Correspondence

An Overview of Character Recognition Focused on Off-Line Handwriting

Nafiz Arica and Fatos T. Yarman-Vural

***Abstract*—Character recognition (CR) has been extensively studied in the last half century and progressed to a level sufficient to produce technology driven applications. Now, the rapidly growing computational power enables the implementation of the present CR methodologies and creates an increasing demand on many emerging application domains, which require more advanced methodologies.**

This material serves as a guide and update for readers working in the CR area. First, the historical evolution of CR systems is presented. Then, the available CR techniques with their superiorities and weaknesses are reviewed. Finally, the current status of CR is discussed, and directions for future research are suggested. Special attention is given to the off-line handwriting recognition since this area requires more research to reach the ultimate goal of machine simulation of human reading.

***Index Terms*—Character recognition (CR), feature extraction, off-line handwriting recognition, segmentation, training and recognition.**

I. Introduction

Machine simulation of human functions has been a very challenging research field since the advent of digital computers. In some areas, which require certain amount of intelligence, such as number crunching or chess playing, tremendous improvements are achieved. On the other hand, humans still outperform even the most powerful computers in relatively routine functions such as vision. Machine simulation of human reading is one of these areas, which has been the subject of intensive research for the last three decades, yet it is still far from the final frontier.

In this overview, character recognition (CR) is used as an umbrella term, which covers all types of machine recognition of characters in various application domains. The overview serves as an update for the state-of-the-art in the CR field, emphasizing the methodologies required for the increasing needs in newly emerging areas, such as development of electronic libraries, multimedia databases, and systems which require handwriting data entry. The study investigates the direction of the CR research, analyzing the limitations of methodologies for the systems, which can be classified based upon two major criteria: 1) the data acquisition process (on-line or off-line) and 2) the text type (machine-printed or handwritten). No matter in which class the problem belongs, in general, there are five major stages in the CR problem:

1) preprocessing;
2) segmentation;
3) representation;
4) training and recognition;
5) post processing.

The paper is arranged to review the CR methodologies with respect to the stages of the CR systems, rather than surveying the complete solutions. Although the off-line and on-line CR techniques have different approaches, they share a lot of common problems and solutions. Since it is relatively more complex and requires more research compared to on-line and machine-printed recognition, off-line handwritten CR is selected as a focus of attention in this article. However, the article also reviews some of the methodologies for on-line CR, as it intersects with the off-line case.

After giving a historical review of the developments in Section II, the methodologies of CR systems are reviewed in Section III. Finally, future research directions are discussed in Section IV. Since it is practically impossible to cite hundreds of independent studies conveyed in the field of CR, we suffice to provide only selective references and avoid an exhaustive list of studies, which can be reached from the references given at the end of this overview. The comprehensive survey on off-line and on-line handwriting recognition in [141], the survey in [162] dedicated to off-line cursive script recognition, and the book in [124] which covers the optical CR methodologies can be taken as good starting points to reach the recent studies in various types and applications of the CR problem.

Manuscript received June 4, 1999; revised March 4, 2001. This paper was recommended by Associate Editor B. J. Oommen.

The authors are with the Computer Engineering Department, Middle East Technical University, Ankara, Turkey (e-mail: nafiz@ceng.metu.edu.tr; vural@ceng.metu.edu.tr).

Publisher Item Identifier S 1094-6977(01)06490-2.

II. History

Writing, which has been the most natural mode of collecting, storing, and transmitting information through the centuries, now serves not only for communication among humans but also serves for communication of humans and machines. The intensive research effort in the field of CR was not only because of its challenge on simulation of human reading but also because it provides efficient applications such as the automatic processing of bulk amount of papers, transferring data into machines, and web interface to paper documents. Historically, CR systems have evolved in three ages.

1900–1980 Early Ages: The history of CR can be traced as early as 1900, when the Russian scientist Tyuring attempted to develop an aid for the visually handicapped [114]. The first character recognizers appeared in the middle of the 1940s with the development of digital computers [42]. The early work on the automatic recognition of characters has been concentrated either upon machine-printed text or upon a small set of well-distinguished handwritten text or symbols. Machine-printed CR systems in this period generally used template matching in which an image is compared to a library of images. For handwritten text, low-level image processing techniques have been used on the binary image to extract feature vectors, which are then fed to statistical classifiers. Successful, but constrained algorithms have been implemented mostly for Latin characters and numerals. However, some studies on Japanese, Chinese, Hebrew, Indian, Cyrillic, Greek, and Arabic characters and numerals in both machine-printed and handwritten cases were also initiated [43], [125], [164].

The commercial character recognizers were available in the 1950s, when electronic tablets capturing the x–y coordinate data of pen-tip movement was first introduced. This innovation enabled the researchers to work on the on-line handwriting recognition problem. A good source of references for on-line recognition until 1980 can be found in [163].

1980–1990 Developments: Studies up until 1980 suffered from the lack of powerful computer hardware and data acquisition devices. With the explosion of information technology, the previously developed methodologies found a very fertile environment for rapid growth

1094–6977/01$10.00 © 2001 IEEE

in many application areas, as well as CR system development [18], [54], [170]. Structural approaches were initiated in many systems in addition to the statistical methods [13], [155]. The CR research was focused basically on the shape recognition techniques without using any semantic information. This led to an upper limit in the recognition rate, which was not sufficient in many practical applications. Historical review of CR research and development during this period can be found in [123] and [163] for off-line and on-line cases, respectively.

After 1990 Advancements: The real progress on CR systems is achieved during this period, using the new development tools and methodologies, which are empowered by the continuously growing information technologies.

In the early 1990s, image processing and pattern recognition techniques were efficiently combined with artificial intelligence (AI) methodologies. Researchers developed complex CR algorithms, which receive high-resolution input data and require extensive number crunching in the implementation phase. Nowadays, in addition to the more powerful computers and more accurate electronic equipments such as scanners, cameras, and electronic tablets, we have efficient, modern use of methodologies such as neural networks (NNs), hidden Markov models (HMMs), fuzzy set reasoning, and natural language processing. The recent systems for the machine-printed off-line [9], [12] and limited vocabulary, user-dependent on-line handwritten characters [67], [116], [140] are quite satisfactory for restricted applications. However, there is still a long way to go in order to reach the ultimate goal of machine simulation of fluent human reading, especially for unconstrained on-line and off-line handwriting.

III. Methodologies of CR Systems

In this section, we focus on the methodologies of CR systems, emphasizing the off-line handwriting recognition problem. A bottom–up approach for most of the systems would be starting the process from the pixel level and ending up with a meaningful text. This approach varies a great deal, depending upon the type of CR system and the methodology used. The literature review in the field of CR indicates that these hierarchical tasks are grouped in the stages of the CR for preprocessing, segmentation, representation, training and recognition, and postprocessing. In some methods, some of the stages are merged or omitted; in others a feedback mechanism is used to update the output of each stage.

A. Preprocessing

The raw data, depending on the data acquisition type, is subjected to a number of preliminary processing steps to make it usable in the descriptive stages of character analysis. Preprocessing aims to produce data that are easy for the CR systems to operate accurately. The main objectives of preprocessing are

1) noise reduction;
2) normalization of the data;
3) compression in the amount of information to be retained.

In order to achieve the above objectives, the following techniques are used in the preprocessing stage.

1) Noise Reduction: The noise, introduced by the optical scanning device or the writing instrument, causes disconnected line segments, bumps and gaps in lines, filled loops, etc. The distortion, including local variations, rounding of corners, dilation, and erosion, is also a problem. Prior to the CR, it is necessary to eliminate these imperfections. Hundreds of available noise reduction techniques can be categorized in three major groups [153], [161].

a) Filtering: This aims to remove noise and diminish spurious points, usually introduced by uneven writing surface and/or poor sampling rate of the data acquisition device. Various spatial and frequency domain filters can be designed for this purpose. The basic idea is to convolute a predefined mask with the image to assign a value to a pixel as a function of the gray values of its neighboring pixels. Filters can be designed for smoothing [104], sharpening [105], thresholding [119], removing slightly textured or colored background [101], and contrast adjustment purposes [142].

b) Morphological Operations: The basic idea behind the morphological operations is to filter the document image replacing the convolution operation by the logical operations. Various morphological operations can be designed to connect the broken strokes [7], decompose the connected strokes [26], smooth the contours, prune the wild points, [153], thin the characters [147], and extract the boundaries [185]. Therefore, morphological operations can be successfully used to remove the noise on the document images due to low quality of paper and ink, as well as erratic hand movement.

c) Noise Modeling: Noise could be removed by some calibration techniques if a model for it were available. However, modeling the noise is not possible in most of the applications. There is very little work on modeling the noise introduced by optical distortion, such as speckle, skew, and blur [10], [92]. Nevertheless, it is possible to assess the quality of the documents and remove the noise to a certain degree, as suggested in [21].

2) Normalization: Normalization methods aim to remove the variations of the writing and obtain standardized data. The following are the basic methods for normalization [40], [55].

a) Skew Normalization and Baseline Extraction: Due to inaccuracies in the scanning process and writing style, the writing may be slightly tilted or curved within the image. This can hurt the effectiveness of later algorithms and, therefore, should be detected and corrected. Additionally, some characters are distinguished according to the relative position with respect to the baseline (e.g., "9" and "g"). Methods of baseline extraction include using the projection profile of the image [78], a form of nearest neighbors clustering [64], cross correlation method between lines [31], and using the Hough transform [189]. In [134], an attractive repulsive NN is used for extracting the baseline of complicated handwriting in heavy noise (see Fig. 1). After skew detection, the character or word is translated to the origin, rotated, or stretched until the baseline is horizontal and retranslated back into the display screen space.

b) Slant Normalization: One of the measurable factors of different handwriting styles is the slant angle between longest stroke in a word and the vertical direction. Slant normalization is used to normalize all characters to a standard form. The most common method for slant estimation is the calculation of the average angle of near-vertical elements (see Fig. 2). In [111], vertical line elements from contours are extracted by tracing chain code components using a pair of one-dimensional (1-D) filters. Coordinates of the start and end points of each line element provide the slant angle. Another study [56] uses an approach in which projection profiles are computed for a number of angles away from the vertical direction. The angle corresponding to the projection with the greatest positive derivative is used to detect the least amount of overlap between vertical strokes and, therefore, the dominant slant angle. In [18], slant detection is performed by dividing the image into vertical and horizontal windows. The slant is estimated based on the center of gravity of the upper and lower half of each window averaged over all the windows. Finally, in [90], a variant of the Hough transform is used by scanning left to right across the image and calculating projections in the direction of 21 different slants. The top three projections for any slant are added and the slant

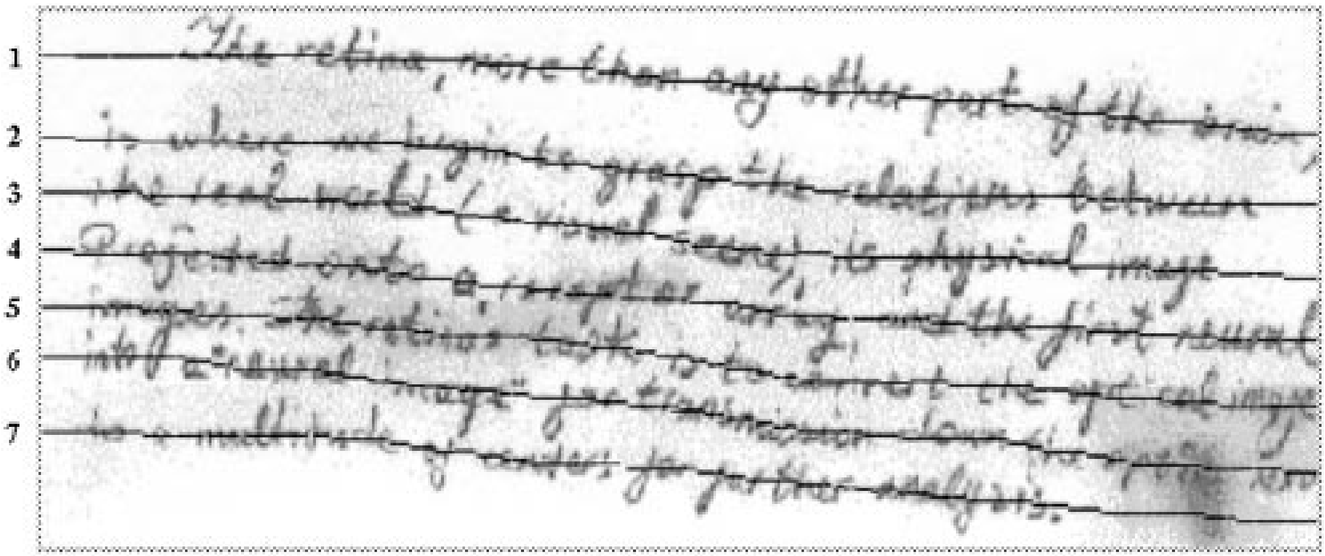
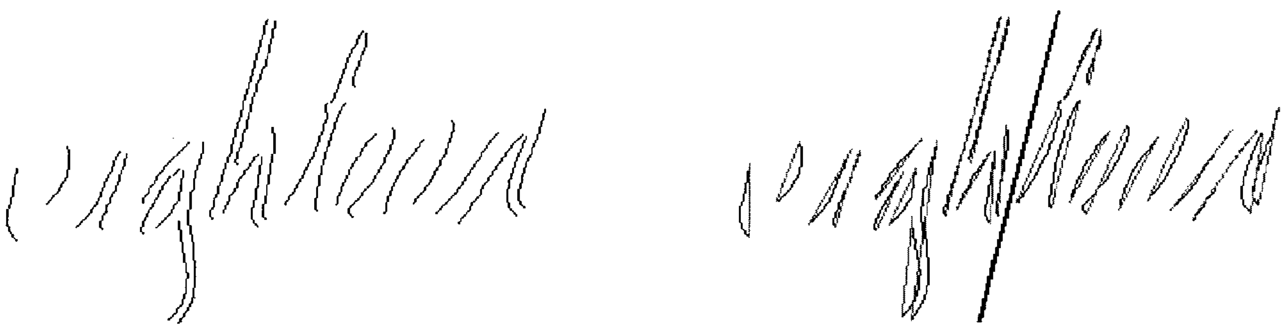


Fig. 1. Baseline extraction using attractive and repulsive network in [145].



