Neural Computing and Applications (2021) 33:3011–3034
https://doi.org/10.1007/s00521-020-05137-6

REVIEW

Automatic recognition of handwritten Arabic characters: a comprehensive review

Hossam Magdy Balaha[1] · Hesham Arafat Ali[1] · Mahmoud Badawy[2,3]

Received: 20 February 2020 / Accepted: 16 June 2020 / Published online: 17 July 2020

Abstract
The paper is a comprehensive review of the current research trends in the area of Arabic language especially state-of-the-art approaches to highlight the current status of diverse research aspects of that area to facilitate the adaption and extension of previous systems into new applications and systems. The Arabic language has deep, widespread and unexplored scope to research although the tremendous effort and researches that had been done previously. Modern state-of-the-art methods and approaches with fewer errors are required according to the high speed of hardware and technology development. The focus of this article will be on the offline Arabic handwritten text recognition as it is one of the most important topics in the Arabic scope. The main objective of this paper is critically analyzing the current researches to identify the problem areas and challenges faced by the previous researchers. This identification is intended to provide many recommendations for future advances in the area. It also compares and contrasts technical challenges, methods and the performances of handwritten text recognition previous researches works. It summarizes the critical problems and enumerates issues that should be considered when addressing these tasks. It also shows some of the Arabic datasets that can be used as inputs and benchmarks for training, testing and comparisons. Finally, it provides a fundamental comparison and discussion of some of the remaining open problems and trends in that field.

Keywords Arabic handwritten character recognition · Artificial intelligence · Classification · Convolutional neural network · Deep learning · Feature extractionNeural network · Optical character recognition · Preprocessing

Abbreviations

Adam	An algorithm and not an acronym
AHCR	Arabic handwritten character recognition
AI	Artificial intelligence
ANN	Artificial neural network
ASCII	American Standard Code for Information Interchange
CER	Character error rate
CS	Computer science
CTC	Connectionist temporal classification
CMATER	Center for Microprocessor Application for Training Education and Research
CNN	Convolutional neural network
CV	Computer vision
CUDA	Compute Unified Device Architecture
DCT	Discrete cosine transformation
DL	Deep learning
DPI	Dots per inch
DWT	Discrete wavelet transform
EBPANN	Error back-propagation artificial neural network
EKG	Electrocardiogram
FC	Fully connected
FCM	Fuzzy C-means clustering
GPU	Graphical processing unit
HMM	Hidden Markov model
HCR	Handwritten character recognition
HOG	Histogram of oriented gradients
HDR	Handwritten digital recognition
HP	Hewlett–Packard

✉ Hossam Magdy Balaha
hossam.m.balaha@mans.edu.eg

Hesham Arafat Ali
h_arafat_ali@mans.edu.eg

Mahmoud Badawy
engbadawy@mans.edu.eg

[1] Computer Engineering and Systems Department, Faculty of Engineering, Mansoura University, Mansoura, Egypt

[2] Computer Engineering and Systems Department, Faculty of Engineering, Mansoura University, Mansoura, Egypt

[3] Department of Computer Science and Informatics, Taibah University Al Medina Al Munawara, Medina, Saudi Arabia

K-NN (KNN)	K-nearest neighbor
LR	Learning rate
LSTM	Long short-term memory network
LVQ	Learning vector quantization
MD-BLSTM	Multi-dimension bi-direction long short-term memory
ML	Machine learning
MLP	Multi-layer perceptron
MSA	Modern standard Arabic
NLP	Natural language processing
OCR	Optical character recognition
OIAHCR	Offline isolated Arabic handwritten character recognition system
PCA	Principal component analysis
PDA	Personal digital assistant
PPM	Pages per minute
PR	Pattern recognition
RBF	Radial basis function
ReLU	Rectified linear unit
RNN	Recurrent neural network
SAE	Stacked autoencoder
SGD	Stochastic gradient descent
SSD	Single shot multi-box detector
SVM	Support vector machine
TPU	Tensor processing unit
UN	United States
UTF	Unicode Transformation Format
WER	Word error rate
XML	eXtensible Markup Language

1 Introduction

Nowadays, Arabic "العربية" is the official language of almost 26 countries and it is spoken by 280 millions worldwide. It is one of the United Nations (UN) six official languages (Chinese, Arabic, English, French, Russian and Spanish). In addition to that, Persian (Farsi), Jawi, Kurdish, Urdu and Pashto languages use some of its words and structures.

Arabic is an extremely descriptive language where some languages have one word only for something. Arabic has many synonyms for different words such as "Faith", "Hope" and "Lion". A feature of the Arabic characters is that its characters do not have a fixed width or height, even in the printed form. The character size varies according to its shape which is affected by its position in the word. All of the previous characteristics represent major challenges for researchers in the field of Arabic language [1–5].

Arabic also plays a crucial role in the Islamic faith and belief as Arabic is the language of the Holy Quran "القران الكريم". Muslims hold the importance of the Arabic language at a very soaring with most of their confidence and thinking being inseparable from Arabic. They regard also that Arabic may be the mother of all found languages in the world.

Arabic is one of the oldest languages in the world with a richness of knowledge that many archeologists till nowadays are still trying to uncover the facts as it has broad roots going back to the sixth century. The Middle East has a rich storytelling history and has produced some of the most memorable stories in the world such as "Ali Baba" and "Alaa El-Din" [6]. Arabs have also made significant improvements in such areas as Literature, Mathematics, Navigation, Astrology and Architecture. In the Liberal Age, Arabic was the most far-reaching study of the modernizing trend of political and social thought in the Arab Middle East [7]. "Muhammad ibn Musa al-Khwarizmi" and "Jabir ibn Hayyan" are two examples of Arabs who made works and inventions in different fields such as Mathematics and Medicine.

Many people nowadays still take notes (for example) traditionally with pen and paper. There are many drawbacks to that approach. Handwritten text is difficult to be stored and accessed efficiently and suitably. Searching through them and sharing with others are tedious tasks. With the absence of that text in a digital form, a lot of important knowledge may get lost and not used effectively.

The process of identifying and recognizing different means on inputs such as images, speeches or streams of characters is called pattern recognition (PR). Stages in that process may involve measurement of the object to identify distinguishing attributes, extraction of features for the defining attributes and comparison with known patterns to determine whether there is a matching between them or not (a mismatching) [8].

Pattern recognition is used in different applications and sciences such as security, medicine, robotics, video editing, filming, remote sensing and many more. PR as mentioned has many applications and fields which can be summarized as follows: Character Recognition, Biometrics: (Face Recognition, Speech Recognition, Finger Print Recognition and Iris Recognition) and Diagnostic Systems: X-ray System and Electrocardiogram (EKG) Analysis.

The rest of this paper is organized as follows: In the next section, we describe the Arabic language in different research areas. In Sect. 3, we discuss, compare and criticize some of the character recognition systems. In Sect. 4, we make a detailed description of the Arabic language's different characteristics. In Sect. 5, we describe the steps, components, methods and techniques that can be used to build a character recognition system. In Sect. 6, we report some of the Arabic datasets that can be used in training, testing and validation. In Sect. 7, we discuss some of the remaining open problems, challenges and trends in that field. Finally, in Sect. 8, we provide a conclusion on this survey.

2 Arabic language research areas

Arabic has many challenges related to the PR approach in computer science (CS). Figure 1 summarizes some of these challenges. It is inspired from these authors [9–11].

2.1 Character recognition

Character recognition [12] is the ability to recognize images of typed, handwritten or printed text into the corresponding machine-encoded text. It can be classified into online and offline character recognition that both of them can be achieved for either printed or handwritten text using several segmentation strategies [13–15].

Handwritten text is a very general term, and it is required to narrow down the scope of the work by specifying a specific meaning of the handwritten text for our purposes. The work takes on the challenge of classifying the image of any handwritten word written in Arabic, which might be in the form of blocks.

It is one of the popular approaches that depends on scanning the textual image with a predefined sliding window, from which the required features are extracted and modeled after that. Handwritten character recognition (HCR) is an important field of research in artificial intelligence (AI), computer vision (CV) and PR. Associated with artificial neural networks (ANNs), such as multi-layer perceptron (MLP) or long short-term memory network (LSTM) and with a language model, these models yield good conversions in simple machine learning (ML) applications. Deep neural networks especially convolutional neural networks (CNNs) consist of several hidden layers with various architectures to train a model that can accurately classify words and produce a significant reduction of error rates.

2.2 Speech to text conversion

Speech to text conversion (speech recognition) is the ability to recognize and classify the speech utterance into words. An utterance is the speaking of a word. It can be a single word, a set of words, a sentence or even multiple sentences [16, 17]. There are three challenging approaches to speech recognition: acoustic phonetic approach, pattern recognition approach and artificial intelligence approach. Text to speech conversion is the opposite operation of the speech to text conversion which is related to how to convert the human voice into words or sentences.

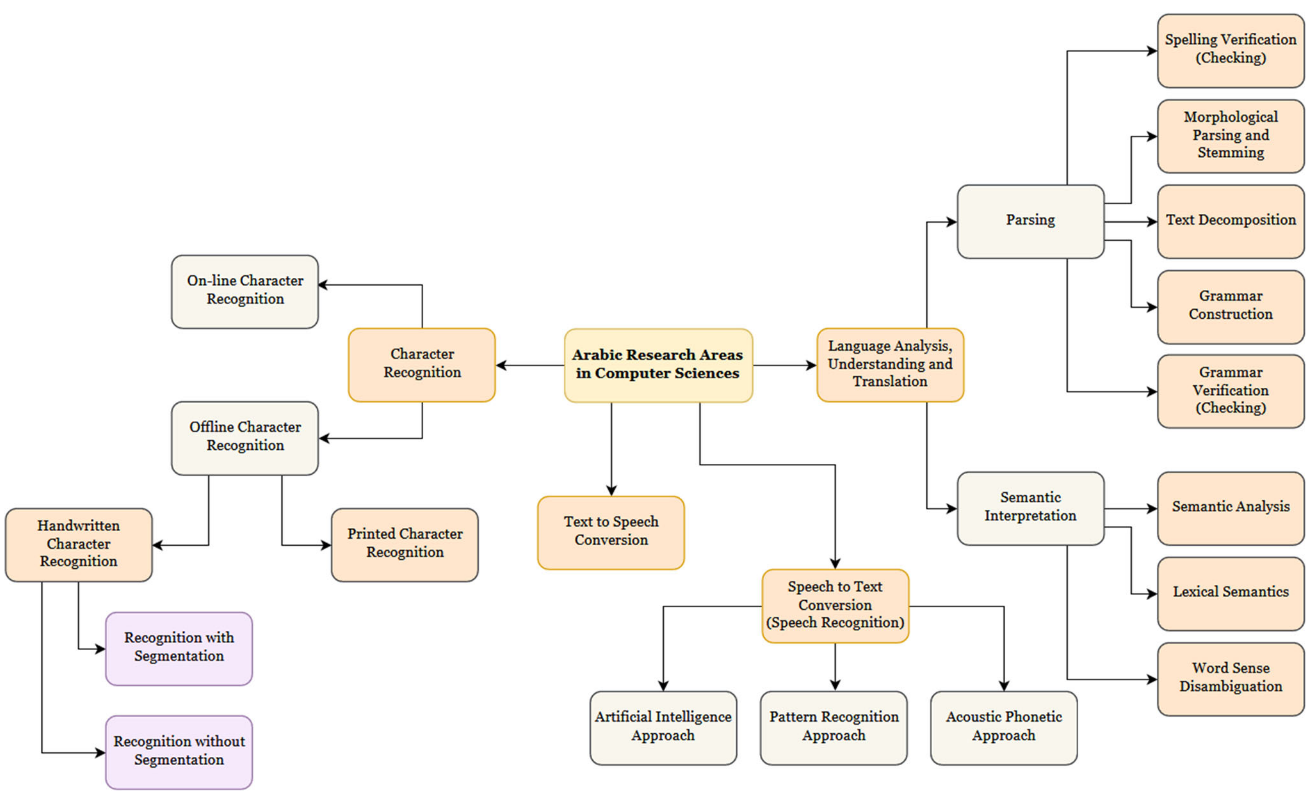


Fig. 1 Arabic challenges in computer science (CS)

2.3 Language analysis, understanding and translation

Natural language processing (NLP) [18, 19] is a large domain in CS that is related to the ability of the different machines such as computers and robots to understand the human language (Arabic in our case) the way it is written or spoken [20]. It covers a wide range of research areas and sub-areas such as parsing and semantics interpretation [21]. Parsing [22, 23] is usually applied to text data, and it is the process of converting that text into a format that can be processed and understood after that [24]. Almost all of the areas following parsing requires decomposition which is the conversion of the data (statements) into sub-words.

