Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2022.024555

check for updates

REVIEW

Arabic Optical Character Recognition: A Review

Salah Alghyaline*

Computer Science Department, The World Islamic Sciences and Education University, Amman, 1101-11947, Jordan

*Corresponding Author: Salah Alghyaline. Email: salah.alghyaline@wise.edu.jo

Received: 01 June 2022 Accepted: 16 August 2022

ABSTRACT

This study aims to review the latest contributions in Arabic Optical Character Recognition (OCR) during the last decade, which helps interested researchers know the existing techniques and extend or adapt them accordingly. The study describes the characteristics of the Arabic language, different types of OCR systems, different stages of the Arabic OCR system, the researcher's contributions in each step, and the evaluation metrics for OCR. The study reviews the existing datasets for the Arabic OCR and their characteristics. Additionally, this study implemented some preprocessing and segmentation stages of Arabic OCR. The study compares the performance of the existing methods in terms of recognition accuracy. In addition to researchers' OCR methods, commercial and open-source systems are used in the comparison. The Arabic language is morphologically rich and written cursive with dots and diacritics above and under the characters. Most of the existing approaches in the literature were evaluated on isolated characters or isolated words under a controlled environment, and few approaches were tested on page-level scripts. Some comparative studies show that the accuracy of the existing Arabic OCR commercial systems is low, under 75% for printed text, and further improvement is needed. Moreover, most of the current approaches are offline OCR systems, and there is no remarkable contribution to online OCR systems.

KEYWORDS

Arabic Optical Character Recognition (OCR); Arabic OCR software; Arabic OCR datasets; Arabic OCR evaluation

1 Introduction

Optical Character Recognition (OCR) detects and recognizes the printed and handwritten text from an image and converts it to editable text. The editable version of the image is usually used for further processing operations like indexing, searching, analyzing, and modification. The OCR has many applications such as data entry [1], vehicle license plate recognition [2], postal address reading [3], bank cheque reading [4], intelligent driving systems [5], and invoice reading [6]. The OCR systems are either printed-based or handwritten-based OCRs. People have different writing styles; therefore, recognizing handwritten scripts is more challenging than recognizing printed text and the expected accuracy is low. There are online and offline OCR systems. The online OCR performs the recognition result in real-time and only uses handwritten scripts. Most OCR approaches are designed to extract the printed script from a single image document. However, few approaches are proposed for video-based OCR.

Most of the existing Arabic OCR systems are either handcrafted or learned-based OCR systems. Deep learned features showed superior results compared with handcrafted features during the last

This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

decade in terms of recognition accuracy and speed in many fields of image processing [7,8]. Deep learned-based OCR systems use Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNN) for feature extraction and classification [9,10]. The CNN architecture consists of a sequence of layers. The CNN layers include convolutional, max pooling, sampling, and a fully connected layer. Arabic handcrafted OCR systems usually extract features from contour features [11], statistical features [12], topological features [13], geometrical features [14], from feature descriptors such as SIFT descriptor [15], Speeded Up Robust Features (SURF) [16] and Histograms of Oriented Gradients (HOGs) descriptor [17].

Most of the existing OCR systems work on Latin, Japanese, Chinese, and Korean scripts [18]. Many authors mentioned that few efforts were made in Arabic OCR compared with Latin and English scripts [19,20]. During the last decade, there has been dramatic advancement in the development of Graphics Processing Units (GPUs). This significantly speeds up the computation time and allows scientists to design deep CNNs that can extract and process more features. The Deep CNNs methods achieved state-of-the-art results in many fields of computer vision and image processing. CNNs extract features directly from raw pixels without any preprocessing operations, are invariant to object class changes, handles inputs with high dimension, and have distinct features [21]. Tian et al. [22] proposed OCR approach to recognize English characters from natural images. The method was inspired by the success of using deep CNNs for object recognition. The architecture is based on the VGG16 CNN model followed by a Bi-Directional Long Short-Term Memory (LSTM). The method achieved state-of-the-art on ICDAR2013 and ICDAR2015 benchmarks. Ye et al. [23] developed an OCR approach called TextFuseNet to recognize English texts with irregular shapes from natural scenes. The CNN architecture uses the ResNet model to fuse and extract characters, words, and global text features. The method achieved state-of-the-art on the following datasets: CTW-1500, ICDAR2015, ICDAR2013, and Total-Text. Mackay et al. [24] proposed OCR to recognize real images that contain English words. The approach is called Rotational You Only Look Once (R-YOLO). The system is based on the YOLO4 CNN object detector architecture. Non-Maximum Suppression is used to eliminate the redundant bounding boxes. The method detects text with arbitrary rotation angles. The method results outperformed the state-of-the-art in the following datasets ICDAR2017-MLT, ICDAR2013, ICDAR2015, and MSRA-TD500.

Each language has different writing structures and styles. The Arabic language is written cursive, from right to left, and diacritics are used and can change the word meaning accordingly. Recognizing Arabic script is more challenging than English script due to many facts [25–27]: the cursive nature of the Arabic language, the high similarity between the Arabic letters, the Arabic language is very rich morphologically, the diacritics in the Arabic language can change the word meaning. Most of the existing Arabic OCR approaches in the literature are tested under a controlled environment with some constraints, such as datasets with high-quality images, isolated characters, or isolated words, and few OCRs were tested on a page-level script. Some approaches were used with specific font types and sizes. Most of the existing Arabic OCR approaches ignore diacritics. According to an experiment [28] that evaluated four well-known OCR systems, Sakhr, ABBYY, RDI, and Tesseract, the recognition accuracies are 51.56%, 75.19%, 46.00%, and 48.61%, respectively (font type was Arabic transparent). The experiments were conducted on the KAFD dataset, a page-level printed text dataset. Hegghammer [29] evaluated the performance of Google Document AI and Tesseract OCR systems in English and Arabic. The systems' performance was lower in the Arabic language compared with English. Despite the evaluated English dataset was challenging and included scripts from old books and different font styles. In contrast, the Arabic dataset includes simple font styles and writings from the internet like Wikipedia. The word recognition rates for Document AI and Tesseract are 85.8% and 65.2% for

English scripts and 80.2% and 58.9% for Arabic scripts, respectively. The OCR for Arabic script is still an unsolved problem for printed and handwritten scripts, especially for page-level scripts [28–30].

This study reviews the main contributions in Arabic optical character recognition during the last decade.

The review will be useful for all readers interested in Arabic OCR: non-expert people can have an overview of the Arabic OCR techniques. At the same time, the expert readers can extend the existing methods and datasets or adapt them to their applications. The contributions of this article are summarized as follows:

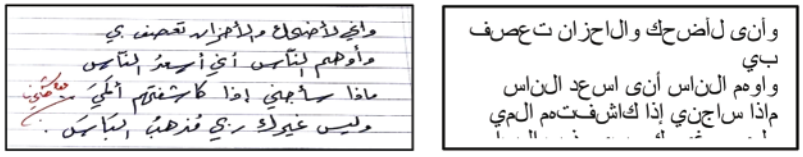
1. Explain the main characteristics of the Arabic language.
2. Report the main phases of the Arabic character recognition system, the different techniques used to handle each stage, and the researchers' contributions in each step.
3. Survey the used datasets for Arabic OCR and their characteristics.
4. Provide a comprehensive comparison between the existing Arabic OCR approaches and software.
5. Describe the evaluation metrics for OCR.
6. Implement some preprocessing and segmentation techniques for Arabic OCR.

2 Literature Review

2.1 Printed and Handwritten OCR

Handwritten OCR converts symbols and characters written by natural hand to editable text. The characters are written on paper or directly on a touch screen using a pen or fingers. Printed OCR takes an image containing printed text as input and converts it into editable text. Recognizing handwritten text is more challenging than recognizing printed text. People have diverse ways of writing, which makes it difficult even for a human to recognize it.

Fig. 1 shows a sample of Arabic scripts. The handwritten script is usually written in cursive, even in Latin scripts. Segmenting cursive scripts is more complicated than segmenting printed scripts [31,32]. Handwritten character recognition has essential applications such as sorting postal mail, processing handwritten forms, and processing bank checks [33]. The handwritten script has many sizes, orientations, and resolutions compared with printed text, and there are no standard font sizes and orientations.



A- Handwritten Arabic Text **B-** Printed Arabic text

Figure 1: Sample of handwritten and printed Arabic text

2.2 Online and Offline OCR

Online OCR is used with handwritten scripts. The word is recognized immediately in real-time after it is written. Usually, a pen is used to write the text on a touch screen. The characters are represented as a point in 2-D space. The time sequence of writing the characters (temporal

information) could help segment the characters, especially for non-cursive characters [34]. Online OCR is usually achieved higher accuracy than offline OCR because much information is captured during the writing time, such as the direction of the stroke, speed, and order. In Offline OCR, the recognition result does not appear immediately and needs some time, depending on the OCR speed. The Offline OCR is used with handwritten and printed scripts [35].

2.3 Image and Video-Based OCR

In image-based OCR, the system receives a single image as input and outputs a single editable text file. In the video-based OCR, the temporal information from different frames is used to recognize the text [19,36]. The exact text appears in a sequence of frames; therefore, the text repetition boosts the recognition accuracy. The same word could have many recognition results, and the result with the highest probability is chosen. Fig. 2 shows three frames taken at different seconds, showing the exact text.