(a) (b)

Fig. 2. Slant angle estimation. (a) Near vertical elements. (b) Average slant angle.

with the largest count is taken as the slant value. On the other hand, in some studies, recognition systems do not use slant correction and compensate it during training stage [6], [36].

c) Size Normalization: This is used to adjust the character size to a certain standard. Methods of CR may apply both horizontal and vertical size normalizations. In [186], the character is divided into number of zones and each of these zones is separately scaled. Size normalization can also be performed as a part of the training stage, and the size parameters are estimated separately for each particular training data [5]. In Fig. 3, two sample characters are gradually shrunk to the optimal size, which maximize the recognition rate in the training data. On the other hand, word recognition, due to the desire to preserve large intraclass differences in the length of words so they may assist in recognition, tends to only involve vertical height normalization or bases the horizontal size normalization on the scale factor calculated for the vertical normalization [90].

d) Contour Smoothing: It eliminates the errors due to the erratic hand motion during the writing. It generally reduces the number of sample points needed to represent the script, thus improving efficiency in remaining preprocessing steps [7], [104].

3) Compression: It is well known that classical image compression techniques transform the image from the space domain to domains, which are not suitable for recognition. Compression for CR requires space domain techniques for preserving the shape information. Two popular compression techniques are thresholding and thinning.

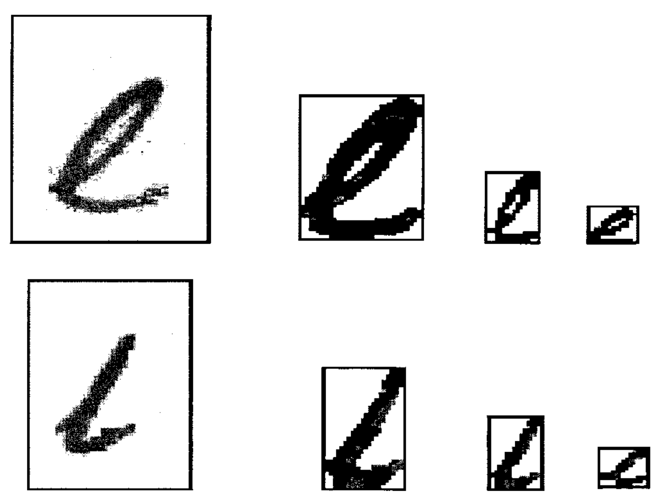


Fig. 3. Normalization of characters "e" and "l" in [6].

a) Thresholding: In order to reduce storage requirements and to increase processing speed, it is often desirable to represent gray-scale or color images as binary images by picking a threshold value. Two categories of thresholding exist: *global* and *local*. Global thresholding picks one threshold value for the entire document image which is often based on an estimation of the background level from the intensity histogram of the image [160]. Local (adaptive) thresholding use different values for each pixel according to the local area information [151]. In [171], a comparison of common global and local thresholding techniques is given by using an evaluation criterion that is goal-directed in the sense that the accuracies of a CR system using different techniques were compared. On those tested, it is shown that Niblack's locally adaptive method [128] produces the best result. Additionally, the recent study [184] develops an adaptive logical method by analyzing the clustering and connection characteristics of the characters in degraded document images.

b) Thinning: While it provides a tremendous reduction in data size, thinning extracts the shape information of the characters. Thinning can be considered as conversion of off-line handwriting to almost on-line like data, with spurious branches and artifacts. Two basic approaches for thinning are 1) *pixel wise* and 2) *nonpixel* wise thinning [97]. Pixel wise thinning methods locally and iteratively process the image until one pixel wide skeleton remains. They are very sensitive to noise and may deform the shape of the character. On the other hand, the nonpixel wise methods use some global information about the character during the thinning. They produce a certain median or centerline of the pattern directly without examining all the individual pixels [11]. In [113], clustering-based thinning method defines the skeleton of character as the cluster centers. Some thinning algorithms identify the singular points of the characters, such as end points, cross points, and loops [187]. These points are the source of problems. In a nonpixel wise thinning, they are handled with global approaches. A survey of pixel wise and nonpixel wise thinning approaches is available in [97].

The iterations for thinning can be performed either in sequential or parallel algorithms. Sequential algorithms examine the contour points by raster scan [4] or contour following [53]. Parallel algorithms are superior to sequential ones since they examine all the pixels simultaneously, using the same set of conditions for deletion [59]. They can be efficiently implemented in parallel hardware. An evaluation of parallel thinning algorithms for CR can be found in [96].

The preprocessing techniques are well explored and applied in many areas of image processing besides CR [24], [161]. Note that the above techniques affect the data and may introduce unexpected distortions to the document image. As a result, these techniques may cause the loss of important information about writing. They should be applied with care.

B. Segmentation

The preprocessing stage yields a "clean" document in the sense that a sufficient amount of shape information, high compression, and low noise on a normalized image is obtained. The next stage is segmenting the document into its subcomponents. Segmentation is an important stage because the extent one can reach in separation of words, lines, or characters directly affects the recognition rate of the script. There are two types of segmentation: external segmentation, which is the isolation of various writing units, such as paragraphs, sentences, or words, and internal segmentation, which is the isolation of letters, especially in cursively written words.

1) External Segmentation: It is the most critical part of the document analysis, which is a necessary step prior to the off-line CR Although document analysis is a relatively different research area with its own methodologies and techniques, segmenting the document image into text and nontext regions is an integral part of the OCR software. Therefore, one who works in the CR field should have a general overview for document analysis techniques.

Page layout analysis is accomplished in two stages: The first stage is the *structural analysis*, which is concerned with the segmentation of the image into blocks of document components (paragraph, row, word, etc.), and the second one is the *functional analysis*, which uses location, size, and various layout rules to label the functional content of document components (title, abstract, etc.) [129].

A number of approaches regard a homogeneous region in a document image as a textured region. Page segmentation is then implemented by finding textured regions in gray-scale or color images. For example, Jain *et al.* use Gabor filtering and mask convolution [75], the Tang *et al.* approach is based on fractal signature [166], and Doermann's method [39] employs wavelet multiscale analysis. Many approaches for page segmentation concentrate on processing background pixels or using the white space in a page to identify homogeneous regions [72]. These techniques include X–Y tree [25], pixel-based projection profile [138], connected component-based projection profile [61], white space tracing [2], and white space thinning [86]. They can be regarded as top–down approaches, which segment a page, recursively, by X-cut and Y-cut from large components, starting with the whole page to small components, eventually reaching individual characters. On the other hand, there are some bottom–up methods which recursively grow the homogeneous regions from small components based on the processing on pixels and connected components. An example of this approach may be the Docstrum method, which uses k-nearest neighbor clustering [129]. Some techniques combine both top–down and bottom–up techniques [108]. A brief survey of the work in page decomposition can be found in [72].

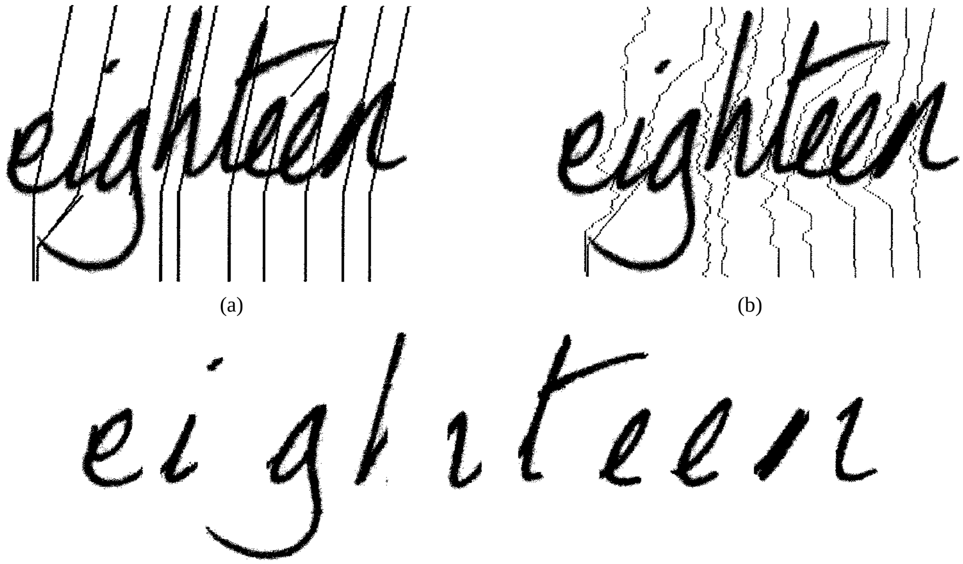
2) Internal Segmentation: Although the methods have developed remarkably in the last decade and a variety of techniques have emerged, segmentation of cursive script into letters is still an unsolved problem. Character segmentation strategies are divided into three categories [22].

a) Explicit Segmentation: In this strategy, the segments are identified based on "character-like" properties. The process of cutting up the image into meaningful components is given a special name: *dissection*. Dissection is a process that analyzes an image without using a specific class of shape information. The criterion for good segmentation is the agreement of general properties of the segments with those expected for valid characters. Available methods based on the dissection of an image use white space and pitch [66], vertical projection analysis [174], connected component analysis [176], and landmarks [63]. Moreover, explicit segmentation can be subjected to evaluation using linguistic context [99].

b) Implicit Segmentation: This segmentation strategy is based on recognition. It searches the image for components that match predefined classes. Segmentation is performed by the use of recognition confidence, including syntactic or semantic correctness of the overall result. In this approach, two classes of methods can be employed: 1) methods that make some search process and 2) methods that segment a feature representation of the image [22].

The first class attempts to segment words into letters or other units without use of feature-based dissection algorithms. Rather, the image is divided systematically into many overlapping pieces without regard to content. Conceptually, these methods originate from schemes developed for the recognition of machine-printed words [23]. The basic principle is to use a mobile window of variable width to provide sequences of tentative segmentations, which are confirmed by CR. Another technique combines dynamic programming and NNs [20]. Finally, the method of selective attention takes NNs even further in the handling of the segmentation problem [45].

The second class of methods segments the image implicitly by classification of subsets of spatial features collected from the image as a whole. This approach can be divided into two categories: HMM-based approaches and non-Markov-based approaches. The survey in [51] provides an introduction to HMM-based approaches in recognition



(a) (b) (c)

Fig. 4. Segmentation by finding the shortest path of a graph formed by gray level image. (a) Segmentation intervals. (b) Segmentation paths. (c) Segments.

applications. In [26] and [121], HMMs are used to structure the entire word recognition process. Non-Markov approaches stem from concepts used in machine vision for recognition of occluded object [28]. This family of recognition-based approaches uses probabilistic relaxation [65], the concept of regularities and singularities [157], and backward matching [98].

c) Mixed Strategies: They combine explicit and implicit segmentation in a hybrid way. A dissection algorithm is applied to the image, but the intent is to "over segment," i.e., to cut the image in sufficiently many places that the correct segmentation boundaries are included among the cuts made. Once this is assured, the optimal segmentation is sought by evaluation of subsets of the cuts made. Each subset implies a segmentation hypothesis, and classification is brought to bear to evaluate the different hypothesis and choose the most promising segmentation [158]. In [100], the segmentation problem is formulated as finding the shortest path of a graph formed by binary and gray-level document image. In [6], the HMM probabilities, obtained from the characters of a dissection algorithm, are used to form a graph. The optimum path of this graph improves the result of the segmentation by dissection and HMM recognition. Fig. 4(a) indicates the initial segmentation intervals obtained by evaluating the local maxima and minima together with the slant angle information. Fig. 4(b) and (c) show the shortest path for each segmentation interval and the resulting candidate characters, respectively. Mixed strategies yield better results compared to explicit and implicit segmentation methods.

The techniques presented above have limited capabilities in segmentation. Error detection and correction mechanisms should be embedded into the systems for which they were developed. As Casey and Lecolinet [22] pointed out, the wise use of context and classifier confidence leads to improved accuracy.

C. Representation

Image representation plays one of the most important roles in a recognition system. In the simplest case, gray-level or binary images are fed to a recognizer. However, in most of the recognition systems, in order to avoid extra complexity and to increase the accuracy of the algorithms, a more compact and characteristic representation is required. For this purpose, a set of features is extracted for each class that helps distinguish it from other classes while remaining invariant to characteristic differences within the class [131]. A good survey on feature extraction methods for CR can be found in [172]. In the following, hundreds of document image representation methods are categorized into three major groups.

1) Global Transformation and Series Expansion: A continuous signal generally contains more information than needs to be represented for the purpose of classification. This may be true for discrete approximations of continuous signals as well. One way to represent a signal is by a linear combination of a series of simpler well-defined functions. The coefficients of the linear combination provide a compact encoding known as transformation or/and series expansion. Deformations like translation and rotations are invariant under global transformation and series expansion. Common transform and series expansion methods used in the CR field are include the following.

a) Fourier Transforms: The general procedure is to choose magnitude spectrum of the measurement vector as the features in an n-dimensional Euclidean space. One of the most attractive properties of the Fourier transform is the ability to recognize the position-shifted characters, when it observes the magnitude spectrum and ignores the phase. Fourier transforms have been applied to CR in many ways [178], [192].

b) Gabor Transform: This is a variation of the windowed Fourier transform. In this case, the window used is not a discrete size but is defined by a Gaussian function [62].

c) Wavelets: Wavelet transformation is a series expansion technique that allows us to represent the signal at different levels of resolution. The segments of document image, which may correspond to letters or words, are represented by wavelet coefficients, corresponding to various levels of resolution. These coefficients are then fed to a classifier for recognition [103], [154]. The representation in multiresolution analysis (MRA) with low resolution can absorb the local variation in handwriting as opposed to MRA with high resolution. However, the representation in low resolution may cause the important details for the recognition stage to be lost.

d) Moments: Moments, such as central moments, Legendre moments, and Zernike moments, form a compact representation of the original document image that make the process of recognizing an object scale, translation, and rotation invariant [34], [83]. Moments are considered as series expansion representation since the original image can be completely reconstructed from the moment coefficients.

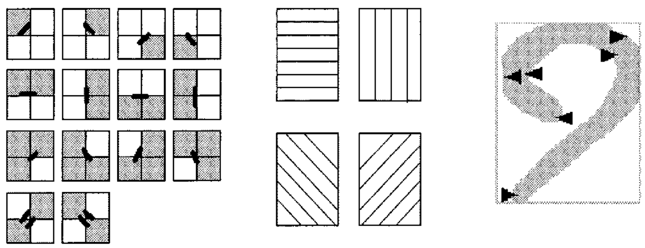


Fig. 5. Contour direction and bending point features with zoning [131].