A spell verifier (checker) [25] is a program that tries to check the spelling of the input textual data automatically. Morphological parsing and stemming are used in the detection of inflectional or derivational forms and correction of the errors related to morphological affixes [26]. One of the examples of the morphological errors is the difference between "دعا" and "دعى." Grammar construction is the process of combining a set of related and correct words to build a phrase. Grammar verification (checking) is used to detect and correct grammar errors. Semantics [27] is the study of the text-linguistic meaning of words, phrases and sentences [28]. It includes the disambiguation of word sense, semantic analysis and lexical semantics. Semantic analysis is the building of meaning representations and assigning them to the textual data. Lexical semantics [29] is the detection of synonyms, homonyms, hyponyms, hypernyms, antonyms, etc.

Word sense disambiguation is an approach to allow the user to find the most appropriate and closest sense of an ambiguous word [30]. The lexical analysis is an approach to concatenate the sequence of characters into appropriate tokens with a suitable meaning which can be used to identify the infected Arabic words, for example [30]. Sentiment analysis is a trending approach to identify and extract the different subjective information in a sentence of a paragraph. It is extremely useful as it facilitates us to know a wide overview of the different public opinions or attitudes toward certain topics, issues, products or services [31, 32]. Semantic analysis is an approach to overcome the problem of gained information overload and how to extract the different semantic relationships between textual elements and deal with the elements redundancy [33].

3 Character recognition systems history

The automatic conversion of the text in Arabic handwritten documents has multiple applications, from real-time document processing to document understanding. Optical character recognition (OCR) is the technique used to identify the different characters from an image and convert them to the corresponding set of characters written as machine codes such as the American Standard Code for Information Interchange (ASCII) or Unicode. OCR had been used in many applications such as check validation, automatic text recognition and text digitalization. Many researchers around the world work on the problem of OCR using different approaches and techniques trying to reach an accuracy near the human reading capabilities. OCR speed should be increased to accommodate itself to the speed of the hardware processing such as a multi-paper scanner which can scan more than 100 to 250 images per minute. According to the available research articles, we can construct Table 1 that shows the different researches with their results, inputs, methods, used features, etc. It is sorted from the latest to the oldest. After the table, some of the pros and cons are mentioned.

3.1 Related work

As stated above, due to the importance of the Arabic language, lots of researches have been presented to solve the problems of the Arabic language. In [34], they proposed their own dataset which contained 560 handwritten character images (20 images for each character where each image was 64×64). Their dataset was divided into 50% for training and 50% for testing. They used image thresholding, noise removal, black space removal, images thinning, Canny edge detection and normalization in the preprocessing phase. They used the structural features, statistical features and global transformation in the feature extraction phase. They used support vector machine (SVM) classifier in the classification phase. Their highest accuracy was 99.64% using linear SVM classifier (with learning parameters: $cost(c) = 4$ and $\gamma = 0.25$). They used polynomial, radial basis function (RBF) kernel, sigmoid SVM classifiers, but the accuracies were 85%, 95% and 94%, respectively. The drawbacks of their system were: The size of their dataset was small, and they compared their system with different systems but with different datasets which were not an effective way of comparison.

In [35], they generated 548,325 images from papers in different font types such as "AdobeArabic", "Arial" and "Tahoma". They used multi-dimension bi-direction long short-term memory (MD-BLSTM) that has five-hidden layers, SoftMax output layer and connectionist temporal classification (CTC) cost function. They used stochastic gradient descent (SGD) with the optimizer, Tanh activation function, learning rate (LR) of 0.0003 and momentum of 0.9. They used the KACST Arabic corpus [47], cleaned it, adjusted the fixed number of characters for a text line and added noise. They gathered about 10 GB files with 19

Table 1 Arabic character recognition systems history

References	Years	Classifier	Feature extraction	Input (image)	Output	Recognition accuracy	Image size	Used dataset
[34]	2019	SVM	Multiple methods	Word	Name of the word	99.64%.	64×64	Proposed dataset
[35]	2018	MD-BLSTM	Automatic	Word	Name of the word	–	–	Proposed dataset
[36]	2017	CNN	Automatic	Number	Name of the number	97.4%	32×32	CMATERDB 3.3.1
[37]	2017	CNN and LSTM	Automatic	Word	Name of the word	38%	–	IAM Handwriting Dataset
[38]	2017	CNN	Automatic	Character	Name of the character	94.8%	28×28 or	AIA9k
						97.6%	32×32	AHCD
[39]	2017	CNN	Automatic	Scene	Name of the character	–	50×50	EAST
[40]	2017	Unsupervised learning with SAE	Automatic	Number	Name of the number	98.5%	28×28	MADBase
[41]	2016	CNN LeNet-5	Automatic	Number	Name of the number	88%	32×32	MADBase
[42, 43]	2016	MLP neural network	Zoning techniques	Character	Name of the character	94.75%	–	CENPARMI
[44]	2014	ANN	Multiple features	Character	Name of the character	66%:100%	–	CENPARMI
[45]	2014	Neural pattern recognition	Centralized Moments	–	–	76%	100×100	33 classes (proposed)
[46]	2003	Neural pattern recognition	Centralized Moments	–	–	73%	640×480	200 characters (proposed)

Arabic font files and corrected the wrong words using a language model. They compared their work with Tesseract 3.0 and Tesseract 4.0. They achieved the least error rates of 0.0988 and 0.2956 using the character error rate (CER) and the word error rate (WER), respectively.

In [36], they used the ANN approach by using the CNN models. They used the Center for Microprocessor Application for Training Education and Research Database (CMATERDB) 3.3.1 Arabic handwritten digit dataset. They worked on 32×32 grayscale images. They applied color inversion and normalization of the images. They used the rectified linear unit (ReLU) activation function in all hidden layers and SoftMax activation function in the output layer. They applied a 25% dropout to avoid overfitting. There were 4 fully connected (FC) layers (including the output one), 3 dropout layers and the input layer in their described MLP model. Two convolutional layers, 2 max-pooling layers, 2 dropout layers, 1 flatten layer and 2 FC layers were described in their CNN model. They applied their model using Keras with Theano backend. Their model configurations were 1000 epochs with a batch size of 128, AdaDelta optimizer and categorical cross-entropy loss function. The drawback of their paper was that they compared their work with a single paper [48] published in 2010, which may seem an old paper.

In [37], they used two main approaches. The first was done by using CNN to classify words, and the second was done by using LSTM to separate (segment) the characters of a word, classify them individually and reconstruct them in a single word again. They considered in their work that CNNs tended to work more efficiently on raw inputs pixels than features of the image. In the first approach (word-level classification), they used VGG-19, ResNet-18 and ResNet-34 CNN architectures, mini-batch size of 50, Adam optimizer, SoftMax activation function in the last layer and cross-entropy loss function. In the second approach (character-level classification), they used the Tesseract LSTM to segment words into characters, Adam optimizer and the CTC loss function. Table 2 shows their different classification results gained from both approaches.

In [38], they used the CNN approach on the AIA9k and AHCD databases. They proposed three ideas to improve the classification process: sparse interaction, parameter sharing and equivalent representation. They used the

Table 2 Classification results in the two used approaches in [37]

#	Architecture	Training accuracy (%)	Validation accuracy (%)	Test accuracy (%)
1	VGG-19	28	22	20
2	ResNet-18	31	23	22
3	ResNet-34	35	27	25
4	Char-Level	38	33	31

graphical processing units (GPUs) to solve the problem of latency since there is a GPU-enabled TensorFlow with support for Compute Unified Device Architecture (CUDA) acceleration. They used batch normalization and proved to cause an improvement in speed and accuracy. Three convolutional layers followed by a fully connected layer as hidden layers were created with the neglection of the max-pooling layers. They used the ReLU activation functions after each convolutional layer and the SoftMax activation function after the output layer. They used categorical cross-entropy as the cost function to update the weights and also used Adam optimizer and data augmentation techniques (such as translation by 3 pixels, rotation by $\pm 10°$ and adding Gaussian noise with a zero mean and standard deviation of 5). They divided the dataset into 70%, 15% and 15% for training, validation and testing datasets, respectively. They used a batch size of 100. Table 3 shows their tries in achieving their final results.

In [39], they used CNN in their paper. The input image was preprocessed at the size of 50×50 and at a grayscale level. Each image was saved in five various orientations. They used ReLU activation functions in the hidden layers, SoftMax activation function in the output layer, max-pooling layer after each convolution layer, and the kernels sizes of the two convolutional layers were 3×3 and 5×5. They segmented the characters of the Arabic scene image manually. Their best-reported accuracy was achieved when the filter size was 3×3, the stride was 2, and the LR was 0.005. The error rate was 18.24%. Table 4 shows their tries in achieving their final results.

They conducted that the convolutional networks were suitable, for instance, learning tasks rather than sequence learning and more appropriate in cursive scripts. They assumed in their paper that the convolutional networks extracted more detailed features through its strong layers compared with other normal feature extraction techniques. The drawback of their work was that they used a small number of data (which was 2450 in training) which impacted their accuracy and error rate.

In [40], they used an unsupervised learning approach with stacked autoencoder (SAE). They trained and tested their model on the MADBase database. They reached an average accuracy of 98.5%. Their first sparse autoencoder configurations were: The input layer size was 784, the hidden layer size was 392, and the output layer size was 784. Their second sparse autoencoder configurations were: The input layer size was 392, the hidden layer size was 196, and the output layer size was 392. Their first and

Table 3 Different training tries of [38]

#	Model	Data augmentation	Dataset	Dropout	No. of epochs	Testing accuracy (%)
1	1st hidden layer: 24	No	AHCD	0.5	10	92
2	2nd hidden layer: 48				20	93
3	3rd hidden layer: 64				28	94.5
	F.C. layer: 200					
4	1st hidden layer: 72	Yes			28	94.7
5	2nd hidden layer: 144	Reached 80 K			18	96.7
	3rd hidden layer: 192					
	F.C. layer: 400					
6	1st hidden layer: 24	No	AIA9K	0.75	17	93.4
7	2nd hidden layer: 48				17	94.2
	3rd hidden layer: 64					
	F.C. layer: 200					
8	1st hidden layer: 72				28	94.65
9	2nd hidden layer: 144			0.8	32	94.8
	3rd hidden layer: 192					
	F.C. layer: 400					

Table 4 Classification results in the different tries by [39]

#	Filter size	Stride	Learning rate	Error rate (%)
1	3×3	1	0.005	14.57
2	3×3	1	0.5	20.93
3	3×3	2	0.005	18.24
4	3×3	2	0.5	25.59
5	5×5	1	0.005	19.75
6	5×5	1	0.5	29.01
7	5×5	2	0.005	22.20
8	5×5	2	0.5	33.97

second sparse autoencoders used L2 regularization. The last layer was a SoftMax layer of an output of 10 classes. The features were extracted from 28×28 pixels of images.

In [41], they worked on the handwritten digital recognition (HDR) system and the LeNet-5 CNN model. They used 8 layers: input, output, four convolutional and two FC layers. The used images were grayscale with a size 32×32. They used the MADBase dataset. They trained the model using 30 iterations but reported their best accuracy in the 12th iteration. They reached misclassification error; 1% in training and 12% in testing. The drawback of their work was: They did not try other models nor modify the current LeNet-5 model. They also compared their work with other works that had different datasets.

In [42, 43], the MLP neural network approach with feature extraction techniques was used and the CENPARMI database was used to apply their model. They used binarization and skeletonization as preprocessing techniques. They used the Z-score normalization method to ensure that the feature vectors were unique. They focused on the zoning techniques for feature extraction. They achieved an overall accuracy of 94.75% for all letters. They compared their work with other works with the same dataset. Their proposed model achieved the highest accuracy, while others were 90.88% [49], 89.2% [50] and 88% [51]. Although the high performance of their model, it leaked from overfitting and they did not use data augmentation (as an example) to help them overcome that problem.

In [44], they used the ANN as their classifier. They depended on binarization, normalization and some noise removing methods as their preprocessing techniques. Their features were extracted from the main body and the secondary components of the character in a way to overcome variations of the handwritten characters. They reached a maximum recognition rate of 100% for some characters and the lowest recognition rate of 66% for some characters. Their used dataset was CENPRMI. They divided the dataset into three groups: 198, 67 and 67 for training, validations and testing, respectively.

4 Arabic language characteristics

The Arabic language is written horizontally from the right direction to the left direction which is different than some other languages such as English and French. It is composed of twenty-eight primary characters with no lower nor upper case characters. The character shape depends on its position in the word, so that a single character may have two to four different shapes. We can divide the character shapes into 3 categories. The first is the main category which contains the 28 characters. The second consists of 8 characters. The third (last and special) contains 4 characters that do not fit into the first 2 categories. Table 5 shows the first category which is the different 28 characters with their different shapes of the Arabic language.