Figure 2: Sample of a printed text at different video frames

3 Characteristics of the Arabic Language

The Arabic language is the official language in all Arab countries. According to World Bank statistics [37], the population in Arab countries will be more than four hundred million people in 2020. Muslims worldwide use the Arabic language to read the Holy Quran written in the Arabic language. The Arabic alphabet is used in many other languages such as Persian, Pashto, Kurdish, Urdu, Punjabi, Brahui, Baluchi, and Kashmiri. The Arabic language consists of twenty-eight letters in addition to Hamza (ء). The Arabic letter shape is changed according to its location in the world. At most, there are four shapes for each letter beginning, middle, end, and alone, as shown in Table 1. Dots are used in the Arabic language and change the letter meaning according to their locations, as shown in Fig. 3. Fig. 4 shows how the shape of the letter Ayan ("ع") changes according to its location in the word. Unlike Latin text, the Arabic language is written in a cursive way and from right to left direction, and there are no capital and small letters. The cursive property makes it difficult to segment the Arabic word into characters. The Arabic language has many diacritics that can change the word meaning accordingly. Arab can understand the word meaning from the sentence context without writing the word's diacritics. Table 2 shows the main Arabic diacritics with their pronunciation. Table 3 shows examples of how diacritics change the word meaning. The Arabic language has a rich and complex morphological structure [25,26]. Table 4 shows some but not all variations of the root Katab (كتب). According to [27], recognizing Arabic script is more challenging than recognizing English script. The Character Error Rate (CER) for Arabic is 3.3% *vs.* 1.1% for English. According to the authors, that is due to the high similarity between Arabic characters, the Arabic language is written in a cursive way, and it has many ligatures and a variety of Arabic fonts and styles.

Table 1: List of Arabic characters

Arabic letter	Alone	Beginning	Middle	End
Alif	ا			ـا
Ayn	ع	عـ	ـعـ	ـع
Baa	ب	بـ	ـبـ	ـب
Daad	ض	ضـ	ـضـ	ـض
Daal	د			ـد
Faa	ف	فـ	ـفـ	ـف
Gaaf	ق	قـ	ـقـ	ـق
Ghayn	غ	غـ	ـغـ	ـغ
Haa	ه	هـ	ـهـ	ـه
Haaa	ح	حـ	ـحـ	ـح
Hamza	ء			
Jiim	ج	جـ	ـجـ	ـج
Kaaf	ك	كـ	ـكـ	ـك
Laam	ل	لـ	ـلـ	ـل
Miim	م	مـ	ـمـ	ـم
Nuun	ن	نـ	ـنـ	ـن
Raa	ر			ـر
Saad	ص	صـ	ـصـ	ـص
Shiin	ش	شـ	ـشـ	ـش
Siin	س	سـ	ـسـ	ـس
Taa	ظ	ظـ	ـظـ	ـظ
Taaa	ت	تـ	ـتـ	ـت
Thaa	ث	ثـ	ـثـ	ـث
Thaaa	ط	طـ	ـطـ	ـط
Thaal	ذ			ـذ
Waaw	و			ـو
Xaa	خ	خـ	ـخـ	ـخ
Yaa	ي	يـ	ـيـ	ـي
Zaay	ز			ـز

Figure 3: Dots at different locations in the word

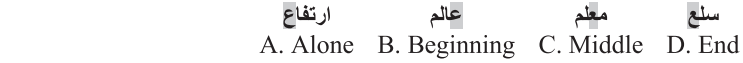


Figure 4: Ayn letter (ع) at different locations in the word

Table 2: Arabic diacritics with their pronunciation

Name	Diacritic	Example	Pronunciation
◌َ	Fatha	مَ	Ma
◌ِ	Kasra	مِ	Me
◌ُ	Damma	مُ	Mo
◌ً	Tanween Fath	مً	Man
◌ٍ	Tanween Kasr	مٍ	Men
◌ٌ	Tanween Dam	مٌ	Mon
◌ْ	Sukun	مْ	M
◌ّ	Shadda	مّ	Mma

Table 3: Examples of how Arabic diacritics change the word meaning accordingly

The Arabic word	The meaning	Pronunciation
حَسَبَ	To calculate	Ha-sa-ba
حَسِبَ	Assume	Ha-se-ba
حَسَب	According to	Ha-sab

Table 4: Some of the variations derived from the root Katab (كتب)

Transliteration	Arabic word	English meaning
Kataba	كتب	Wrote
KitaAb	كتاب	Book
Makotuwb	مكتوب	Written
Kateb	كاتب	Writer
Yakitob	يكتب	Write
Makotabaph	مكتبة	library
Maktab	مكتب	Office
Maktabat	مكتبات	libraries

4 Main Steps of the OCR System

The OCR system includes 4 stages: preprocessing, segmentation, feature extraction and classification as shown in Fig. 5.

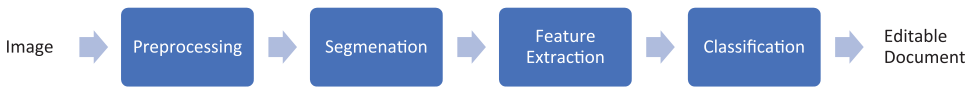


Figure 5: OCR main stages

4.1 Preprocessing

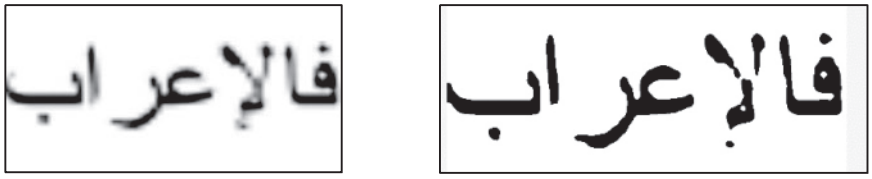
The captured images could suffer from noise interference; therefore, preprocessing operations are applied to the digital image to improve the image quality. Improving image quality is crucial, primarily if this image is used for further processing, such as feature extraction, object recognition, and action recognition. A comparative study [38] on four known OCR systems, Google Docs, ABBY FineReader, Tesseract, and Transym, showed that basic image preprocessing operations such as converting image color to grayscale, brightness, and contrast adjustment improved the recognition accuracy of all systems up to 9%. Illumination adjustment includes brightness and contrast operations to increase the object's sharpness and show the contours clearly. Shen et al. [39] used preprocessing to enhance the image quality for character recognition. The image contrast is adjusted using non-linear transformation, then the image color is changed to a gray color, and a threshold is used to filter pixels. The experiments proved that the preprocessing improves the recognition accuracy by 21%, 8%, and 2% for Hanwang, ABBY FineReader, and Tesseract OCR systems, respectively. Binarization operation includes converting the color or grayscale image pixels to two colors, black and white. Usually, a specific threshold is applied to classify pixels. The Binary image is easier to process due to reducing the color range from three channels, each channel with 256 values to two values, and in character recognition, the color of the character is not essential for character classification. Fig. 6 shows converting the colored image into grayscale and binary images.

A- Colored image **B-** Grayscale image **C-** Binary image (white and black)

Figure 6: Covering image color to grayscale and binary colors (using rgb2gray and im2bw MATLAB functions)

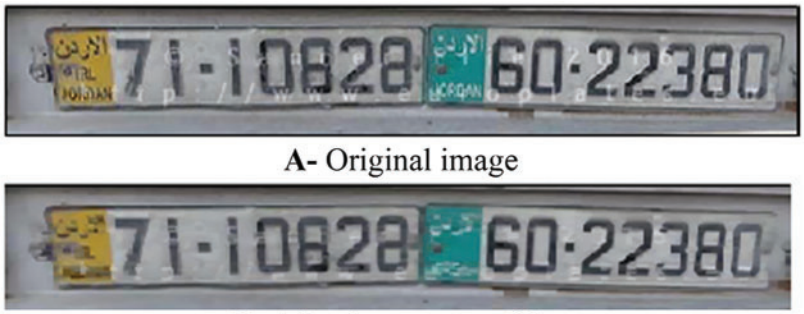
Many techniques are applied to remove the noise in the image, such as statistical noise removal and morphological operations. Morphological operations include removing, filling, dilation, erosion, closing, and opening operations. Statistical operations like median filter reduce the noise by replacing a set of neighbor pixels with their median pixel, as shown in Fig. 7. Removing operation replaces 1 s by 0 s, where the surrounding pixels are 0 s. The filling operation replaces 0 s by 1 s, where the surrounding pixels are 1 s. Dilation is used to add pixels to the boundaries of the object. As a result, the boundaries will become thicker, it will fill the small holes in the object, and the object will become more visible. Erosion operation removes pixels from the object boundaries, and this could remove isolated pixels that are not related to the crucial objects in the image (noise). The closing operation performs a dilation operation with a specific kernel size followed by an erosion operation. The opening operation performs an erosion operation with a specific kernel size followed by a dilation operation. Opening and closing operations are combined to smooth the image contour lines and remove the image's small holes. Therefore, it removes the background objects and keeps the script in the image, as shown in Fig. 8. Scanned images or images taken from a phone could suffer from the skew problem and positioning the image in the correct direction enhances the recognition rate of the OCR system. Malik et al. [32] and Ahmad et al. [40] proposed Arabic OCR that corrects the skewness in the input

image reported that the skewness correction enhanced the character segmentation. In Fig. 9, Hough Transform is applied to correct the image skewness. Thinning is used to reduce the number of pixels to one pixel. As a result of the thinning operation, the skeleton of the text will remain, as shown in Fig. 10. The character's skeleton includes valuable information that can be used to distinguish the character. The disadvantage of the thinning operation is losing character information by reducing the number of pixels, and the shape of the skeleton could be different from the original character shape. Alghamdi et al. [41] proposed thinning technique for Arabic text that preserves the dots above the Arabic characters and the connectivity between characters. The background of the image contains many noises and unnecessary features for OCR. Therefore removing the background and keeping the script will improve the OCR recognition rate [42,43]. Fig. 11 shows that removing the background eliminates many unwanted features for OCR and keeps the text-only. Nosseir et al. [44] proposed OCR system to extract the numbers from Egyptian identity cards. The preprocessing operations include image cropping, converting the image into grayscale color, color reversing, converting the image into a black-white, and dilation operation. Talaat et al. [13] proposed an Arabic OCR approach based on a set of preprocessing operations. The input image is converted to a black-white, the skew is corrected, a set of morphological operations is applied: filling, bridging, removing, and dilation, and finally, the image pixels are normalized by applying a median after. Figs. 6 to 11 show a sample of preprocessing operations implemented using built-in MATLAB functions.