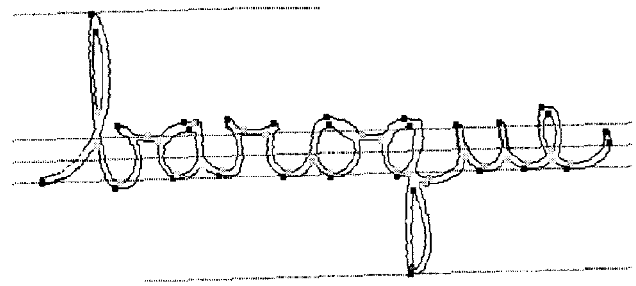


Fig. 6. Topological features: Maxima and minima on the exterior and interior contours, reference lines, ascenders, and descenders [119].

e) Karhunen–Loeve Expansion: This is an eigenvector analysis which attempts to reduce the dimension of the feature set by creating new features that are linear combinations of the original ones. It is the only optimal transform in terms of information compression. Karhunen–Loeve expansion is used in several pattern recognition problems such as face recognition. It is also used in the National Institute of Standards and Technology (NIST) OCR system for form-based handprint recognition [49]. Since it requires computationally complex algorithms, the use of Karhunen–Loeve features in CR problems is not widespread. However, by the increase of the computational power, it will become a realistic feature for CR systems in the next few years [172].

2) Statistical Representation: Representation of a document image by statistical distribution of points takes care of style variations to some extent. Although this type of representation does not allow the reconstruction of the original image, it is used for reducing the dimension of the feature set providing high speed and low complexity. The following are the major statistical features used for character representation:

a) Zoning: The frame containing the character is divided into several overlapping or nonoverlapping zones. The densities of the points or some features in different regions are analyzed and form the representation. For example, contour direction features measure the direction of the contour of the character [122], which are generated by dividing the image array into rectangular and diagonal zones and computing histograms of chain codes in these zones. Another example is the bending point features, which represent high curvature points, terminal points, and fork points [165]. Fig. 5 indicates contour direction and bending point features.

b) Crossings and Distances: A popular statistical feature is the number of crossing of a contour by a line segment in a specified direction. In [5], the character frame is partitioned into a set of regions in various directions and then the black runs in each region are coded by the powers of two. Another study [121] encodes the location and number of transitions from background to foreground pixels along vertical lines through the word. Also, the distance of line segments from a given boundary, such as the upper and lower portion of the frame, can be used as statistical features [19]. These features imply that a horizontal threshold is established above, below, and through the center of the normalized script. The number of times the script crosses a threshold becomes the value of that feature. The obvious intent is to catch the ascending and descending portions of the script.

c) Projections: Characters can be represented by projecting the pixel gray values onto lines in various directions. This representation creates a 1-D signal from a two-dimensional (2-D) image, which can be used to represent the character image [168], [179].

3) Geometrical and Topological Representation: Various global and local properties of characters can be represented by geometrical and topological features with high tolerance to distortions and style variations. This type of representation may also encode some knowledge about the structure of the object or may provide some knowledge as to what sort of components make up that object. Hundreds of topological and geometrical representations can be grouped into the following categories.

a) Extracting and Counting Topological Structures: In this group of representation, a predefined structure is searched in a character or word. The number or relative position of these structures within the character forms a descriptive representation. Common primitive structures are the strokes, which make up a character. These primitives can be as simple as lines (l) and arcs (c) which are the main strokes of Latin characters and can be as complex as curves and splines making up Arabic or Chinese characters. In on-line CR, a stroke is also defined as a line segment from pen-down to pen-up [163]. Characters and words can be successfully represented by extracting and counting many topological features such as the extreme points, maxima and minima, cusps above and below a threshold, openings to the right, left, up, and down, cross (x) points, branch (T) points, line ends (J), loops, direction of a stroke from a special point, inflection between two points, isolated dots, a bend between two points, symmetry of character, horizontal curves at top or bottom, straight strokes between two points, ascending, descending, and middle strokes and relations among the stroke that make up a character, etc. [58], [111], [112]. Fig. 6 indicates some of the topological features.

b) Measuring and Approximating the Geometrical Properties: In many studies (e.g., [36] and [94]), the characters are represented by the measurement of the geometrical quantities such as the ratio between width and height of the bounding box of a character, the relative distance between the last point and the last y-min, the relative horizontal and vertical distances between first and last points, distance between two points, comparative lengths between two strokes, width of a stroke, upper and lower masses of words, and word length. A very important characteristic measure is the curvature or change in the curvature [133]. Among many methods for measuring the curvature information, one is suggested by [130] for measuring the directional distance and measures local stroke direction distribution for directional decomposition of the character image.

The measured geometrical quantities can be approximated by a more convenient and compact geometrical set of features. A class of methods includes polygonal approximation of a thinned character [149]. A more precise and expensive version of the polygonal approximation is the cubic spline representation [118].

c) Coding: One of the most popular coding schema is Freeman's chain code. This coding is essentially obtained by mapping the strokes of a character into a 2-D parameter space, which is made up of codes, as shown in Fig. 7. There are many versions of chain coding. As an example, in [57], the character frame is divided to left–right sliding window and each region is coded by the chain code.

d) Graphs and Trees: Words or characters are first partitioned into a set of topological primitives, such as strokes, loops, cross points, etc. Then, these primitives are represented using attributed or relational graphs [106]. There are two kinds of image representation by graphs.

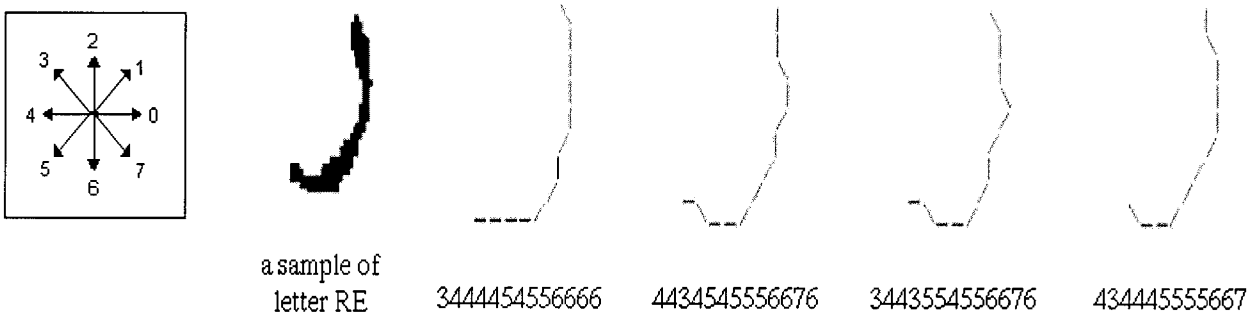


Fig. 7. Sample Arabic character and the chain codes of its skeleton.

The first kind uses the coordinates of the character shape [33]. The second kind is an abstract representation with nodes corresponding to the strokes and edges corresponding to the relationships between the strokes [109]. Trees can also be used to represent the words or characters with a set of features, which have a hierarchical relation [110].

The feature extraction process is performed mostly on binary images. However, binarization of a gray level image may remove important topological information from characters. In order to avoid this problem, some studies attempt to extract features directly from grayscale character images [102].

In conclusion, the major goal of representation is to extract and select a set of features, which maximizes the recognition rate with the least amount of elements. In [87], feature extraction and selection is defined as extracting the most representative information from the raw data, which minimizes the within class pattern variability while enhancing the between class pattern variability.

Feature selection can be formulated as a dynamic programming problem for selecting the k-best features out of N features, with respect to a cost function such as Fishers discrimant ratio. Feature selection can also be accomplished by using principal component analysis or a NN trainer. In [35], the performance of several feature selection methods for CR are discussed and compared. Selection of features using a methodology as mentioned here requires expensive computational power and most of the time yields a suboptimal solution [47]. Therefore, the feature selection is, mostly, done by heuristics or by intuition for a specific type of the CR application.

D. Training and Recognition Techniques

CR systems extensively use the methodologies of pattern recognition, which assigns an unknown sample into a predefined class. Numerous techniques for CR can be investigated in four general approaches of pattern recognition, as suggested in [76]:

1) template matching;
2) statistical techniques;
3) structural techniques;
4) neural networks (NNs).

The above approaches are neither necessarily independent nor disjointed from each other. Occasionally, a CR technique in one approach can also be considered to be a member of other approaches.

In all of the above approaches, CR techniques use either holistic or analytic strategies for the training and recognition stages: holistic strategy employs top–down approaches for recognizing the full word, eliminating the segmentation problem. The price for this computational saving is to constrain the problem of CR to limited vocabulary. Also, due to the complexity introduced by the representation of whole cursive word (compared to the complexity of a single character or stroke), the recognition accuracy is decreased.

TABLE I
STRATEGIES OF THE CHARACTER RECOGNITION

Holistic Strategy	Analytic Strategy
Whole Word Recognition	Sub-word or Letter Recognition
Limited Vocabulary	Unlimited Vocabulary
Vulnerable to Recognition of Long Words	Vulnerable to Segmentation Errors
No Segmentation	Requires Explicit or Implicit Segmentation

On the other hand, the analytic strategies employ bottom–up approaches starting from stroke or character level and going toward producing a meaningful text. Explicit or implicit segmentation algorithms are required for this strategy, not only adding extra complexity to the problem, but also introducing segmentation error to the system. However, with the cooperation of segmentation stage, the problem is reduced to the recognition of simple isolated characters or strokes, which can be handled for unlimited vocabulary with high recognition rates (see Table I).

1) Template Matching: CR techniques vary widely according to the feature set selected from the long list of features described in the previous section for image representation. Features can be as simple as the gray-level image frames with individual characters or words or as complicated as graph representation of character primitives. The simplest way of CR is based on matching the stored prototypes against the character or word to be recognized. Generally speaking, matching operation determines the degree of similarity between two vectors (group of pixels, shapes, curvature, etc.) in the feature space. Matching techniques can be studied in three classes.

a) Direct Matching: A gray-level or binary input character is directly compared to a standard set of stored prototypes. According to a similarity measure (e.g., Euclidean, Mahalanobis, Jaccard, or Yule similarity measures, etc.), a prototype matching is done for recognition. The matching techniques can be as simple as one-to-one comparison or as complex as decision tree analysis in which only selected pixels are tested. A template matcher can combine multiple information sources, including match strength and k-nearest neighbor measurements from different metrics [46], [175]. Although direct matching method is intuitive and has a solid mathematical background, the recognition rate of this method is very sensitive to noise.

b) Deformable Templates and Elastic Matching: An alternative method is the use of deformable templates, where an image deformation is used to match an unknown image against a database of known images. In [73], two characters are matched by deforming the contour of one to fit the edge strengths of the other. A dissimilarity measure is derived from the amount of deformation needed, the goodness of fit of the edges, and the interior overlap between the deformed shapes (see Fig. 8).

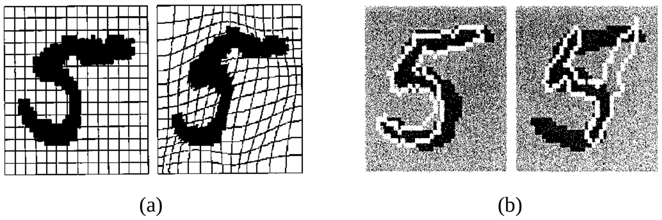


Fig. 8. Deformable templates. (a) Deformations of a sample digit. (b) Deformed template superimposed on target image with dissimilarity measures in [78].

The basic idea of elastic matching is to optimally match the unknown symbol against all possible elastic stretching and compression of each prototype. Once the feature space is formed, the unknown vector is matched using dynamic programming and a warping function [68], [169]. Since the curves obtained from the skeletonization of the characters could be distorted, elastic-matching methods cannot deal with topological correlation between two patterns in the off-line CR. In order to avoid this difficulty, a self-organization matching approach is proposed in [107] for hand-printed CR using thick strokes. Elastic matching is also popular in on-line recognition systems [127].

c) Relaxation Matching: It is a symbolic level image matching technique that uses feature-based description for the character image. First, the matching regions are identified. Then, based on some well-defined ratings of the assignments, the image elements are compared to the model. This procedure requires a search technique in a multidimensional space for finding the global maximum of some functions [144]. In [182], Xie *et al.* proposed a handwritten Chinese character system, where a small number of critical structural features, such as end points, hooks, T-shape, cross, and corner are used. Recognition is done by computing the matching probabilities between two features by a relaxation method.

The matching techniques mentioned above are sometimes used individually or combined in many ways as part of the CR schemes.

2) Statistical Techniques: Statistical decision theory is concerned with statistical decision functions and a set of optimality criteria, which maximizes the probability of the observed pattern given the model of a certain class [38]. Statistical techniques are mostly based on three major assumptions.

1) Distribution of the feature set is Gaussian or in the worst case uniform.
2) There are sufficient statistics available for each class.
3) Given ensemble of images $\{I\}$, one is able to extract a set of features $\{f_i\} \in F$, $i \in \{1, \ldots, n\}$, which represents each distinct class of patterns.

The measurements taken from n features of each word unit can be thought to represent an n-dimensional vector space and the vector, whose coordinates correspond to the measurements taken, represents the original word unit. The major statistical approaches applied in the CR field are listed below.

a) Nonparametric Recognition: This method is used to separate different pattern classes along hyperplanes defined in a given hyperspace. The best known method of nonparametric classification is the NN and is extensively used in CR [159]. It does not require *a priori* information about the data. An incoming pattern is classified using the cluster, whose center is the minimum distance from the pattern over all the clusters.

b) Parametric Recognition: Since *a priori* information is available about the characters in the training data, it is possible to obtain a parametric model for each character [14]. Once the parameters of the model, which are based on some probabilities, are obtained, the characters are classified according to some decision rules such as maximum likelihood or Bayes method.

c) Clustering Analysis: The clusters of character features, which represent distinct classes, are analyzed by way of clustering methods. Clustering can be performed either by agglomerative or divisive algorithms. The agglomerative algorithms operate step-by-step merging of small clusters into larger ones by a distance criterion. On the other hand, the divisive methods split the character classes under certain rules for identifying the underlying character [188].

d) Hidden Markov Modeling (HMM): HMMs are the most widely and successfully used technique for handwritten CR problems [26], [27], [89], [120], [121]. It is defined as a stochastic process generated by two interrelated mechanisms: a Markov Chain having a finite number of states and a set of random functions, each of which is associated with a state [145]. At discrete instants of time, the process is assumed to be in some state, and an observation is generated by the random function corresponding to the current state. The underlying Markov chain then changes states according to its transitional probabilities. Here, the job is to build a model that explains and characterizes the occurrence of the observed symbols [77]. The output corresponding to a single symbol can be characterized as discrete or continuous. Discrete outputs may be characters from a finite alphabet or quantized vectors from a codebook, while continuous outputs are represented by samples from a continuous waveform. In generating a word or a character, the system passes from one state to another, each state emitting an output according to some probabilities until the entire word or character is out. There are two basic approaches to CR systems using HMM.