Table 5 Arabic characters first category

#	Name	Isolated	Initial	Middle	End	#	Name	Isolated	Initial	Middle	End
1	Alif	ا			ـا	15	Dad	ضـ	ضـ	ـضـ	ـض
2	Baa	بـ	بـ	ـبـ	ـب	16	Taa	ط	ط	ـط	ـط
3	Taa	تـ	تـ	ـتـ	ـت	17	Zaa	ظ	ظ	ـظ	ـظ
4	Thaa	ثـ	ثـ	ـثـ	ـث	18	Ain	عـ	عـ	ـعـ	ـع
5	Gem	جـ	جـ	ـجـ	ـج	19	Gin	غـ	غـ	ـغـ	ـغ
6	Haa	حـ	حـ	ـحـ	ـح	20	Faa	فـ	فـ	ـفـ	ـف
7	Khaa	خـ	خـ	ـخـ	ـخ	21	Qaf	قـ	قـ	ـقـ	ـق
8	Dal	د			ـد	22	Kaf	كـ	كـ	ـكـ	ـك
9	Zal	ذ			ـذ	23	Lam	لـ	لـ	ـلـ	ـل
10	Raa	ر			ـر	24	Mem	مـ	مـ	ـمـ	ـم
11	Zai	ز			ـز	25	Noon	نـ	نـ	ـنـ	ـن
12	Sin	سـ	سـ	ـسـ	ـس	26	Haa	ه	هـ	ـهـ	ـه
13	Shin	شـ	شـ	ـشـ	ـش	27	Waw	و			ـو
14	Sad	صـ	صـ	ـصـ	ـص	28	Yaa	يـ	يـ	ـيـ	ـي

Table 6 Arabic characters with Noqtah (dots)

Type	Characters									
One dot	ب	ج	خ	ذ	ز	ض	ظ	غ	ف	ن
Two dots	ت	ق	ي							
Three dots	ث	ش								

The first three columns represent: a counter for the characters, the name (reading) of the character and the character shape in its isolated form. The last three columns represent the character's shape in their initial (beginning), middle and end forms, respectively. The dot or what so-called Noqtah plays a role in the consistency and structure of the character shape. The maximum count of the "Noqtah" in a single shape is three such as the character "ث". Table 6 shows the different characters that depend on the dots as sub-components to them.

Ten characters have one dot, three have two dots, and two have three dots. Dots can be above the character such as "ن", in the middle such as "ج" or below such as "ب". The "Hamzah" plays also a role in the character shape. It may appear above the character such as "ؤ", it may appear inside the character such as "ئ", it may appear below the character such as "إ", or it may appear as a single isolated character such as "ء". If the "Hamzah" appears above, inside or below the character, it is called a sub-component to it, and if it appears separately (isolate), it is a component. Table 7 shows the second and third categories.

"Tashkeel" (or "Tushkeel") plays also a role if it appeared with a character. It does not affect the character but the pronunciation and word meaning. The "Fathah" appears above the characters as a small rotated dash. The "Sokon" appears above the characters as a small circle. The "Damah" appears above the characters as a small comma. The "Shaddah" appears above the characters as two small rotated dashes. The "Kasrah" appears below the characters as a small rotated dash. "Tanween" is a bit similar to "Tashkeel" in the shape; however, it comes in doubles. There are three types of it: double "Damah", double "Fathat" and double "Kasrah", "Tashkeel" and "Tanween" can be omitted from the text, and the Arabic readers are expected to understand the meaning from the context itself. The punctuation characters are used also in the Arabic language such as the questions and exclamation marks. Table 8 shows the different "Tashkeel" and punctuation characters.

Arabic text is cursive [52] which means that there is an imaginary horizontal line called a "baseline" that connects a word characters. Due to that cursive nature, Arabic is considered a very challenging problem compared to Latin, Chinese or Japanese languages. Some lines appear above that baseline and named "ascenders". Some lines appear below it and named "descenders" as shown in Fig. 2.

Spaces can separate words and tiny spaces separate characters which can be named sub-words. The sub-words that form a word are not connected as shown in Table 9.

Ligature [53] is the vertical overlapping between the neighboring characters. In other words, the following character appears before the current character. It may appear when the characters (خ, ح, ج, ل,م) are used after other some other characters. Ligatures differ from normal overlapping in that a ligature is an overlap with a touch, while normal overlapping is without a touch. Besides the Arabic characters, there are Arabic numerals (digits) from 0 to 9, which are written in a way that is different than other languages. They are shown in Table 10.

Some characters may become similar in handwritten Arabic texts, although they are not the same in the meaning. The similarity between them makes the recognition and classification a challenging task to the human eye and computers as well [54].

Table 7 Arabic characters second and third categories

#	Name	Isolated	Initial	Middle	End	Category
1	Hamzah				ء	Second
2	Waw-Hamzah	ؤ			ـؤ	Second
3	Yaa-Hamzah	ئ		ـئـ	ـئ	Second
4	Alif-Hamzah (above)	أ			ـأ	Second
5	Alif-Hamzah (below)	إ			ـإ	Second
6	Taa-Marbotah	ة			ـة	Second
7	Alif-Yaa	ى			ـى	Second
8	Alif-Maad	آ			ـآ	Second
9	Lam-Alif	لا			ـلا	Third
10	Lam-Alif-Hamzah (above)	لأ			ـلأ	Third
11	Lam-Alif-Hamzah (below)	لإ			ـلإ	Third
12	Lam-Alif Alif-Maad	لآ			ـلآ	Third

Table 8 Arabic characters Tashkeel (Tushkeel) cases and punctuation characters

#	Name	Shape	Type
1	Fathah	َ	Tashkeel
2	Sokon	ْ	Tashkeel
3	Damah	ُ	Tashkeel
4	Shaddah	ّ	Tashkeel
5	Kasrah	ِ	Tashkeel
6	Comma	،	Punctuation
7	Question mark	؟	Punctuation
8	Single quotation mark	’	Punctuation
9	Double quotation mark	”	Punctuation
10	Exclamation mark	!	Punctuation
11	Left parentheses	)	Punctuation
12	Right parentheses	(	Punctuation
13	Semicolon	؛	Punctuation
14	Dot (period)	.	Punctuation

Fig. 2 Baseline, ascenders and descenders

Table 9 Words with multiple sub-words

Type	Examples		
Single sub-word	محمد	محمود	مصطفى
Two sub-words	احمد	خالد	اسلام
Three sub-words	ضرائب	اشتراك	اصرار
Four sub-words	اتراك	اقرار	ابرار
Five sub-words	اوراق	ارزاق	اوروبا

5 Arabic handwritten character recognition systems

Character recognition is one of the leading applications of pattern recognition despite its many limitations. This process's main purpose is distinguishing between the individual characters and between the words. Arabic character recognition systems can be either online or offline character recognition [15] as shown in Fig. 3.

Figure 3 shows the different paths for Arabic handwritten character recognition systems (AHCR). The online character recognition is the process of classifying the character while writing [55]. This technique required a special type of pens and tablet [56] such as a personal digital assistant (PDA) and a mobile phone. The mechanism of working is that the pressure is used in the digital display to create a series of connected points that are traced by the pen. It cannot be used in recognizing offline or pre-written documents. One the other hand, offline character recognition systems deal with pre-scanned images and documents.

Table 10 Arabic digits versus English digits

Arabic digit	English digit	Arabic digit	English digit
٠ صفر	0	٥ خمسة	5
١ واحد	1	٦ ستة	6
٢ اثنين	2	٧ سبعة	7
٣ ثلاثة	3	٨ ثمانية	8
٤ اربعة	4	٩ تسعة	9

Offline character recognition can be divided into three categories. The first is the recognition of printed characters whose styles, sizes and fonts are the same which leads to making it the simplest between the different offline recognition systems. The second is the recognition of handwritten characters where the styles for a single user are different. The third is typeset character recognition.

Handwritten text can be recognized in two ways: recognition without segmentation techniques and recognition based on segmentation techniques. Because of the ligatures and cursive nature of Arabic scripts, it is preferred to recognize the whole word to avoid a high raise in error rates. The typeset writing style is used regularly in printing newspapers and books. It is more difficult than the normal printed one because of the ligatures as shown in Fig. 4.

AHCR system can be divided into a set of stages from image data retrieval (acquisition) to image classification and recognition. Figure 5 shows the broad steps for Arabic handwritten text/character recognition systems.

5.1 Image retrieval (acquisition) phase

This is the initial phase (task) in the Arabic handwritten text recognition system. The main objective of this phase is to retrieve the data (images) from different sources and convert them into the digitalized form. Digital cameras, scanners and tablets are examples of the text acquisition methods. To increase the speedup of the scanning process, it is required to select a suitable device (i.e., scanner) with high throughput, sensing tool, transport mechanism and document feeder. Pages per minute (PPM) [58] is a factor used to decide whether the device has a high feeder than others or not. Dots per inch (DPI) [59] is another factor used to decide whether the retrieved image has a high resolution or not. The output of this phase is the captured image of the original file or document.

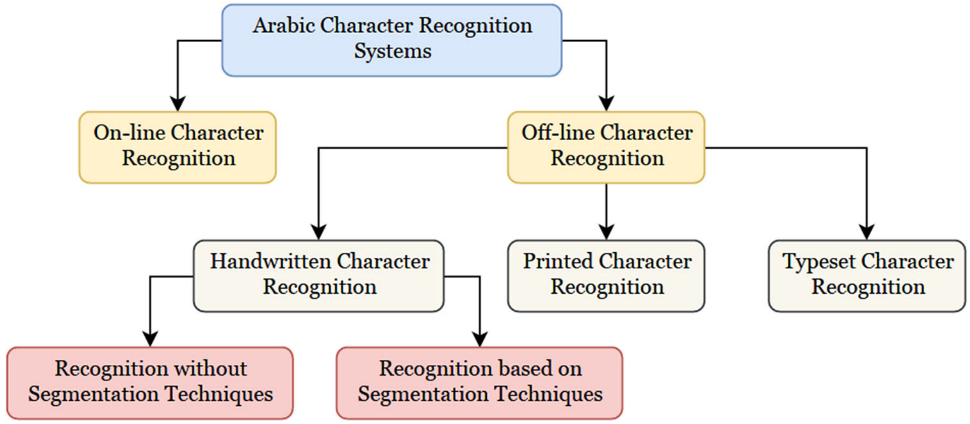


Fig. 3 Classification of the Arabic handwritten character recognition systems

تأميم شركة قناة السويس

الأهرام

الرئيس يعلن باسم الأمة :

أموالنا وحقوقنا ردت الينا

سنبنى السد العالى معتمدين على سواعدنا واتحادنا ودمائنا

قرار رئيس الجمهورية بتأميم شركة القناة

Fig. 4 Old newspaper page (Al-Ahram Newspaper 1956-06-27) [57]

5.2 Preprocessing phase

The preprocessing phase is the second task which is very important in any recognition system [60]. The purpose of this step in handwritten text recognition is to improve and enhance the readability of the textual image and remove unnecessary details from it [61].

