Offline Arabic Handwriting Recognition: A Survey

Liana M. Lorigo, *Member*, *IEEE Computer Society*, and
Venu Govindaraju, *Member*, *IEEE Computer Society*

Abstract—The automatic recognition of text on scanned images has enabled many applications such as searching for words in large volumes of documents, automatic sorting of postal mail, and convenient editing of previously printed documents. The domain of handwriting in the Arabic script presents unique technical challenges and has been addressed more recently than other domains. Many different methods have been proposed and applied to various types of images. This paper provides a comprehensive review of these methods. It is the first survey to focus on Arabic handwriting recognition and the first Arabic character recognition survey to provide recognition rates and descriptions of test data for the approaches discussed. It includes background on the field, discussion of the methods, and future research directions.

Index Terms—Computer vision, document analysis, handwriting analysis, optical character recognition.

1 Introduction

Offline handwriting recognition is the task of determining what letters or words are present in a digital image of handwritten text. It is of significant benefit to man-machine communication and can assist in the automatic processing of handwritten documents. It is a subtask of Optical Character Recognition (OCR), whose domain can be machine-print or handwriting but is more commonly machine-print. The recognition of Arabic handwriting presents unique challenges and benefits and has been approached more recently than the recognition of text in other scripts. This paper describes the state of the art of this field.

A recognition system can be either "online" or "offline." It is "online" if the temporal sequence of points traced out by the pen is available, such as with electronic personal data assistants that require the user to "write" on the screen using a stylus. It is "offline" if it is applied to previously written text, such as any images scanned in by a scanner. The online problem is usually easier than the offline problem since more information is available. This survey is restricted to offline systems.

1.1 Motivation

Arabic is spoken by 234 million people [1] and important in the culture of many more. While spoken Arabic varies across regions, written Arabic, sometimes called "Modern Standard Arabic" (MSA), is a standardized version used for official communication across the Arab world [1]. The characters of Arabic script and similar characters are used by a much higher percentage of the world's population to write languages such as Arabic, Farsi (Persian), and Urdu. Thus, the ability to automate the interpretation of written Arabic would have widespread benefits.

Arabic handwriting recognition can also enable the automatic reading of ancient Arabic manuscripts. Since written Arabic has changed little over time, the same techniques developed for MSA can be applied to many manuscripts. Automatic processing can greatly increase the availability of their content. Because the writing in manuscripts is usually neater than free handwriting, the recognition task is arguably simpler. However, image degradation, unexpected markings, and previously unseen writing styles provide challenges.

1.2 Arabic Writing

The Arabic alphabet contains 28 letters. Each has between two and four shapes and the choice of which shape to use depends on the position of the letter within its word or subword. The shapes correspond to the four positions: beginning of a (sub)word, middle of a (sub)word, end of a (sub)word, and in isolation. Table 1 shows each shape for each letter. Letters without initial or medial shapes shown cannot be connected to the following letter, so their "initial" shapes are simply their isolated shapes and their "medial" shapes are their final shapes.

Additional small markings called "diacritical marks" or "diacritics" represent short vowels or other sounds, such as syllable endings and nunation (the addition of an "n" or "nuun" sound). The diacritics fat-ha, dumma, and kesra indicate short vowels, sukkun indicates a syllable stop, and the nunation diacritic can accompany fat-ha, dumma, or kesra (Fig. 1). They are normally omitted from handwriting. Other markings (sometimes called "diacritics," too) indicate doubled consonants or different sounds. Examples are "hamza," "shadda," and "madda" (Fig. 2). Some publications on Arabic character recognition use the term "diacritics" even more broadly to also include dots of letters, but that practice is not standard and is not used here. Some letters have "descenders" or "ascenders," which are portions that extend below the primary line on which the letters sit or above the

- *The authors are with the Department of Computer Science and Engineering, State University of New York at Buffalo, 520 Lee Entrance, Suite 202, UB Commons, Amherst, NY 14228.*
E-mail: {lmlorigo, govind}@ buffalo.edu.

Manuscript received 19 Nov. 2004; revised 21 Sept. 2005; accepted 26 Sept. 2005; published online 13 Mar. 2006.
Recommended for acceptance by Y. Amit.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0619-1104.

0162-8828/06/$20.00 © 2006 IEEE Published by the IEEE Computer Society

TABLE 1
The Arabic Alphabet

Name	Isolated	Initial	Medial	Final
alif	ا	-		ـا
baa	ب	بـ	ـبـ	ـب
taa	ت	تـ	ـتـ	ـت
thaa	ث	ثـ	ـثـ	ـث
jiim	ج	جـ	ـجـ	ـج
Haa	ح	حـ	ـحـ	ـح
khaa	خ	خـ	ـخـ	ـخ
daal	د	-		ـد
dhaal	ذ	-		ـذ
raa	ر	-		ـر
zaay	ز	-		ـز
siin	س	سـ	ـسـ	ـس
shiin	ش	شـ	ـشـ	ـش
Saad	ص	صـ	ـصـ	ـص
Daad	ض	ضـ	ـضـ	ـض
Taa	ط	طـ	ـطـ	ـط
Dhaa	ظ	ظـ	ـظـ	ـظ
ayn	ع	عـ	ـعـ	ـع
ghayn	غ	غـ	ـغـ	ـغ
faa	ف	فـ	ـفـ	ـف
qaaf	ق	قـ	ـقـ	ـق
kaaf	ك	كـ	ـكـ	ـك
laam	ل	لـ	ـلـ	ـل
miim	م	مـ	ـمـ	ـم
nuun	ن	نـ	ـنـ	ـن
haa	ه	هـ	ـهـ	ـه
waaw	و	-		ـو
yaa	ي	يـ	ـيـ	ـي

Position-dependent shapes are shown for each letter.

height of most letters (Fig. 3). There is no upper or lower case, but only one case.

Arabic script is written from right to left and letters within a word are normally joined even in machine-print. Letter shape and whether or not to connect depend on the letter and its neighbors (Fig. 4). Letters are connected at the same relative height. The "baseline" is the line at the height at which letters are connected and it is analogous to the line on which an English word sits. Letters are wholly above it except for descenders and some markings. For handwriting, the baseline is an ideal concept and a simplification of actual writing. In practice, connections occur near, but not necessarily on, such a line (Fig. 4c).

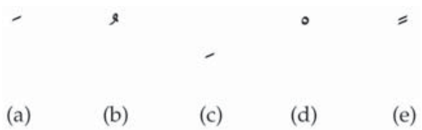


Fig. 1. Diacritical marks: (a) fat-ha, (b) dumma, (c) kesra, (d) sukkun, and (e) nunation diacritic with fat-ha.

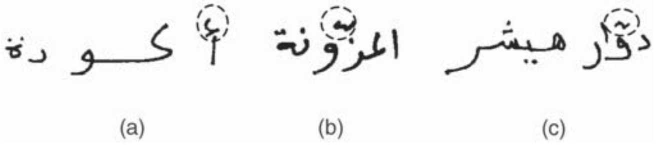


Fig. 2. Handwritten words with secondary markings circled: hamza, shadda, and madda.

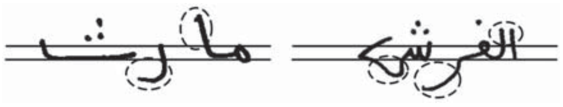


Fig. 3. Ascenders and descenders are circled; horizontal lines are shown for reference.

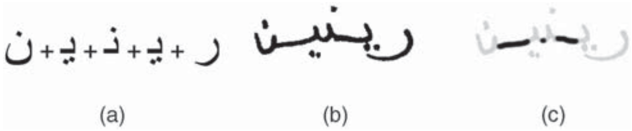


Fig. 4. Example of connecting letters. (a) Individual letters. (b) The original word image. (c) Letter connections along the baseline shown in black.

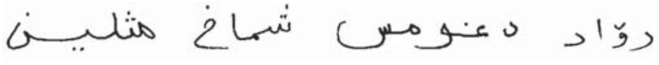
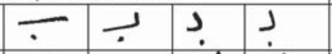
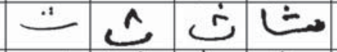
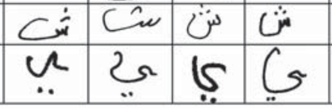
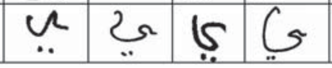


Fig. 5. Words comprised of one, two, three, and four subwords, respectively.

Fig. 6. Four handwritten examples of laam-alif suggest allowable variation.

TABLE 2
Variation in Handwritten Letters

Printed, Handwritten		Remarks
ـبـ		Vert. line may be missing
ث		Dot pattern varies
ش		Dots, curve shape vary
ي		Curves' angles, sizes vary

There is no connection between separate words, so word boundaries are always represented by a space. Six letters, however, can be connected only on one side. When they occur in the middle of a word, the word is divided into multiple subwords separated by spaces (Fig. 5). Some publications call subwords "pieces of Arabic words" or "PAWs."

A "ligature" is a character formed by combining two or more letters in an accepted manner. Arabic has several standard ligatures, which are exceptions to the above rules for joining letters. Most common is "laam-alif," the combination of "laam" and "alif," and others include "yaa-miim" and "laam-miim." In machine-print laam-alif appears as لا and, in handwriting, as in Fig. 6. Exact shapes are font-dependent in print and writer-dependent in handwriting.

Segmentation is the task of separating a word into its component characters. The connected nature of Arabic renders it more difficult than for nonconnected writing methods such as printed Latin. Handwriting exhibits variation in slope, stretch, skew, relative size, and letter appearance. Table 2 shows examples of variation in letter appearance. Another challenge is that sometimes one letter appears above or below the previous letter (Fig. 7a). Also difficult is the rare situation when a preceding letter appears to the left of a succeeding letter (Fig. 7b).