- *Model-Discriminant HMM*: A model is constructed for each class (word, character, or segmentation unit) in the training phase. States represent cluster centers for the feature space. The goal of classification is then to decide on the model, which produces the unknown observation sequence. In [50] and [121], each column of the word image is represented as a feature vector by labeling each pixel according to its location. A separate model is trained for each character using word images, where the character boundaries have been identified. Then, the word matching process uses models for each word in a supplied lexicon. The advantage of this technique is that the word needs not to be segmented into characters for the matching process. In segmentation-based approaches, the segmentation process is applied to the word image and several segmentation alternatives are proposed [6], [89]. Each segmentation alternative is embedded to the HMM recognizer which assigns a probability for each segment. In postprocessing, a search algorithm takes into account the segmentation alternatives and their recognition scores in order to find the final result. Model-discriminant HMMs also used for isolated CR in various studies [5], [90]. Fig. 9 shows a model discriminant HMM with five states left-to-right topology constructed for each character class.
- *Path-Discriminant HMM*: In this approach, a single HMM is constructed for the whole language or context. Modeling is supported by the initial and transitional probabilities on the basis of observations from a random experiment, generally by using a lexicon. Each state may signify a complete character, a partial character, or joint characters. Recognition consists of estimation of the optimal path for each class using Viterbi algorithm, based on dynamic programming. In [26] and [27], the input word image is first segmented into a sequence of segments in which an individual segment corresponds to a state. The HMM parameters

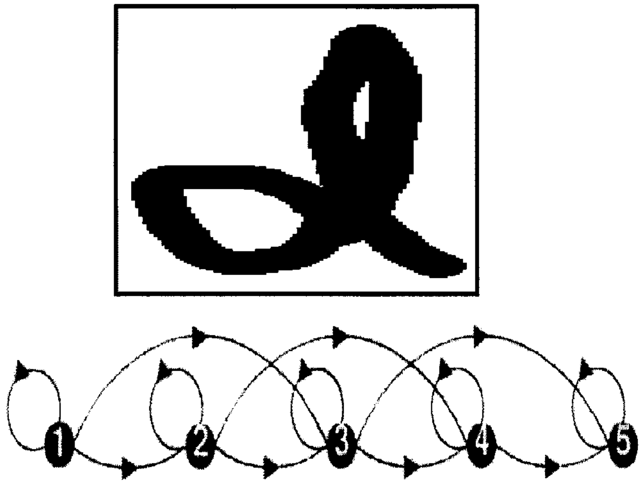


Fig. 9. Model discriminant HMM with five states for the character "d."

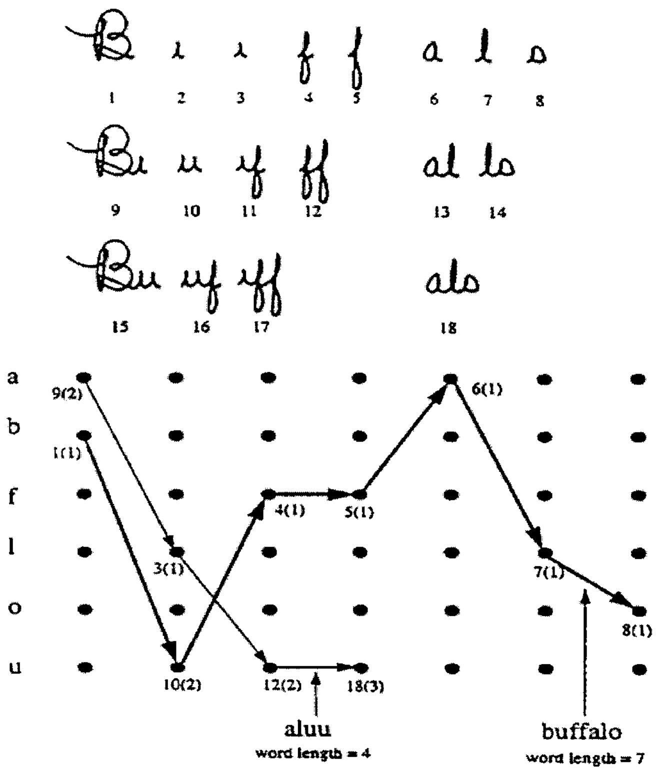


Fig. 10. Path discriminant HMM: All block images of up to three segments (top) and two particular paths or state sequences.

are estimated from the lexicon and the segmentation statistics of the training images. Then, a modified Viterbi algorithm is used to find the best L-state sequences in recognition of a word image (see Fig. 10).

The performances of these two approaches are compared in various experiments by utilizing different lexicon sizes in [93]. The major design issue in the HMM problem is the selection of the feature set and HMM topology. These two tasks are strongly related to each other, and there is no systematic approach developed for this purpose.

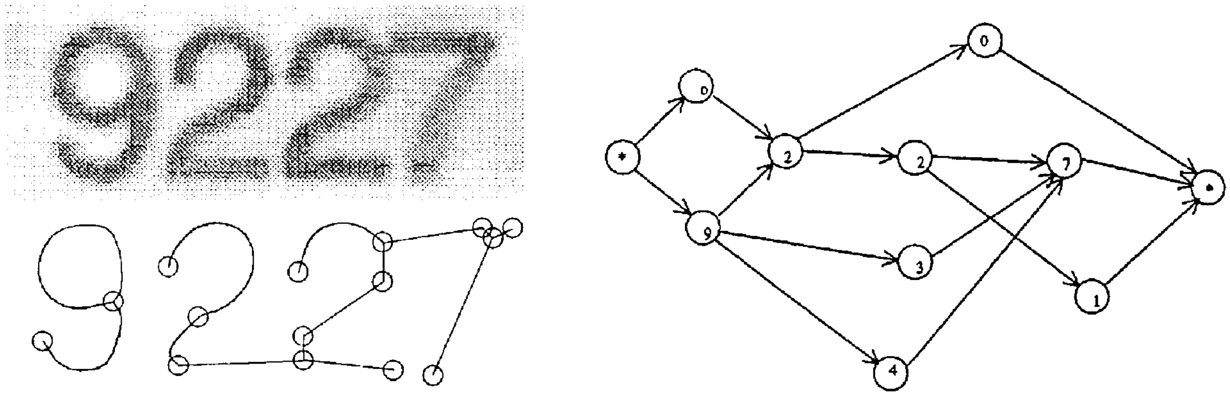
e) Fuzzy Set Reasoning: Instead of using a probabilistic approach, this technique employs fuzzy set elements in describing the similarities between the features of the characters. Fuzzy set elements give more realistic results when there is no *a priori* knowledge about the data, and therefore, the probabilities cannot be calculated. The characters can be viewed as a collection of strokes, which are compared to reference patterns by fuzzy similarity measures. Since the strokes under consideration are fuzzy in nature, the concept of fuzziness is utilized in the similarity measure. In order to recognize a character, an unknown input character is matched with all the reference characters and is assigned to the class of the reference character with the highest score of similarity among all the reference characters. In [32], fuzzy similarity measure is utilized to define fuzzy entropy for off-line handwritten Chinese characters. An off-line handwritten CR system is proposed in [1] using a fuzzy graph theoretic approach, where each character is described by a fuzzy graph. A fuzzy graph-matching algorithm is then used for recognition. Wang and Mendel propose an off-line handwritten recognition algorithm system, which generates crisp features first and then fuzzify the characters by rotating or deforming them [177]. The algorithm uses average values of membership for final decision. In handwritten word recognition, Gader *et al.* uses the choquet fuzzy integral as the matching function [48]. In on-line handwriting recognition, Plamondon proposes a fuzzy-syntactic approach to allograph modeling [135].

3) Structural Techniques: The recursive description of a complex pattern in terms of simpler patterns based on the shape of the object was the initial idea behind the creation of structural pattern recognition. These patterns are used to describe and classify the characters in the CR systems. The characters are represented as the union of the structural primitives. It is assumed that the character primitives extracted from writing are quantifiable, and one can find the relations among them. The following structural methods are applied to the CR problems.

a) Grammatical Methods: In the mid-1960s, researchers started to consider the rules of linguistics for analyzing the speech and writing. Later, various orthographic, lexicographic, and linguistic rules were applied to the recognition schemes. The grammatical methods create some production rules in order to form the characters from a set of primitives through formal grammars. These methods may combine any type of topological and statistical features under some syntactic and/or semantic rules [137], [155]. Formal tools, like language theory, allow us to describe the admissible constructions and to extract the contextual information about the writing by using various types of grammars, such as string grammars, graph grammars, stochastic grammars, and picture description language [173].

In grammatical methods, training is done by describing each character by a grammar G_i. In the recognition phase, the string, tree or graph of any writing unit (character, word, or sentence) is analyzed in order to decide to which pattern grammar it belongs [13]. Top–down or bottom–up parsing does syntax analysis. Given a sentence, a derivation of the sentence is constructed and the corresponding derivation tree is obtained. The grammatical methods in the CR area are applied in character [158], word [81], and sentence [60] levels. In character level, picture description language (PDL) is used to model each character in terms of a set of strokes and their relationship. This approach is used for Indian CR, where Devanagari characters are presented by a PDL [158]. The system stores the structural description in terms of primitives and the relations. Recognition involves a search for the unknown character, based on the stored description. In word level, bi-gram and three-gram statistics are used to form word generation grammars. Word and sentence representation uses knowledge bases with linguistic rules. Grammatical methods are mostly used in the postprocessing stage for correcting the recognition errors [17], [152].

b) Graphical Methods: Writing units are represented by trees, graphs, di-graphs, or attributed graphs. The character primitives (e.g., strokes) are selected by a structural approach, irrespective of how the



(a) (b)

Fig. 11. Graph matching. (a) Feature graph of digit string "9227." (b) Its corresponding directed net [163].

final decision making is made in the recognition [82], [180]. For each class, a graph or tree is formed in the training stage to represent strokes, letters, or words. Recognition stage assigns the unknown graph to one of the classes by using a graph similarity measure.

There are a great variety of approaches that use the graphical methods. Hierarchical graph representation approach is used for handwritten Korean and Chinese CR in [82] and [109], respectively. Pavlidis and Rocha use homoemorphic subgraph matching method for complete word recognition in [150]. First, the word image is converted into a feature graph. Edges of the feature graph are the skeletons of the input strokes. Graph nodes correspond to singularities (inflection points, branch points, sharp corners, and ending points) on the strokes. Then, the meaningful subgraphs of the feature graph are recognized, matching the previously defined character prototypes. Finally, each recognized subgraph is introduced as a node in a directed net that compiles different interpretations of the features in the feature graph. A path in the net represents a consistent succession of characters in the net (Fig. 11).

In [157], Simon proposes an off-line cursive script recognition scheme. The features are regularities, which are defined as uninformative parts and singularities, which are defined as informative strokes about the characters. Stroke trees are obtained after skeletonization. The goal is to match the trees of singularities.

Although it is computationally expensive, relaxation matching is also a popular method in graphical approaches to the CR problem [33], [182].

4) Neural Networks (NNs): An NN is defined as a computing architecture that consists of a massively parallel interconnection of adaptive "neural" processors. Because of its parallel nature, it can perform computations at a higher rate compared to the classical techniques. Because of its adaptive nature, it can adapt to changes in the data and learn the characteristics of input signal. An NN contains many nodes. The output from one node is fed to another one in the network and the final decision depends on the complex interaction of all nodes. In spite of the different underlying principles, it can be shown that most of the NN architectures are equivalent to statistical pattern recognition methods [148].

Several approaches exist for training of NNs [74]. These include the error correction, Boltzman, Hebbian, and competitive learning. They cover binary and continuous valued input, as well as supervised and unsupervised learning. On the other hand, NN architectures can be classified into two major groups, namely, feedforward and feedback (recurrent) networks. The most common NNs used in the CR systems are the multilayer perceptron of the feedforward networks and the Kohonen's self organizing map (SOM) of the feedback networks.

Multilayer perceptron, proposed by Rosenblatt [15] and elaborated by Minsky and Papert [117], is applied in CR by many authors. One example is the feature recognition network proposed by Hussain and Kabuka [70], which has a two-level detection scheme. The first level is for detection of subpatterns, and the second level is for detection of characters. Mohiuddin *et al.* use a multinetwork system in hand-printed CR by combining contour direction and bending point features [115], which is fed to various NNs. Neocognitron of Fukushima *et al.* [44] is a hierarchical network consisting of several layers of alternating neuron-like cells. S-cells are used for feature extracting and C-cells allow for positional errors in the features. The last layer is the recognition layer. Some of the connections are variable and can be modified by learning. Each layer of S and C cells are called *cell planes*. This study proposes some techniques for selecting training patterns useful for deformation-invariant recognition of a large number of characters. The feedforward NN approach to the machine-printed CR problem is proven to be successful in [9], where the NN is trained with a database of 94 characters and tested in 300 000 characters generated by a postscript laser printer, with 12 common fonts in varying size. No errors were detected. In this study, Garland *et al.* propose a two-layer NN, trained by a centroid dithering process. The modular NN architecture is used for unconstrained handwritten numeral recognition in [132]. The whole classifier is composed of subnetworks. A subnetwork, which contains three layers, is responsible for a class among ten classes. Another study [152] uses recurrent NN in order to estimate probabilities for the characters represented in the skeleton of word image (see Fig. 12). A recent study proposed by Maragos and Pessoa incorporates the properties of multilayer perceptron and morphological rank NNs for handwritten CR. They claim that this unified approach gives higher recognition rates than a multilayer perceptron with smaller processing time [139].

Most of the recent developments on handwritten CR research are concentrated on Kohonen's SOM [88]. SOM integrates the feature extraction and recognition steps in a large training set of characters. It can be shown that it is analogous to k-means clustering algorithm. An example of SOM on CR systems is the study of Liou and Yang [107], which presents a self-organization matching approach to accomplish the recognition of handwritten characters drawn with thick strokes. In [146], Reddy and Nagabhushan propose a combination of modified SOM and learning vector quantization to define a three-dimensional (3-D) NN model for handwritten numeral recognition. They report higher recognition rates with shorter training time than other SOMs reported in the literature. Jabri *et al.* realized the adaptive-subspace SOM map to build a modular classification system for handwritten digit recognition [190].