The preprocessing phase usually includes several operations such as binarization, thinning, alignment, padding, centering, slant correction, noise removal, smoothing, baseline detection, skeletonization, skewing and normalization. The system can apply one or more from the preprocessing operations. It can also bypass this phase without applying any of the operations if the input data are

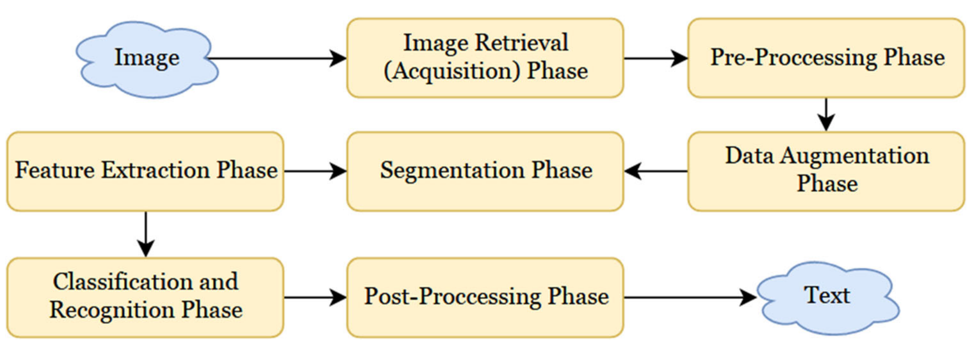


Fig. 5 General steps for Arabic handwritten text/character recognition

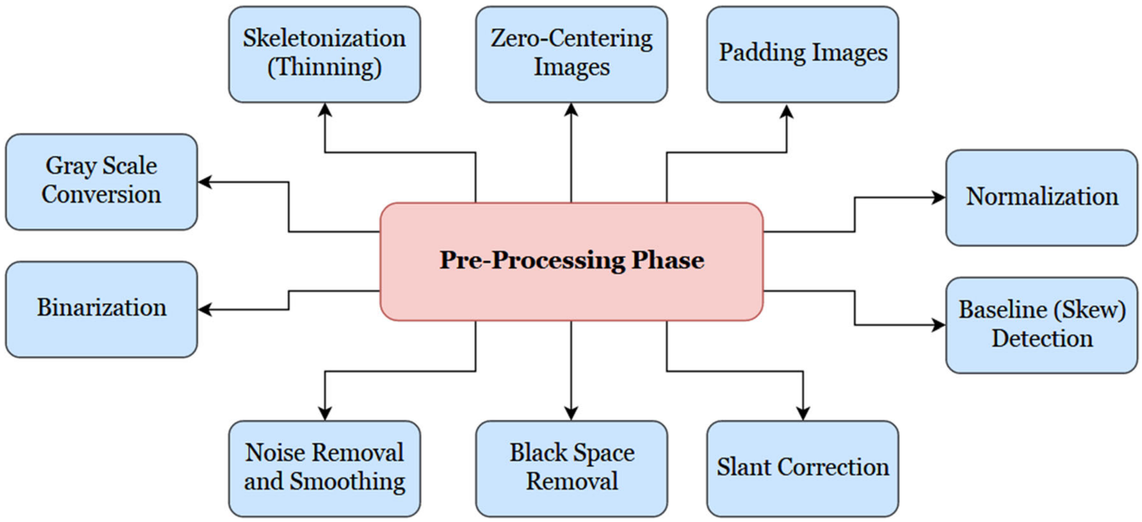


Fig. 6 Summarization of the different preprocessing techniques

previously preprocessed. Figure 6 shows a summary of the different preprocessing techniques.

Grayscale conversion: The colored image in this process is converted to a grayscale image. So, instead of working with a three-channel image, we can work only with a one-channel image. There are multiple ways such as averaging, human eye correction (weighted), desaturation, decomposition and single-color channel methods.

Binarization: It is also called thresholding which is the process of converting a text image into a binary (bi-level) format [62]. In other words, it is called a black and white image. The values of background pixels are 1 for white and 0 for black. This process improves processing speed [63]. They are many methods with different categories used into image binarization [64] such as the fixed threshold method [65], mean value threshold method and Otsu method [66]. Figure 7 shows two sample images after applying the three methods.

The fixed threshold method deals with assigning a fixed value of threshold (i.e., $threshold = 100$). If the pixels' values are more than or equal to that threshold value, the output pixels value is 1 and 0 otherwise according to Eq. (1).

$$g(x,y) = \begin{cases} 1 \rightarrow f(x,y) \geq threshold \\ 0 \rightarrow otherwise \end{cases} \tag{1}$$

The mean value threshold method is similar to the previous one unless that the threshold is not predetermined by the user but is determined by taking the mean value of all of the pixels according to Eq. (2).

$$threshold = \frac{\sum_{i=1}^{width} \sum_{j=1}^{height} f(i,j)}{(width * height)} \tag{2}$$

The Otsu threshold method is used in the automatic binarization based on the histogram shape. It involves iterating through all of the possible threshold values and calculating the measure of spread for the pixel levels in each side of the threshold. The algorithm target is to find the threshold that minimizes the sum of the foreground and background spreads. For more details, see [67–69].

Noise removal and smoothing: All the data acquisition methods are affected by noise; hence, there is no ideal situation for no noise attached to data. Noise removal is the process of removing distortions and unrequired small objects that are not part of the writing [70]. Smoothing is the process of removing noise and data variations by using mathematical morphology operations [71]. There are multiple methods used to reduce image noise and perform smoothing such as filtering and morphological operations [72]. In image processing, filters are used to suppress the high frequencies in the image (i.e., smoothing the input image) or the low frequencies (i.e., enhancing or detecting edges in the input image). Filtering is a neighborhood operation that uses the kernel for performing that.

A kernel is a small window (matrix) that passes through all of the image pixels where the pixel in the center is replaced by a value according to the selected filter [73]. The mathematical expression in Eq. (3) is used to apply the filters in Fig. 8.

$$g(x,y) = \sum_{i=-\frac{M}{2}}^{\frac{M}{2}} \sum_{j=-\frac{M}{2}}^{\frac{M}{2}} h(i,j) * f(x-i, y-j) \tag{3}$$

where M is the size of the kernel (a kernel size is an odd number), $h(i,j)$ is the kernel value at row i and column j, f is the original image, and g is the resultant image.

Fig. 7 Two sample images after applying the three methods. **a** Original. **b** Using fixed threshold (threshold=100). **c** Using mean threshold. **d** Using Otsu threshold

Multiple filters can be used for that such as median filter, Wiener filter [74] and Gaussian filter. The corresponding kernel for the median filter with a size of 3 is shown in Eq. (4):

$$kernel = \frac{1}{9}\begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix} \tag{4}$$

Fig. 8 A sample image after applying the median filter. **a** Upper: original. **b** Lower: using median filter (kernel size=5)

Fuzzy C-means clustering (FCM) [75] is a proposed noise removal algorithm that can be used in thresholding and removing the unrequired noise from the image.

Black space removal: In this process, the unrequired black pixels are removed from the image. The count of black pixels is calculated from the image borders until the white pixel is found. The bounding box can be used to eliminate the space around the character image.

Slant correction: It is a way to correct and eliminate the different character slants [76]. A shear transformation is applied after the average slant of the character.

Baseline (skew) detection: The baseline is the imaginary horizontal line that connects the words' characters together [77]. This process is helpful in determining the words' structural features such as dots, ascenders and descenders [78]. There are multiple methods that can be used for detecting the baseline such as the horizontal project methods or the rotation angle. The rotated angle in the lines requires to be fixed; hence, baseline (skew) correction is used. Skew is considered one of the image distortions forms. Skew correction is used to correct the orientation angle of the image [79]. Some methods are based on the principal component analysis (PCA) technique and word contours [80]. Table 11 shows a comparative study of skew detection and possible correction techniques.

Normalization: it is the process of reducing the variations between the images and adjust the size, position and shape of each character in the word as well [82]. Z-score normalization [83] is a normalization technique. Batch normalization is a normalization and regularization technique [84]. It is used to solve the following issues that appear during the training process:

Internal covariate shift: This means that a small change in the input distribution may affect significantly the entire batch and training.

Vanishing gradient: By using nonlinear activation functions, training may get stuck in the saturation region.

Padding images: Images in many datasets are not the same size as the width, height or both may differ. The input images' sizes to most of the recognizers in the different phases must be the same. The first approach is to crop large images to the size of the minimum image, but in this case, data will be lost. The second approach is to resize the image to a suitable aspect ratio, but in this case, large images data may be distorted. The third approach is to pad the image with white to the maximum width and height existing in the dataset. The white is added on both sides equally, so the image is stilling in the center. The latter approach is better than the first two approaches as it does not lose nor distort the data of the image.

Zero-centering images: Centering the image can be achieved by subtracting the dataset mean pixel values from the pixel values of the images as shown in the following two Eqs. (5) and (6):

$$X_{center} = X - \frac{1}{Height * Width} * \sum_{i=1}^{Height} \sum_{j=1}^{Width} X_{i,j} \tag{5}$$

$$Y_{center} = Y - \frac{1}{Height * Width} * \sum_{i=1}^{Height} \sum_{j=1}^{Width} Y_{i,j} \tag{6}$$

This approach is helpful when learning factors are used. If the image is not centered, it may be updated and amplified in the wrong direction.

Skeletonization (thinning): The skeletonization (also known as thinning) [85] is one of the morphological operations that can be applied to images. It is based on reducing the foreground regions in a binary image. The pixels on the boundaries of the image are removed, but the main content of the image should outbreak apart.

5.3 Data augmentation

Data augmentation is an approach that enables us to increase the diversity and amount of data available for training models without the collection of any new data. There are different techniques that can be applied such as rotating images, brightness change, zooming, shearing, shifting, cropping, padding and flipping [86].

Rotating images: A copy of the image is rotated toward the right or the left by a predefined small angle. Performing this to the whole dataset to make the dataset more robust to the issue of different writers can write the same word in different angles on blank papers.

Flipping images: A copy of the image is flipped horizontally, vertically or both according to a predefined Boolean flag (true or false). This can affect some of the symmetric characters positively such as "ب". This may also affect some of the characters negatively such as "ج" which will be meaningless if it is flipped horizontally. Hence, a special treatment should be considered using the flipping technique.

Cropping images: A cropped image copy is generated and rescaled to the original shape to match other images' shapes. Applying cropping and rescaling on an image can be called zooming.

Generative adversarial networks (GANs) can be also used in data augmentation [87]. They have been proved in many cases to be able to produce augmented data that are alike the real data. They also have been successfully

Table 11 Comparative study of skew detection and correction techniques [81]

Author	Technique used	Script	Advantage	Disadvantage	Result
S. Y. Hashemi	Center of gravity (COG)	Arabic Persian	Efficient and fast algorithm. Works with different resolutions.	Show better results for English script rather than Arabic/Persian	93% with error rate less than 1%
L. Shah et al.	Linear regression	Gujarati	Simple and fast algorithm.	Fail to handle minor variations	59.63% for printed and 45.58% for handwritten
E. kavallieratoo	Projection profile	English	Reduce computation complexity	Unsuccessful for same orientation. It can handle non-parallel text line.	Success rate 100% within confidence range $\pm 0.3\%$
J. Sadri el. al.	Particle swarm optimization (PSO)	Latin Arabic	Better results for gray and binary scale images	Need to improve speed and performance for larger collection of documents	96.34% accuracy and error range less than $\pm 1\%$

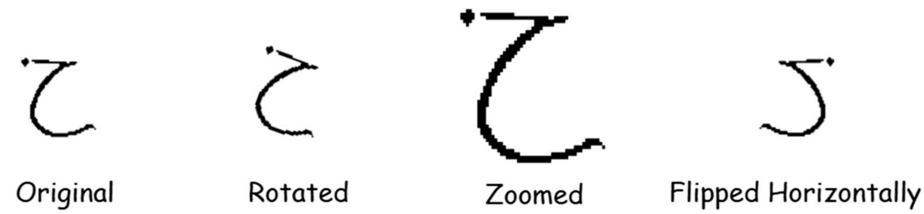


Fig. 9 Rotation, flipping and zooming appliance on the "ح" character

applied to various image generation tasks as a useful approach for data augmentation [88].

Figure 9 shows a simple appliance of rotation, flipping and cropping (with zooming) on the "ح" character. It showed that the horizontal flipping changed the meaning of the character, while zooming increased the number of black pixels that can be used in feature extraction.

5.4 Segmentation phase

The segmentation phase performs segmentation of texts into smaller units. They can be lines, words and characters. It is an important phase as it can after the recognition accuracy rate. There are many types of segmentations such as segmenting a page into line, segmenting a line into words and segmenting a word into characters. Figure 10 shows the different steps in segmenting a page.

Segmenting a page into lines: This process is used to divide the whole text into separated simpler lines. There are multiple used methods for this purpose; one of them is the single shot multi-box detector (SSD) [89] as shown in Fig. 11.

Segmenting a line into words: After processing the previous stage, each line is segmented into words. This stage depends on the space between words. Longer spaces separate words, while shorter ones separate sub-words [91]. Hence, the assumption taken by most researchers is that the spaces separating the words are bigger than the spaces separating the sub-words [92]. Working with words only can be less effective because of their number both in the languages and in the dataset. The dataset will not cover the whole language. There are more than 12 million words in the Arabic language and no datasets covering these whole words [93].

Segmenting a word into characters: This stage uses the output of the previous stage and performs segmentation of each word and converts them into characters [94]. The cursive nature and ligature of the Arabic language make this stage a tedious task. Character segmentation can be based on contour tracing, ANNs, morphological operations and vertical projection. This approach is more suitable and more promising than the two others because there are few distinct character symbols in the Arabic language.

Tesseract [95] is an open-source text recognizer OCR Engine. It was originally developed by Hewlett–Packard (HP). In 2005, Tesseract was open-sourced by HP. Since 2006, it is developed by Google. It has the Unicode Transformation Format (UTF-8) support and can recognize more than 100 languages out of the box. It is now an open-source for everyone. It can be used to segment the words into characters by generating the bounding boxes of each character into an eXtensible Markup Language (XML) file. In the current version, they added the ANN with LSTM, so there is no demand for segmentation rules.