Many of the above aspects render the Arabic recognition task more difficult than that of Latin script. However, there are also aspects that could make it easier such as lack of

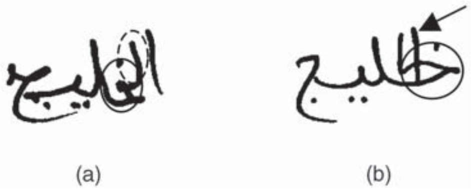


Fig. 7. Two images of the same name: (right-to-left) alif, laam, khaa, laam, yaa, jiim. (a) khaa (solid circle) is below laam (dashed) and (b) khaa (circle) is before alif (arrow).

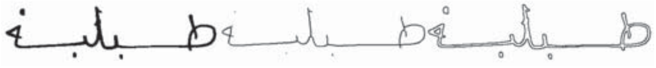


Fig. 8. A word image, its skeleton (algorithm from [2]), and its contour.

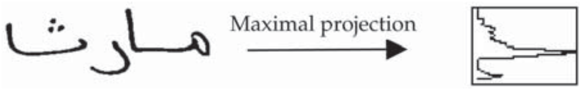


Fig. 9. Vertical projection of a word used to detect the baseline.

case, strong baseline, short average word length, discriminatory dots and markings, and systematic variants on letter shape. The frequent assumption that Arabic is more difficult may be due to the fact that less effort has been devoted to it and, so, the state of the art is less advanced.

The remainder of this paper is organized as follows: Section 2 presents background on the field including common techniques, an overview of Arabic machine-print recognition, and databases available for research. Section 3 defines a framework for the recognition task and analyzes specific systems in its context. It also includes tables that classify the systems according to several algorithmic aspects. Section 4 discusses directions for future work and conclusions.

2 Background

This section summarizes major aspects of recognition approaches: preprocessing, structural and statistical features, and recognition strategies. It gives a brief overview of machine-print recognition since many handwriting approaches follow from work on machine-print and it describes databases created for recognition research.

2.1 Prerecognition Tasks

The image is often converted to a more concise representation prior to recognition (Fig. 8). A *skeleton* is a one-pixel thick representation showing the centerlines of the text. Skeletonization, or "thinning," facilitates shape classification and feature detection. An alternative is the Freeman chain code of the border ("contour") of the text [3], [4]. Chain code stores the absolute position of the first pixel and the relative positions of successive pixels along the contour. Difficulties with thinning include possible mislocalization of features and ambiguities particular to each thinning algorithm. The contour approach avoids these difficulties since no shape information is lost.

A common step is the detection of the baseline. The standard approach is vertical projection, which is the projection of the binary image of a word or line of text onto a vertical line. The baseline can be detected as the maximal peak (Fig. 9). This approach is ineffective for some single words or short sequences of words; so, in 2002, Pechwitz and Märgner approximated the skeleton by piecewise linear curves and detected the baseline as the line that best fit the relevant edges of that approximation [5].

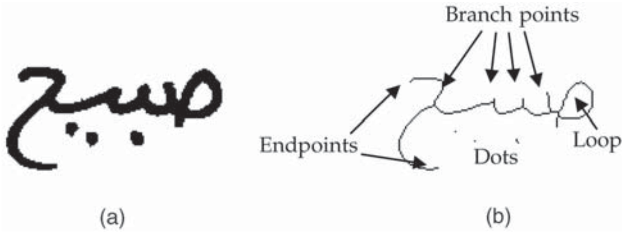


Fig. 10. (a) Original image. (b) Structural features shown on "skeleton" image [2].

Noise removal and slope and slant correction are often needed. In this survey, these steps are discussed only when warranted by the specific method. Farooq et al. [6] presented a method for baseline detection and methods for slant normalization, slope correction, and line and word separation. The first method calculated an approximate baseline from linear regression on local minima on the contour of the word. The approximation was refined by a second linear regression on only those minima that were close to it. The system was tested on a set of images used in [5] and comparisons with that method were given.

For unconstrained images, it is necessary to locate the handwriting in the image. In 2003, Soleymani and Razzazi presented a system to find isolated characters handwritten on forms [7]. It detected letter boundaries in the presence of noise, separated the main body of each letter from other markings, and extracted a skeleton. On a database of 220,000 handwritten forms by more than 50,000 writers, it cropped and processed 96.4 percent of the characters with no error. The majority of the errors occurred when there were discontinuities in the main body of a character.

Motivated by online recognition [8], [9], [10], in 1993 Abuhaiba and Ahmed presented a method to restore the writing order of strokes to offline word images [11]. Line segments were fit to thinned words and ordering was hypothesized from knowledge of the script. Using text by two writers, with 728 and 877 strokes, respectively, stroke-ordering success rate was 92 percent with some errors from the line-approximation step.

2.2 Structural and Statistical Features

Structural features are intuitive aspects of writing, such as loops, branch-points, endpoints, and dots. They are often, but not necessarily, computed from a skeleton of the text image, as shown in Fig. 10. Many Arabic letters share common primary shapes, differing only in the number of dots and whether the dots are above or below the primary shape. Structural features are a natural method for capturing dot information explicitly, which is required to differentiate such letters. This perspective may be a reason that structural features remain more common for the recognition of Arabic script than for that of Latin script. *Statistical* features are numerical measures computed over images or regions of images. They include, but are not limited to, pixel densities, histograms of chain code directions, moments, and Fourier descriptors.

2.3 Recognition Methodologies

Artificial neural networks (ANNs), or "neural networks," consist of simple processing elements and a high degree of interconnection [12]. The weights within the elements are

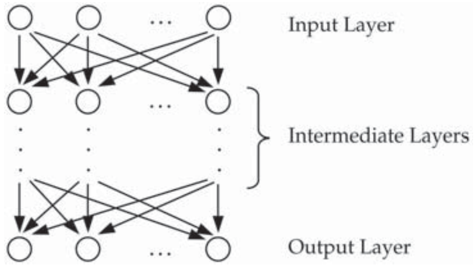


Fig. 11. General neural network architecture. There may be an arbitrary number of processing elements (circles) in each layer and an arbitrary number of intermediate layers.

learned from training data. The elements are organized into an initial input layer, intermediate "hidden" layers, and a final output layer (Fig. 11). Information proceeds from the first to the final layer, which gives a character or word choice in this task.

Hidden Markov models (HMMs) are also appropriate for learning characteristics that are difficult to describe intuitively [13]. Conventional HMMs model one-dimensional sequences of data and contain states and probabilities for transitioning between them according to an observed sequence of data or "observations." Assume that, at each time step, the system was in one of n possible states and produced one of m possible observation symbols, the choice depending on probabilities associated with that state (Fig. 12). The goal is to reconstruct the state sequence ("path") from the observations to learn the meaning of the data. For text recognition, the observations could be sets of pixel values and states could represent parts of letters. An alternative to finding a path in a single model is accepting the most probable of several models. See [14] for a 2000 survey of the use of HMMs in Arabic character recognition.

Finally, recognition approaches can be either "holistic" or segmentation-based. "Holistic" means that words are processed as a whole without segmentation into characters or strokes [15]. In segmentation-based approaches, whole or partial characters are recognized individually after they have been extracted from the text image.

2.4 Machine-Print Recognition

Earlier surveys discussed both machine-print and handwriting, with much more discussion of machine-print [16], [17], [18], [19]. In 1980, Nouh et al. suggested a standard Arabic character set to facilitate computer processing [20], [21]. Parhami and Taraghi presented an approach to Arabic OCR in 1981, demonstrated on newspaper headlines [22]. Subwords were segmented and recognized according to features such as concavities, loops, and connectivity. Increasing the tolerance to font variations, in 1986 Amin and Masini proposed a system for segmentation and recognition that used horizontal and vertical projections and shape-based primitives [23]. On 100 multifont words, it achieved a character recognition rate of 85 percent and a word recognition rate of 95 percent. In a 1988 recognition system by El-Sheikh and Guindi, segmentation points were based on minimal heights of word contours and character classification used Fourier descriptors [24].

A 1990 approach by Sami El-Dabi et al. based on invariant moments segmented characters only after they were recognized. Recognition was attempted on regions of increasing width until a match was found [25]. Persian and Arabic characters are almost the same and a 1995 paper on Persian OCR emphasized a segmentation algorithm that traced word contours to separate disconnected overhanging characters and used a sliding-window approach for most characters [26]. In 1996, Ymin and Aoki presented a two-step segmentation system which used vertical projection onto a horizontal line followed by feature extraction and measurements of character width [27]. Al-Badr and Haralick presented a holistic recognition system based on shape primitives that were detected with mathematical morphology operations (1996, 1998) [28], [29]. Alherbish et al. presented a parallel OCR algorithm which achieved a speed-up of 5.34 in 1997 [30]. A 1999 system by Khorsheed and Clocksin used features from a word's skeleton for recognition without prior segmentation [31]. BBN developed a script-independent methodology for OCR, which has been tested on English, Arabic, Chinese, Japanese, and other languages [32], [33], [34]. It used their HMM-based speech recognizer with features from image frames (vertical strips). Only the lexicon, language model, and training data depended on the language. In 1999, they presented a system for English and Arabic in which the lexicon was unlimited [35]. The DARPA Arabic OCR Corpus was used for testing. Trenkle et al. presented a method that used ensembles of decision trees for recognition on low-quality, low-resolution images in 2001 [36]. Prior methods are discussed in [37], [38], [39]. In 2002, Hamami and Berkani developed a structural approach to handle many fonts and it included rules to prevent oversegmentation [40]. Al-Qahtani and Khorsheed presented a system based on the portable Hidden Markov Model Toolkit in 2004 [41], [42].