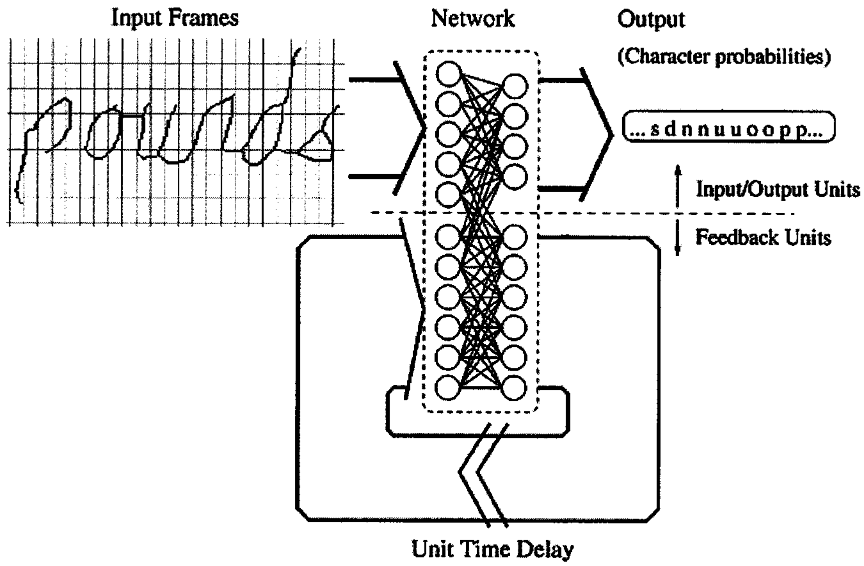


Fig. 12. Recurrent NN in [167].

5) Combined CR Techniques: The above review indicates that there are many training and recognition methods available for the CR systems. All of the methods have their own superiorities and weaknesses. Now, the question of "May these methods be combined in a meaningful way to improve the recognition results?" appears. In order to answer this question, various strategies are developed by combining the CR techniques. Hundreds of these studies can be classified either according to the algorithmic point of view, the representational point of view, or according to the architecture they use [3]. In this study, we suffice to classify the combination approaches in terms of their architecture in three classes:

1) serial architecture;
2) parallel architecture;
3) hybrid architecture.

Serial architecture feeds the output of a classifier into the next classifier. There are four basic methodologies used in the serial architecture, namely

1) sequential;
2) selective;
3) boosting;
4) cascade;

methodologies.

In the sequential methodology, the goal of each stage is to reduce the number of classes in which the unknown pattern may belong. Initially, the unknown pattern belongs to one of the total number of classes. The number of probable classes reduces at each stage, yielding the label of the pattern in the final stage [126]. In the selective methodology the initial classifier assigns the unknown pattern into a group of characters that are similar to each other. These groups are further classified in later stages in a tree hierarchy. At each level of the tree, the children of the same parent are similar with respect to a similarity measure. Therefore, the classifiers work from coarse to fine recognition methods in small groups [52]. In the boosting method each classifier handles the classes, which cannot be handled by the previous classifiers [41]. Finally, in the cascade method, the classifiers are connected from simpler to more complex ones. The patterns, which do not satisfy a certain confidence level, are passed to a costlier classifier in terms of features and/or recognition scheme [136].

Parallel architectures combine the result of more than one independent algorithm by using various methodologies. Among many methodologies, the voting [95], Bayesian [79], Dempster–Shafer (D–S) [183], behavior–knowledge space [69], mixture of experts [71], and stacked generalization [181] are the most representative.

Voting is a democracy-behavior approach based on "the opinion of the majority wins" [95]. It treats classifiers equally without considering their differences in performance. Each classifier represents one score that is either as a whole assigned to one class label or divided into several labels. The label, which receives more than half of the total scores, is taken as the final result. While voting methods are only based on the label without considering the error of each classifier, Bayesian and D–S approaches take these errors into consideration.

The Bayesian approach uses the Bayesian formula to integrate classifiers' decisions. Usually, it requires an independence assumption in order to tackle the computation of the joint probability [79]. The D–S method deals with uncertainty management and incomplete reasoning. It aggregates committed, uncommitted, and ignorant beliefs [183]. It allows one to attribute belief to subsets, as well as to individual elements of the hypothesis set. Bayesian and D–S approaches utilize the probabilistic reasoning for classification.

The behavior–knowledge space method has been developed in order to avoid the independence assumption of the individual classifiers [69]. In order to avoid this assumption, the information should be derived from a knowledge space, which can concurrently record the decisions of all classifiers on each learned sample. The knowledge space records the behavior of all classifiers, and the method derives the final decisions from the behavior knowledge space.

The mixture of experts' method is similar to the voting method, where decision is weighted according to the input. The experts partition the input space and a gating system decides on the weights for a given input. Thus, the experts are allowed to specialize on local regions of the input space. This is dictated through the use of a cost function, based on the likelihood of a normal mixture [71].

Stacked generalization is another extension of the voting method [181], where the outputs of the classifiers are not linearly combined. Instead, a combiner system is trained for the final decision.

Finally, the hybrid architecture is a crossbreeding between the serial and parallel architectures. The main idea is to combine the power of both architectures and to prevent the inconveniences [52].

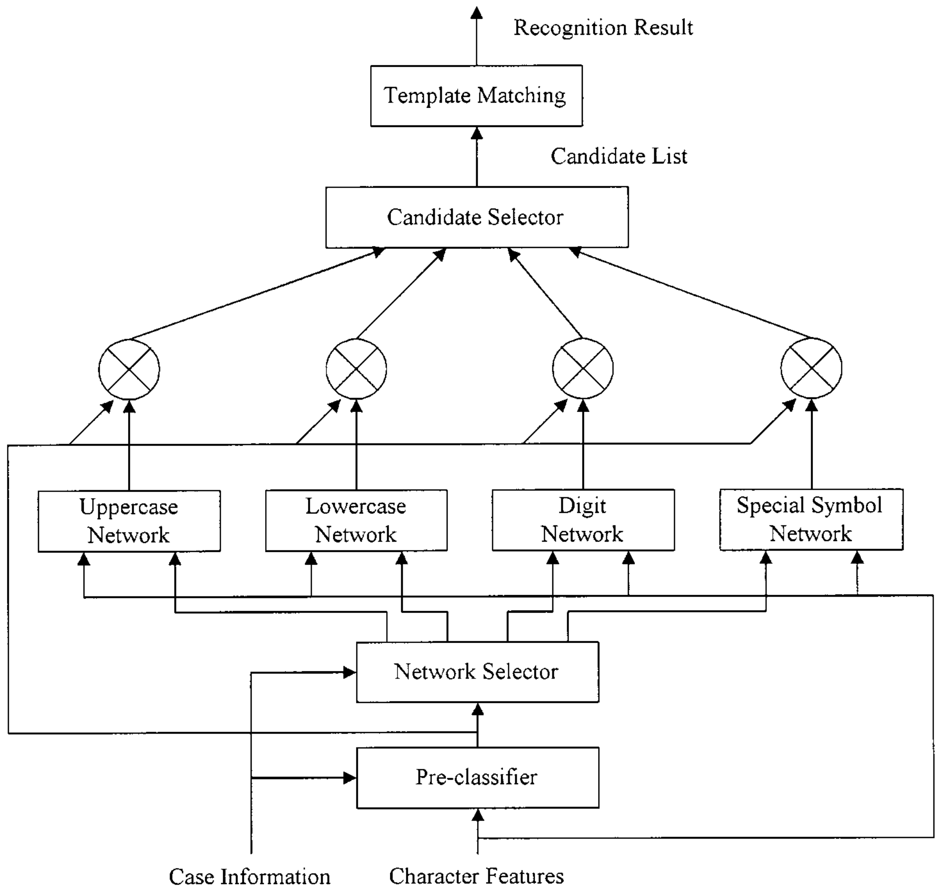


Fig. 13. Combined CR system in [56].

Examples of the combined approaches may be given as follows.

In [167], a sequential approach based on multifeature and multilevel classification is developed for handwritten Chinese characters. First, a group of classifiers breaks down all the characters into a smaller number of groups; hence, the number of candidates for the process in the next step drops sharply. Then, the multilevel character classification method, which is composed of five levels, is employed for the final decision. In the first level, a Gaussian distribution selector is used to select a smaller number of candidates from several groups. From the second level to the fifth one, matching approaches using different features are performed, respectively.

Another example is given by the study of Shridhar and Kimura, where two algorithms are combined for the recognition of unconstrained isolated handwritten numerals [156]. In the first algorithm, a statistical classifier is developed by optimizing a modified quadratic discriminant function derived from the Bayes rule. The second algorithm implements a structural classifier; a tree structure is used to express each number. Recognition is done in a two-pass procedure with different thresholds. The recognition is made either using parallel or serial decision strategies.

In [183], Xu *et al.* studied the methods of combining multiple classifiers and their application to handwritten recognition. They proposed a serial combination of structural classification and relaxation matching algorithm for the recognition of handwritten zip codes. It is reported that the algorithm has very low error rate and high computational cost.

In [69], Suen *et al.* derive the best final decision by using the behavior–knowledge space method for the recognition of unconstrained handwritten numerals. They use three different classifiers. Their experiments show that the method achieves very promising performances and outperforms voting, Bayesian, and D–S approaches.

As an example of the stacked generalization method, a study is proposed in [143] for on-line CR. This method combines two classifiers with an NN. The training ability of NN works well in managing the conflicts appearing between the classifiers. Here, weighting is subtler as it is applied differently on each class and not equally on all the classifiers score.

In [191], a new kind of NN—quantum neural network (QNN) is proposed and tested on the recognition of handwritten numerals. QNN combines the advantages of neural modeling and the fuzzy theoretic approach. An effective decision fusion system is proposed with high reliability.

A good example of the hybrid method is proposed in [52], where IBM research group combines NN and template matching methods in a complete CR scheme (see Fig. 13). First, the two-stage multinetwork (TSMN) classifier identifies the top three candidates. TSMN consists of a bank of specialized networks, each of which is designed to recognize a subset of the entire character set. A preclassifier and a network selector are employed for selectively invoking the necessary specialized networks. Next, the template matching (TM) classifier is invoked to match the input pattern with only those templates in the three categories selected by the TSMN classifier. Template matching distances are used to reorder choices only if the TSMN is not sure about its decision.

E. Postprocessing

Until this point, no semantic information is considered during the stages of CR. It is well known that humans read by context up to 60%

for careless handwriting. While preprocessing tries to "clean" the document in a certain sense, it may remove important information, since the context information is not available at this stage. The lack of context information during the segmentation stage may cause even more severe and irreversible errors since it yields meaningless segmentation boundaries. It is clear that if the semantic information were available to a certain extent, it would contribute a lot to the accuracy of the CR stages. On the other hand, the entire CR problem is for determining the context of the document image. Therefore, utilization of the context information in the CR problem creates a chicken and egg problem. The review of the recent CR research indicates minor improvements when only shape recognition of the character is considered. Therefore, the incorporation of context and shape information in all the stages of CR systems is necessary for meaningful improvements in recognition rates. This is done in the postprocessing stage with a feedback to the early stages of CR.

The simplest way of incorporating the context information is the utilization of a dictionary for correcting the minor mistakes of the CR systems. The basic idea is to spell check the CR output and provide some alternatives for the outputs of the recognizer that do not take place in the dictionary [16]. Spelling checkers are available in some languages, like English, German, and French, etc. String matching algorithms can be used to rank the lexicon words using a distance metric that represents various edition costs [91]. Statistical information derived from the training data and the syntactic knowledge such as N-grams improves the performance of the matching process [60], [81]. In some applications, the context information confirms the recognition results of the different parts in the document image. In automatic reading of bank checks, the inconsistencies between the legal and the courtesy amount can be detected, and the recognition errors can be potentially corrected [80].

However, the contextual postprocessing suffers from the drawback of making unrecoverable OCR decisions. In addition to the use of a dictionary, a well-developed lexicon and a set of orthographic rules contribute a great deal to the recognition rates in word and sentence levels. In word level, lexicon-driven matching approaches avoid making unrecoverable decisions at the postprocessing stage by bringing the context information earlier in the segmentation and recognition stages. A lexicon of words with a knowledge base is used during or after the recognition stage for verification and improvement purpose. A common technique for lexicon driven recognition is to represent the word image by a segmentation graph and match each entry in the lexicon against this graph [8], [30], [84]. In this approach, the dynamic programming technique is often used to rank every word in the lexicon. The word with the highest rank is chosen as the recognition hypothesis. In order to speed up the search process, the lexicon can be represented by a tree data structure or by hash tables [29], [37]. Fig. 14 shows an example of segmentation result, corresponding segmentation graph, and the representation of the words in the lexicon by a tree structure.

In sentence level, the resulting sentences obtained from the output of the recognition stage can be further processed through parsing in the postprocessing stage to increase the recognition rates [56], [85]. The recognition choices produced by a word recognizer can be represented by a graph as in the case of word level recognition. Then, the grammatically correct paths over this graph are determined by using syntactic knowledge. However, postprocessing in sentence level is rather in its infancy, especially for languages other than English, since it requires extensive research in linguistics and formalism in the field of AI.

IV. Discussion

In this study, we have overviewed the main approaches used in the CR field. Our attempt was to bring out the present status of CR research. Although each of the methods summarized above have their own superiorities and drawbacks, the presented recognition results of different methods seem very successful. Most of the recognition accuracy rates reported are over 85%. However, it is very difficult to make a judgment about the success of the results of recognition methods, especially in terms of recognition rates, because of different databases, constraints, and sample spaces. In spite of all the intensive research effort, numerous journal articles, conference proceedings, and patents, none of the proposed methods solve the CR problem out of the laboratory environment without putting constraints. The answer to the question "Where are we standing now?" is summarized in Table II.