5.5 Feature extraction phase

In this phase, features are extracted and captured. This phase is considered as one of the riskiest phases in OCR. The features are the extracted information from the text image [96]. A single feature describes the characteristics of an underlying character, its partial structure or the whole word structure. This information is passed onto the classifier in the following phase to support its classification process [97]. This phase remains important as it affects the recognition accuracy; hence, it is important to select the best method to achieve the highest value. It can be classified into the following categories: statistical (quantitative) features, structural (qualitative) features and global transformations.

For Arabic text, the features can be dots, dots positions, dots widths, dots heights, dots directions, characters loops and intersections of lines. Extracting features process tries to maximize the interclass variability while minimizing the intra-class variability, and the results of that phase are fed to the classifier. Selecting the most suitable features can

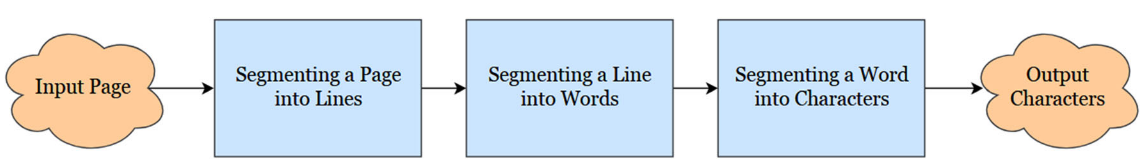
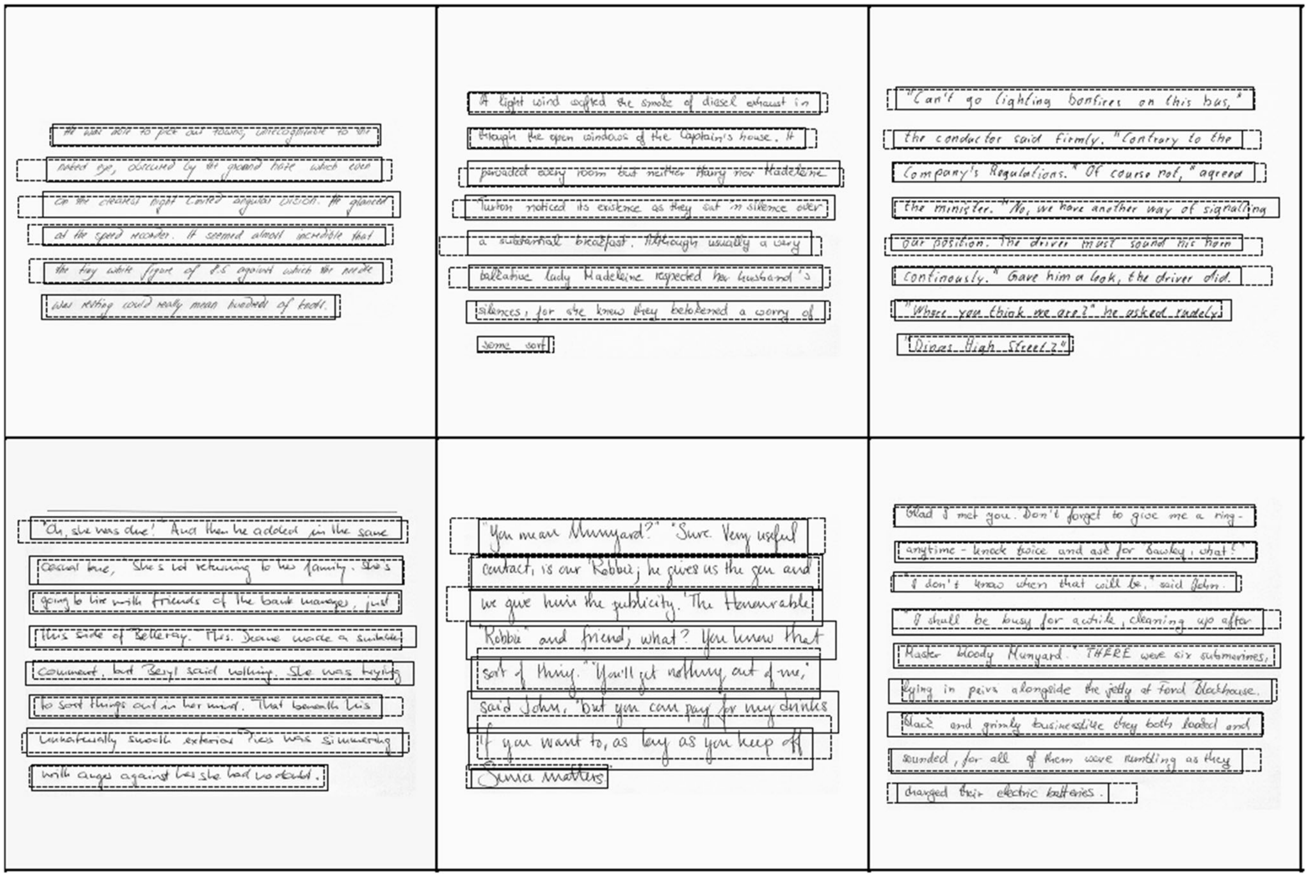
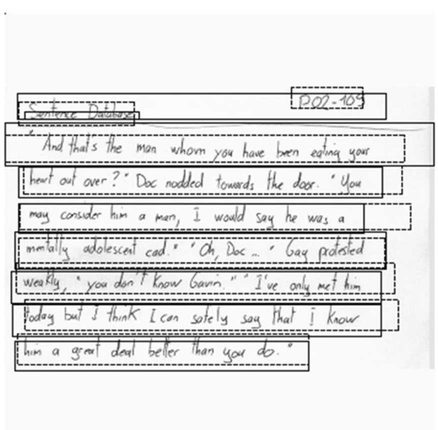


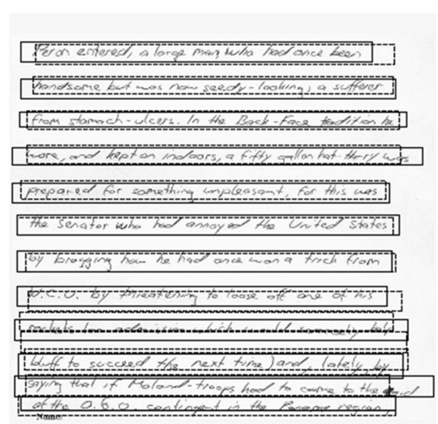
Fig. 10 Steps of segmenting a page



(a) Output images (dotted lines are predicted bounding boxes, solid lines are labelled bounding boxes)



(b) Ignored incorrectly labelled smudge (second line from the top)



(c) Failure cases: misalligned predicted box (forth line from the bottom)

Fig. 11 Three subplots. **a** Predicted bounding boxes from the network, **b** example of where an incorrectly labelled line was predicted correctly, **c** example of where the predicted box was misaligned [90]

play an important role in the performance of the classification stage. Two of the disadvantages of this phase are: It is time-consuming and data-dependent if the manual or traditional way of extraction is followed.

Statistical (quantitative) features: They analyze the distribution of pixels by accumulating the local features at each pixel and generating statistics from these distributions of the local features. Moments, Fourier descriptors and histograms of gray-level distribution are examples of the statistical features.

Structural (qualitative) features: They analyze the geometric and topological characteristics of the image by

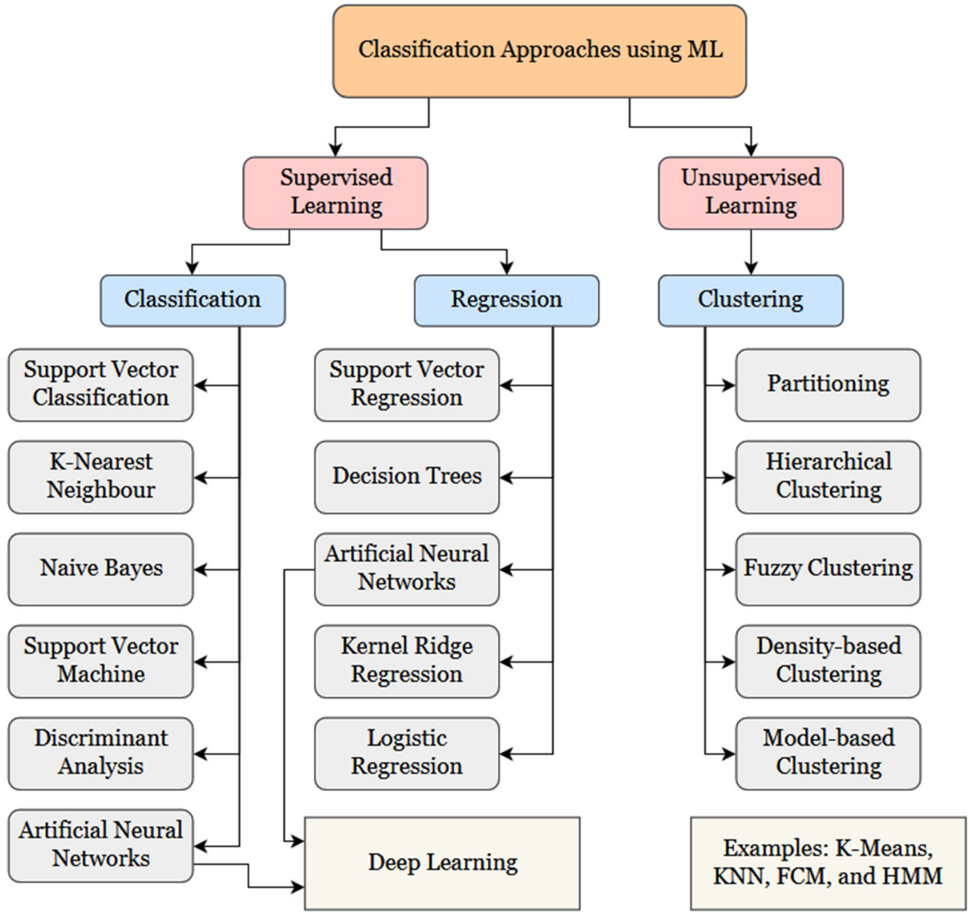


Fig. 12 Summarization of the classification approaches using machine learning

describing their local and global properties. The presence/absence of loops, branch-points, endpoints, dots and orientation of the curves are examples of the structural features.

Global transformations: They are converting the pixel representation into a more compact form. The most common techniques used are: wavelets and adaptive wavelets [98], Hough transform and adaptive Hough transform [99] and Fourier transform.

Topological features: They include pixel ratio, width-to-height ratio (aspect ratio) and the endpoints.

5.6 Classification phase

Classification is the process of identifying and recognizing the object by comparing its features to one of the classes. It is assumed that the classes and model are inputs as the classification is a supervised learning approach. The training process is performed to teach the classifier model on the given features. The output model should have the ability to recognize unseen objects. Classification methods can be based on one of the following: machine learning techniques, graph-theoretic methods such as decision trees, syntactical methods, statistical methods or mathematical methods.

ML is a huge branch of AI inspired by the psychology and biology that deals with learning from a given set of data and can be applied to solve a wide range of problems. It plays an important role in classifications. Supervised machine learning models deal with taking the inputs and outputs and start to build and modify its architecture using algorithms such as gradient descent to reach an optimal point. When the learning completes training, the model is able not only to provide answers to the given data it has learned on but also to yet unseen data with high precision and less error.

Many techniques can be classified into model-based and instance-based techniques. The first one is related to learning a model based on training the input examples to determine the decision boundaries between the different classes. The latter is related to assigning the class of the closest training example to classify a new example. According to the previous taxonomy, each one has different techniques as follows: model-based techniques: ANNs [100], learning vector quantization (LVQ) [101], DL [102], SVM [103] and hidden Markov model (HMM) [104] (can be used also as a post-processing method) and instance-based techniques: K-nearest neighbor (KNN) [105, 106]. Figure 12 shows a summarization of the classification approaches using ML.

Artificial neural networks (ANNs): They are learning models used in ML widely. Their aim is to mimic the learning processes that occur in a bird, an animal or a human neural system to extract the input data features. Being one of the most powerful and important learning models, they are useful in the automation of tasks where the decision of a human being takes too long or may become imprecise. A neural network is swift compared with the human neural network at delivering results. ANNs may detect different connections between the seen instances of data that humans cannot see precisely.

Deep learning (DL): Recently, deep neural networks have got effective enhancements in the text classification process [107, 108], detection [109], learning [109–111] and so on. DL is to establish a network model similar to the human brain information processing appliance. It uses competent learning strategies to abstract features step by step to fit the complex nonlinear functions. Conversely, at present, AHCR methods based on DL still need research development and improvement. As far as we know, there are a few related documents.