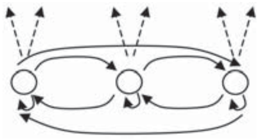
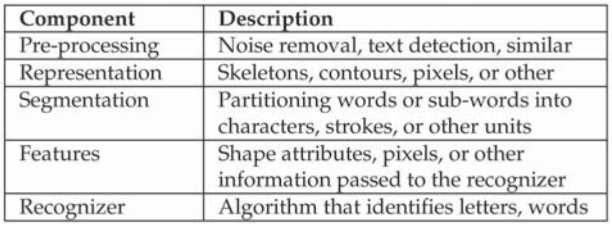


Fig. 12. HMM with three states and two possible observation symbols at each.

Both image-level [32], [35] and structural [31], [41], [42] features have been applied to handwriting recognition (respective examples: [43]; [44], [45], [46]). The former places the burden of processing on the recognizer, while the latter involves more processing at the feature detection stage. The pixel-based approach is more common for machine-print than for handwritten Arabic, which often uses structural or hybrid approaches. This situation may be due to the greater variability in handwriting. More training data would be needed for image-level features to model handwritten character shapes than printed character shapes. Conclusions about the comparative efficacy of the approaches for handwriting are not yet possible because large testing databases have only been available for a short time and are not yet used throughout the field.

There are several commercial Arabic OCR products. In 2000, the performance of the Sakhr and OmniPage products was evaluated using the DARPA/SAIC database. Average page accuracy rates of 90.33 percent and 86.89 percent were observed, with differences in precision, speed, and the effects of changes in image resolution [47], [48]. In 2003, Abuhaiba proposed a disconnected Arabic font to increase these rates toward the higher rates achieved on noncursive scripts such as Latin and Chinese [49]. Ciyasoft and Novodynamics also offer Arabic OCR products. As of this writing, no commercial system exists for offline Arabic handwriting recognition.

TABLE 3
Components of Handwriting Recognition Framework



Component	Description
Pre-processing	Noise removal, text detection, similar
Representation	Skeletons, contours, pixels, or other
Segmentation	Partitioning words or sub-words into characters, strokes, or other units
Features	Shape attributes, pixels, or other information passed to the recognizer
Recognizer	Algorithm that identifies letters, words

2.5 Databases

Ten to 15 years ago, large databases were developed for the recognition of handwriting in Latin scripts. For example, the CEDAR database by our group was released in 1994 and spurred intense research in the Latin OCR field [50]. It contains images of approximately 5,000 city names, 5,000 state names, 10,000 ZIP codes, and 50,000 alphanumeric characters. Recently released databases for Arabic handwriting recognition have similar size and scope.

One widespread domain of the handwriting recognition problem is writing on personal checks. In 2002, Alma'adeed et al. presented the AHDB, a database of samples from 100 different writers, including words used for numbers and in bank checks [51]. It also contains the most popular words in Arabic writing and free handwriting pages on any topic of the writer's choosing. In 2003, Al-Ohali et al. of the Centre for Pattern Recognition and Machine Intelligence (CENPARMI) in Montréal developed databases of images from 3,000 checks provided by a banking corporation. These databases are subwords, Indian digits, legal amounts (numeric amounts written in words), and courtesy amounts (numeric amounts written with Indian digits) [52]. "Indian digits" are the numeric digits normally used in Arabic writing, as opposed to "Arabic numerals" used in Latin script. The subwords database contains 29,498 samples, the Indian digits database 15,175, and the legal and courtesy databases 2,499 each.

The recognition of city names can be used for mail sorting, data entry, and other tasks. Thus, the IFN/ENIT database was created by the Institute of Communications Technology (IFN) at Technical University Braunschweig in Germany and the Ecole Nationale d'Ingénieurs de Tunis (ENIT) in Tunisia. It consists of 26,459 images of the 937 names of cities and towns in Tunisia, written by 411 different writers. The images are partitioned into four sets so that researchers can use and discuss training and testing data in this context.

3 Analysis of Handwriting Methods

This section presents a general framework for the handwriting recognition task. It includes the frequent components of recognition algorithms, namely, preprocessing, representation, stroke or character segmentation, features, and recognizer (Table 3). Some approaches do not use all of these elements but only a subset.

Fig. 13 illustrates the components of the framework organized as in most algorithms. First, an image is cleaned with image processing techniques. It may be converted to a more concise representation, then features are detected from words or characters. With the features as input, a recognizer returns the identified text string. The term "features" does not necessarily refer to structural or precomputed items, but any quantities passed to the recognizer. They may be precomputed for use in segmentation, computed on individual letters after segmentation, or both.

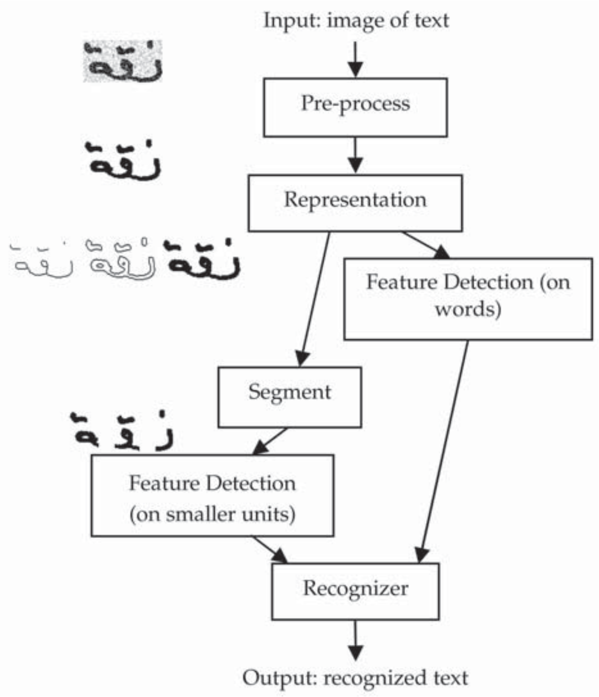


Fig. 13. Generalized framework for Arabic handwriting recognition.

In this section, system descriptions are organized according to which components represent the systems' primary contributions. Many systems demonstrate contributions in multiple areas and other components are also stated in each description. Preprocessing is discussed only when warranted. The other four components represent dominant aspects of the algorithms and provide the organization of this section. A task description is given for each system. It includes style or neatness constraints, the lexicon if applicable, and whether the domain was characters, words, or pages. Recognition rates and the size and type of test data are also given. Following the system descriptions is a summary (Section 3.5) of the first international competition in this field.

3.1 Representation

Most methods extract a skeleton or list of contours from the image. Table 4 categorizes approaches according to the representation used. The first three approaches in this section extended skeletons to graph models, using line segments and links explicitly [53], [54], [55]. Contours of projections [56] and points along trajectories [57] were used by others.

In 1994, Abuhaiba et al. proposed a set of character graph models to recognize isolated letters [53]. Each model was a state machine with transitions corresponding to directions of segments in the character and with additional "fuzzy" constraints to distinguish some characters. Each letter's skeleton was converted to a tree structure which was matched to a model by a rule-based recognizer. Test data was written by four people. Recognition rates depended on tuning the models after experiments on letters by each writer and thinning errors caused recognition errors.

TABLE 4
Representations

Skeleton	Contour	Pixels
Mozaffari 2005 [58]	Safabakhsh 2005 [68]	El-Hajj 2005 [78]
Alma'adeed 2002, 2004 [59, 60]	Souici-Meslati, Farah 2004 [45, 69, 70]	Clocksin 2003 [79]
Haraty 2002-2004 [61-64]	Sari 2002 [71]	Pechwitz: blurred skel. image 2003 [43]
Amin 2003, 1996 [44, 54]	Snoussi Maddouri 2002 [72]	Al-Badr: morphology 1998 [29]
Khorsheed 2003 [46]	Dehghan 2001 [73]	Motawa: morphology 1997 [80]
Soleymani 2003 [7]	Dehghani: cont. of projections 2001 [56]	Al-Yousefi: projections 1992 [81]
Fahmy 2001 [65]	Olivier, Miled 1996, 2001 [74, 75]	
Abuhaiba 1993-1998 [11, 53, 55]	Mostafa 1999 [76]	
Goraine 1992 [66]	Ymin 1996 [27]	
Almuallim 1987 [67]	Romeo-Pakker 1995 [77]	

A problem of skeletons is that there may be "hairs" (short spurious lines) in the thinned image or related difficulties [11], [53], [54], [59], [65].

TABLE 5
Segmentation-Based and Holistic Approaches

Segmentation-based	Holistic
Safabakhsh 2005 [68]	El-Hajj 2005 [78]
Haraty 2002-2004 [61-64]	Alma'adeed 2002, 2004 [59, 60]
Sari 2002 [71]	Souici-Meslati, Farah 2004 [45, 69, 70]
Fahmy 2001 [65]	Khorsheed 2003 [46]
Miled 2001 [75]	Pechwitz 2003 [43]
Abuhaiba 1998 [55]	Snoussi Maddouri 2002 [72]
Goraine 1992 [66]	Dehghan 2001 [73]
Almuallim 1987 [67]	Al-Badr 1998 [29]

Hybrids (e.g., [72] in which some breakpoints are hypothesized) are placed according to the authors' judgment.

Amin et al. also used a skeleton-based graph representation for the recognition of single letters (1996) [54]. Structural features including curves were fed into a five-layer neural network. The network was trained with 2,000 characters, retrained with 528 of the 2,000 and tested with another 1,000 by 10 writers. A 92 percent recognition rate was obtained. Difficulties included spurious thinned lines, incorrect curve directions, and the need to modify rules during testing.