A- Original image **B-** After Binarization and applying the median filter.

Figure 7: Applying median filter (using im2bw and medfilt2 MATLAB functions)



A- Original image

B- After image smoothing

Figure 8: The opening operation is followed by the closing operation to smooth a license plate (using imopen and imclose MATLAB functions)

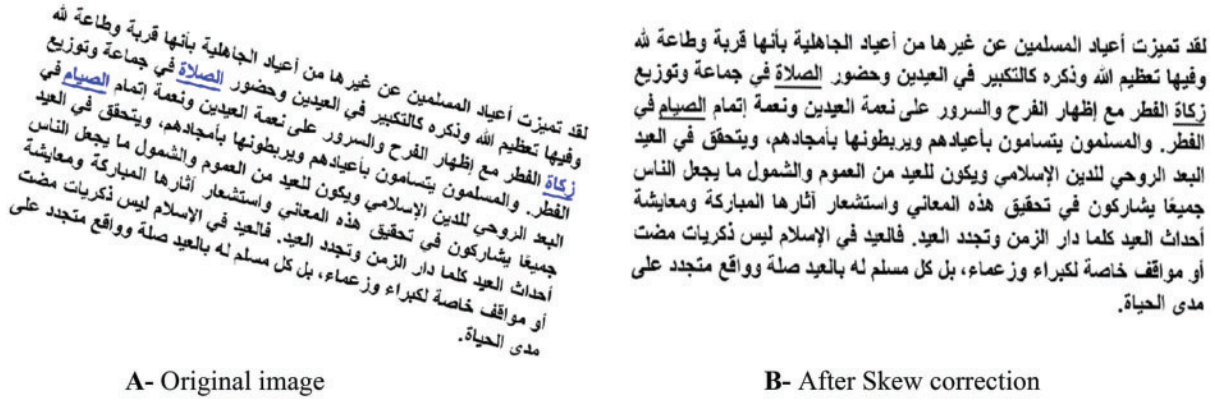


Figure 9: Using hough transform in MATLAB for text skew detection & correction

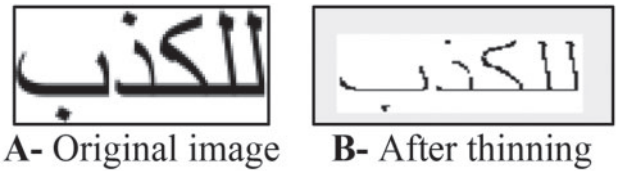


Figure 10: Thinning morphological operation (using bwmorph MATLAB function)

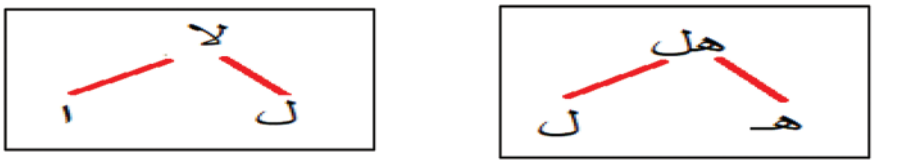
Figure 11: Removing image background

4.2 Segmentation

The goal of this stage is to detect different words or characters in the script. Then, the features of these words or characters are extracted and classified into various labels (classes). There are three segmentation methods for the Arabic script, the segmentation-free method, the segmentation-based method, and the segmentation-hybrid method. The segmentation-free method (holistic method) [11] segments the script into words, and then a template matching technique is used to find the similarity between these templates and the predefined dictionary templates. Template matching works well with printed text, sensitive to font size and style [45]. The holistic method usually works with OCR systems that recognize a limited number of words, like a set of cities' names in a country. The segmentation-based method (analytical approaches) segments the script into characters. The traditional way of the analytical approach is to segment the script into lines, then segment the lines into words, and finally

segment the words into characters. Some analytical approaches segment the script into characters directly using the sliding windows with the template matching method. The analytical approach is applicable to recognize most language words.

However, developing such systems needs more effort and processing time. The segmentation-hybrid method segments the script into ligatures and characters. The ligature is a sequence of connected characters that always come together. Fig. 12 shows an example of two Arabic ligatures. Each ligature consists of two letters. Many approaches proposed Arabic OCR [46–48] based on ligature segmentation instead of character segmentation. Ligature segmentation would be a suitable alternative, and it is used instead of using character segmentation. However, the number of ligatures is large, and there is no way to count them accurately. The incorrect segmentation will result in false recognition results. Usually, the segmentation process passes through three stages: line, word, and character segmentation [49]. Segmenting the Arabic script is a challenging problem, and this is due to the cursive nature of the Arabic language. One of the popular segmentation methods is using histogram projection. There are two types of pixel projections. The horizontal project segments the script into lines, and the vertical projection is used for words and character segmentation. The image is converted into greyscale color. Then, the sum of pixel values at each row is calculated for horizontal projection, whereas in the vertical projection, the sum of pixel values at each column is calculated. The horizontal projection can segment the script into different lines, where the white color in the histogram represents the new lines. In Fig. 13, the horizontal projection can detect the white spaces between the paragraph lines. The red lines represent the segmentation point. In Fig. 14, the vertical projection can detect the white spaces between the words. However, due to the cursive nature of Arabic, it is not easy to segment the characters. MATLAB is used to implement the segmentation parts in Figs. 13 and 14. The sum of pixels of each column in the image is calculated to find the vertical image histogram. Whereas the sum of pixels of each row in the image is calculated to find the horizontal image histogram. It is clear from the figures that the white areas between lines and words are shown with high intensity and can be identified after thresholding pixels intensity. Kiaei et al. [50] proposed Arabic an OCR method based on horizontal and vertical pixels projection. A template matching technique and the sliding window were used to find the similarity between the template bank and the image part.

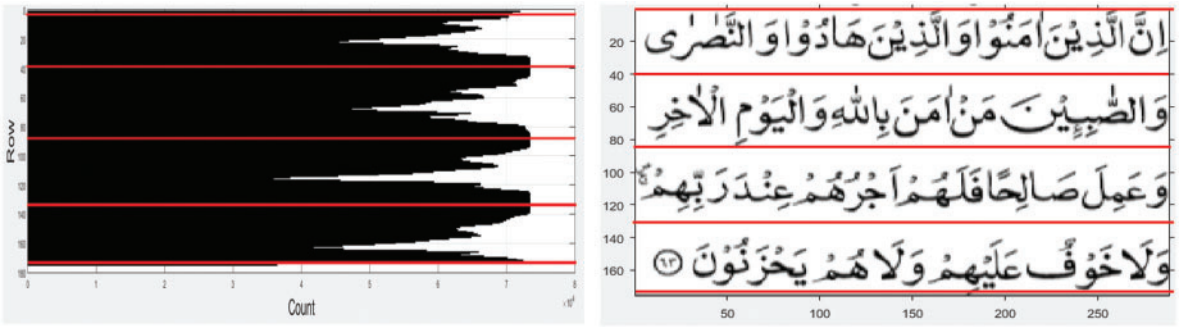


A-" La" ligature (means "No" in English) **B-** Hal(means "do you" in English)

Figure 12: Examples of Arabic ligatures consist of two letters

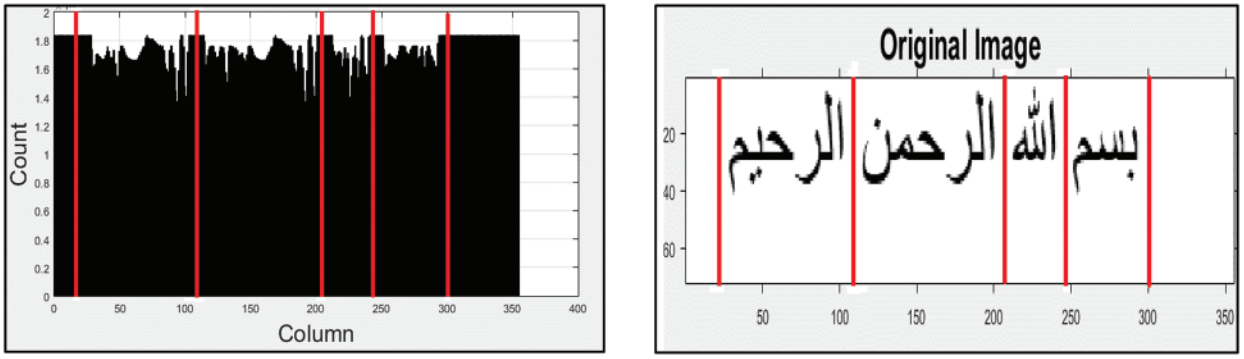
Thinning and contour tracking are used to segment the Arabic script. Skeleton includes character strokes, the direction of strokes, extreme points, characters intersections, and characters' dots [51]. Segmenting the characters from the skeleton is easier than segmenting the original character [48–52]. The contour has valuable information, the contour geometrical information is used to recognize the character, or descriptors such as SIFT, HOG, and SURF are used to describe the character features, especially the character corners and strokes. Osman et al. [53] proposed an Arabic OCR approach that segments the Arabic script into words, and then thinning operation is applied. Finally, the contours of the thinned words are tracked to identify the points where the contour switches from horizontal to vertical and consider these points as segmentation points. Qaroush et al. [20]

proposed an algorithm to segment Arabic text. The input is an image of Arabic line text (APTI dataset), and the output is a set of images containing a single Arabic character. Image profile Project and the Interquartile Range method segment the text line into words. Image profile Project, statistical and topological information are used to segment words into characters. Elkhayati et al. [54] proposed an approach to segment Handwritten Arabic characters based on morphological operations (erosion and Dilation) and directed CNN architecture. The segmentation approach achieved 97.35% accuracy on IFN/ENIT dataset.