DL is a new and trending field of ML. It is one of the most researched areas in the last few years and its models and techniques reached the state-of-the-art performance [112]. It learns and extracts features automatically from the given data. It took the top place in the object recognition field because of the performance, improvements and results [113]. DL can be used with text, images, audios, speeches and videos. Recently, it has won many contests in pattern recognition [102]. TensorFlow and Keras are two famous frameworks that facilitate dealing with DL and provide a high level of abstraction that makes writing programs and scripts easier. Deep neural networks, CNN, recurrent neural networks (RNNs) and stacked autoencoders are some of the DL architectures.

CNN is a type of deep neural networks that is used widely in analyzing visual imagery. It is a type of supervised learning used in classification and depends on the concept of the mathematical operation called convolutions and parallel processing. From the ANN point of view, CNN consists of one or more convolutional layers with one or more fully connected layers. CNN is easier to train and have fewer parameters than the ANN fully connected networks.

CNN requires a lot of training samples (at an average of $8K \rightarrow 10K$) to reach acceptable results and accurate decisions and computational requirements such as GPUs or tensor processing units (TPUs) to reach less and acceptable training times. CNN can work with large-scale, complex, multi-dimension and nonlinear mapped data. The key advantage of using the convolutional operation is generating multiple images from the input image that will lead to enhanced automatic feature extraction from that original input image. Hence, the classification process becomes more powerful. Unlike RNNs, convolutional networks are more focused on single instance learner than a sequence learner. The context is not important in training using convolutional networks.

To reach the automatic feature extraction, CNNs applies different filters in the whole original image. Each filter will represent a specific feature. Pooling layers are used after the convolution layers to reduce the dimensionality but keep the most important features from the previous layer. Maximum, average and sum are types of these pooling layers. The last layer is a fully connected layer that uses one-dimensional vector instead of the two-dimensional arrays. The output from that layer is the selected class with the highest probability. Some of the features that can be extracted using CNNs are: the width and height of the character, the edges of the Arabic character, the positions of the Arabic characters sub-components and the edges of characters.

Figure 13 shows the main building blocks of the CNN architecture. Each coming layer size is calculated according to the number of filters f, the size of stride s, the amount of zero padding p and width (or height) as shown in Eqs. (7) and (8):

$$w_{i+1} = \frac{(w_i + \mathrm{f_i} + 2 * \mathrm{p_i})}{s_i} + 1 \quad (7)$$

$$h_{i+1} = \frac{(h_i + \mathrm{f_i} + 2 * \mathrm{p_i})}{s_i} + 1 \quad (8)$$

The stride means the number of move pixels that will be applied to the kernel. So, if the stride is 2, it means that the filter will move two pixels to the right on the same row and two pixels to the bottom on the next row.

RNN is used to handle the work of sequence prediction problems and to be able to recognize patterns in these problems such as handwritings, speaks, genomes and time series functions. They come into multiple forms; one-to-many, many-to-one and many-to-many. One-to-many: An input observation is mapped to a sequence with multiple output steps. Many-to-one: A multiple inputs sequences are mapped to a class or a quantity prediction. Many-to-many: A multiple inputs sequences are mapped to a sequence with multiple output steps (Fig. 13).

Autoencoder is an unsupervised ANN technique used for learning efficient encoding. The input layer and output layer have the same size. The hidden layers have smaller sizes and perform the compression of the input data [115, 116]. It is a simple 3-layer neural network that consists of an input layer, a hidden encoding layer and an output

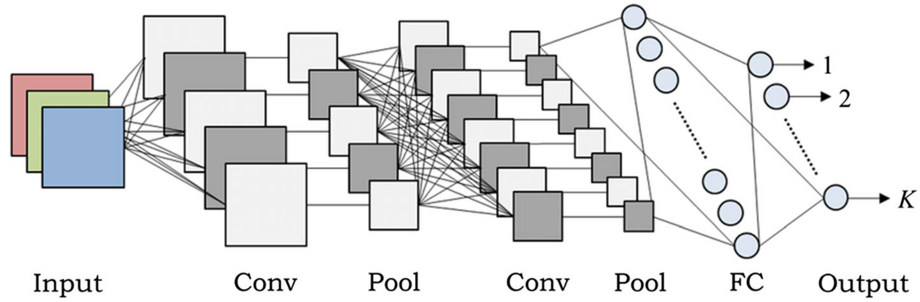


Fig. 13 CNN main building blocks [114]

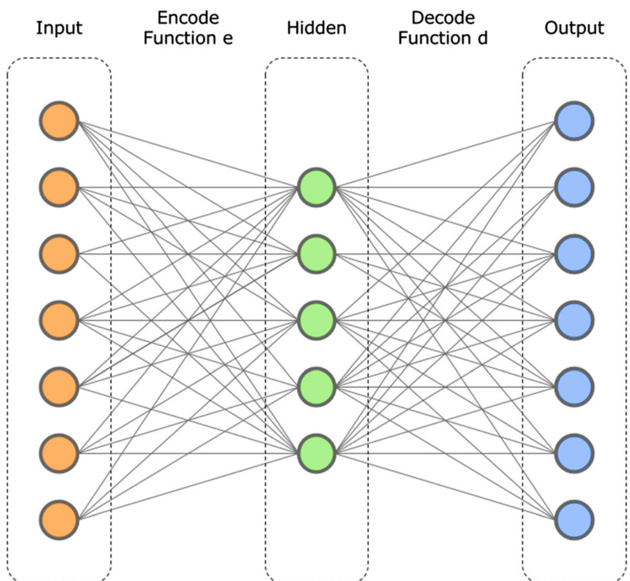


Fig. 14 Autoencoder sample architecture [117]

decoding layer. Figure 14 shows an example of the autoencoder inner architecture.

Support vector machine: Vapnik and Cortes developed the statistical classifier SVM in the later 1990s [118]. It can be used in PR, object tracking, document analysis, bioinformatics, OCR, etc. Linear, RBF, sigmoid and polynomial are widely used kernels used with the SVM classifier. It can be used with multiclass classification and is considered to produce high classification rates. LIBSVM [119] is one of the libraries that can be used for classifying using SVM.

5.7 Post-processing phase

The post-preprocessing stage improves the output from the previous stage by refining the taken decisions. This can be done by using the context. This stage is responsible for outputting the best solution and is often implemented as a set of techniques and methods that rely mainly on the character lexicons, frequencies and other information [120]. Grouping, error detection and error correction can also be applied in the post-preprocessing phase.

5.8 Affecting factors (challenges)

There are many factors (challenges) affecting the development of any recognitions system especially Arabic (our case). Figure 15 shows a summarization of the affecting factors in AHCR.

Random factors: They can affect the document quality while being scanned such as paper quality, dirt, ink quality, ink amount, lighting, shadows, digitalization errors and distortions. The mood of the writer, the writing position and the writing situation are factors that affect the written character also [121, 122]. For normal people, the mentioned considerations do not make much difference, but they are important for optical character recognition systems as a single dot may affect the recognized character.

Linguistic factors: Such as the cardinality of the Arabic alphabet used in the system and the character shape variation within a word [123, 124].

Corrupted images (documents): Classification may become difficult to be done efficiently with corrupted or ancient images that do not have only textual information. Noise can be soaring in them, and applying preprocessing leads to information and content loss [125].

Different styles: Document or image may be written in a font that contains many overlapping characters which will make it more difficult on the classifier to recognize and classify it correctly [126].

Blurred documents: Blurred documents or images are another factor in losing information. The correct sharpness of the document is required [127].

Complex views: The complexity of the view plays an important role [128]. The image may not contain only textual elements but may contain other elements such as homes, carts and other objects.

6 Arabic handwritten character databases

To have the ability to compare the performance between the systems, a standard and a large database is required for training, testing and validation steps. Some of the authors yield high accuracy because they are using small databases

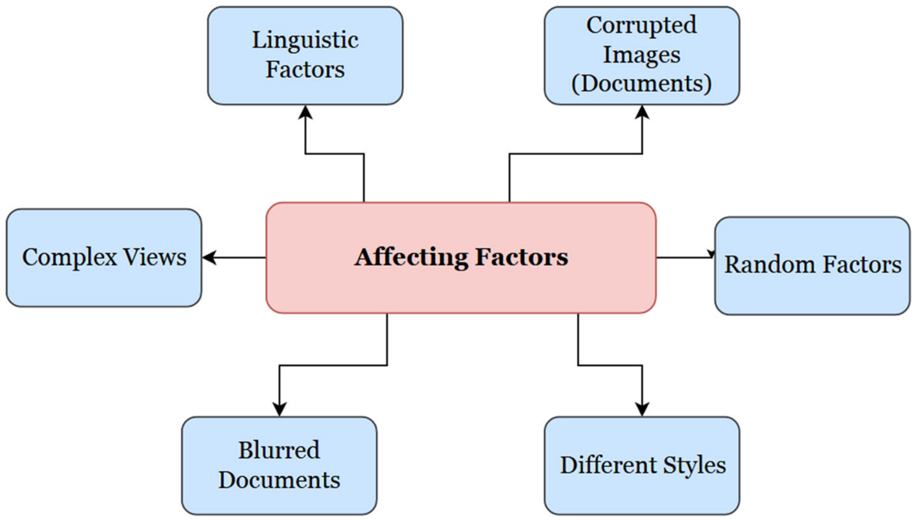


Fig. 15 Summarization of the affecting factors in AHCR

and not standard ones. In the handwritten text, there is no type of control over the user's writing style. Compared to the English dataset, the Arabic datasets are limited and many of them are generated, created and developed for specific reasons.

IFN/ENIT database contains 26,459 city words that was written by 411 different writers. Each image represents a 300 DPI binary handwritten word. It is available free for non-commercial purposes (http://www.ifnenit.com/).

The ADBase and MADBase are two large Arabic handwritten digits databases. The first database is the Arabic Digits dataBase (ADBase) which is composed of 70,000 digits images in *bmp* format divided into 60,000 for training (6000 per digit) and 10,000 for testing (1000 per digit). The second database is the Modified ADBase (or the MADBase) which is a modified version of the ADBase. Both have the same format as the MNIST database. The authors of MADBase wrote each digit ten times. It is freely available for non-commercial research (http://datacenter.aucegypt.edu/shazeem/).

KHATT (KFUPM Handwritten Arabic TexT) database is a database of unconstrained handwritten Arabic Text that was written by 1000 different writers. The database includes 2000 similar-text paragraph images and 2000 unique-text paragraph images and their extracted text line images. It is freely available for non-commercial reasons (http://khatt.ideas2serve.net/index.php).

The APTI Database is the large-scale benchmarking of open-vocabulary, multi-font, multi-size and multi-style text recognition systems in Arabic. The database is called APTI for Arabic Printed Text Image. It is synthetically generated using a lexicon of 113,284 words, 10 Arabic fonts, 10 font sizes and 4 font styles. The database contains 45,313,600 single word images totaling to more than 250 million characters. It is available at (https://diuf.unifr.ch/main/diva/APTI/index.html/).

The AIA9k [129] dataset is another database used for Arabic handwritten character recognition. It is part of a research project, pending availability of further funding. The current version of the dataset contains about 8737 valid samples of the 28 Arabic alphabet letters. The samples were collected from 107 volunteers between 18 and 28 years old who were BSc or MSc students. They were 62 females and 45 males. Each one of them wrote the character 3 times. It is available at (http://www.eng.alexu.edu.eg/~mehussein/AIA9k/index.html).

The AlexU-Word dataset for Isolated-Word Closed-Vocabulary Offline Arabic Handwriting Recognition is a part of a research project, pending availability of further funding. The current version of the dataset contains 25,114 samples of 109 unique Arabic words that cover all possible shapes of all Arabic alphabet letters. The words samples were collected from 907 writers. It is available at (http://www.eng.alexu.edu.eg/~mehussein/alexu-word/index.html).

The Arabic Handwritten Character Dataset (AHCD) [130, 131] is composed of 16,880 characters that were written by 60 participants, the age ranges from 19 to 40 years and 90% of the participants are right hand. Each participant wrote each character (from the "alef" or "أ" character to the "yeh" or "ي" character) ten times. The written characters were scanned after that at the resolution of 300 DPI. The database is partitioned into two sets: a training set (13,440 characters to 480 images per class) and a test set (3360 characters to 120 images per class). It is available at (https://www.kaggle.com/mloey1/ahcd1).

The CENPARMI [132] is a dataset composed of offline Arabic handwritten isolated characters. It contains 21,426 characters that were written by 328 writers.