Extending the work in [53], Abuhaiba et al. proposed a system for the recognition of free handwritten text in 1998 [55]. It used the skeleton representation and segmented subwords into strokes that were further segmented into "tokens." Tokens are single vertices representing dots or loops or sequences of vertices. The recognizer was a "fuzzy sequential machine" which consisted of classes to be recognized, sets of initial and terminal states, stroke directions used for entering states, and a function for transitioning between states. Tokens were recognized if possible or else used to augment the recognizer. When needed, the user interactively grouped tokens into meaningful "token strings." To detect lines of text, strokes from the entire page were partitioned using a minimal spanning tree algorithm. Another graph algorithm grouped strokes into characters and subwords. Thirteen pages by 13 writers were used for training, and another 20 pages by 20 writers were used for testing. Writers were asked to write in a particular style, to write the main stroke without lifting the pen, to omit diacritics, and to avoid generating blobs, but most did not comply with these constraints. Subword and character recognition rates of 55.4 percent and 51.1 percent were obtained. No lexicon was used. In addition to the technical method, this publication is important since it generalized the domain to free handwriting.

The uncommon representation of "contour of projections" was employed by Dehghani et al. in 2001 [56]. The task was Persian character recognition and preprocessing included median and mathematical morphological filtering, binarization, scaling, and centering. Regional projection contour transformation (RPCT) was used [82], so the image was projected in multiple directions (here, horizontal and vertical) and the chain-code contour of each projection was obtained. The contour was sampled and features were obtained for each section using a two-dimensional pattern, the number of active pixels, and slope and curvature. Separate feature vectors from the contours of horizontal and vertical projections were computed and modeled by individual HMMs, yielding two HMMs per character. During recognition, scores from individual classifiers were integrated to improve performance. The size of the training and testing sets was not provided. Recognition rates were 92.76 percent on the training set and 71.82 percent on the test set.

Al-Shaher and Hancock considered the recognition problem from a different perspective (2002, 2003) [57], [83]. They chose seven basic shape classes found in Arabic characters, each of which consisted of only one trajectory, which could be obtained from online writing information or stroke analysis of text. Their system distributed 20 points uniformly along each trajectory to train point distribution models (PDMs) in the style of Cootes and Taylor [84]. The recognizer was the expectation-maximization (EM) algorithm. Testing 100 samples for each class showed that mixtures of PDMs achieved significantly better performance than did a single PDM.

3.2 Segmentation

This component refers to the segmentation of words into characters, strokes, or other units. Higher-level segmentation, such as segmenting pages into lines of text or lines into words, is discussed separately when needed. Table 5 classifies word recognition methods according to whether or not they use explicit segmentation and this section describes segmentation methods.

In 1995, Romeo-Pakker et al. published two methods to segment handwritten cursive text into characters [77]. They were applied to handwritten Arabic text and cursive Latin text for segmentation into lines, words, and characters. The methods used horizontal and vertical projections, a Freeman chain code representation, and rules. The higher rate of 99.3 percent successful segmentation on 1,383 words was obtained using the method based on the text's upper contour. The authors collaborated with Olivier to segment words into portions of characters called "graphemes" (1996) [74]. It determined the segmentation points from the upper half of the border of the letters and generated a description of each grapheme inspired by human perception. On 6,000 city-names by 20 different writers, it attained a 98.52 percent rate of good segmentation.

In 1997, Motawa et al. suggested an algorithm for segmenting Arabic words into characters [80]. They applied mathematical morphological techniques based on the assumptions that characters are usually connected by horizontal lines and that these lines are "regularities," as opposed to (vertical) "singularities," when considering the connected

TABLE 6
Features

Publication	Features
El-Hajj 2005 [78]	Pixel densities, density transitions, and concavity configurations along frames and with respect to baselines; frame derivatives; centers of gravity with respect to baselines
Mozaffari 2005 [58]	Average and variance of X and Y changes in portions of the skeleton
Safabakhsh 2005 [68]	Fourier descriptors, numbers of loops, height/width ratios, pixel densities, positions of right and left connections
Alma'adeed 2004 [60]	Ascenders, descenders, structural features, frame-based features
Haraty 2002-2004 [61-64]	Loops, end/turning/junction points, widths, heights, row/column transitions and densities, contour heights
Souici-Meslati, Farah 2004 [45, 69, 70]	Loops, dots, connected components, ascenders, descenders
Amin 2003 [44]	Loops, dots, hamza, lines, open curves, and their relationships
Clocksin 2003 [79]	Moment functions applied to image and polar transform image
Khorsheed 2003 [46]	Dots, loops, endpoints, branch/cross/turning points, lines
Pechwitz 2003 [43]	Pixel intensites from pre-processed image
Snoussi Maddouri 2002 [72]	Ascenders, descenders, loops, dots, sub-word positions, normalized Fourier descriptors
Dehghan 2001 [73]	Histograms of slopes along contour
Dehghani 2001 [56]	Slope, curvature, number of active pixels using a two-dimensional pattern
Fahmy 2001 [65]	End/turning/junction points, complementary characters, loops
Miled 2001 [75]	Concavities and inclinations for ascenders and descenders, densities and shapes for diacritics
Abuhaiba 1994, 1998 [53, 55]	Dots, loops, strokes, links, directions, intersections
Amin 1996 [54]	Loop positions and types, line positions and directions
Olivier 1996 [74]	Loops and relative locations, heights, and sizes of parts of characters
Al-Yousefi 1992 [81]	Statistics from moments of horizontal and vertical projections
Goraine 1992 [66]	Stroke directions, sub-word positions, loops, secondary strokes, dots
Almuallim 1987 [67]	Strokes' endpoints, lengths, frames, connection points, others

word or subword as a function or curve. The algorithm was tested on a few hundred words written by different writers and achieved an 81.88 percent rate of good segmentation.

Mostafa and Darwish presented baseline-independent algorithms to detect lines and words, to segment words into primitives and to extract diacritics in handwritten text (1999) [76]. Using the chain code representation, the segmentation algorithm oversegmented words, then applied rules to remove extra points. On 7,922 characters written by 14 writers, the system achieved a 97.7 percent rate of correct segmentation.

A character segmentation system was proposed by Sari et al. in 2002 [71]. It used the contour representation and detected segmentation points by applying rules to local minima of the lower contour of each subword. Characters that overlapped vertically due to writing style or slant were addressed in a subsequent contour-processing step. Segmentation success rate was 86 percent on 100 words. The authors combined this system with their previously published recognizer, RECAM [85]. RECAM used four three-layer neural networks, one for each character position (beginning, middle, end, isolated).

Lorigo and Govindaraju presented a segmentation system which used derivative information in a region around the baseline to oversegment words [86]. It used rules based on allowable shapes to discard extra points. The test set was 200 images from the IFN/ENIT database and excluded images containing several letters and markings. The correctly detected segmentation points were 92.3 percent and the oversegmentation points remaining were 5.1 percent.

3.3 Features

Features are the information passed to the recognizer, such as pixels, shape data, or mathematical properties. They are sometimes used for segmentation. Table 6 lists the features used in the various algorithms and system descriptions follow.

In 1987, Almuallim and Yamaguchi proposed one of the first methods for Arabic handwriting recognition [67]. It used the skeleton representation and structural features for word recognition. Words were segmented into "strokes" which were classified and combined into characters according to the features. The recognizer was the set of classification rules. The method achieved a recognition rate of 91 percent on 400 by two writers. To our knowledge, the method was the first to focus on text that was not presegmented.

In 1992, Al-Yousefi and Udpa introduced a statistical approach for the recognition of isolated Arabic characters [81]. It included the segmentation of each character into primary and secondary parts (such as dots and small markings) and normalization by moments of vertical and horizontal projections. The features were nine measurements of kurtosis, skew, and relationships of moments, and the recognizer was a quadratic Bayesian classifier. Test data included machine-printed and handwritten characters, but only 10 samples were used on the handwritten side.

Goraine et al. presented a structural approach in 1992 [66]. It operated on whole words and was applied to typewritten and handwritten words. After segmentation points were estimated from skeletons, structural features and a rule-based recognizer identified each letter. A dictionary was used to confirm or correct the results. In the handwriting recognition test, the system obtained a 90 percent recognition rate on 180 words comprised of about 600 characters. The three writers were asked to write neatly in a prespecified font.

Related work by Clocksin and Fernando in 2003 addressed the domain of Syriac manuscripts [79]. Also, a West Semitic language, Syriac is less grammatically complex than Arabic and was a primary language for theology, science, and literature from the third century AD to the seventh century AD. The system used full image representation of individual characters and sets of features based on moments. A segmentation method based on vertical and horizontal projections and run-lengths was described, but recognition rates were given for presegmented characters. The recognizer was a support vector machine with tenfold cross-validation. The highest rate attained was 91 percent using features from the character image and its polar transform image.

In 2005, El-Hajj et al. demonstrated the benefit of features based on upper and lower baselines, within the context of frame-based features with an HMM recognizer [78]. This context was used in the BBN system for machine-print discussed above [32]. El-Hajj et al., however, included features measuring densities, transitions, and concavities in zones defined by the detected baselines. The system was tested on the IFN/ENIT database minus those names that have fewer than eight images, leaving 21,500 images for testing. For each of four experiments, the system was trained on three of the four image sets and tested on the remaining set. Recognition rates ranged from 85.45 percent to 87.20 percent. In their experiments, the addition of the baseline-dependent features to similar measurements that do not use those zones significantly improved recognition.

Also, in 2005, Mozaffari et al. proposed a method for the recognition of Arabic numeric characters which is structural and also uses statistical features [58]. Endpoints and intersection points were detected on a skeleton then used to partition it into primitives. Eight statistical features were computed on each primitive, the features for all primitives were concatenated, and the result was normalized for length. Nearest-neighbor was used for classification. Eight digits were tested, and 280 image of each were used for training and 200 for testing. The digits were written by over 200 writers collectively. The recognition rate was 94.44 percent.