A- Horizontal Histogram Projection B- Segmentation Result

Figure 13: Histogram of the horizontal projection



A- Vertical Histogram Projection **B-** Segmentation Result

Figure 14: Histogram of the vertical projection

4.3 Feature Extraction

It represents the most critical stage in the OCR system. In this stage, the features of the segmented parts are collected. The language characters have different features that distinguish characters from each other. There are two types of features: Handcrafted features and learned features.

4.3.1 Handcrafted Features

Represent the features that a data scientist designs [55]. Scale-Invariant Feature Transform (SIFT) descriptor is used widely in Arabic OCR. It is robust to image scaling and rotation. The key points are detected using the Difference of Gaussians (DoG), and then each key point is described with a 128-dimensional descriptor. The descriptor includes information about the pixel's gradient and the magnitude of the gradient. Chergui et al. [56] used SIFT descriptor to train a model to classify words

based on matching interest points descriptors. Twenty-five images from IFN/ENIT dataset are used to train each class label. Zahedi et al. [57] and Stolyarenko et al. [58] proposed Arabic OCR based on SIFT features and keypoint matching. Hassan et al. [59] used SIFT descriptor for feature extraction and bag-of-words framework with SVM to classify features into different words' labels. Histograms of Oriented Gradients (HOGs) [60] descriptor divides the image into small regions and then calculates each region's gradient and orientation. HOG is used in some Arabic OCRs. Jebril et al. [17] used HOG to build an Arabic OCR system to recognize handwritten Jordanian cities' names. The system performance was evaluated on 13,000 images and achieved a 99% recognition rate. Khaissidi et al. [61] used HOG to detect and describe features of handwritten scripts from the Ibn Sina dataset. The system achieved a 68.4% recognition rate. Speeded-Up Robust Feature (SURF) descriptor [62] is three times faster than SIFT. SURF uses the Hessian matrix to locate the interest points. Then, each interest point is divided into sub-regions. Haar wavelet is taken from each sub-region to represent the SURF descriptor. Alsimry et al. [63] proposed an Arabic OCR system to find duplicate words in the image. SURF descriptor is used to extract the features, and the Euclidean Distance is used to find the similarity between different words.

Bagasi et al. [64] proposed Arabic image retrieval system based on image content. SURF and Binary Robust Invariant Scalable Keypoints (BRISK) [65] descriptors extract and describe image features. Hamming distance is used to find the similarity between the descriptors of the stored image with newly tested mages. Torki et al. [66] performed a comparative study on the performance of handcrafted descriptors on Arabic OCR. The experiments show that SIFT outperformed SUFR and HOG descriptors in recognition Rate. SIFT achieved 94.28%, whereas HOG and SURF achieved 90.46% and 7.21% recognition rates. The geometrical and statistical features were used to extract character features in many Arabic OCR approaches [14,67,68]. Geometrical features include height and width of the character, distances, and area calculation. At the same time, statistical features include the number of white and black pixels, the number of transitions for horizontal and vertical pixels, pixel density, and probability distribution. Zernike moments, Legendre moments, and Fourier descriptors represent statistical features. They were used by many Arabic OCR systems [69–71]. Zoning is used to extract features by dividing the image into regions of equal sizes horizontally and vertically, and then the regions with the black pixels are used as a feature by calculating the sum of pixels intensities at each zone [72,73].

4.3.2 Learned Features

During the last decade, Convolutional Neural Networks (CNNs) features [74–77] achieved state-of-the-art results in object detection compared with handcrafted features. The deep network includes many layers. The number of layers exceeds 100 layers for the very deep CNN architectures. Each layer convolves a set of filters on the image pixels, as in Fig. 15. Many popular CNNs architectures such as DenseNet, AlexNet, VGGNet, MobileNet, SqueezeNet, ResNet, and GoogLeNet can be trained on a dataset of images to detect custom objects like language characters. Radwan et al. [78] proposed an Arabic OCR approach based on three neural networks. The first neural network is to detect the font size, and then the script is modified to an 18 pt font size. The second network segments the words into characters. Finally, the third convolutional network is used to recognize the characters. The CNN consists of two convolutional layers and two max pooling layers. The max pooling kernel size is 2×2. The first and the second ConvNets have the same structure. Each ConvNet includes 64 filters with a 3×3 kernel size. The last layer consists of a fully connected layer with a dropout of 25% of the nodes. Ahmed et al. [9] proposed CNN architecture for printed Arabic OCR. The input image is converted to a grayscale image, and then it is resized to 50×50 pixels. The network includes two convolutional layers (3×3 kernel size), and a stride value is 2, each one followed by a max pooling

layer and one fully connected layer. Butt et al. [10] proposed a video text recognition approach for the Arabic language. It is based on CNN with Recurrent Neural Networks (RNN). The convolutional layers are similar to VGG architecture [79]. Elleuch et al. [80] proposed an OCR approach based on CNN and Support Vector Machine (SVM). The CNN includes two convolutional layers and two subsampling (4×4 kernel size) layers. The first ConvNet contains six filters (5×5 kernel size), and the second ConvNet includes 12 filters (8×8 kernel size). The last layer includes a fully connected layer with a dropout of 50% of the nodes. The output from the last layer represents the last features that are input into the SVM classifier. Mustafa et al. [81] proposed CNN architecture for handwritten Arabic OCR. The architecture includes four convolutional layers, two max pooling layers, and a fully connected layer with a dropout of 20% of the nodes. Naz et al. [82] proposed Pashto ligature (subword) OCR approach. The approach achieved 99.31% using the DenseNet CNN architecture. They used the FAST-NU dataset of Pashto ligatures to evaluate the approach accuracy. Sokar et al. [83] proposed Arabic OCR based on Siamese CNN architecture. The Architecture includes two CNNs with a similar design. The CNN architecture contains two convolutional layers. The first ConvNet contains 100 filters (5×5 kernel size), and the second ConvNet includes 150 filters (5×5 kernel size). Each ConvNet is followed by a max pooling layer (2×2 kernel size). The architecture of the last layer is a fully connected layer. It reported that the architecture is robust to noise and can be applied to any new dataset without retraining the CNN on the new dataset. Ashiquzzaman et al. [84] proposed CNN to recognize handwritten Arabic numbers. The CNN architecture contains two convolutional layers. The first ConvNet contains 30 filters (5×5 kernel size), and the second ConvNet includes 15 filters (3×3 kernel size). Each ConvNet is followed by a max pooling layer (2×2 kernel size). The last layer is a fully connected layer with a drop out of 15% of neurons.

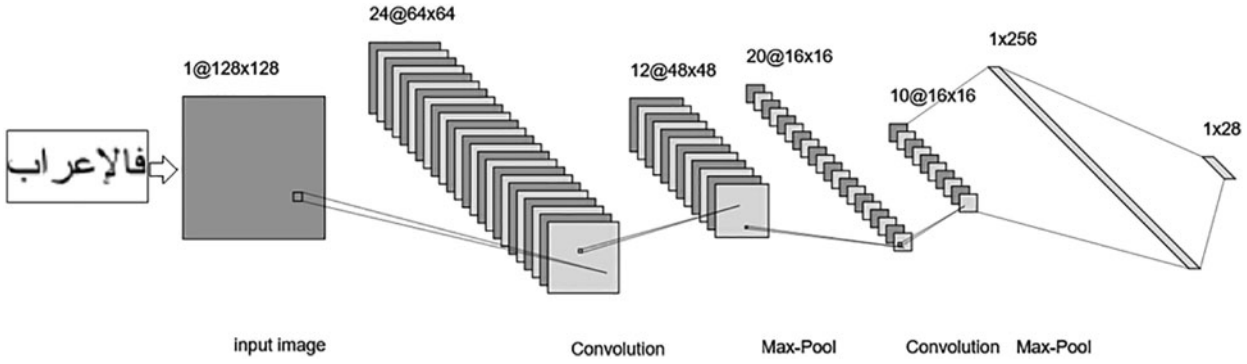


Figure 15: A sample of deep learning architecture: the input is an image of $1 \times 128 \times 128$ dimensions. The first ConvNet contains 24 filters (3×3 kernel size), and the second ConvNet includes 20 filters (3×3 kernel size). Each ConvNet is followed by a max pooling layer (2×2 kernel size). The last two layers are fully connected layers

4.4 Feature Classification

The extracted features from the Arabic script represent valuable information and are used to distinguish characters and words from each other. The feature classification is the last stage in the OCR system and is used to identify the character's label or class based on the collected features.