Table 12 Arabic handwritten character databases

Database	Data type	Dataset size	Image size	DPI
IFN/ENIT	Words	26,459	–	300
ADBase	Numbers	70,000	Variable	300
MADBase				
KHATT	Text	4000	–	200, 300 and 600
APTI	Words	45,313,600	–	–
AIA9K	Letters	8737	32×32	–
AlexU-Word	Words	25,114	–	–
AHCD	Images	16,800	–	300
CENPARMI	–	21,426	–	–
CMATERDB v.3.3.1	Numbers	3000	32×32	–

The CMATERDB v.3.3.1 is a PR dataset used for research purposes. The Arabic version of it was last updated on August 8, 2011. It contains 3000 Arabic digits images with a size of 32×32 each. Each digit from 0 to 9 has 300 images. It is free and has a small size. It is available at (https://code.google.com/archive/p/cmaterdb/downloads).

Table 12 lists the available Arabic database that can be used in character recognition.

7 Challenges and modern trends

There are many challenges for researchers to work on in this field, and there is a demand for new methods to emerge as the computational technology is increasing and resource limitations are decreasing [133]. Some possible challenges and future directions may include, and Fig. 16 summarizes them:

Arabic handwritten characters datasets: The number of available datasets in Arabic for handwritten characters is not too many compared to the English language. Also, some of the available datasets have not many records. There is a need for a dataset with a large amount of data with different font sizes, styles, illuminations, users and words.

Noise in preprocessing: Some of the captured records (images) may contain unrequired noise of data that should be removed or it may lead to misclassification. Some of the works depended on removed them manually. There is a need to find an easy and fast way to remove them automatically especially for large datasets.

State-of-the-art techniques: Some of the works depended on unsupervised learning, and some of them depended on non-state-of-the-art techniques, however, their accuracy. The state-of-the-art supervised techniques provide scalable and high accuracy to different datasets compared to other techniques. There is a need to depend on these techniques in future researches.

Low resolution and quality documents: Ancient papers and documents are a great challenge as they contain unrequired factors such as noise, and some of the characters are removed. It is an open area problem for researchers to work on.

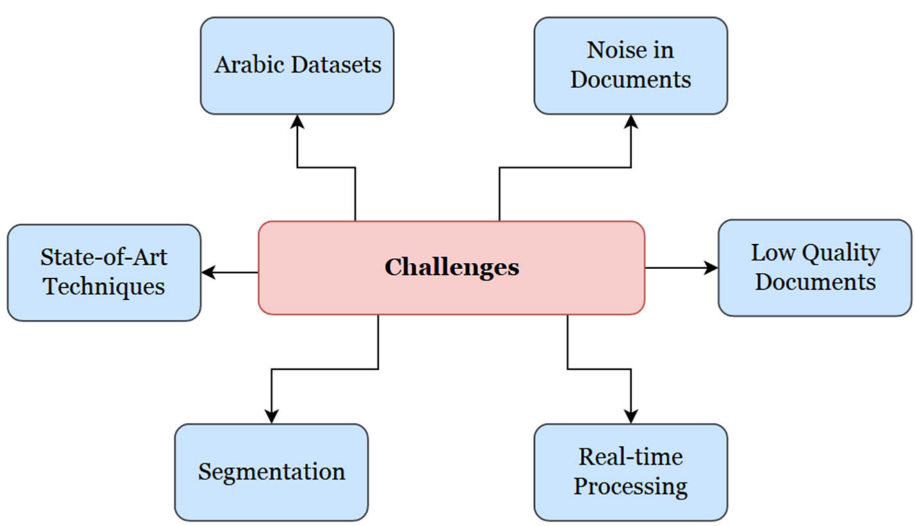


Fig. 16 Some of the AHCR challenges, trends and future directions

Table 13 Summarization of the proposed research topics

Item	Proposed direction	Improvement
Preprocessing	Applying binarization, thinning, alignment, padding, centering, slant correction, noise removal, smoothing, baseline detection, skeletonization, skewing and normalization	Decreasing errors and noise
Segmentation	Applying segmentation on page, lines and words	Increasing accuracy
Feature Extraction	Using deep learning (state of art) techniques with new architectures can be used for automatic feature extraction, learning and classification	Reducing errors
Learning		Increasing accuracy
Classification		Reducing resources
Dataset	Collecting new datasets, merging different datasets and using existing datasets	Reducing overfitting
Overfitting	Applying data augmentation, dropout and normalization	Reducing overfitting

Segmentation: Some of the previous works depended on the manual segmentation of the datasets, some of the available datasets are not segmented, and some of the works depended on segmented datasets. There is a need to find a scalable way to segment the documents into lines then into words (or characters) automatically especially for large and old datasets. Another challenge with segmentation is how to deal with ligatures and the huge number of Arabic words.

Real-time systems: Most of the papers applied their testing and validation on portions of their used datasets. These previous systems are not tested on real-time processing. There is a demand to find a system that works with real-time scanning especially for large amounts of papers and documents.

Table 13 summarizes the proposed research topics to increase the efficiency of the Arabic text comprehension system.

8 Conclusions

This survey showed the current research trends in the area of the Arabic language. It highlighted the current status of diverse research aspects of that area. This can foster and facilitate the adaption and extension of previous systems into new applications and systems. Arabic has a widespread and unexplored scope; however, less research was done previously in that field.

The focus of this article was on offline Arabic handwritten text recognition, and the main objective of this paper was critically analyzing the current researches to identify the problem areas and challenges faced by the previous researchers.

We showed some of the previous work that is related to modern state-of-the-art methods and showed fewer errors and the high level of abstraction of their works. This identification is intended to provide recommendations for future advances in the area as shown in the challenges and trends section. We discussed also some of the Arabic datasets that can be used as benchmarks for training, testing and validation comparisons.

References

1. Versteegh K (2014) Arabic language. Edinburgh University Press, Edinburgh
2. Suleiman Y (2003) The Arabic language and national identity. Edinburgh University Press, Edinburgh
3. Shaalan K, Al-Sheikh S, Oroumchian F (2012) Query expansion based-on similarity of terms for improving Arabic information retrieval. In: International conference on intelligent information processing. Springer, pp 167–176
4. El-Desouky AI, Salem MM, El-Gwad AOA, Arafat H (1991) A handwritten Arabic character recognition technique for machine reader. In: Third international conference on software engineering for real time systems, 1991, pp 212–216
5. Shirko O, Omar N, Arshad H, Albared M (2010) Machine translation of noun phrases from Arabic to English using transfer-based approach. J Comput Sci 6(3):350
6. Importance of the Arabic language. https://www.importanceoflanguages.com/importance-arabic-language/. Accessed 25 Nov 2019
7. Hourani A (1983) Arabic thought in the liberal age 1798–1939. Cambridge University Press, Cambridge
8. Pattern recognition. https://www.britannica.com/technology/pattern-recognition-computer-science. Accessed on 01 Jan 2019
9. Lensu A (2002) Computationally intelligent methods for qualitative data analysis. University of Jyväskylä, Jyväskylä
10. Vadwala MA, Suthar MK, Karmakar MY, Thakkar N (2017) Survey paper on different speech recognition algorithm: challenges and techniques. Int J Comput Appl 175:31–36
11. Lawgali A (2015) A survey on Arabic character recognition. Int J Signal Process Image Process Pattern Recognit 8:401–426
12. Govindan V, Shivaprasad A (1990) Character recognition—a review. Pattern Recognit 23(7):671–683
13. Biadsy F, Saabni R, El-Sana J (2011) Segmentation-free online Arabic handwriting recognition. Int J Pattern Recognit Artif Intell 25(07):1009–1033

14. Tappert CC, Suen CY, Wakahara T (1990) The state of the art in online handwriting recognition. IEEE Trans Pattern Anal Mach Intell 12(8):787–808
15. Plamondon R, Srihari SN (2000) Online and off-line handwriting recognition: a comprehensive survey. IEEE Trans Pattern Anal Mach Intell 22(1):63–84
16. Klatt DH (1987) Review of text-to-speech conversion for English. J Acoust Soc Am 82(3):737–793
17. Bijl D, Hyde-Thomson H (2001) Speech to text conversion. Google Patents, ed
18. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. J Mach Learn Res 12:2493–2537
19. Manning CD, Manning CD, Schütze H (1999) Foundations of statistical natural language processing. MIT Press, Cambridge
20. Shaalan K, Siddiqui S, Alkhatib M, Abdel Monem A (2018) Challenges in Arabic natural language processing. World Scientific, Singapore, pp 59–83
21. Nikolajeva M (2014) "Reading for learning," cognitive approaches to children's literature. John Benjamins Publishing Company, Amsterdam
22. Nasukawa T (1998) Parsing method and system for natural language processing. Google Patents, ed
23. Busch JE, Lin AD, Graydon PJ, Caudill M (2006) Ontology-based parser for natural language processing. Google Patents, ed
24. Mitchell DC (1994) Sentence parsing. In: Gernsbacher MA (ed) Handbook of psycholinguistics. Academic Press, New York, pp 375–409
25. Ueda H (1989) Word processor including spelling verifier and corrector. Google Patents, ed
26. Zitouni I, Sorensen J, Luo X, Florian R (2005) The impact of morphological stemming on Arabic mention detection and coreference resolution. In: Proceedings of the ACL workshop on computational approaches to semitic languages. Association for Computational Linguistics, pp 63–70
27. Lyons J (1977) Semantics. Cambridge University Press, Cambridge
28. Jackendoff R (1983) Semantics and cognition. MIT Press, Cambridge
29. Geeraerts D (2010) Theories of lexical semantics. Oxford University Press, Oxford
30. Zouaghi A, Zrigui M, Antoniadis G, Merhbene L (2012) Contribution to semantic analysis of Arabic language. Adv Artif Intell 2012
31. Boudad N, Faizi R, Thami ROH, Chiheb R (2018) Sentiment analysis in Arabic: a review of the literature. Ain Shams Eng J 9 (4):2479–2490
32. Tartir S, Abdul-Nabi I (2017) Semantic sentiment analysis in Arabic social media. J King Saud Univ - Comput Inf Sci 29 (2):229–233
33. Alami N, El Adlouni Y, En-nahnahi N, Meknassi M (2018) Using statistical and semantic analysis for Arabic text summarization. In: International conference on information technology and communication systems. Springer, Cham, pp 35–50
34. Salam M, Hassan AA (2019) Offline isolated Arabic handwriting character recognition system based on SVM. Int Arab J Inf Technol 16(3):467–472
35. Ko D, Lee C, Han D, Ohk H, Kang K, Han S (2018) Approach for machine-printed Arabic character recognition: the-state-of-the-art deep-learning method. Electron Imaging 2018(2):1–8
36. Ashiquzzaman A, Tushar AK (2017) Handwritten Arabic numeral recognition using deep learning neural networks. In: 2017 IEEE international conference on imaging, vision & pattern recognition (icIVPR). IEEE, pp 1–4
37. Balci B, Saadati D, Shiferaw D (2017) Handwritten text recognition using deep learning. CS231n: convolutional neural networks for visual recognition, Stanford University, Course Project Report, Spring
38. Younis KS (2017) Arabic handwritten character recognition based on deep convolutional neural networks. Jordanian J Comput Inf Technol 3(3):186–200
39. Ahmed SB, Naz S, Razzak MI, Yousaf R (2017) Deep learning based isolated Arabic scene character recognition. In: 2017 1st international workshop on Arabic script analysis and recognition (ASAR). IEEE, pp 46–51
40. Loey M, El-Sawy A, EL-Bakry H (2017) Deep learning autoencoder approach for handwritten Arabic digits recognition. arXiv preprint arXiv:1706.06720
41. El-Sawy A, Hazem E-B, Loey M (2016) CNN for handwritten Arabic digits recognition based on LeNet-5. In: International conference on advanced intelligent systems and informatics. Springer, pp 566–575
42. Abdalkafor A, Alhamouz S (2016) Arabic offline handwritten isolated character recognition system using neural network. Int J Bus ICT 2:41–50
43. Abdalkafor AS, Sadeq A (2016) Arabic offline handwritten isolated character recognition system using neural network. Int J Bus ICT 2(3):41–50
44. Sahloul A, Suen C (2014) OFF-line system for the recognition of handwritten Arabic character. In: Fourth international conference on computer science & information technology, pp 227–244
45. Nawaz SN, Sarfraz M, Zidouri A, Al-Khatib WG (2004) An approach to offline Arabic character recognition using neural networks, vol 3, pp 1328–1331
46. Sarfraz M, Nawaz SN, Al-Khuraidly A (2003) Offline Arabic text recognition system, pp 30–35
47. Al-Thubaity AO (2015) A 700 M+ Arabic corpus: KACST Arabic corpus design and construction. Langu Resources Eval J Artic 49(3):721–751
48. Das N, Mollah AF, Saha S, Haque SS (2010) Handwritten Arabic numeral recognition using a multi layer perceptron. arXiv preprint arXiv:1003.1891
49. Jamal AT, Nobile N, Suen CY (2014) End-shape recognition for Arabic handwritten text segmentation. In: IAPR workshop on artificial neural networks in pattern recognition. Springer, pp 228–239
50. Sahlol AT, Suen CY, Elbasyoni MR, Sallam AA (2014) Investigating of preprocessing techniques and novel features in recognition of handwritten Arabic characters. In: IAPR workshop on artificial neural networks in pattern recognition. Springer, pp 264–276
51. Sahlol AT, Suen CY, Elbasyouni MR, Sallam AA (2014) A proposed OCR algorithm for the recognition of handwritten Arabic characters. J Pattern Recognit Intell Syst 2:8–22
52. Al-Muhtaseb HA, Mahmoud SA, Qahwaji RS (2008) Recognition of off-line printed Arabic text using Hidden Markov Models. Sig Process 88(12):2902–2912
53. Hamid A, Haraty R (2001) A neuro-heuristic approach for segmenting handwritten Arabic text. In: Proceedings ACS/IEEE international conference on computer systems and applications. IEEE, pp 110–113
54. Pal U, Chaudhuri B (2004) Indian script character recognition: a survey. Pattern Recognit 37(9):1887–1899
55. Ward JR, Kuklinski T (1988) A model for variability effects in handprinting with implications for the design of handwriting character recognition systems. IEEE Trans Syst Man Cybern 18 (3):438–451
56. Kholmatov A, Yanikoglu B (2005) Identity authentication using improved online signature verification method. Pattern Recognit Lett 26(15):2400–2408