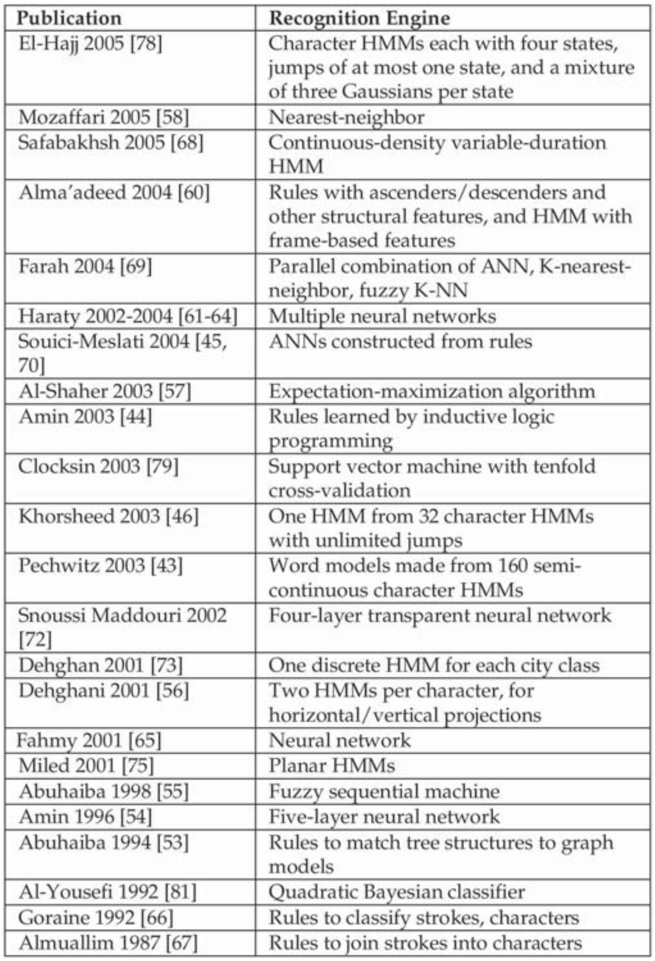
3.4 Recognition Engine

The recognition engine can be rule-based, probabilistic, or a combination. Lexical information is usually incorporated at this stage. Table 7 lists the recognizers used by the respective systems. The methods in this section use artificial neural networks, hidden Markov models, rules, or hybrids of rules with statistical methods.

3.4.1 Rules

Several previously described systems used rules for recognition [53], [60], [66], [67]. To automate this paradigm, Amin presented an automatic technique to learn rules for isolated characters (2003) [44]. Structural features including open curves in several directions were detected from the Freeman code representation of the skeleton of each character and the relationships were determined with Inductive Logic Programming (ILP). The reader is referred to [87] for further information on ILP. Test data consisted of 40 samples of 120 different characters by different writers with 30 character samples used for training and 10 for testing for most experiments. A character recognition rate of 86.65 percent was obtained.

TABLE 7
Recognition Engines



Publication	Recognition Engine
El-Hajj 2005 [78]	Character HMMs each with four states, jumps of at most one state, and a mixture of three Gaussians per state
Mozaffari 2005 [58]	Nearest-neighbor
Safabakhsh 2005 [68]	Continuous-density variable-duration HMM
Alma'adeed 2004 [60]	Rules with ascenders/descenders and other structural features, and HMM with frame-based features
Farah 2004 [69]	Parallel combination of ANN, K-nearest-neighbor, fuzzy K-NN
Haraty 2002-2004 [61-64]	Multiple neural networks
Souici-Meslati 2004 [45, 70]	ANNs constructed from rules
Al-Shaher 2003 [57]	Expectation-maximization algorithm
Amin 2003 [44]	Rules learned by inductive logic programming
Clocksin 2003 [79]	Support vector machine with tenfold cross-validation
Khorsheed 2003 [46]	One HMM from 32 character HMMs with unlimited jumps
Pechwitz 2003 [43]	Word models made from 160 semi-continuous character HMMs
Snoussi Maddouri 2002 [72]	Four-layer transparent neural network
Dehghan 2001 [73]	One discrete HMM for each city class
Dehghani 2001 [56]	Two HMMs per character, for horizontal/vertical projections
Fahmy 2001 [65]	Neural network
Miled 2001 [75]	Planar HMMs
Abuhaiba 1998 [55]	Fuzzy sequential machine
Amin 1996 [54]	Five-layer neural network
Abuhaiba 1994 [53]	Rules to match tree structures to graph models
Al-Yousefi 1992 [81]	Quadratic Bayesian classifier
Goraine 1992 [66]	Rules to classify strokes, characters
Almuallim 1987 [67]	Rules to join strokes into characters

3.4.2 Artificial Neural Networks

Many variations of ANNs have been used in this field. In 2001, Fahmy and Al Ali proposed a system based on ANNs with structural features [65]. Preprocessing steps included slope correction and slant correction. Features were detected from skeletons and fed into a neural network. A 69.7 percent word recognition rate was obtained on 600 words written by one writer.

Snoussi Maddouri et al. used a "transparent" four-layer neural network on images of words from bank checks (2002) [72]. Here, "transparency" means that the layers had intuitive meanings: primitives, letters, subwords, and words. Preprocessing included slant correction before baseline detection. Global features including ascenders, descenders, loops, dots, and position were used for "contextual" segmentation, which refers to segmentation into zones based on features. Global features and local Fourier descriptors were fed to the neural network. A 97 percent word-level recognition rate was achieved on 2,070 images with a lexicon of 70 words. The combination of global and local features was like the mechanism by which people read. We recognize many words without examining individual letters and, if that fails, we examine letters.

Haraty and Ghaddar proposed the use of two neural networks to classify previously segmented characters (2003)

[63], [64]. Their method used a skeleton representation and structural and quantitative features such as the number and density of black pixels and the numbers of endpoints, loops, corner points, and branch points. On 2,132 characters, the recognition rate was over 73 percent. A prior segmentation system used the same representation and similar features to oversegment words and a neural network to confirm or reject the proposed breakpoints [61]. Also addressed was the task of horizontal segmentation, which is needed when one letter is above another (here, "ligatures") [62]. Two networks were used and success rates were 79 percent for ligature identification and 91 percent for validation of potential horizontal segmentation points.

3.4.3 Hidden Markov Models

HMMs have also been applied in diverse ways. Miled and Ben Amara combined the algorithm of [74] with a planar hidden Markov model (PHMM) to recognize machine-printed and handwritten words in 2001 [75]. They chose the planar model to handle the planar nature of writing and the specific situation in which one letter is directly above another.

In 2001, Dehghan et al. presented an HMM-based system whose features were histograms of Freeman chain code directions in regions of vertical frames [73]. No segmentation was used. There was one discrete HMM for each city class. The system achieved a 65 percent word-level recognition rate without the use of contextual information on a database of more than 17,000 images of the 198 names of cities in Iran.

A 2003 approach by Pechwitz and Märgner used 160 semi-continuous HMMs representing the characters or shapes [43]. It thinned each word and used columns of pixels in the blurred thinned image as features. The models were combined into a word model for each of 946 valid city names. The system obtained an 89 percent word-level recognition rate using the IFN/ENIT database (26,459 images of Tunisian city-names).

Khorsheed applied an HMM recognizer with image skeletonization to the recognition of text in an ancient manuscript (2003) [46]. No segmentation was done. One HMM was constructed from 32 individual character HMMs, each with unrestricted jump margin. Structural features were used and the recognition rate was 87 percent (72 percent) with (without) spell-check. The rate for the correct result being in the top five choices was 97 percent (81 percent). The test set was 405 character samples of a single font, extracted from a single manuscript.

Safabakhsh and Adibi applied a continuous-density variable-duration hidden Markov model [88] to the recognition of handwritten Persian words in the Nastaaligh style (2005) [68]. This style contains many vertically overlapping letters and sloped letter sequences, which present problems for the ordering of characters and for baseline detection. Their system removed ascenders and descenders before the primary recognition stage to avoid incorrect orderings and was baseline-independent. Words were oversegmented into pseudocharacters using local minima of their upper contour, similar to [74]. Eight features were computed for each pseudocharacter (Table 6). The HMM was path-discriminant and included 25 character states, each of which was divided into up to four substates to indicate position-dependent shapes. The lexicon consisted of 50 words chosen to include all characters and compound forms and the training set contained two 50-word scripts from each of seven writers. On a test set of two 50-word scripts from two different writers and omitting words that showed error in an earlier stage of the method, the system achieved a 69 percent recognition rate with five iterations of the recognition step and a 91 percent rate with 20 iterations. The rates were 52.38 percent and 90.48 percent on 21 words not in the lexicon.

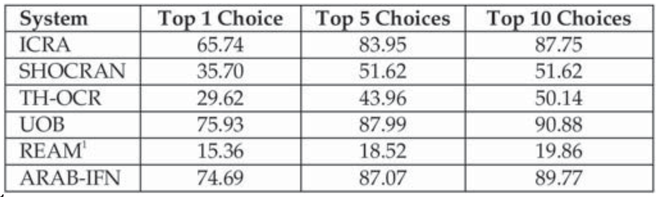
3.4.4 Hybrids

In 2004, Alma'adeed et al. combined a rule-based recognizer with a set of HMMs to recognize words in a bank-check lexicon of 47 words [60]. Preprocessing normalized the text with respect to slant, slope, and letter height. A skeleton representation normalized for stroke width and no segmentation was done. The rule-based engine used ascenders, descenders, and other structural features to separate the data into groups of words (reduce the lexicon) and an HMM classifier for each group used frame-based features to determine the word. To train the HMMs, words were separated into letters or subletters that were transformed into feature vectors and partitioned by a clustering algorithm. In testing, the feature vectors were obtained by vector-quantizing observation vectors obtained from frames of the image. The HMMs had 55 possible states, corresponding to letters or subletters in the data set and codebook sizes between 80 and 120. The system was tested on about 4,700 words collectively written by 100 writers, excluding about 10 percent of the words due to errors in baseline detection and preprocessing. A near 60 percent recognition rate was achieved. An earlier version obtained a 45 percent recognition rate (2002) [59].