4.4.1 Template Matching

The simplest way to classify characters is using template matching between the testing sample with an unknown class label and the dataset with a known class label. It is used when there is a

limited number of words in the dataset, such as signages, cities, and numbers. Hamming distance [64], Euclidean distance [63,85], Cosines and scalar product [86], and Normalized Cross-Correlation (NCC) [87] were used to find the similarity between templates in many Arabic OCR approaches. Farhat et al. [88] segmented the image of the Qatari Plate into characters, then each character image is divided into four zones, and finally, template matching is used to recognize the character label. Nosseir et al. [44] extracted SURF features from the Egyptian ID Cards, then template matching is used to classify the characters. Hairuman et al. [89] used template matching to recognize signage images. The disadvantages of template matching are that the templates of each character must be stored in the memory. It has low accuracy compared with other classification approaches. Finally, it is sensitive to image noise and could fail if there is a variation with input image scaling and rotation [90].

4.4.2 Naïve Bayes Classifiers

It is based on the Bayes probability theorem. It calculates the probability of the class label using previously known probabilities about the event. Bayes classifier assumes that all features are independent and have no relations between them. Eq. (1) shows the formula of the Bayes theorem. Abdalkafor et al. [91] proposed a handwritten Arabic OCR approach, the character's image is divided into 3×3 zones for feature extraction, and then Naïve Bayes is used for classification. The approach was evaluated on the CENPARMI dataset and achieved a 97.05% recognition rate. Saeed et al. [92] proposed Arabic OCR to classify cities name, the Maximally Stable Extremal Regions (MSERs) were used for extracting features, and Naïve Bayes with support vector machine (SVM) were used for classification. The method achieved a 92.64% recognition rate on IFN/ENIT dataset. Jayech et al. [93] used Bayesian Network to classify segmented handwritten Arabic characters. The approach was evaluated on IFN/ENIT and achieved an 82% recognition rate.

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{1}$$

$X = \{x_1, x_2, \ldots, x_n\}$ set of feature set.

$Y = \{y_1, y_2, \ldots, y_n\}$ set of class label set.

4.4.3 Support Vector Machine (SVM)

It is one of the most successful and used classifiers in many machine learning applications. It plots the data features in n-dimensional space and then finds the line or the plane that differentiates classes from each other, as shown in Fig. 16. The LIBSVM tool [94] is a free library that implements most SVM techniques and supports many programming languages. Elleuch et al. [95] proposed Arabic handwritten OCR based on Gabor filter for features extraction and one-against-all RBF kernel SVM for classification. The rate of classification error is 11.23% on the HACDB database. Yamina et al. [12] proposed Arabic printed OCR-based fifty-six statistical features extracted from the image, and one-against-all SVM is used for multi-class classifications. The approach was evaluated on a private dataset of 7623 characters and achieved a 95.03% accuracy rate. Elzobi et al. [96] classified Gabor transform features using the SVM classifier. The recognition rate is 74% on a set of 5436 Arabic characters from the IESK-arDB dataset.

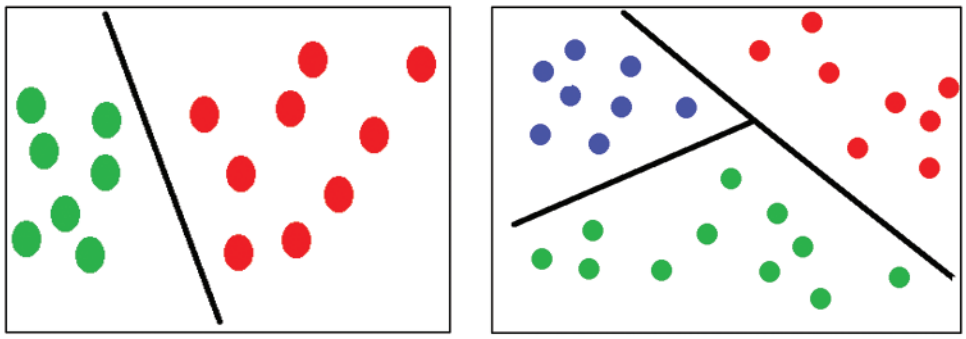


Figure 16: Illustration of the support vector machine classification: a set of classes are represented by different colors and are linearly separable

4.4.4 SoftMax

Artificial Neural Networks are used for feature extraction and classification simultaneously. The SoftMax classifier represents the last layer in the CNN architecture. SoftMax classifier is used to normalize the vector values from the output of the neural network between 0 and 1. The normalized value represents a probability for each class label. Eq. (2) shows the formula of the SoftMax function. SoftMax gave superior results during the last decade compared with other feature extractors and classifiers [81,97]. Sokar et al. [83] compared the performance of using three classifiers, SVM, KNN, and SoftMax, to classify license plate characters. They reported that the recognition rates were 95.6%, 95.67%, and 98.0% for SVM, KNN, and SoftMax classifiers, respectively.

$$\sigma\left(\vec{x}\right)_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{2}$$

x	Input vector
$\sigma\left(\vec{x}\right)_i$	SoftMax value for i-th element of the vector x
e^{x_i}	Exponential value for i-th element of the vector x
n	Number of classes
$\sum_{j=1}^{n} e^{x_j}$	Total of exponential values for all elements of the vector x

4.4.5 Hidden Markov Model (HMM)

It is a statistical model introduced in the late 1960s and is still used in many applications such as OCR, face recognition, and speech recognition gesture recognition [98]. The model includes a set of states, and each state has a probability. The transition between states is based on transition probability, and the transition from one state to another is called the Markov process [99]. The HMM classifier is used by many Arabic OCR approaches, such as [100–103].

4.4.6 K-Nearest Neighbor (KNN)

It checks the closest K neighbors around the object and then assigns the object to the class with the majority votes. If $K=1$, and then the object class is like the class of its nearest neighbor, K should be an odd number. Usually, Euclidean distance is used to measure the distance between points. Darwish et al. [104] proposed a printed Arabic OCR approach based on the second-order statistics and Fuzzy KNN used for classification. Kessab et al. [105] proposed OCR for Arabic numbers recognition based on zoning for features extraction and KNN and HMM for classification. Fig. 17 shows how the

KNN checks the k-th nearest neighbors of the blue object (nearest neighbor surrounding by a blue circle) extraction and KNN and HMM for classification.

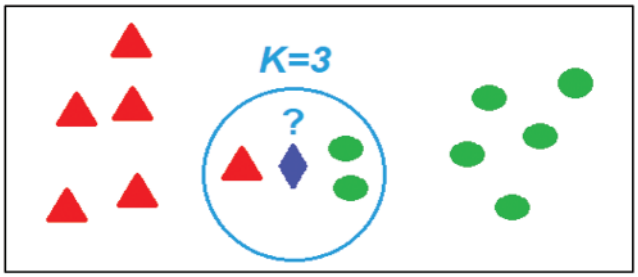


Figure 17: Illustration of the K-Nearest Neighbor classification: the figure shows two classes. A red triangle represents the first class, and a green circle represents the second class. The blue diamond represents a new data point, and the points inside the circle are used to predict the class for that point

4.4.7 Random Forest Tree (RFT)

The decision tree classifier is sensitive to the order of features inside the dataset. Changing the order of inserted features into the decision tree will build different decision trees; each one could have a different classification result. RFT solved this problem by randomly sampling the original dataset to generate different datasets. A decision tree is built for each generated dataset, and majority voting is used to choose the final class label [106], as shown in Fig. 18. Hassanien et al. [107] extracted statistical features based on the character shapes, then two classifiers are used for recognizing isolated Arabic characters, KNN and RFT. According to the authors, RFT recognition accuracy outperformed KNN by 11%. Sahlol et al. [108] used RFT to classify Arabic handwritten features. The extracted features are based on gradient, zoning, and Number of Holes. The approach was evaluated on the CENPRMI dataset and achieved a 91.66% recognition rate.

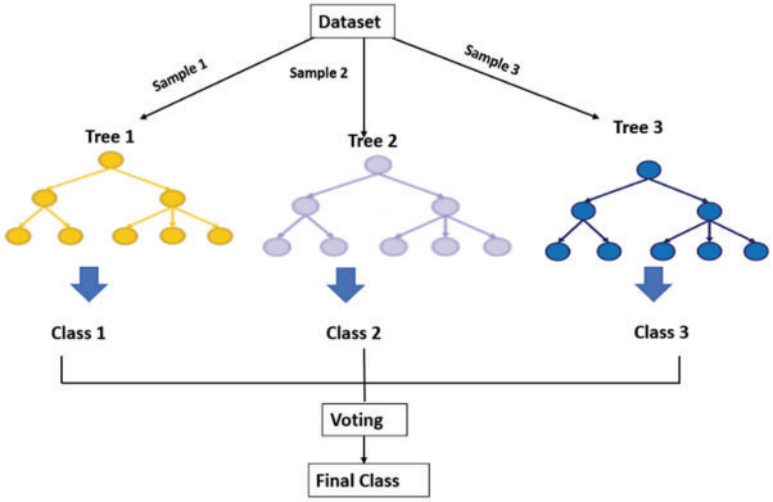


Figure 18: Illustration of the Random Forest Tree (RFT) classification. Three datasets are generated by randomly sampling the original dataset. Each dataset is used to build a separate decision tree classifier. The different decision trees vote to decide the final class label

4.5 Postprocessing

The Arabic language has many letters with similar shapes. Samples of characters with similar shapes are shown in Table 5. This problem can be solved by developing techniques to correct the words that have wrong recognition results after the classification stage. Bassil et al. [109] proposed an OCR post-processing algorithm based on google search engine suggestions to correct the spelling of the OCR false classified words. Doush et al. [110] developed an Arabic post-processing technique based on statistics and rule-based models. They reported that the model reduced the word error rate by 5% [111]. Reported that using the spell checker of both Microsoft word processor and google website corrected 49% of the falsely classified words.

