang W, Yu Z, Xu Y (2002) Imaged document text retrieval without OCR. IEEE Trans Pattern Anal Mach Intell 24:838–844
38. Rath TM, Manmatha R, Lavrenko V (2004) A search engine for historical manuscript images. In: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval, Sheffield. ACM, pp 369–376
39. Cao H, Farooq F, Govindaraju V (2007) Indexing and retrieval of degraded handwritten medical forms. In: Proceedings of the workshop on multimodal information retrieval at IJCAI-2007, Hyderabad
40. Cao H, Bhardwaj A, Govindaraju V (2009) A probabilistic method for keyword retrieval in handwritten document images. Pattern Recognit 42(12):3374–3382
41. Milewski RJ, Govindaraju V, Bhardwaj A (2009) Automatic recognition of handwritten medical forms for search engines. Int J Doc Anal Recognit 11(4):203–218
42. Bhardwaj A, Farooq F, Cao H, Govindaraju V (2008) Topic based language models for ocr correction. In: Proceedings of the second workshop on analytics for noisy unstructured text data, Singapore. ACM, pp 107–112

43. Marinai S (2006) A survey of document image retrieval in digital libraries. In: 9th colloque international Francophone Sur l'Ecrit et le document (CIFED), Fribourg, pp 193–198.
44. Aschenbrenner S (2005) Jstor: adapting lucene for new search engine and interface. DLib Mag Vol. 11, no. 6
45. Agam G, Argamon S, Frieder O, Grossman D, Lewis D (2007) Content-based document image retrieval in complex document collections. In: Proceedings of the SPIE, vol 6500. Document Recognition & Retrieval XIV, San Jose.
46. Zhu G, Zheng Y, Doermann D (2008) Signature-based document image retrieval. In: Computer vision–ECCV 2008, Marseille, pp 752–765
47. Zhu G, Zheng Y, Doermann D, Jaeger S (2007) Multi-scale structural saliency for signature detection. In: 2007 IEEE conference on computer vision and pattern recognition, Minneapolis. IEEE, pp 1–8
48. Belongie S, Malik J, Puzicha J (2002) Shape matching and object recognition using shape contexts. IEEE Trans Pattern Anal Mach Intell 24:509–522
49. Zheng Y, Doermann D (2006) Robust point matching for nonrigid shapes by preserving local neighborhood structures. IEEE Trans Pattern Anal Mach Intell 28:643–649
50. Srihari SN, Shetty S, Chen S, Srinivasan H, Huang C, Agam G, Frieder O (2006) Document image retrieval using signatures as queries. In: Second international conference on document image analysis for libraries 2006, DIAL'06, Lyon. IEEE, p 6
51. Jain AK, Vailaya A (1998) Shape-based retrieval: a case study with trademark image databases. Pattern Recognit 31(9):1369–1390
52. Terrades OR, Valveny E (2003) Radon transform for lineal symbol representation. Doc Anal Recognit 1:195
53. Weber M, Liwicki M, Dengel A (2010) a.Scatch-a sketch-based retrieval for architectural floor plans. In: 2010 12th international conference on frontiers in handwriting recognition, Kolkata. IEEE, pp 289–294
54. Vajda S, Plotz T, Fink GA (2009) Layout analysis for camera-based whiteboard notes. J Univers Comput Sci 15(18):3307–3324
55. Burzan T, Burzan B (2003) The mind map book. BBC Worldwide, London
56. Liwicki M, Bunke H (2005) Handwriting recognition of whiteboard notes. In: Proceedings of the 12th conference of the international graphonomics society, Salerno, pp 118–122.
57. Marti UV, Bunke H (2001) Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system. IJPRAI 15(1):65–90
58. Plotz T, Thurau C, Fink GA (2008) Camera-based whiteboard reading: new approaches to a challenging task. In: Proceedings of the 11th international conference on frontiers in handwriting recognition, Montreal, pp 385–390
59. Yoshida D, Tsuruoka S, Kawanaka H, Shinogi T (2006) Keywords recognition of handwritten character string on whiteboard using word dictionary for e-learning, International Conference on Hybrid Information Technology, Cheju Island, Vol. 1, pp 140–145
60. Konidaris T, Gatos B, Ntzios K, Pratikakis I, Theodoridis S (2007) Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. Int J Doc Anal Recognit 9(2):167–177
61. Lu Y, Tan CL (2004) Chinese word searching in imaged documents. Int J Pattern Recognit Artif Intell 18(2):229–246
62. Zhang H, Wang DH, Liu CL (2010) Keyword spotting from online chinese handwritten documents using one-vs-all trained character classifier. In: 2010 12th international conference on frontiers in handwriting recognition, Kolkata. IEEE, pp 271–276
63. Senda S, Minoh M, Ikeda K (1993) Document image retrieval system using character candidates generated by character recognition process. In: Proceedings of the second international conference on document analysis and recognition, 1993, Tsukuba. IEEE, pp 541–546
64. Sagheer MW, Nobile N, He CL, Suen CY (2010) A novel handwritten Urdu word spotting based on connected components analysis. In: 2010 international conference on pattern recognition, Istanbul. IEEE, pp 2013–2016

65. Moghaddam RF, Cheriet M (2009) Application of multi-level classifiers and clustering for automatic word spotting in historical document images. In: 2009 10th international conference on document analysis and recognition, Barcelona. IEEE, pp 511–515
66. Leydier Y, Le Bourgeois F, Emptoz H (2005) Omnilingual segmentation-free word spotting for ancient manuscripts indexation. In: Proceedings of the eighth international conference on document analysis and recognition, 2005, Seoul. IEEE, pp 533–537
67. Mitra M, Chaudhuri BB (2000) Information retrieval from documents: a survey. Inf Retr 2(2):141–163
68. Murugappan A, Ramachandran B, Dhavachelvan P (2011) A survey of keyword spotting techniques for printed document images. Artif Intell Rev 1–18
69. 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. Springer, Berlin/Heidelberg, pp 181–204

Further Reading

Methods that have been introduced are mainly proposed and experimented on English documents. Other languages or scripts, such as Chinese and Arabic, have quite different patterns and writing styles from English and have special characteristics which cannot be retrieved by methods for English. For example, some features used to differentiate different characters or words in English documents may not be applicable for Chinese words. Readers who are interested in non-English documents may refer to [60–64] which deal with retrieval of words or documents in languages such as Arabic, Chinese, Japanese, and Urdu. Besides, [65, 66] report language independent methods. One method is highlighted below for multilingual documents retrieval as an illustration.

The Character Shape Coding method proposed by Lu et al. [10] can be used for different languages, namely, multilingual tasks. There are five codes: hole, ascender, descender, leftward water reservoir, and rightward water-reservoir. A word is represented by encoding these features from left to right, top to bottom. The same coding scheme is applied in document similarity measurement task for five languages: English, French, German, Italian, and Spanish.

Imaged-based retrieval and keyword spotting in documents have been researched for many years with many more works not possibly covered in this chapter. Readers may wish to consult the following three survey papers for comparative views of different methods. Among these papers, an earlier one [67] covers a broad area of document-based information retrieval which gives some of the historical perspectives of this field, including retrieval of OCR'ed text. The other two are much more recent surveys. One [68] presents different keyword spotting techniques for printed documents, while the other [69] provides a comparative study of document image retrieval in the context of digital libraries involving printed and handwritten documents.