Souici-Meslati and Sellami presented a hybrid approach to the recognition of literal amounts on bank checks in 2004 [45]. The recognizer was a neural network whose structure was defined by a rule-based method. Preprocessing included binarization, smoothing, normalization for word size, and baseline detection. The representation was Freeman chain code of the text's contour, and the features were loops, dots, connected components, ascenders, and descenders. Segmentation was not performed. Data for training and testing consisted of 480 and 1,200 words, respectively. The system obtained a 93 percent recognition rate, outperforming separate neural network and rule-based systems which each obtained a rate of approximately 85 percent. Also, in 2004, this group proposed another method for this task [69]. The features were still structural and the representation chain code, but the recognizer differed. Three classifiers ran in parallel: neural network, k-nearest-neighbor, and fuzzy k-nearest-neighbor. The outputs were combined by word-level score summation and syntactic postprocessing to obtain a valid phrase. One thousand two hundred words by 100 writers were used for training and 3,600 words for testing. The recognition rate was 96 percent, about 4 percent higher than the average of the individual classifiers. For both methods, the lexicon contained 48 words. Third, this group presented a system to recognize city names in Algerian postal addresses (Souici et al. 2004 [70]). The recognizer was a knowledge-based neural network such as in [45]. The recognition rate was 92 percent with a 55-word lexicon. Separate training and testing data sets each contained 550 words (each of 55 words written by 10 writers).

In 2005, Farah et al. extended the work of [69] to study the effects of multiclassifier systems on recognition rates [89]. Separate ANNs processed structural and statistical features and eight classifier combination methods were tested. The highest rate of 95.2 percent was observed with an ANN for combination. This rate surpassed the 89.3 percent observed

TABLE 8
Recognition Rates from ICDAR 2005 Competition, in Percent



System	Top 1 Choice	Top 5 Choices	Top 10 Choices
ICRA	65.74	83.95	87.75
SHOCRAN	35.70	51.62	51.62
TH-OCR	29.62	43.96	50.14
UOB	75.93	87.99	90.88
REAM[1]	15.36	18.52	19.86
ARAB-IFN	74.69	87.07	89.77

[1] REAM was tested on a reduced set of 3,000 images due to a failure on a full set of images.

with one ANN processing the two feature types collectively. Also, higher rates were achieved when both initial ANNs were trained on the same 2,400-word data set than when each was trained on half of that set. Test data was a separate set of 2,400 words, and all words were literal amounts.

3.5 Competition

The International Conference on Document Analysis and Recognition held the first international Arabic handwriting recognition competition in 2005 [90]. Five groups submitted systems trained on the IFN/ENIT database. Those systems were tested on a portion of that database and on a set of 6,033 new images. The "ICRA" system by Kader was based on subwords and recognized subwords, then words using neural networks. There are no publications on it yet. "SHOCRAN" was sent by a group from Egypt, and no further information is available due to a confidentiality request. Based on a machine-print OCR system [91], "TH-OCR" by Jin et al. performs segmentation using structural or geometrical characteristics and character recognition using statistical methods. "UOB" by Mokbel is a pure HMM system based on a speech recognition system [92] and uses the feature extraction module of [78]. Finally, "REAM" by Touj et al. uses the planar Markov model as described previously ([93]). Recognition rates of the competition hosts' "ARAB-IFN" system (Pechwitz and Märgner 2003 [43]) were also shown. For all six systems and both test sets, recognition rates were shown for the correct answer being in the top 1, 5, and 10 choices. The rates for the new images are shown in Table 8. UOB and ARAB-IFN scored the highest perhaps due to the power of the frame-based HMM strategy [32] in which features computed on vertical strips of the image are fed into an HMM. Further, the baseline-dependent features give consistent improvement, supporting the thesis of [78]. The high performance of ICRA is less expected, but insufficient detail is provided to understand the algorithm since even the features are not stated.

4 Future Work and Conclusions

Some problems are application-specific, with the goal of increasing recognition rates on bank checks, mail addresses, forms, and manuscripts. The surveyed studies have demonstrated the feasibility of these applications. Current limitations include restrictive lexicons and restrictions on the appearance of the text. The highest rates were achieved on restricted tasks, such as the 97 percent rate achieved by [72] on a 70-word lexicon of words from bank checks and the 97 percent rate achieved by [46] on 405 characters of a single font. Future work includes the development of algorithms for use with larger lexicons and more variability in the appearance of words.

Open problems are also related to applications that have been only preliminarily considered, including the recognition of "free handwriting" such as found in a handwritten correspondence. A full solution would require linguistic information like co-occurrence frequency of adjacent words and labeling of words according to parts of speech. The n-gram model, which is used in continuous speech recognition systems, uses knowledge of the prior n words. To our knowledge, such techniques have not been applied to Arabic handwriting. They remain largely unexplored for any script, but successes with the Latin script suggest their benefit. Related issues include the use of very large lexicons (~50,000 words) of the common words in a language which would necessitate lexicon-reduction techniques in recognition.

Also, knowledge of word morphology can enable a system to recognize a word that is not in the lexicon [94], [95]. Morphology is the area of linguistics that investigates word formation, including affixes, roots, and patterns. Arabic is a Semitic language and, as such, exhibits a systematic yet complex structure. A morphological system for analysis and generation was presented in 1989 [96]. See [97] for a comprehensive survey of Arabic morphological analysis techniques. In 2005, Kanoun et al. [98] presented a system that used such knowledge to recognize machine-printed words. The image analysis side was simplified to a domain of one font and recognition by Euclidean distance to templates so that the novel morphological strategy could be explored. Besides assisting OCR, related applications include Web-based translation [99], search engines [100], and information retrieval [101].

Also begun but open is the interpretation of large classes of manuscripts without font customization. It may require new models of character shapes that can be generalized over many fonts. People can read poor-quality writing and fonts they have never seen before, but many systems use vast training sets instead of attempting to incorporate this knowledge. A second requirement to advance the recognition of manuscripts is publicly available imagery for research purposes. Like those discussed in Section 2.5, databases for this domain must contain corresponding text and must cover a wide range of writing styles.

This paper has described research on the automatic recognition of Arabic handwriting. It has discussed methods and classified them according to several criteria. It is the first Arabic character recognition survey to give testing procedures and recognition rates for as many systems as possible and the first to focus on handwriting. Research in this area has progressed much in the past 20 years and algorithm styles have changed as computational power has increased and as related fields have developed: for example, the increased use of statistical techniques. However, current systems are applied to restricted domains or have only been tested on small data sets. Future research and testing are needed to develop systems for widespread use.

Acknowledgments

All images of handwritten Arabic words and letters were obtained from the IFN/ENIT database. This work was supported by a DCI Postdoctoral Fellowship. The authors would like to thank Dr. Paul Thouin and Faisal Farooq for assisting with this paper and the anonymous reviewers for suggestions that much improved it.

REFERENCES

[1] *Ethnologue: Languages of the World,* 14th ed. SIL Int'l, 2000.

[2] Y.S. Chen and W.H. Hsu, "A New Parallel Thinning Algorithm for Binary Image," *Proc. Nat'l Computer Symp.*, pp. 295-299, 1985.

[3] H. Freeman, "On the Encoding of Arbitrary Geometric Configurations," *IRE Trans. Electronic Computing,* vol. 10, pp. 260-268, 1961.

[4] S. Madhvanath, G. Kim, and V. Govindaraju, "Chain Code Processing for Handwritten Word Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, pp. 928-932, 1999.

[5] M. Pechwitz and V. Märgner, "Baseline Estimation for Arabic Handwritten Words," *Proc. Eighth Int'l Workshop Frontiers in Handwriting Recognition,* pp. 479-484, 2002.

[6] F. Farooq, V. Govindaraju, and M. Perrone, "Preprocessing Methods for Handwritten Arabic Documents," *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 267-271, 2005.

[7] M. Soleymani and F. Razzazi, "An Efficient Front-End System for Isolated Persian/Arabic Character Recognition of Handwritten Data-Entry Forms," *Int'l J. Computational Intelligence,* vol. 1, pp. 193-196, 2003.

[8] H.Y. Abdelazim, "Arabic Script Recognition Using Hopfield Networks," *Int'l J. Computers and Their Applications,* vol. 2, pp. 43-49, 1995.

[9] M.S. El-Wakil and A. Shoukry, "On-Line Recognition of Handwritten Isolated Arabic Characters," *Pattern Recognition,* vol. 22, pp. 97-105, 1989.

[10] S. Al-Emami and M. Usher, "On-Line Recognition of Handwritten Arabic Characters," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, pp. 704-710, 1990.

[11] I.S.I. Abuhaiba and P. Ahmed, "Restoration of Temporal Information in Off-Line Arabic Handwriting," *Pattern Recognition,* vol. 26, pp. 1009-1017, 1993.

[12] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification,* second ed. John Wiley and Sons, 2001.

[13] L.R. Rabiner and B.H. Juang, "An Introduction to Hidden Markov Models," *IEEE Acoustics, Speech, and Signal Processing Magazine,* vol. 3, pp. 4-16, 1986.