Table 5: Sample of some Arabic letters with similar shapes

Group 1	“ت”“ث”,“ب”, and “ن”
Group 2	“ظ” and “ط”,
Group 3	“ف” and “ق”
Group 4	“ج”, “ح”, and “خ”
Group 5	“س” and “ش”
Group 6	“ز”, “ذ” and “ر”
Group 7	“ه” and “ة”
Group 8	“لا”, “لإ” and “لأ”

5 Arabic OCR Datasets

The APTI dataset was developed in 2009 [112]. It represents a Large-scale printed Arabic dataset for OCR evaluation. The dataset contains 45,313,600 images. Each image contains one Arabic word, with about 250 million Arabic characters. The dataset is synthetic and generated from a distinct 113,284 words. It includes ten font types, ten font sizes (6 pt–24 pt), and four font styles. The dataset is divided into five sets. Set number 5 is used for testing, and the other sets are used for training. The author publicly published the first four sets. Fig. 19 shows a sample of 15 images from the APTI dataset. The dataset is available at https://diuf.unifr.ch/main/diva/APTI/download.html.

Figure 19: Sample pictures from the APTI dataset

The **MMAC** is a printed Arabic text dataset developed in 2010 [113]. The number of unique words and PAWS are 282,593 and 66,725, respectively. The number of images is increased by a factor of three by skewing and adding noise to the images. The dataset was collected from old books, Arabic research,

and the Holy Quran. Fig. 20 shows sample images from the MMAC dataset. The dataset is available at http://www.ashrafraouf.com/mmac.

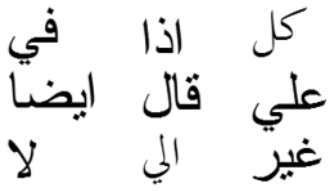


Figure 20: Sample picture from the MMAC dataset

The **HACDB** is an Arabic handwritten dataset [114]. Fifty writers collected it from ages 14 and 50. Each writer writes 132 shapes of characters. The shapes represent the way of writing the Arabic characters at different locations in the word (beginning, middle, end, and isolated). The total number of characters is 6,600. Fig. 21 shows sample images from the HACDB dataset.

Figure 21: Sample picture from the HACDB dataset

The **AcTiV** is a video-based OCR dataset [115]. It includes 80 videos (850,000 frames) collected from four news Arabic channels: Aljazeera, France 24, Russia Today, and EI Wataniya. The dataset includes texts with different sizes, colors, positions, and fonts. Additionally, the background is complex and has many objects with shapes like Arabic characters. Fig. 22 shows sample images from the AcTiV dataset.

Figure 22: Sample picture from the AcTiV dataset

The **Hijja** dataset represents a handwritten Arabic dataset developed in 2020 [116]. 591 children write it under 12 from Riyadh, Saudi Arabia. The dataset contains 47,434 characters that can be used with real-life applications to teach children spelling and handwriting skills. The author reported that the dataset is complex, and it is challenging to train a model to fit the data. Fig. 23 shows sample images from the Hijja dataset. The dataset is available at https://github.com/israksu/Hijja2.

The **KAFD** dataset was developed by king Fahd University and Qassim University in 2014 [117]. It includes 15,068 images and 2,576,024 lines of printed text. Images have different resolutions 100 dpi,

200 dpi, 300 dpi, and 600 dpi. The dataset includes four different Arabic fonts, ten font sizes ranging from 8 to 24 points, and four font styles; Normal, Bold, Italic, and Bold Italic. The dataset images are divided into training, testing, and validation. A sample picture from the dataset is shown in Fig. 24. The dataset is available at http://kafd.ideas2serve.net/.

Figure 23: Sample picture from the Hijja dataset

Figure 24: Sample picture from the KAFD dataset

The **KHATT** is a handwritten text dataset developed in 2013 [118]. The dataset contains one thousand forms written by one thousand different writers. The dataset includes two thousand paragraphs with 9,327 lines taken from forty-six sources. It includes three image resolutions 200 dpi, 300 dpi, and 600 dpi. Seventy percent of the dataset is used for training, 15% for testing, and 15% for validation. Fig. 25 shows a sample of three images from the KHATT dataset. The dataset is available at http://khatt.ideas2serve.net/KHATTDownload.php.

Figure 25: Sample pictures from the KHATT dataset

The **IFN/ENIT** database is one of the oldest handwritten text datasets developed in 2002 [119]. The database includes 2,200 images with 300 dpi resolution for Tunisian cities, the images contain 26,459 words, and the total of Arabic characters is 212,211. The dataset achieved the highest number of citations (640) compared with the existing Arabic OCR database. Fig. 26 shows a sample of three images from the dataset. The dataset is available at http://www.ifnenit.com/.

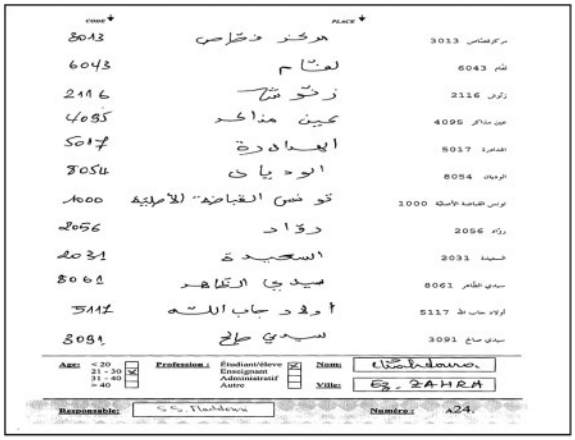


Figure 26: (Continued)

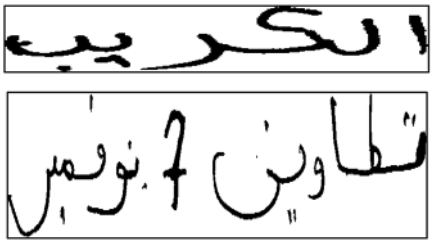


Figure 26: Sample pictures from IFN/ENIT dataset

The **Yarmouk** Arabic OCR dataset is a printed Arabic text dataset developed in 2018 [120]. It includes 8,994 images with 300 dpi resolution. The images contain 436,921 words extracted from the Wikipedia website. Fig. 27 shows a sample image from the dataset. The dataset is available at https://drive.google.com/drive/folders/0B4Kx3iMuktgsdC12Ui1neklnMzQ?resourcekey=0-dX3YkFT4xArRrT81wQ2wSw.

الصادق ثريا - ويكيبيديا، الموسوعة الحرة

الصادق ثريا (القيروان, 4 مارس 1920 - تونس, 12 ديسمبر 2003) هو فنان وملحن تونسي ولد بمدينة الأغالبة، يقول الصادق ثريا عن حياته الفنية ان البداية كانت وعمره 12 سنة.

يقول كنت وأنا في هذه السن أحب الفن. وكنت أستمع إلى غناء عمي الذي كان فنانا وعازفا...وبعد أن درست سنوات في الجامع الكبير بالقيروان، جئت إلى تونس العاصمة لأبدأ عمليا رحلتي مع الفن هنا تعرّفت عل على مجموعة من شعراء تحت السور وأولى أغنياتي كتبها علي الدوعاجي وعبد الرزاق كرباكة والهادي العبيدي ومحمود بورقيبة.

وابتداء من سنة 1938 كنت أنتج الكثير من الأغاني للعديد من المؤلفين بعضهم نسيت اسمه. كانت فترة ذهبية في الإنتاج الفني وكان الجوّ يلهم أحلى الألحان وأجمل الأفكار... وكنت أحيانا أوحي للمؤلف حتى بشكل الأغنية، والإنتاج آنذاك كان دائما مشتركا. التأليف يولد بحضور الملحن واللحن بدوره يولد بحضور المؤلف.

في تلك السنوات اشتغلت مع مجموعة كبيرة من نجوم ذلك العصر (1939 – 1940) فعملت مع محمد التريكي. حسيبة رشدي، شافية وفتحية خيري وعندما تأسست الإذاعة التونسية، بدأت العمل فيها وكان الأستاذ عثمان الكعاك هو الذي انتدبني. وفي سنة 1952 بدأت مرحلة أخرى من حياتي الفنية أردت أن أجرب حظي في فرنسا فسافرت و بدأت أشتغل في إحدى الملاهي وهنا تعرّفت على مجموعة كبيرة من الفنانين الجزائريين والمغاربة. كنا نعمل معا في هذا الملهى ومن هؤلاء أحمد وهبي وإسماعيل أحمد الذي كان آنذاك يعزف على الكمنجة وفي تلك الفترة كان لصاحب الملهى ابنة صغيرة كلفني والدها بأن أعلمها أداء بعض أغاني أم كلثوم. وبالفعل...أذكر. أني بدأت أعلم هذه الطفلة أغنية "سلوا قلبي" لأم كلثوم ورويدا رويدا بدأ صوت هذه الطفلة ينمو وينضج، حتّى أصبحت ألحّن لها أغنيات خاصّة... وأصبحت هي نفسها تشارك في فقرات ملهى أبيها وبقيت أساعدها حتّى رسخت قدميها في الفنّ واشتهرت بعد ذلك. نعم إنها وردة الجزائرية.

وفي المغرب عملت كذلك أشهرا عديدة ذهبت إلى هناك سنة 1958 لأقوم بتنفيذ عقد مدته ستّة أشهر، وهناك قمت بتدريب الفرقة المغربيّة على العديد من الموشحات ومن ضمن الفرقة يوجد الفنانان عبد الوهاب الدكالي وعبد الهادي بلخياط وفي سنة 1962 ذهبت إلى الجزائر وكلّفت بتعليم الموسيقى لأطفال الشهداء، اشتغلت في عنابة وسيدي بلعباس وكنت أشارك كذلك في الإذاعة بتقديم إنتاجي الموسيقي والغنائي.