57. https://www.pinterest.com/pin/349943833544343951/. Accessed on 25 May 2019
58. PPM (pages per minute). https://whatis.techtarget.com/definition/PPM-pages-per-minute. Accessed on 25 May 2019
59. Mackay DG, Zeller C, Cordery RA, Brunk HL (2003) Method for determining a printer's signature and the number of dots per inch printed in a document to provide proof that the printer printed a particular document. Google Patents, ed
60. Farooq F, Govindaraju V, Perrone M (2005) Pre-processing methods for handwritten Arabic documents. In: Eighth international conference on document analysis and recognition (ICDAR'05). IEEE, pp 267–271
61. Tsang V, Jacob D, Shein F (2014) System and method for enhancing comprehension and readability of text. Google Patents, ed
62. Trier OD, Taxt T (1995) Evaluation of binarization methods for document images. IEEE Trans Pattern Anal Mach Intell 17 (3):312–315
63. Long J, Jin L (2004) An image binarization method based on global mean and local standard deviation. Comput Eng 2
64. Sezgin M, Sankur B (2004) Survey over image thresholding techniques and quantitative performance evaluation. J Electron Imaging 13:146–166
65. Baligar VP, Patnaik LM, Nagabhushana G (2006) Low complexity, and high fidelity image compression using fixed threshold method. Inf Sci 176(6):664–675
66. Xu X, Xu S, Jin L, Song E (2011) Characteristic analysis of Otsu threshold and its applications. Pattern Recognit Lett 32(7):956–961
67. Greensted A (2019) Otsu thresholding. http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html. Accessed on 25 May 2019
68. Yousefi J (2015) Image binarization using Otsu thresholding algorithm. University of Guelph, Guelph
69. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
70. Buades A, Coll B, Morel J-M (2005) A non-local algorithm for image denoising. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol 2. IEEE, pp 60–65
71. Roushdy M (2006) Comparative study of edge detection algorithms applying on the grayscale noisy image using morphological filter. GVIP J 6(4):17–23
72. Verma R, Ali J (2013) A comparative study of various types of image noise and efficient noise removal techniques. Int J Adv Res Comput Sci Softw Eng 3(10):617–622
73. Lehmann TM, Gonner C, Spitzer K (1999) Survey: interpolation methods in medical image processing. IEEE Trans Med Imaging 18(11):1049–1075
74. Chen J, Benesty J, Huang Y, Doclo S (2006) New insights into the noise reduction Wiener filter. IEEE Trans Audio Speech Lang Process 14(4):1218–1234
75. Bezdek JC, Ehrlich R, Full W (1984) FCM: the fuzzy c-means clustering algorithm. Comput Geosci 10(2–3):191–203
76. Sun C, Si D (1997) Skew and slant correction for document images using gradient direction. In: Proceedings of the fourth international conference on document analysis and recognition, vol 1. IEEE, pp 142–146
77. Nagabhushan P, Alaei A (2010) Tracing and straightening the baseline in handwritten Persian/Arabic text-line: a new approach based on painting-technique. Int J Comput Sci Eng 2(4):907–916
78. Atallah A-S, Omar K (2008) Methods of Arabic language baseline detection—the state of art. IJCSNS 8(10):137
79. Sansom-Wai CY, Williams IH, Tretter DR (2001) Image processing system with image cropping and skew correction. Google Patents, ed
80. Kurniawan F, Khan AR, Mohamad D (2009) Contour vs non-contour based word segmentation from handwritten text lines. An experimental analysis. Int J Digital Content Technol Appl 3 (2):127–131
81. Mahajan N, Jaidka K (2015) Various skew detection and correction techniques: a survey. Int J Adv Res Comput Sci Softw Eng 5:4
82. Pei S-C, Lin C-N (1995) Image normalization for pattern recognition. Image Vis Comput 13(10):711–723
83. Jain A, Nandakumar K, Ross A (2005) Score normalization in multimodal biometric systems. Pattern Recognit 38(12):2270–2285
84. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167
85. Abu-Ain W, Abdullah SNHS, Bataineh B, Abu-Ain T, Omar K (2013) Skeletonization algorithm for binary images. Procedia Technol 11:704–709
86. Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. J Big Data 6(1):60
87. dos Santos Tanaka FHK, Aranha C (2019) Data augmentation using GANs. Proc Mach Learn Res XXX:1–16
88. Shao S, Wang P, Yan C (2019) Generative adversarial networks for data augmentation in machine fault diagnosis. Comput Ind 106:85–93
89. Liu W et al. (2016) Ssd: single shot multibox detector. In: European conference on computer vision. Springer, pp 21–37
90. Chung J (2018). Handwriting OCR: line segmentation with Gluon. https://medium.com/apache-mxnet/handwriting-ocr-line-segmentation-with-gluon-7af419f3a3d8. Accessed on 11 May 2019
91. Papavassiliou V, Stafylakis T, Katsouros V, Carayannis G (2010) Handwritten document image segmentation into text lines and words. Pattern Recognit 43(1):369–377
92. Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Assoc Comput Linguist 5:135–146
93. How many words does the Arabic language have? https://www.quora.com/How-many-words-does-the-Arabic-language-have. Accessed on 30 Aug 2019
94. Lu Y, Shridhar M (1996) Character segmentation in handwritten words—an overview. Pattern Recognit 29(1):77–96
95. Tesseract OCR. https://github.com/tesseract-ocr/tesseract. Accessed on 30 Aug 2019
96. Hong Z-Q (1991) Algebraic feature extraction of image for recognition. Pattern Recognit 24(3):211–219
97. Trier ØD, Jain AK, Taxt T (1996) Feature extraction methods for character recognition-a survey. Pattern Recognit 29(4):641–662
98. Wickerhauser MV (1996) Adapted wavelet analysis: from theory to software. AK Peters/CRC Press, Boca Raton
99. Illingworth J, Kittler J (1987) The adaptive hough transform. IEEE Trans Pattern Anal Mach Intell PAMI-9(5):690–698
100. Xin Y (1999) Evolving artificial neural networks. Proc IEEE 87 (9):1423–1447
101. Kohonen T, Hynninen J, Kangas J, Laaksonen J, Torkkola K (1996) LVQ PAK: the learning vector quantization program package. Technical report, Laboratory of Computer and Information Science
102. Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117
103. Abe S (2005) Support vector machines for pattern classification. Springer, Berlin

104. Fine S, Singer Y, Tishby N (1998) The hierarchical hidden Markov model: analysis and applications. Mach Learn 32 (1):41–62
105. Keller JM, Gray MR, Givens JA (1985) A fuzzy K-nearest neighbor algorithm. IEEE Trans Syst Man Cybern SMC-15 (4):580–585
106. Dudani SA (1976) The distance-weighted k-nearest-neighbor rule. IEEE Trans Syst Man Cybern SMC-6(4):325–327
107. Korns MF, May T (2019) Strong typing, swarm enhancement, and deep learning feature selection in the pursuit of symbolic regression-classification. In: Genetic programming theory and practice XVI. Springer, pp 59–84
108. Howard J, Ruder S (2018) Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146
109. Deng D, Liu H, Li X, Cai D (2018) Pixellink: detecting scene text via instance segmentation. In: Thirty-second AAAI conference on artificial intelligence
110. Wang Y, Xu W (2018) Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud. Decis Support Syst 105:87–95
111. Chatterjee A, Gupta U, Chinnakotla MK, Srikanth R, Galley M, Agrawal P (2019) Understanding emotions in text using deep learning and big data. Comput Hum Behav 93:309–317
112. Akkus Z, Galimzianova A, Hoogi A, Rubin DL, Erickson BJ (2017) Deep learning for brain MRI segmentation: state of the art and future directions. J Digit Imaging 30(4):449–459
113. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436
114. Hidaka A, Kurita T (2017) Consecutive dimensionality reduction by canonical correlation analysis for visualization of convolutional neural networks. Proceedings of the ISCIE International Symposium on Stochastic Systems Theory and its Applications, Vol 2017. The ISCIE symposium on stochastic systems theory and Its applications, pp 160–167
115. Baldi P (2012) Autoencoders, unsupervised learning, and deep architectures. In: Proceedings of ICML workshop on unsupervised and transfer learning, pp 37–49
116. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P-A (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 11:3371–3408
117. Oppermann A (2018) Deep autoencoders for collaborative filtering. https://towardsdatascience.com/deep-autoencoders-for-collaborative-filtering-6cf8d25bbf1d. Accessed 18 Sept 2019
118. Deng N, Tian Y, Zhang C (2012) Support vector machines: optimization based theory, algorithms, and extensions. Chapman and Hall/CRC Press, Boca Raton
119. Chang C-C, Lin C-J (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol (TIST) 2(3):27
120. Shatnawi M (2015) Off-line handwritten Arabic character recognition: a survey. In: Proceedings of the international conference on image processing, computer vision, and pattern recognition (IPCV), p 52. The Steering Committee of the World Congress in Computer Science, Computer
121. Dale R (2000) Guides to quality in visual resource imaging-imaging systems: the range of factors affecting image quality
122. Russ JC (2016) The image processing handbook. CRC Press, Boca Raton
123. Taha HY (2013) Reading and spelling in Arabic: linguistic and orthographic complexity. Theory Pract Lang Stud 3(5):721
124. Daher J (1998) Gender in linguistic variation: the variable (q) in Damascus Arabic. Amst Stud Theory Hist Linguist Sci Ser 4:183–208
125. Russo F, Ramponi G (1996) A fuzzy filter for images corrupted by impulse noise. IEEE Signal Process Lett 3(6):168–170
126. Slimane F, Kanoun S, Hennebert J, Alimi AM, Ingold R (2013) A study on font-family and font-size recognition applied to Arabic word images at ultra-low resolution. Pattern Recognit Lett 34(2):209–218
127. Vogel CR, Oman ME (1998) Fast, robust total variation-based reconstruction of noisy, blurred images. IEEE Trans Image Process 7(6):813–824
128. Awel MA, Abidi AI (2019) Review on optical character recognition. Int Res J Eng Technol 6:3666–3669
129. Torki M, Husseiny ME, Elsallamy A, Fayyaz M, Yaser S (2014) Window-based descriptors for Arabic handwritten alphabet recognition: a comparative study on a novel dataset
130. Loey M (2019) Arabic handwritten characters dataset. https://www.kaggle.com/mloey1/ahcd1. Accessed on 31 Aug 2019
131. El-Sawy A, Loey M, Hazem E (2017) Arabic handwritten characters recognition using convolutional neural network. WSEAS Trans Comput Res 5:11–19
132. Alamri H, Sadri J, Suen CY, Nobile N (2008) A novel comprehensive database for Arabic off-line handwriting recognition. In: Proceedings of 11th international conference on frontiers in handwriting recognition, ICFHR, vol 8, pp 664–669
133. Eikvil L (1993) OCR-optical character recognition