[14] N. BenAmara, A. Belaïd, and N. Ellouze, "Utilisation des Modèles Markoviens en Reconnaissance de l'Écriture Arabe: Etat de L'art," *Proc. Colloque Int'l Francophone sur l'Ecrit et le Document,* 2000.

[15] S. Madhvanath and V. Govindaraju, "The Role of Holistic Paradigms in Handwritten Word Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, pp. 149-164, 2001.

[16] B. Al-Badr and S.A. Mahmoud, "Survey and Bibliography of Arabic Optical Text Recognition," *Signal Processing,* vol. 41, pp. 49-77, 1995.

[17] A. Amin, "Offline Arabic Character Recognition: The State of the Art," *Pattern Recognition,* vol. 31, pp. 517-530, 1998.

[18] M.S. Khorsheed, "Off-Line Arabic Character Recognition—A Review," *Pattern Analysis and Applications,* vol. 5, pp. 31-45, 2002.

[19] A.S. Eldin and A.S. Nouh, "Arabic Character Recognition: A Survey," *Proc. SPIE Conf. Optical Pattern Recognition,* pp. 331-340, 1998.

[20] A. Nouh, A. Sultan, and R. Tolba, "An Approach for Arabic Characters Recognition," *J. Eng. Science,* vol. 6, pp. 185-191, 1980.

[21] A. Nouh, A. Sultan, and R. Tolba, "On Feature Extraction and Selection for Arabic Character Recognition," *Arab Gulf J. Scientific Research,* vol. 2, pp. 329-347, 1984.

[22] B. Parhami and M. Taraghi, "Automatic Recognition of Printed Farsi Texts," *Pattern Recognition,* vol. 14, pp. 395-403, 1981.

[23] A. Amin and G. Masini, "Machine Recognition of Multi-Font Printed Arabic Texts," *Proc. Int'l Conf. Pattern Recognition,* pp. 392-395, 1986.

[24] T.S. El-Sheikh and R.M. Guindi, "Computer Recognition of Arabic Cursive Scripts," *Pattern Recognition,* vol. 21, pp. 293-302, 1988.

[25] S. Sami El-Dabi, R. Ramsis, and A. Kamel, "Arabic Character Recognition System: A Statistical Approach for Recognizing Cursive Typewritten Text," *Pattern Recognition,* vol. 23, pp. 485-495, 1990.

[26] M.R. Hashemi, O. Fatemi, and R. Safavi, "Persian Cursive Script Recognition," *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 869-873, 1995.

[27] A. Ymin and Y. Aoki, "On the Segmentation of Multi-Font Printed Uygur Scripts," *Proc. 13th Int'l Conf. Pattern Recognition,* vol. 3, pp. 215-219, 1996.

[28] B. Al-Badr and R. Haralick, "Segmentation-Free Word Recognition with Application to Arabic," *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 355-359, 1995.

[29] B. Al-Badr and R. Haralick, "A Segmentation-Free Approach to Text Recognition with Application to Arabic Text," *Int'l J. Document Analysis and Recognition,* vol. 1, pp. 147-166, 1998.

[30] J. Alherbish, R.A. Ammar, and M. Abdalla, "Arabic Character Recognition in a Multiprocessing Environment," *Proc. IEEE Symp. Computers and Comm.*, pp. 286-292, 1997.

[31] M.S. Khorsheed and W.F. Clocksin, "Structural Features of Cursive Arabic Script," *Proc. British Machine Vision Conf.*, pp. 422-431, 1999.

[32] J. Makhoul, R. Schwartz, C. Lapre, and I. Bazzi, "A Script-Independent Methodology for Optical Character Recognition," *Pattern Recognition,* vol. 31, pp. 1285-1294, 1998.

[33] P. Natarajan, M. Decerbo, T. Keller, R. Schwartz, and J. Makhoul, "Porting the BBN BYBLOS OCR System to New Languages," *Proc. Symp. Document Image Understanding Technology,* pp. 47-52, 2003.

[34] M. Decerbo, P. Natarajan, R. Prasad, E. MacRostie, and A. Ravindran, "Performance Improvements to the BBN Byblos OCR System," *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 411-415, 2005.

[35] I. Bazzi, R. Schwartz, and J. Makhoul, "An Omnifont Open-Vocabulary OCR System for English and Arabic," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, pp. 495-504, 1999.

[36] J. Trenkle, A. Gillies, E. Erlandson, S. Schlosser, and S. Cavin, "Advances in Arabic Text Recognition," *Proc. Symp. Document Image Understanding Technology,* 2001.

[37] J. Trenkle, A. Gillies, E. Erlandson, and S. Schlosser, "Arabic Character Recognition," *Proc. Symp. Document Image Understanding Technology,* pp. 191-195, 1995.

[38] A. Gillies, E. Erlandson, J. Trenkle, and S. Schlosser, "Arabic Text Recognition System," *Proc. Symp. Document Image Understanding Technology,* 1999.

[39] J. Trenkle, A. Gillies, and S. Schlosser, "An Off-Line Arabic Recognition System for Machine-Printed Arabic Documents," *Proc. Symp. Document Image Understanding Technology,* pp. 155-161, 1997.

[40] L. Hamami and D. Berkani, "Recognition System for Printed Multifont and Multisize Arabic Characters," *The Arabian J. Science and Eng.*, vol. 27, pp. 57-72, 2002.

[41] S.A. Al-Qahtani and M.S. Khorsheed, "An Omni-Font HTK-Based Arabic Recognition System," *Proc. Eighth IASTED Int'l Conf. Artificial Intelligence and Soft Computing,* 2004.

[42] S.A. Al-Qahtani and M.S. Khorsheed, "A HTK-Based System to Recognise Arabic Script," *Proc. Fourth IASTED Int'l Conf. Visualization, Imaging, and Image Processing,* 2004.

[43] M. Pechwitz and V. Märgner, "HMM Based Approach for Handwritten Arabic Word Recognition Using the IFN/ENIT-Database," *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 890-894, 2003.

[44] A. Amin, "Recognition of Hand-Printed Characters Based on Structural Description and Inductive Logic Programming," *Pattern Recognition Letters,* vol. 24, pp. 3187-3196, 2003.

[45] L. Souici-Meslati and M. Sellami, "A Hybrid Approach for Arabic Literal Amounts Recognition," *The Arabian J. Science and Eng.*, vol. 29, pp. 177-194, 2004.

[46] M.S. Khorsheed, "Recognising Handwritten Arabic Manuscripts Using a Single Hidden Markov Model," *Pattern Recognition Letters,* vol. 24, pp. 2235-2242, 2003.

[47] R. Davidson and R. Hopely, "Arabic and Persian OCR Training and Test Data Sets," *Proc. Symp. Document Image Understanding Technology,* pp. 303-307, 1997.

[48] T. Kanungo, G. Marton, and O. Bulbul, "OmniPage vs. Sakhr: Paired Model Evaluation of Two Arabic OCR Products," *Proc. SPIE Conf. Document Recognition and Retrieval (VI),* pp. 109-121, 1999.

[49] I.S. Abuhaiba, "A Discrete Arabic Script for Better Automatic Document Understanding," *The Arabian J. Science and Eng.*, vol. 28, pp. 77-94, 2003.

[50] J.J. Hull, "A Database for Handwritten Text Recognition Research," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, pp. 550-554, 1994.

[51] S. Alma'adeed, D. Elliman, and C.A. Higgins, "A Data Base for Arabic Handwritten Text Recognition Research," *Proc. Eighth Int'l Workshop Frontiers in Handwriting Recognition,* pp. 485-489, 2002.

[52] Y. Al-Ohali, M. Cheriet, and C. Suen, "Databases for Recognition of Handwritten Arabic Cheques," *Pattern Recognition,* vol. 36, pp. 111-121, 2003.

[53] I.S.I. Abuhaiba, S.A. Mahmoud, and R.J. Green, "Recognition of Handwritten Cursive Arabic Characters," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, pp. 664-672, 1994.

[54] A. Amin, H. Al-Sadoun, and S. Fischer, “Hand-Printed Arabic Character Recognition System Using an Artificial Network,” *Pattern Recognition,* vol. 29, pp. 663-675, 1996.

[55] I.S.I. Abuhaiba, M.J.J. Holt, and S. Datta, “Recognition of Off-Line Cursive Handwriting,” *Computer Vision and Image Understanding,* vol. 71, pp. 19-38, 1998.

[56] A. Dehghani, F. Shabani, and P. Nava, “Off-Line Recognition of Isolated Persian Handwritten Characters Using Multiple Hidden Markov Models,” *Proc. Int'l Conf. Information Technology: Coding and Computing,* pp. 506-510, 2001.

[57] A.A. Al-Shaher and E.R. Hancock, “Learning Mixtures of Point Distribution Models with the EM Algorithm,” *Pattern Recognition,* vol. 36, pp. 2805-2818, 2003.

[58] S. Mozaffari, K. Faez, and M. Ziaratban, “Structural Decomposition and Statistical Description of Farsi/Arabic Handwritten Numeric Characters,” *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 237-241, 2005.

[59] S. Alma'adeed, C. Higgens, and D. Elliman, “Recognition of Off-Line Handwritten Arabic Words Using Hidden Markov Model Approach,” *Proc. 16th Int'l Conf. Pattern Recognition,* vol. 3, pp. 481-484, 2002.

[60] S. Alma'adeed, C. Higgens, and D. Elliman, “Off-Line Recognition of Handwritten Arabic Words Using Multiple Hidden Markov Models,” *Knowledge-Based Systems,* vol. 17, pp. 75-79, 2004.