خلف الفقيد رصيدا غنائيا ثريا كان نتاج شخصيته الموسيقية الأصيلة التي تميزت بالمزج بين المقامات الشرقية و الطبوع التونسية. وتحتفظ خزينة الإذاعة الوطنية بعدد كبير من إبداعاته الغنائية أداء ولحنا نذكر منها :

توفي الصادق ثريا 12 ديسمبر 2003 عن سن تناهز 83 سنة كانت زاخرة بالعطاء والإبداع الفني خدمة لإشعاع الأغنية التونسية.

Figure 27: Sample picture from Yarmouk dataset

The **APTID/MF** is a printed Arabic text dataset developed in 2013 [69]. It includes 1,845 images with 300 dpi resolution. The images contain 27,402 characters. The images are taken from 387 pages of Arabic documents. The images include ten font types, two font styles (normal and bold), and four

font sizes (12 pt, 14 pt, 16 pt, and 18 pt). Fig. 28 shows a sample of three images from the dataset. The dataset is available upon request.

Figure 28: Sample pictures from APTID/MF dataset

The **ARABASE** dataset is a printed and handwritten Arabic text dataset developed in 2005 [121]. More than 400 writers wrote handwritten images, most of them from Tunisia. The printed text was obtained from daily newspapers and the book published by the Tunisian national library on the internet. The image's resolution ranged from 200 dpi to 600 dpi. Fig. 29 shows a sample of two images from the dataset. The dataset is available upon request.

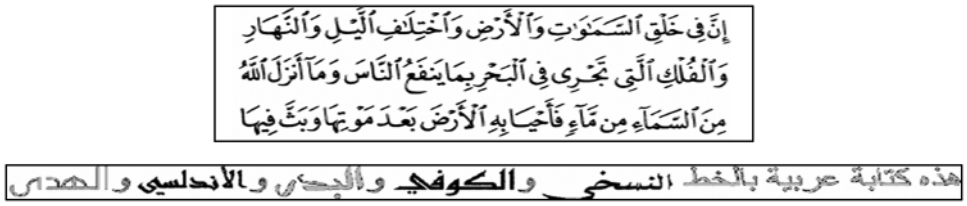


Figure 29: Sample pictures from the ARABASE dataset

6 Recent Arabic OCR Techniques

This section summarizes the techniques and dataset proposed for Arabic OCR during the last decade. Initially, Google Scholar was used to find related articles. A broad set of keywords were used to identify the list of related articles. The used articles are peer-reviewed and have an excellent citation number. The selected articles are relevant to the Arabic OCR, and the methodology of the proposed approaches and datasets were clearly described. In addition to the Google Scholar database, the snowball method was used to find related articles using references from some relevant articles. The search stage includes the following terms: *Arabic OCR, Arabic optical character recognition, Arabic OCR dataset, Arabic OCR Database, handwritten Arabic characters, printed Arabic recognition, CNN Arabic OCR, Handcrafted Arabic OCR, and deep learning Arabic OCR.*

Rosenberg et al. [15] proposed OCR called SOCR to recognize printed Arabic text. The approach used sliding window techniques with SIFT descriptor to segment a line of Arabic text into paws and letters. In addition to SIFT, a set of features are extracted: Mass Center, Black color Histogram, Crosshair, Ratio, and Outline Features. The extracted features are classified using Hidden Markov Model (HMM). The authors reported that the training set is small compared with previous methods. The approach evaluated the PATS dataset and outperformed the PATS [119] results on five out of eight fonts. Additionally, the approach evaluated six font sizes (6, 8, 10, 12, 18, 204) of the set4 of the APTI dataset and achieved a 99.6%-character recognition rate.

Sabbour et al. [11] proposed a Printed Arabic OCR approach called Naboc. It was trained to recognize two types of fonts: Arabic Naskh and Urdu Nastaleeq. A dataset called UPTI was created for Printed Urdu Text. The system's main steps are to segment the page script into lines. After that, the lines are segmented into ligatures. A descriptor is used to describe the features of ligatures. Finally, classify the ligatures into a predefined class label. Contour pixels' intensity and Shape Context are used to describe the ligatures. K-Nearest Neighbor is used for classification. The recognition error rate for Arabic ligature is 16.1%, whereas Tesseract's error rate is 16.2%.

Talaat et al. [13] used preprocessing operations to enhance image quality. The preprocessing operation includes Binarization, Slant Correction, Normalization, Statistical Noise removal, and Morphological operations (filling, Dilation Bridging). The extracted features include Lower and Upper image profiles, Vertical and Horizontal profiles, connected components, and Topological features—Neural Networks (NNs) area used for classification. The input vector for NNs is 133 elements, whereas the output is 28 neurons (number of Arabic letters). The approach achieved 88% accuracy on the CENPRMI dataset. Characters (ـہ and ۓ) achieved low accuracy rates (61% and 66%).

Hafiz et al. [122] performed preprocessing operations: binarized, slant removal resizing, and dilation to improve image quality. For feature extraction, the images are divided vertically into 6-pixel widths. Then, Images are manually segmented into paw images. The following features are extracted: foreground color density, the transition of white and black pixels, the sum of pixels in a vertical column, and concavity features. A combination of HMMS and K-NN is used for classification. The approach achieved 82.67%, 86%, and 94% on splits A, B, and C of the IFN/ENIT database.

El-Sawy et al. [123] developed CNNs to recognize isolated handwritten Arabic letters. The CNN architecture contains two Conv2D and two max pooling with Relu Activation, a fully connected layer, and SoftMax Activation. The input is an image of 32×32 pixels, and the output is a fully connected layer with 28 numbers, representing the probabilities for 28 Arabic letters. The approach achieved 94.9% accuracy on a private dataset.

The OpenITI team developed OCR for printed classical Arabic scripts collected from the old book "Ibn al-Faqīh's Kitāb al-Buldān [124]." The approach is based on a customized Kraken open-source OCR software. The overall recognition rate of the method is 97.56%.

Mudhsh et al. [125] proposed deep CNNs to recognize handwritten characters. The Alphanumeric method is based on the VGGNet architecture. The number of filters was reduced by 8, reducing the time complexity to run the VGGNet. The approach achieved 97.32% and 99.66% accuracies on HACDB and ADBase datasets.

Nashwan et al. [67] proposed a holistic Arabic OCR. The whole world is recognized without the need to segment it into letters. Compared with previous works, the approach used many vocabularies and reduced the recognition time. Clustering techniques cluster similar words' shapes to reduce the recognition time. Two features are extracted: Discrete Cosine Transform (DCT) and local block features. Then the features are clustered, and Euclidean distance is used to find the best possible matched word. The accuracy of evaluating the approach on 3465 words is 84.8%.

Doush et al. [110] reported that their proposed approach represents the first end-to-end Arabic post-processing approach. For many Arabic OCR systems, the OCR output does not match the ground truth text for some characters due to high similarity between Arabic characters. Therefore, the author proposed using post-processing to correct the word spelling. The method is based on: Language Model (LM) to check whether the word is correct or not, the Error Model (EM), and Google to correct the

wrong words. The proposed approach reduced the error rate from 14.95 to 14.42. the approach was evaluated on a dataset of 500 images.

Radwan et al. [78] developed an OCR system to recognize printed Arabic words. The architecture is based on three CNNs. The CNNs were used for three tasks: font size detection, character segmentation, and character recognition. The authors reported that OCR accuracy is 94.38% on APTI synthetic dataset.

Darwish et al. [104] developed printed Arabic OCR. The approach performed preprocessing operations: converting image into grayscale, median filter, morphological operations, correct rotation, and image resizing. Gray Level Co-Occurrence Matrix (GLCM) was used for feature extraction. A genetic algorithm was used to choose the best features and reduced the running time by two. K-Nearest Neighbor classifier is used for classification. The approach was evaluated on PATS-A01 (650 images) and APTI (550 images) datasets and achieved 98.69% and 95.37% recognition rates, respectively.

Fasha et al. [126] proposed a hybrid OCR approach for printed Arabic text. The approach includes five CNNs layers followed by Bi-Directional Short LSTM. The CNN architecture includes five Conv2D layers with Relu for activation and five max pooling layers. The output from the CNN is passed to Bi-Directional LSTM (BDLSTM). The BDLSTM consists of two layers of LSTM. Each layer contains two cells, and each cell has 256 hidden units. The approach was evaluated on the APTI dataset and a chevied 85.15%-character recognition rate and 76.3%-word recognition rate.

Shams et al. [127] proposed OCR for handwritten Arabic text. The Proposed CNNs architecture consists of three convolutional layers and three max pooling layers to recognize handwritten Arabic isolated characters. CNNs extract features, dropout operation, is applied to reduce the running time, and SVM is used to classify the features into 28 classes (number of Arabic letters). The approach achieved 95.07% accuracy.

Altwaijry et al. [116] proposed CNNs architecture to recognize isolated handwritten Arabic characters. The CNNs include three Conv2D layers, each followed by ReLU activation and max pooling. The output is flattened into a fully connected layer followed by two fully connected layers with an 80% dropout rate. The approach was evaluated on AHCD and Hijja datasets and achieved 97% and 88% recognition rates.

Balaha et al. [21] proposed 14 different CNNs architectures for handwritten Arabic OCR. The proposed architecture includes three Conv2D layers and three max pooling layers. The difference between the different architectures is in the number of filters and the fully connected layers. Additionally, combinations of VGG16, VGG19, and MobileNetV2 are evaluated. The accuracy of the proposed CNN-5 architecture is 91.96% on the HMBD dataset, and it requires less memory and processing time compared with VGG16, which achieved 92.74%.