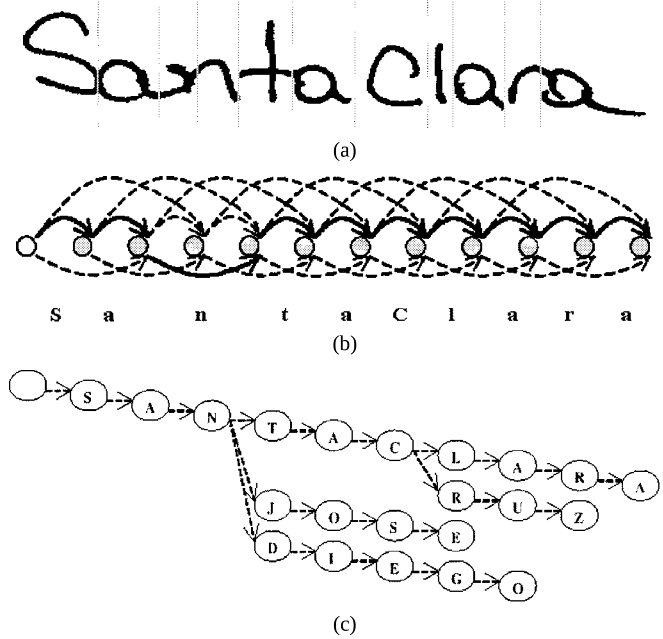
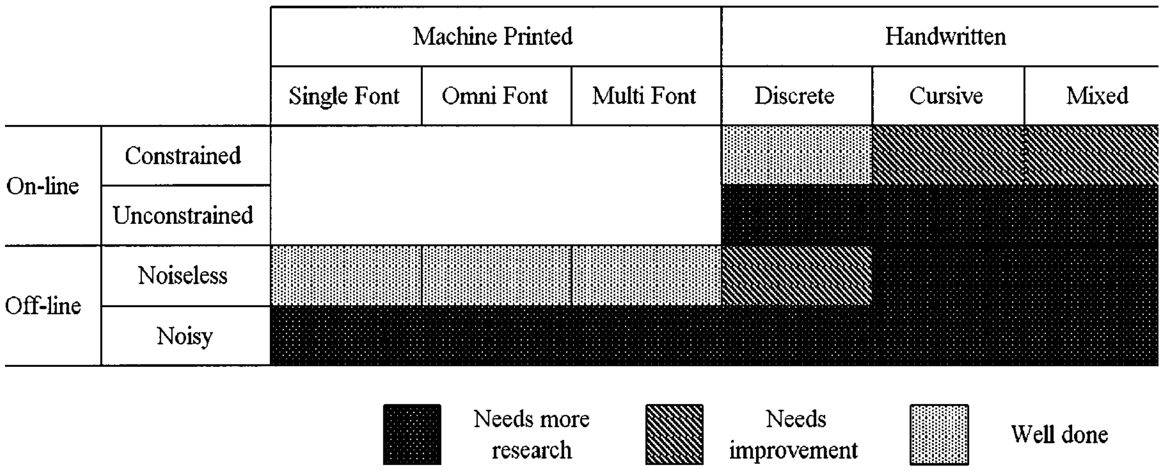


Fig. 14. Lexicon driven matching. (a) Segmentation result. (b) Corresponding segmentation graph. (c) Representation of the lexicon words by a tree structure in [32].

For texts which are handwritten under poor conditions or for free-style handwriting, there is still an intensive need in almost all the stages of CR research. The proposed methods shift toward the hybrid architectures, which utilize, basically, HMM together with the support of structural and statistical techniques. The discrete and clean handwriting on high-quality paper or on tablet can be recognized with a rate above 85%, when there is a limited vocabulary. A popular application area is number digit or limited vocabulary form (bank checks, envelopes, and forms designed for specific applications) recognition. These applications may require intensive preprocessing, for skew detection, thinning, and baseline extraction purposes. Most of the systems use either NNs or HMMs in the recognition stage. Finally, the output of a laser-quality printing device and neat discrete handwriting can be recognized with a rate above 95% (vertical areas). Two leading commercial products Omni-Page Pro and Recognita can read complex pages printed on high-quality papers, containing the mixture of fonts, without user training.

The best OCR packages in the market use combined techniques based on NNs for machine-printed characters. Although the input is clean, they require sophisticated preprocessing techniques for noise reduction and normalization. External segmentation is the major task for page layout decomposition. Projection profile is sufficient for character segmentation. Language analysis is used for final confirmation of the recognition stage, providing alternatives in case of a mismatch. A few of the systems, such as Recognita Plus, work on Greek and Cyrillic in addition to the Latin alphabet for limited

TABLE II
CURRENT STUDIES IN CR STUDIES



vocabulary applications. OCR for alphabets other than Latin and for languages other than English, French, and German remains mostly in the research arena, even for Chinese and Japanese, which have some commercial products.

A number of weaknesses, which exist in the proposed systems, can be summarized as follows.

1) In all of the proposed methods, the studies on the stages of CR have come to a point where the improvements are marginal with the current research directions. The stages are mostly based on the shape extracting and recognition techniques and ignore the semantic information. Incorporation of the semantic information is not well explored. In most cases, it is too late to correct all the errors, which propagates through the stages of the CR, in the postprocessing stage. This situation implies the need of a global change in the approaches for freestyle handwriting.
2) A major difficulty lies behind the lack of the noise model, over all the stages. Therefore, many assumptions and parameters of the algorithms are set by trial and error at the initial phase. Unless there are strict constraints about the data and the application domain, the assumptions are not valid even for small changes out of the laboratory environment.
3) Handwriting generation involves semantic, syntactic, and lexical information, which is converted into a set of symbols to generate the pen-tip trajectory from a predefined alphabet. The available techniques suffer from the lack of characterizing the handwriting generation and the perceptual process in reading, which consists of many complicated phenomena. For example, none of the proposed systems take into account contextual anticipatory phenomena, which lead to co-articulations and the context effects on the writing [141]. The sequence of cursive letters is not produced in a serial manner, but parallel articulatory activity occurs. The production of a character is thus affected by the production of the surrounding characters and thus the context.
4) In most of the methods, recognition is isolated from training. The large amount of data is collected and used to train the classifier prior to the classification. Therefore, it is not easy to improve the recognition rate using the knowledge obtained from the analysis of the recognition errors.
5) Selection of the type and the number of features is done by heuristics. It is well known that design of the feature space depends on the training and recognition method. On the other hand, the performance of the recognizer highly depends on the selected features. This problem cannot be solved without evaluating the system with respect to the feature space and the recognition scheme. There exist no evaluation tools for measuring the performance of the stages as well as the overall performance of the system, indicating the source of errors for further improvements.
6) Although color scanners and tablets enable data acquisition with high resolution, there is always a tradeoff between the data acquisition quality and complexity of the algorithms, which limits the recognition rate.

Let us evaluate the methods according to the information utilization.

1) Neither the structural nor the statistical information can represent a complex pattern alone. Therefore, one needs to combine statistical and structural information supported by the semantic information.
2) NNs or HMMs are very successful in combining statistical and structural information for many pattern recognition problems. They are comparatively resistant to distortions, but they have a discrete nature in the matching process, which may cause drastic mismatching. In other words, they are flexible in training and recognizing samples in classes of reasonably large within-class variances, but they are not continuous in representing the between-class variances. The classes are separate with their own samples and have models without any relations among them.
3) Template matching methods deal with a character as a whole in the sense that an input plane is matched against a template constrained on and X–Y plane. This makes the procedure very simple, and the complexity of character shape is irrelevant, but, it suffers from the sensitivity to noise and is not adaptive to variations in writing style. The capabilities of human reasoning are better captured by flexible matching techniques than by direct matching. Some features are tolerant to distortion and take care of style variations, rotation, and translation to some extent.
4) The characters are natural entities. They require complex mathematical descriptions to obey the mathematical constraint set of the formal language theory. Imposing a strict mathematical rule on the pattern structure is not particularly practical in CR, where intraclass variations are very large.

The following are some suggestions for future research directions.

1) Five stages of the CR given in this study reached a plateau, namely, that no more improvement could be achieved by using the current image processing and pattern recognition methodolo-

gies. In order to improve the overall performance, higher abstraction levels are to be used for modeling and recognizing the handwriting. This is possible by adopting the AI methodologies to the CR field. The stages should be governed by a set of advanced shells utilizing the decision-making, expert systems, and knowledge-base tools, which incorporate semantic and linguistic information. These tools should provide various levels of abstractions for characters, words, and sentences.

2) For a real-life CR problem, we need techniques to describe a large number of similar structures of the same category while allowing distinct descriptions among categorically different patterns. It seems that a combined CR model is the only solution to practical CR problems.
3) HMM is very suitable for modeling the linguistic information as well as the shape information. The best way of approaching the CR problem is to combine the HMMs at various levels for shape recognition and generating grammars for words and sentences. The HMM classifiers of the hybrid CR system should be gathered by an intelligent tool for maximizing the recognition rate. This may be possible if one achieves integration of the top–down linguistic models and bottom–up CR schemes by combining various HMMs, which are related to each other through this tool. The tool may consist of a flexible high-level NN architecture, which provides continuous description of the classes.
4) Design of the training set should be handled systematically, rather than putting up the available data in the set. Training sets should have considerable size and contain random samples, including poorly written ones. Decision of the optimal number of samples from each class which are the samples that minimize within class variance and maximize between class variance, should be selected according to a set of cost functions. The statistics, relating to the factors such as character shape, slant, and stroke variations and the occurrences of characters in a language and interrelations of characters, should be gathered.
5) The feature set should represent the models for handwriting generation and perception, which is reflected to the structural and statistical properties properly so that they are sensitive to intraclass variances and insensitive to interclass variances.
6) Training and recognition processes should use the same knowledge base. This knowledge base should be built incrementally with several analysis levels by a cooperative effort from the human expert and the computer. Methodologies should be developed to combine knowledge from different sources and to discriminate between correct and incorrect recognition. Evaluation tools should be developed to measure and improve the results of the CR system.
7) The methods on the classical shape-based recognition techniques have reached a plateau for the maximum recognition rate, which is lower than the practical needs. Generally speaking, future research should focus on the linguistic and contextual information for further improvements.

In conclusion, a lot of effort has been spent in the research on CR. Some major improvements have been obtained. Can the machines read human writing with the same fluency as humans? Not yet.

References

[1] I. Abulhaiba and P. Ahmed, "A fuzzy graph theoretic approach to recognize the totally unconstrained handwritten numerals," *Pattern Recognit.*, vol. 26, no. 9, pp. 1335–1350, 1993.

[2] O. Akindele and A. Belaid, "Page segmentation by segment tracing," in *Proc. 2nd Int. Conf. Document Anal. Recognit.*, Japan, 1993, pp. 314–344.

[3] E. Alpaydin, "Techniques for combining multiple learners," in *Proc. Eng. Intell.*, 1998, vol. 2, pp. 6–12.

[4] C. Arcelli, "A one pass two operation process to detect the skeletal pixels on the 4-distance transform," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 411–414, Apr. 1989.

[5] N. Arica and F. T. Yarman-Vural, "One dimensional representation of two dimensional information for HMM based handwritten recognition," *Pattern Recognit. Lett.*, vol. 21, no. 6–7, pp. 583–592, 2000.

[6] ——, "A new scheme for off-line handwritten connected digit recognition," in *Proc. 14th Int. Conf. Pattern Recognit.*, Brisbane, Australia, 1998, pp. 1127–1131.

[7] A. Atici and F. Yarman-Vural, "A heuristic method for Arabic character recognition," *Signal Process.*, vol. 62, pp. 87–99, 1997.

[8] E. Augustin *et al.*, "Legal amount recognition on French bank checks using a neural network—Hidden Markov model hybrid," in *Proc. 6th Int. Workshop Frontiers Handwriting Recognit.*, Korea, 1998.

[9] H. I. Avi-Itzhak, T. A. Diep, and H. Gartland, "High accuracy optical character recognition using neural network with centroid dithering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 218–228, Feb. 1995.

[10] H. S. Baird, "Document image defect models," in *Proc. Int. Workshop Syntactical Structural Pattern Recognit.*, 1990, pp. 38–47.

[11] O. Baruch, "Line thinning by line following," *Pattern Recognit. Lett.*, vol. 8, no. 4, pp. 271–276, 1989.

[12] I. Bazzi, R. Schwartz, and J. Makhoul, "An omnifont open vocabulary OCR system for English and Arabic," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 495–504, June 1999.

[13] A. Belaid and J. P. Haton, "A syntactic approach for handwritten mathematical formula recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 105–111, 1984.

[14] S. O. Belkasim, M. Shridhar, and M. Ahmadi, "Pattern recognition with moment invariants: A comparative survey," *Pattern Recognit.*, vol. 24, no. 12, pp. 1117–1138, 1991.

[15] H. D. Block, B. W. Knight, and F. Rosenblatt, "Analysis of a four layer serious coupled perceptron," *Rev. Mod. Phys.*, vol. 34, pp. 135–152, 1962.

[16] M. Bokser, "Omnifont technologies," *Proc. IEEE*, vol. 80, pp. 1066–1078, 1992.

[17] D. Bouchaffra, V. Govindaraju, and S. N. Srihari, "Postprocessing of recognized strings using nonstationary Markovian models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 990–999, Oct. 1999.

[18] R. M. Bozinovic and S. N. Srihari, "Off-line cursive script word recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 68–83, Jan. 1989.

[19] M. K. Brown and S. Ganapathy, "Preprocessing techniques for cursive script word recognition," *Pattern Recognit.*, vol. 16, no. 5, pp. 447–458, 1983.

[20] C. J. C. Burges, J. I. Be, and C. R. Nohl, "Recognition of handwritten cursive postal words using neural networks," in *Proc. U. S. Postal Service 5th Adv. Technol. Conf.*, 1992, pp. 117–129.

[21] M. Cannon, J. Hockberg, and P. Kelly, "Quality assessment and restoration of typewritten document images," *Int. J. Document Anal. Recognit.*, vol. 2, no. 2/3, pp. 80–89, 1999.

[22] R. G. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 690–706, July 1996.

[23] R. G. Casey and G. Nagy, "Recursive segmentation and classification of composite patterns," in *Proc. 6th Int. Conf. Pattern Recognit.*, München, Germany, 1982, pp. 1023–1031.

[24] K. R. Castleman, *Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[25] F. Cesarini, M. Gori, S. Mariani, and G. Soda, "Structured document segmentation and representation by the modified X–Y tree," in *Proc. 5th Int. Conf. Document Anal. Recognit.*, Bangalore, India, 1999, pp. 563–566.

[26] M. Y. Chen, A. Kundu, and J. Zhou, "Off-line handwritten word recognition using a hidden Markov model type stochastic network," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 481–496, May 1994.

[27] M. Y. Chen, A. Kundu, and S. N. Srihari, "Variable duration hidden Markov model and morphological segmentation for handwritten word recognition," *IEEE Trans. Image Processing*, vol. 4, pp. 1675–1688, Dec. 1995.

[28] C. Chen and J. DeCurtins, "Word recognition in a segmentation-free approach to OCR," in *Proc. 2nd Int. Conf. Document Anal. Recognit.*, 1993, pp. 573–579.