[61] R. Haraty and A. Hamid, “Segmenting Handwritten Arabic Text,” *Proc. Int'l Conf. Computer Science, Software Eng., Information Technology, e-Business, and Applications,* 2002.

[62] R. Haraty and H. El-Zabadani, “Abjad: An Off-Line Arabic Handwritten Recognition System,” *Proc. Int'l Arab Conf. Information Technology,* 2002.

[63] R. Haraty and C. Ghaddar, “Neuro-Classification for Handwritten Arabic Text,” *Proc. ACS/IEEE Int'l Conf. Computer Systems and Applications,* 2003.

[64] R. Haraty and C. Ghaddar, “Arabic Text Recognition,” *Int'l Arab J. Information Technology,* vol. 1, pp. 156-163, 2004.

[65] M.M.M. Fahmy and S. Al Ali, “Automatic Recognition of Handwritten Arabic Characters Using Their Geometrical Features,” *Studies in Informatics and Control J.,* vol. 10, 2001.

[66] H. Goraine, M. Usher, and S. Al-Emami, “Off-Line Arabic Character Recognition,” *Computer,* vol. 25, pp. 71-74, 1992.

[67] H. Almuallim and S. Yamaguchi, “ A Method of Recognition of Arabic Cursive Handwriting,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 9, pp. 715-722, 1987.

[68] R. Safabakhsh and P. Adibi, “Nastaaligh Handwritten Word Recognition Using a Continuous-Density Variable-Duration HMM,” *The Arabian J. Science and Eng.,* vol. 30, pp. 95-118, 2005.

[69] N. Farah, L. Souici, L. Farah, and M. Sellami, “Arabic Words Recognition with Classifiers Combination: An Application to Literal Amounts,” *Proc. Artificial Intelligence: Methodology, Systems, and Applications,* pp. 420-429, 2004.

[70] L. Souici, N. Farah, T. Sari, and M. Sellami, “Rule Based Neural Networks Construction for Handwritten Arabic City-Names Recognition,” *Proc. Artificial Intelligence: Methodology, Systems, and Applications,* pp. 331-340, 2004.

[71] T. Sari, L. Souici, and M. Sellami, “Off-Line Handwritten Arabic Character Segmentation Algorithm: ACSA,” *Proc. Int'l Workshop Frontiers in Handwriting Recognition,* pp. 452-457, 2002.

[72] S. Snoussi Maddouri, H. Amiri, A. Belaid, and C. Choisy, “Combination of Local and Global Vision Modeling for Arabic Handwritten Words Recognition,” *Proc. Int'l Conf. Frontiers in Handwriting Recognition,* pp. 128-135, 2002.

[73] M. Dehghan, K. Faez, M. Ahmadi, and M. Shridhar, “Handwritten Farsi (Arabic) Word Recognition: A Holistic Approach Using Discrete HMM,” *Pattern Recognition,* vol. 34, pp. 1057-1065, 2001.

[74] G. Olivier, H. Miled, K. Romeo, and Y. Lecourtier, “Segmentation and Coding of Arabic Handwritten Words,” *Proc. 13th Int'l Conf. Pattern Recognition,* vol. 3, pp. 264-268, 1996.

[75] H. Miled and N.E. Ben Amara, “Planar Markov Modeling for Arabic Writing Recognition: Advancement State,” *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 69-73, 2001.

[76] K. Mostafa and A.M. Darwish, “Robust Base-Line Independent Algorithms for Segmentation and Reconstruction of Arabic Handwritten Cursive Script,” *Proc. IS&T/SPIE Conf. Document Recognition and Retrieval VI,* vol. 3651, pp. 73-83, 1999.

[77] K. Romeo-Pakker, H. Miled, and Y. Lecourtier, “A New Approach for Latin/Arabic Character Segmentation,” *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 874-877, 1995.

[78] R. El-Hajj, L. Likforman-Sulem, and C. Mokbel, “Arabic Handwriting Recognition Using Baseline Dependant Features and Hidden Markov Modeling,” *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 893-897, 2005.

[79] W.F. Clocksin and P.P.J. Fernando, “Towards Automatic Transcription of Syriac Handwriting,” *Proc. Int'l Conf. Image Analysis and Processing,* pp. 664-669, 2003.

[80] D. Motawa, A. Amin, and R. Sabourin, “Segmentation of Arabic Cursive Script,” *Proc. Int'l Conf. Document Analysis and Recognition,* vol. 2, pp. 625-628, 1997.

[81] H. Al-Yousefi and S.S. Udpa, “Recognition of Arabic Characters,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, pp. 853-857, 1992.

[82] H.S. Park and S.W. Lee, “Off-Line Recognition of Large-Set Handwritten Characters with Multiple Hidden Markov Models,” *Pattern Recognition,* vol. 29, pp. 231-244, 1996.

[83] A.A. Al-Shaher and E.R. Hancock, “Arabic Character Recognition Using Shape Mixtures,” *Proc. British Machine Vision Conf.,* pp. 497-506, 2002.

[84] T.F. Cootes and C.J. Taylor, “A Mixture Model for Representing Shape Variation,” *Image and Vision Computing,* vol. 17, pp. 567-573, 1999.

[85] M. Sellami, L. Souici, T. Sari, and Z. Zemirli, “Contribution à la Reconnaissance de Mots Arabes Manuscrits,” *Proc. Colloque Africain de Recherche en Informatique,* pp. 122-124, 1998.

[86] L. Lorigo and V. Govindaraju, “Segmentation and Pre-Recognition of Arabic Handwriting,” *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 605-609, 2005.

[87] J.R. Quinlan, “Learning Logical Definitions from Relations,” *Machine Learning,* vol. 5, pp. 239-266, 1990.

[88] 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, 1995.

[89] N. Farah, A. Ennaji, T. Khadir, and M. Sellami, “Benefits of Multi-Classifier Systems for Arabic Handwritten Words Recognition,” *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 222-226, 2005.

[90] V. Märgner, M. Pechwitz, and H. ElAbed, “ICDAR 2005 Arabic Handwriting Recognition Competition,” *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 70-74, 2005.

[91] J. Jin, H. Wang, X. Ding, and L. Peng, “Printed Arabic Document Recognition System,” *Proc. SPIE-IS&T Electronic Imaging,* vol. 5676, pp. 48-55, 2005.

[92] C. Mokbel, H. Abi Akl, and H. Greige, “Automatic Speech Recognition of Arabic Digits over Telefone Network,” *Proc. Int'l Conf. Research Trends in Science and Technology,* 2002.

[93] S.M. Touj and N. Ben Amara, “Arabic Handwritten Words Recognition Based on a Planar Hidden Markov Model,” *Int'l Arab J. Information Technology,* vol. 2, 2005.

[94] H. Bunke, S. Bengio, and A. Vinciarelli, “Off-Line Recognition of Unconstrained Handwritten Texts Using HMMS and Statistical Language Models,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, pp. 709-720, 2004.

[95] G. Kim, V. Govindaraju, and S. Srihari, “Architecture for Handwritten Text Recognition Systems,” *Advances in Handwriting Recognition, Series in Machine Perception and Artificial Intelligence,* pp. 163-172, 1999.

[96] T.A. El-Sadany and M.A. Hashish, “An Arabic Morphological System,” *IBM Sytems J.,* vol. 28, pp. 600-612, 1989.

[97] I.A. Al-Sughaiyer and I.A. Al-Kharashi, “Arabic Morphological Analysis Techniques: A Comprehensive Survey,” *J. Am. Soc. Information Science and Technology,* vol. 55, pp. 189-213, 2004.

[98] S. Kanoun, A.M. Alimi, and Y. Lecourtier, “*Affixal* Approach for Arabic Decomposable Vocabulary Recognition: A Validation on Printed Word in Only One Font,” *Proc. Int'l Conf. Document Analaysis and Recognition,* pp. 1025-1029, 2005.

[99] M. Yaseen, B. Haddad, H. Papageorgiou, S. Piperidis, M. Hattab, N. Theophilopoulos, and S. Krauwer, “A Term Base Translator over the Web,” *Proc. ACL/EACL 2001 Workshop—ARABIC Language Processing: Status and Prospects,* pp. 58-65, 2001.

[100] I.A. Al-Kharashi, “A Web Search Engine for Indexing, Searching and Publishing Arabic Bibliographic Databases,” *Proc. Internet Global Summit,* 1999.

[101] I.A. Al-Kharashi and M.W. Evens, “Comparing Words, Stems, and Roots as Index Terms in an Arabic Information Retrieval System,” *J. Am. Soc. Information Science,* vol. 45, pp. 548-560, 1994.

Liana M. Lorigo received the BA degree in computer science and mathematics from Cornell University in 1994, an MS degree in computer science from MIT in 1996, and the PhD degree in computer science from MIT in 2000. She was affiliated with Teradyne, Inc. from 2000 to 2003 and is currently affiliated with the Department of Computer Science and Engineering at the University at Buffalo. In 1999, she received the François Erbsmann Award for best student presentation at the IEEE International Conference on Information Processing in Medical Imaging. She is a member of the IEEE Computer Society. Her research interests include handwriting recognition and medical image analysis.

Venu Govindaraju received the B-Tech degree (honors) from the Indian Institute of Technology (IIT), Kharagpur, India, in 1986, and the PhD degree in computer science from University at Buffalo (UB), State University of New York in 1992. He is a professor of computer science and engineering at UB. Dr. Govindaraju's research is focused on pattern recognition applications in the areas of biometrics and digital libraries. He is a recipient of the ICDAR Outstanding Young Investigator Award (2001) and the MIT Global Indus Technovators Award (2004). He was elected a fellow of the International Association of Pattern Recognition (IAPR) in 2004. He is a senior member of the IEEE Computer Society.