[29] D. Y. Chen, J. Mao, and K. Mohiuddin, "An efficient algorithm for matching a lexicon with a segmentation graph," in *Proc. 5th Int. Conf. Document Anal. Recognit.*, Bangalore, India, 1999, pp. 543–546.

[30] W. T. Chen, P. Gader, and H. Shi, "Lexicon-driven handwritten word recognition using optimal linear combination of order statistics," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 77–82, Jan. 1999.
[31] M. Chen and X. Ding, "A robust skew detection algorithm for gray-scale document image," in *Proc. 5th Int. Conf. Document Anal. Recognit.*, Bangalore, India, 1999, pp. 617–620.
[32] F. H. Cheng, W. H. Hsu, and C. A. Chen, "Fuzzy approach to solve the recognition problem of handwritten Chinese characters," *Pattern Recognit.*, vol. 22, no. 2, pp. 133–141, 1989.
[33] H. Cheng, W. H. Hsu, and M. C. Kuo, "Recognition of handprinted Chinese characters via stroke relaxation," *Pattern Recognit.*, vol. 26, no. 4, pp. 579–593, 1993.
[34] Y. C. Chim, A. A. Kassim, and Y. Ibrahim, "Character recognition using statistical moments," *Image Vis. Comput.*, vol. 17, pp. 299–307, 1999.
[35] K. Chung and J. Yoon, "Performance comparison of several feature selection methods based on node pruning in handwritten character recognition," in *Proc. 4th Int. Conf. Document Anal. Recognit.*, Ulm, Germany, 1997, pp. 11–15.
[36] M. Cote *et al.*, "Reading of cursive scripts using a reading model and perceptual concepts, the PERCEPTO system," *Int. J. Document Anal. Recognit.*, vol. 1, no. 1, pp. 3–17, 1998.
[37] A. Dengel *et al.*, "Techniques for improving OCR results," in *Handbook of Character Recognition and Document Image Analysis*, H. Bunke and P. S. P. Wang, Eds. Singapore: World Scientific, 1997, pp. 227–254.
[38] P. A. Devijer and J. Kittler, *Pattern Recognition: A Statistical Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
[39] D. Doermann, "Page decomposition and related research," in *Proc. Symp. Document Image Understanding Technol.*, Bowie, MD, 1995, pp. 39–55.
[40] C. Downtown and C. G. Leedham, "Preprocessing and presorting of envelope images for automatic sorting using OCR," *Pattern Recognit.*, vol. 23, no. 3–4, pp. 347–362, 1990.
[41] H. Drucker, R. Schapire, and P. Simard, "Improving performance in neural networks using a boosting algorithm," in *Advances in NIPS*, S. J. Hanson, J. Cowan, and L. Giles, Eds. San Mateo, CA: Morgan Kaufmann, 1993, pp. 42–49.
[42] L. D. Earnest, "Machine reading of cursive script," in *Proc. IFIP Congr.*, Amsterdam, The Netherlands, 1963, pp. 462–466.
[43] El-Sheikh and R. M. Guindi, "Computer recognition of Arabic cursive scripts," *Pattern Recognit.*, vol. 21, no. 4, pp. 293–302, 1988.
[44] K. Fukushima and N. Wake, "Handwritten alphanumeric character recognition by neocognition," *IEEE Trans. Neural Networks*, vol. 2, pp. 355–365, June 1991.
[45] K. Fukushima and T. Imagawa, "Recognition and segmentation of connected characters with selective attention," *Neural Networks*, vol. 6, no. 1, pp. 33–41, 1993.
[46] P. D. Gader *et al.*, "Recognition of handwritten digits using template and model matching," *Pattern Recognit.*, vol. 24, no. 5, pp. 421–431, 1991.
[47] P. D. Gader and M. A. Khabou, "Automatic feature generation for handwritten digit recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 1256–1262, Dec. 1996.
[48] P. D. Gader, M. A. Mohamed, and J. M. Keller, "Dynamic programming based handwritten word recognition using the Choquet fuzzy integral as the match function," *J. Electron. Imaging*, vol. 5, no. 1, pp. 15–24, 1996.
[49] M. D. Garris *et al.*, "NIST form based handprinted recognition system," Nat. Inst. Standard Technol., Gaithersburg, MD, Tech. Rep. NISTIR 546 920 899, July 1994.
[50] A. Gillies, "Cursive word recognition using hidden Markov models," in *Proc. U.S. Postal Service Adv. Technol. Conf.*, Washington, DC, 1992, pp. 557–563.
[51] E. Gilloux, "Hidden Markov models in handwriting recognition," in *Fundamentals in Handwriting Recognition*. ser. NATO ASI F, S. Impedovo, Ed. New York: Springer-Verlag, 1994, vol. 124.
[52] S. Gopisetty *et al.*, "Automated forms-processing software and services," *IBM J. Res. Develop.*, vol. 40, no. 2, pp. 211–230, 1996.
[53] V. K. Govindan and A. P. Shivaprasad, "A pattern adaptive thinning algorithm," *Pattern Recognit.*, vol. 20, no. 6, pp. 623–637, 1987.
[54] ——, "Character recognition—A review," *Pattern Recognit.*, vol. 23, no. 7, pp. 671–683, 1990.
[55] W. Guerfaii and R. Plamondon, "Normalizing and restoring on-line handwriting," *Pattern Recognit.*, vol. 26, no. 3, pp. 418–431, 1993.
[56] D. Guillevic and C. Y. Suen, "Cursive script recognition: A sentence level recognition scheme," in *Proc. Int. Workshop Frontiers in Handwriting Recognit.*, 1994, pp. 216–223.
[57] ——, "HMM–KNN word recognition engine for bank cheque processing," in *Proc. 14th Int. Conf. Pattern Recognit.*, Brisbane, Australia, 1998, pp. 1526–1529.
[58] ——, "Recognition of legal amounts on bank cheques," *Pattern Anal. Applicat.*, vol. 1, no. 1, pp. 28–41, 1998.
[59] Z. Guo and R. W. Hall, "Fast fully parallel thinning algorithms," *CVGIP, Image Understanding*, vol. 55, no. 3, pp. 317–328, 1992.
[60] I. Guyon and F. Pereira, "Design of a linguistic postprocessor using variable memory length Markov models," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, Montreal, QC, Canada, 1995, pp. 454–457.
[61] J. Ha, R. Haralick, and I. Philips, "Document page decomposition by bounding-box projection technique," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, Montreal, QC, Canada, 1995, pp. 119–122.
[62] Y. Hamamoto *et al.*, "Recognition of handprinted Chinese characters using Gabor features," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, Montreal, QC, Canada, 1995, pp. 819–823.
[63] L. D. Harmon, "Automatic recognition of print and scripts," *Proc. IEEE*, vol. 60, pp. 1165–1177, 1972.
[64] A. Hashizume, P. S. Yeh, and A. Rosenfeld, "A method of detecting the orientation of aligned components," *Pattern Recognit. Lett.*, vol. 4, pp. 125–132, 1986.
[65] K. C. Hayes, "Reading handwritten words using hierarchical relaxation," *Comput. Graph. Image Process.*, vol. 14, pp. 344–364, 1980.
[66] R. B. Hennis, "The IBM 1975 optical page reader, system design," *IBM J. Res. Develop.*, pp. 346–353, 1968.
[67] J. Hu, S. G. Lim, and M. K. Brown, "Writer independent on-line handwriting recognition using an HMM approach," *Pattern Recognit.*, vol. 33, no. 1, pp. 133–147, 2000.
[68] J. Hu and T. Pavlidis, "A hierarchical approach to efficient curvilinear object searching," *Comput. Vis. Image Understanding*, vol. 63, no. 2, pp. 208–220, 1996.
[69] Y. S. Huang and C. Y. Suen, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 90–94, Jan. 1995.
[70] B. Hussain and M. Kabuka, "A novel feature recognition neural network and its application to character recognition," *IEEE Trans Pattern Anal. Machine Intell.*, vol. 16, pp. 99–106, Jan. 1994.
[71] R. A. Jacobs *et al.*, "Adaptive mixtures of local experts," *Neural Comput.*, vol. 3, pp. 79–87, 1991.
[72] A. K. Jain and B. Yu, "Document representation and its application to page decomposition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 294–308, Mar. 1998.
[73] A. K. Jain and D. Zongker, "Representation and recognition of handwritten digits using deformable templates," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 1386–1391, Dec. 1997.
[74] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *IEEE Comput.*, pp. 31–44, Mar. 1996.
[75] A. K. Jain and K. Karu, "Page segmentation using texture analysis," *Pattern Recognit.*, vol. 29, pp. 743–770, 1996.
[76] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 4–38, Jan. 2000.
[77] B. H. Huang and L. R. Rabiner, "The segmental K-means algorithm for estimating parameters of hidden Markov models," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 1639–1641, Sept. 1990.
[78] J. Kanai and A. D. Bagdanov, "Projection profile based skew estimation algorithm for JPIG compressed images," *Int. J. Document Anal. Recognit.*, vol. 1, no. 1, pp. 43–51, 1998.
[79] H. J. Kang and S. W. Lee, "Combining classifiers based on minimization of a Bayes error rates," in *Proc. 5th Int. Conf. Document Anal. Recognit.*, Bangalore, India, 1999, pp. 398–401.
[80] G. Kaufmann and H. Bunke, "Detection and correction of recognition errors in check reading," *Int. J. Document Anal. Recognit.*, vol. 2, no. 4, pp. 211–221, 2000.
[81] F. G. Keenan, L. J. Evett, and R. J. Whitrow, "A large vocabulary stochastic syntax analyzer for handwriting recognition," in *Proc. 1st Int. Conf. Document Anal. Recognit.*, 1991, pp. 794–802.
[82] H. Y. Kim and J. H. Kim, "Handwritten Korean character recognition based on hierarchical random graph modeling," in *Proc. Int. Workshop Frontiers Handwriting Recognit.*, 1998, pp. 577–586.
[83] W. Y. Kim and P. Yuan, "A practical pattern recognition system for translation, scale and rotation invariance," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, June 1994.
[84] G. Kim and V. Govindaraju, "A lexicon driven approach to handwritten word recognition for real-time applications," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 366–379, Apr. 1997.
[85] G. Kim, V. Govindaraju, and S. Srihari, "Architecture for handwritten text recognition systems," in *Proc. Int. Workshop Frontiers Handwriting Recognit.*, 1998, pp. 113–122.

[86] K. Kise, O. Yanagida, and S. Takamatsu, "Page segmentation based on thinning of background," in *Proc 13th Int. Conf. Pattern Recognit.*, Vienna, Austria, 1996, pp. 788–792.

[87] J. Kitler, "Feature selection and extraction," in *Handbook of Pattern Recognition and Image Processing*, T. Young and K. Fu, Eds. New York: Academic, 1986, pp. 59–83.

[88] T. Kohonen, *Self Organizing Maps*. New York: Springer, 1995, vol. 30.

[89] A. Kornai, K. M. Mohiuddin, and S. D. Connell, "An HMM-based legal amount field OCR system for checks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern., Intell. Syst. 21st Century*, vol. 3, 1995, pp. 2800–2805.

[90] ——, "Recognition of cursive writing on personal checks," in *Proc. Int. Workshop Frontiers Handwriting Recognit.*, Essex, U.K., 1996, pp. 373–378.

[91] K. Kukich, "Techniques for automatically correcting words in text," *ACM Comput. Surv.*, vol. 24, no. 4, pp. 377–439, 1992.

[92] T. Kunango, R. Haralick, and I. Phillips, "Nonlinear local and global document degradation models," *Int. J. Imaging Syst. Technol.*, vol. 5, no. 4, pp. 274–282, 1994.

[93] A. Kundu, Y. He, and M. Y. Chen, "Alternatives to variable duration HMM in handwriting recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 1275–1280, Nov. 1998.

[94] A. Kundu and Y. He, "On optimal order in modeling sequence of letters in words of common language as a Markov chain," *Pattern Recognit.*, vol. 24, no. 7, pp. 603–608, 1991.

[95] L. Lam and C. Y. Suen, "Increasing experts for majority vote in OCR: Theoretical considerations and strategies," in *Proc. Int. Workshop Frontiers Handwriting Recognit.*, 1994, pp. 245–254.

[96] ——, "An evaluation of parallel thinning algorithms, for character recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 914–919, Sept. 1995.

[97] L. Lam, S. W. Lee, and C. Y. Suen, "Thinning methodologies—A comprehensive survey," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 869–885, Sept. 1992.

[98] E. Lecolinet, "A new model for context driven word recognition," in *Proc. SDAIR*, 1993, pp. 135–139.

[99] E. Lecolinet and J.-P. Crettez, "A grapheme based segmentation technique for cursive script recognition," in *Proc. 1st Int. Conf. Document Anal. Recognit.*, Saint-Malo, France, 1991, pp. 740–744.

[100] S. W. Lee, D. J. Lee, and H. S. Park, "A new methodology for gray-scale character segmentation and recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 1045–1050, Oct. 1996.

[101] W. L. Lee and K.-C. Fan, "Document image preprocessing based on optimal Boolean filters," *Signal Process.*, vol. 80, no. 1, pp. 45–55, 2000.

[102] S. W. Lee and Y. J. Kim, "Direct extraction of topographic features for gray-scale character recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 724–729, July 1995.

[103] ——, "Multiresolutional recognition of handwritten numerals with wavelet transform and multilayer cluster neural network," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, Montreal, QC, Canada, 1995, pp. 1010–1014.

[104] R. Legault and C. Y. Suen, "Optimal local weighted averaging methods in contour smoothing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 690–706, July 1997.

[105] J. G. Leu, "Edge sharpening through ramp width reduction," *Image Vis. Comput.*, vol. 18, no. 6–7, pp. 501–514, 2000.

[106] X. Li, W. Oh, J. Hong, and W. Gao, "Recognizing components of handwritten characters by attributed relational graphs with stable features," in *Proc. 4th Int. Conf. Document Anal. Recognit.*, 1997, pp. 616–620.

[107] C. Y. Liou and H. C. Yang, "Handprinted character recognition based on spatial topology distance measurements," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 941–945, Sept. 1996.

[108] J. Liu, Y. Tang, Q. He, and C. Suen, "Adaptive document segmentation and geometric relation labeling: Algorithms and experimental results," in *Proc 13th Int. Conf. Pattern Recognit.*, Vienna, Austria, 1996, pp. 763–767.

[109] W. Lu, Y. Ren, and C. Y. Suen, "Hierarchical attributed graph representation and recognition of handwritten chinese characters," *Pattern Recognit.*, vol. 24, no. 7, pp. 617–632, 1991.

[110] S. Madhvanath, E. Kleinberg, V. Govindaraju, and S. N. Srihari, "The HOVER system for rapid holistic verification of off-line handwritten phrases," in *Proc. 4th Int. Conf. Document Anal. Recognit.*, Ulm, Germany, 1997, pp. 855–890.

[111] S. Madhvanath, G. Kim, and V. Govindaraju, "Chaincode contour processing for handwritten word recognition," *IEEE Pattern. Anal. Machine Intell.*, vol. 21, pp. 928–932, Sept. 1999.

[112] S. Madhvanath and V. Govindaraju, "Reference lines for holistic recognition of handwritten words," *Pattern Recognit.*, vol. 32, no. 12, pp. 2021–2028, 1999.

[113] S. A. Mahmoud, I. Abuhabia, and R. J. Green, "Skeletonization of Arabic characters using clustering based skeletonization algorithm," *Pattern Recognit.*, vol. 24, no. 5, pp. 453–464, 1991.

[114] J. Mantas, "An overview of character recognition methodologies," *Pattern Recognit.*, vol. 19, no. 6, pp. 425–430, 1986.

[115] J. Mao, K. M. Mohiuddin, and T. Fujisaki, "A two stage multi-network OCR system with a soft pre-classifier and a network selector," in *Proc. 3rd Int. Conf. Document Analysis and Recognit.*, Montreal, QC, Canada, 1995, pp. 821–828.

[116] A. Meyer, "Pen computing: A technology overview and a vision," *SIGCHI Bull.*, vol. 27, no. 3, pp. 46–90, 1995.

[117] M. L. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA: MIT Press, 1969.

[118] K. T. Miura, R. Sato, and S. Mori, "A method of extracting curvature features and its application to handwritten character recognition," in *Proc. 4th Int. Conf. Document Anal. Recognit.*, Ulm, Germany, 1997, pp. 450–454.

[119] S. Mo and V. J. Mathews, "Adaptive, quadratic preprocessing of document images for binarization," *IEEE Trans. Image Processing*, vol. 7, pp. 992–999, July 1998.

[120] M. A. Mohamed and P. Gader, "Generalized hidden Markov models—Part II: Application to handwritten word recognition," *IEEE Trans. Fuzzy Syst.*, vol. 8, pp. 82–95, Feb. 2000.

[121] ——, "Handwritten word recognition using segmentation-free hidden Markov modeling and segmentation based dynamic programming techniques," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 548–554, May 1996.

[122] K. M. Mohiuddin and J. Mao, "A comparative study of different classifiers for handprinted character recognition," *Pattern Recognit. Practice IV*, pp. 437–448, 1994.

[123] S. Mori, C. Y. Suen, and K. Yamamoto, "Historical review of OCR research and development," *Proc. IEEE*, vol. 80, pp. 1029–1057, July 1992.

[124] S. Mori, H. Nishida, and H. Yamada, *Optical Character Recognition*. New York: Wiley, 1999.

[125] S. Mori, K. Yamamoto, and M. Yasuda, "Research on machine recognition of handprinted characters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 386–404, Apr. 1984.

[126] M. Nakagava, T. Oguni, and A. Homma, "A coarse classification of on-line handwritten characters," in *Proc. 5th Int. Workshop Frontiers Handwriting Recognit.*, Essex, U.K., 1996, pp. 417–420.

[127] M. Nakagawa and K. Akiyama, "A linear-time elastic matching for stroke number free recognition of on-line handwritten characters," in *Proc. Int. Workshop Frontiers Handwriting Recognit.*, Taiwan, R.O.C., 1994, pp. 48–56.

[128] W. Niblack, *An Introduction to Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[129] L. O'Gorman, "The document spectrum for page layout analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 162–173, 1993.

[130] I. S. Oh and C. Y. Suen, "Distance features for neural network-based recognition of handwritten characters," *Int. J. Document Anal. Recognit.*, vol. 1, no. 2, pp. 73–88, 1998.

[131] I. S. Oh, J. S. Lee, and C. Y. Suen, "Analysis of class separation and combination of class-dependent features for handwriting recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 1089–1094, Oct. 1999.

[132] I. S. Oh *et al.*, "Class-expert approach to unconstrained handwritten numeral recognition," in *Proc. 5th Int. Workshop Frontiers Handwriting Recognit.*, Essex, U.K., 1996, pp. 95–102.

[133] M. Okamoto and K. Yamamoto, "On-line handwriting character recognition method with directional features and direction change features," in *Proc. 4th Int. Conf. Document Anal. Recognit.*, Ulm, Germany, 1997, pp. 926–930.

[134] E. Oztop *et al.*, "Repulsive attractive network for baseline extraction on document images," *Signal Process.*, vol. 74, no. 1, 1999.

[135] M. Parizeau and R. Plamondon, "A fuzzy-syntactic approach to allograph modeling for cursive script recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 702–712, July 1995.

[136] J. Park, V. Govindaraju, and S. N. Srihari, "OCR in a hierarchical feature space," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 400–407, Apr. 2000.

[137] T. Pavlidis, "Recognition of printed text under realistic conditions," *Pattern Recognit. Lett.*, p. 326, 1993.

[138] T. Pavlidis and J. Zhou, "Page segmentation by white streams," in *Proc. 1st Int. Conf. Document Anal. Recognit.*, Saint-Malo, France, 1991, pp. 945–953.

[139] L. F. C. Pessoa and P. Maragos, "Neural networks with hybrid morphological/rank/linear nodes: A unifying framework with applications to handwritten character recognition," *Pattern Recognit.*, vol. 33, pp. 945–960, 2000.

[140] R. Plamandon *et al.*, "On-line handwriting recognition," in *Encyclopedia of Electrical and Electronics Engineering*, J. G. Webster *et al.*, Eds. New York: Wiley, 1999, vol. 15, pp. 123–146.

[141] R. Plamandon and S. N. Srihari, "On-line and off-line handwriting recognition: A comprehensive survey," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 63–85, Jan. 2000.

[142] A. Polesel, G. Ramponi, and V. Matthews, "Adaptive unsharp masking for contrast enhancement," in *Proc. Int. Conf. Image Process.*, vol. 1, 1997, pp. 267–271.

[143] L. Prevost and M. Milgram, "Automatic allograph selection and multiple expert classification for totally unconstrained handwritten character recognition," in *Proc. 14th Int. Conf. Pattern Recognit.*, Brisbane, Australia, 1998, pp. 381–383.

[144] K. E. Price, "Relaxation matching techniques comparison," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, pp. 617–623, May 1985.

[145] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–286, 1989.

[146] N. V. S. Reddy and P. Nagabhushan, "A three dimensional neural network model for unconstrained handwritten numeral recognition: A new approach," *Pattern Recognit.*, vol. 31, no. 5, pp. 511–516, 1998.

[147] J. M. Reinhardt and W. E. Higgins, "Comparison between the morphological skeleton and morphological shape decomposition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 951–957, Sept. 1996.

[148] B. Ripley, "Statistical aspects of neural networks," in *Networks on Chaos: Statistical and Probabilistic Aspects* , U. Bornndorff-Nielsen, J. Jensen, and W. Kendal, Eds. London, U.K.: Chapman & Hall, 1993.

[149] J. Rocha and T. Pavlidis, "A shape analysis model," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 394–404, Apr. 1994.

[150] ——, "Character recognition without segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 903–909, Sept. 1995.

[151] J. Saula and M. Pietikainen, "Adaptive document image binarization," *Pattern Recognit.*, vol. 33, no. 2, pp. 225–236, 2000.

[152] A. W. Senior and A. J. Robinson, "An off-line cursive handwriting recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 309–322, Mar. 1998.

[153] J. Serra, "Morphological filtering: An overview," *Signal Process.*, vol. 38, no. 1, pp. 3–11, 1994.

[154] T. Shioyama, H. Y. Wu, and T. Nojima, "Recognition algorithm based on wavelet transform for handprinted Chinese characters," in *Proc. 14th Int. Conf. Pattern Recognit.*, vol. 1, 1998, pp. 229–232.

[155] M. Shridhar and A. Badreldin, "High accuracy syntactic recognition algorithm for handwritten numerals," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, pp. 152–158, Jan. 1985.

[156] M. Shridhar and F. Kimura, "Handwritten numerical recognition based on multiple algorithms," *Pattern Recognit.*, vol. 24, no. 10, pp. 969–983, 1991.

[157] J. C. Simon, "Off-line cursive word recognition," *Proc. IEEE*, vol. 80, p. 1150, July 1992.

[158] R. M. K. Sinha, B. Prasada, G. F. Houle, and M. Babourin, "Hybrid contextual text recognition with string matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 915–925, Sept. 1993.

[159] S. Smith *et al.*, "Handwritten character classification using nearest neighbor in large databases," *IEEE. Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 915–919, Sept. 1994.

[160] Y. Solihin and C. G. Leedham, "Integral ratio: A new class of global thresholding techniques for handwriting images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 761–768, Aug. 1999.

[161] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*, 2nd ed. Pacific Grove, CA: Brooks/Cole, 1999.

[162] T. Steinherz, E. Rivlin, and N. Intrator, "Off-line cursive script word recognition—A survey," *Int. J. Document Anal. Recognit.*, vol. 2, no. 2, pp. 90–110, 1999.

[163] C. Y. Suen, C. C. Tappert, and T. Wakahara, "The state of the art in on-line handwriting recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 787–808, Aug. 1990.

[164] C. Y. Suen, M. Berthod, and S. Mori, "Automatic recognition of handprinted characters—The state of the art," *Proc. IEEE*, vol. 68, pp. 469–487, Apr. 1980.

[165] H. Takahashi, "A neural net OCR using geometrical and zonal-pattern features," in *Proc. 1st Int. Conf. Document Anal. Recognit.*, 1991.

[166] Y. Tang *et al.*, "A new approach to document analysis based on modified fractal signature," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, Montreal, QC, Canada, 1995, pp. 567–570.

[167] Y. Tang *et al.*, "Off-line recognition of Chinese handwriting by multifeature and multilevel classification," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 556–561, May 1998.

[168] Y. Tao and Y. Y. Tang, "The feature extraction of Chinese character based on contour information," in *Proc. 5th Int. Conf. Document Anal. Recognit.*, Bangalore, India, 1999, pp. 637–640.

[169] C. C. Tappert, "Cursive script recognition by elastic matching," *IBM J. Res. Develop.*, vol. 26, no. 6, pp. 765–771, 1982.

[170] Q. Tian *et al.*, "Survey: Omnifont printed character recognition," *Visual Commun. Image Process: Image Process.*, pp. 260–268, 1991.

[171] Ø. D. Trier and A. K. Jain, "Goal directed evaluation of binarization methods," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 1191–1201, Dec. 1995.

[172] Ø. D. Trier, A. K. Jain, and T. Taxt, "Feature extraction method for character recognition—A survey," *Pattern Recognit.*, vol. 29, no. 4, pp. 641–662, 1996.

[173] W. H. Tsai and K. S. Fu, "Attributed grammar—A tool for combining syntactic and statistical approaches to pattern recognition," *IEEE Syst. Man, Cybern.*, vol. SMC-10, pp. 873–885, Dec. 1980.

[174] S. Tsujimoto and H. Asada, "Major components of a complete text reading system," *Proc. IEEE*, vol. 80, p. 1133, July 1992.

[175] D. Tubbs, "A note on binary template matching," *Pattern Recognit.*, vol. 22, no. 4, pp. 359–365, 1989.

[176] J. Wang and J. Jean, "Segmentation of merged characters by neural networks and shortest path," *Pattern Recognit.*, vol. 27, no. 5, p. 649, 1994.

[177] L. Wang and J. Mendel, "A fuzzy approach to handwritten rotation invariant character recognition," *Comput. Mag.*, vol. 3, pp. 145–148, 1992.

[178] S. S. Wang, P. C. Chen, and W. G. Lin, "Invariant pattern recognition by moment Fourier descriptor," *Pattern Recognit.*, vol. 27, pp. 1735–1742, 1994.

[179] K. Wang, Y. Y. Tang, and C. Y. Suen, "Multi-layer projections for the classification of similar Chinese characters," in *Proc. 9th Int. Conf. Pattern Recognit.*, 1988, pp. 842–844.

[180] T. Watanabe, L. Qin, and N. Sugie, "Structure recognition methods for various types of documents," *Mach. Vis. Appl.*, vol. 6, pp. 163–176, 1993.

[181] D. H. Wolperk, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241–259, 1992.

[182] S. L. Xie and M. Suk, "On machine recognition of hand-printed Chinese characters by feature relaxation," *Pattern Recognit.*, vol. 21, no. 1, pp. 1–7, 1988.

[183] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their application to handwritten recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, pp. 418–435, Mar. 1992.

[184] Y. Yang and H. Yan, "An adaptive logical method for binarization of degraded document images," *Pattern Recognit.*, vol. 33, no. 5, pp. 787–807, 2000.

[185] J. Yang and X. B. Li, "Boundary detection using mathematical morphology," *Pattern Recognit. Lett.*, vol. 16, no. 12, pp. 1287–1296, 1995.

[186] B. A. Yanikoglu and P. Sandon, "Recognizing off-line cursive handwriting," in *Proc. Int. Conf. Comput. Vision Pattern Recognit.*, Seattle, WA, 1994, pp. 397–403.

[187] F. T. Yarman-Vural and A. Atici, "A segmentation and feature extraction algorithm for Ottoman cursive script," in *Proc. 3rd Turkish Symp. Artif. Intell. Neural Networks*, 1994.

[188] F. T. Yarman-Vural and E. Ataman, "Noise histogram and cluster validity for Gaussian mixtured data," *Pattern Recognit.*, vol. 20, no. 4, pp. 385–401, 1987.

[189] B. Yu and A. K. Jain, "A robust and fast skew detection algorithm for generic documents," *Pattern Recognit.*, vol. 29, 1996.

[190] B. Zhang *et al.*, "Handwritten digit recognition by adaptive-subspace self organizing map (ASSOM)," *IEEE Trans. Neural Networks*, vol. 10, pp. 939–945, July 1999.

[191] J. Zhou *et al.*, "Recognition of handwritten numerals by quantum neural network with fuzzy feature," *Int. J. Document Anal. Recognit.*, vol. 2, no. 1, pp. 30–36, 1999.

[192] X. Zhu, Y. Shi, and S. Wang, "A new algorithm of connected character image based on Fourier transform," in *Proc. 5th Int. Conf. Document Anal. Recognit.*, Bangalore, India, 1999, pp. 788–791.