Ahmed et al. [128] proposed CNNs architecture that includes 9 Conv2D layers with 3×3 kernels and five max-pooling layers with 2×2 kernels. Batch normalization is used after each Conv2D layer. Dropout with rates from 0.1 to 0.4 is used after each max pooling layer. The tensor flattens by a fully connected layer followed by another fully connected layer. The authors reported that the proposed method achieved a super result (99.94%) compared with VGGNet-19.

Jbrail et al. [129] developed four CNNs architectures to recognize isolated handwritten Arabic characters. The architectures use a different number of layers (3, 9, 13 layers). Different activations functions (Relu and Softmax) and optimizations (Gradient descent and Adam). It includes a deep neural network with nine hidden layers. The layers contain Conv2D with 3×3 kernel and max pooling

with 3×3 kernel with Relu and SoftMax activation. The approach achieved 99.3% accuracy on the Hijja dataset.

Table 6 summarizes the mentioned above Arabic OCR approaches and Table 7 summarizes the Arabic datasets. The citations column is obtained in May 2022 from Google Scholar. In the last two rows of Table 6, ABBYY FineReader Engine and Tesseract open-source library were used in MATLAB and evaluated on a sample of 1,000 images from the APTI dataset.

Table 6: Summary of some Arabic character recognition technique (2012–2022)

Ref.	Year	Method description	Online/ Offline	Dataset	Handcrafted/ Learned features	Accuracy	Citations	Printed/ Handwritten
[15]	2012	The approach is based on SIFT descriptor for features extraction and description, and HMMs are used for the classification stage.	Offline	PATS and APTI	Handcrafted	98.87%–100%	17	Printed
[11]	2013	The contour features are extracted and then described using the Shape Context descriptor. Features are classified using K-Nearest Neighbor.	Offline	Private 20,000 ligatures	Handcrafted	86%	105	Printed
[13]	2014	Recognizing isolated handwritten Arabic characters. Preprocessing operations are performed. Extracting statistical and topological features. Classification using Neural Networks.	Offline	CENPRMI dataset (includes 11620 characters)	Handcrafted	88%	22	Handwritten
[122]	2016	A set of preprocessing operations are performed to enhance image quality. A set of statistical and topological features was extracted. HMMs and KNN are used for classification.	Offline	IFN/ENIT database	Handcrafted	86.0% (Average of A, B and C splits)	6	Handwritten
[123]	2017	Recognizing isolated handwritten Arabic characters. The approach is based on a CNNs architecture.	Offline	Private 16800 images	deep learned	94.9%	154	Handwritten
[124]	2017	Developed by Benjamin Kiessling et al. from Leipzig University. The open-source Kraken library is trained to recognize historical Arabic books. It is designed based on Convolutional Neural Networks.	Offline	Gold Standard-Book: Ibn al-Faqīh's Kitāb al-Buldān	deep learned	97.56%	19	Printed
[125]	2017	CNN architecture based on VGGNet. The VGGNet filers were reduced by a factor of eight to reduce the time complexity.	Offline	HACDB, ADBase	Handcrafted	97.32%, 99.66%	34	Handwritten
[67]	2018	A holistic Arabic word recognition method based on Discrete Cosine Transform (DCT) and local block features. The features are clustered, and Euclidean distance is used to find the best possible matched word.	Offline	Private around 356,000 words	Handcrafted	84.8%	21	Printed
[110]	2018	Post-processing systems called rule-based and hybrid systems are used to improve OCR accuracy.	Offline	Private 9000 images	Handcrafted	85.5%	16	Printed

(Continued)

Table 6 (continued)

Ref.	Year	Method description	Online/ Offline	Dataset	Handcrafted/ Learned features	Accuracy	Citations	Printed/ Handwritten
[78]	2018	Used multiple Convolutional Neural Networks for character segmentation and recognition.	Offline	APTI	deep learned	94.38%	17	Printed
[104]	2020	The second-order statistics are used for feature extraction, and the Genetic Algorithm is used for feature selection. K-Nearest Neighbor classifier is used for classification.		PATS-A01 (650 images) and APTI (550 images)	Handcrafted	98.69% and 95.37%	4	Printed
[126]	2020	Hybrid DCNN to recognize printed Arabic Text. the architecture includes five layers of CNNS followed by Bi-Directional Short LSTM.	Offline	APTI	deep learned	76.3%	3	Printed
[127]	2020	CNNs architecture is used to extract the features and SVM is used for classification.	Offline	[123]	deep learned	95.07%	11	Handwritten
[116]	2021	Deep CNNs architecture to recognize isolated handwritten Arabic characters.	Offline	AHCD, Hijja	deep learned	97%, 88%	81	Handwritten
[21]	2021	Proposed 14 different deep CNNs architectures for handwritten Arabic OCR. Additionally, combinations of VGG16, VGG19, AND MobileNetV2 are evaluated.	Offline	HMBD	deep learned	92.88%	8	Handwritten
[128]	2021	Several sacked CNNs are used to design Arabic OCR to recognize numbers, characters, and words. Achieved superior results compared with VGG-19 net on MNIST.	Offline	HACDB	deep learned	99.91%	23	Handwritten
[129]	2022	Four deep CNNs architectures were proposed. The architectures use different layers, activation functions, and optimization techniques.	Offline	Hijja dataset	deep learned	99.3%	1	Handwritten
[130]	2022	ABBYY (commercial).	Online/ Offline	APTI dataset	-	74.8%	-	Printed
[131]	2022	Tesseract (open source).	Offline	APTI dataset	-	71%	-	Printed

Table 7: Summary of Arabic databases

Ref.	Year	Database	Printed/ Handwritten	Description	Letter/word/ line/paragraph/ page level images	Citations	Availability
[112]	2009	APTI	Printed	45,313,600 word-images (250 million characters)	Word-level	153	Public
[113]	2010	MMAC	Printed	847,779 paragraph-images (552 paragraphs)	Paragraph-level	41	Public

(Continued)

Table 7 (continued)

Ref.	Year	Database	Printed/ Handwritten	Description	Letter/word/ line/paragraph/ page level images	Citations	Availability
[118]	2012	KHATT	Handwritten	2,000 paragraph-images (9,327 lines)	Paragraph-level	87	Public
[119]	2012	IFN/ENIT	Handwritten	2,265 word-images (26,459 words 212,211 characters)	Word-level	651	Public
[69]	2013	APTID/MF	printed	1,845 line-images (27,402 characters)	Line-level	29	Upon request
[114]	2013	HACDB	Handwritten	6,600 characters	Letter-level	57	Upon request
[117]	2014	KAFD	Printed	15,068page-images (2,576, 024 lines)	Page-level	32	Public
[132]	2015	ALIF	Printed	6,532 line-images (52,410 paws 18,041 words 89,819 characters)	Line-level	42	Upon request
[115]	2015	AcTiV	Printed	80 videos (850,000 frames)	Paragraph-level	52	Upon request
[133]	2016	SmartATID	Printed/ Handwritten	9,088 page-images	Page-level	14	Upon request
[134]	2016	BCE-Arabic-v1	Printed	1,833 page-images (from 180 books)	Page-level	14	Public
[120]	2018	Yarmouk	Printed	8,994 page-images 436,921 words	Page-level	6	Public
[135]	2019	EASTR-42 K	Printed	8,915 line-images (2,593 words 12,000 characters)	Line-level	20	Upon request
[136]	2021	HMBD	Handwritten	54,115 characters	Letter-level	19	Public
[116]	2001	Hijja	Handwritten	47,434 characters	Letter-level	81	Public

7 Commercial and Open-Source Arabic OCR Software

This section summarizes the leading commercial and open-source software for Arabic OCR.

Table 8 shows the recognition rates for some well-known OCR software that support the Arabic language. Four software, Tesseract, Abbyy FineReader, Sakhr, and Readiris, are used in the comparison. The authors used different datasets to evaluate the four software. It is clear from the table that the recognition rates for the software vary from one dataset to another, and no one software outperformed all software on all tested datasets. The average accuracy rate for these softer ranges from 70% to 80%.

Tesseract [131] is an open-source OCR. It supported more than 100 languages and was developed by Hewlett-Packard (HP) as a PhD project from 1985 to 1994. In 2006, it was sponsored and redeveloped by Google till 2018. The latest available version of Tesseract is version 5. It supports different operating systems Windows, Linux, and Mac. Tesseract converted the input image into a binary image. Fuzzy space is used to divide the text into words. The connected component is used to identify the layout of the characters. Finally, a classifier is trained to find the labels of the character.

Abbyy FineReader [130] is commercial software that supports more than 200 languages. It supports different operating systems Windows, Linux, and Mac. It accepts documents in different formats such as PDF, TIFF, and JPEG, printed and handwritten. The image is preprocessed to enhance its quality. The preprocessing operations include banalization, rotation, and deskewing. The document analysis stage is used to identify the image structure and the formats of its elements, such as the location of

the header, footer, tables, diagrams, and text fields. Since its commercial software, the details of the recognition approach are unknown. Abbyy OCR has an online trial version with 1000 pages for each registered user.

Sakhr [137] is commercial software it supports the Arabic language or the languages that use Arabic characters such as Farsi, Urdu Pashto, and Jawi. It supports both printed and printed scripts and provides online and offline recognition. It runs on Windows operating systems, and there is no trial version. Sakhr claimed to be the best available OCR for the Arabic system, according to US government evaluators. Sakhr claimed 99.8% accuracy for the documents with high-quality images.

Readiris [138] is commercial software. It supports around 130 languages (Including Arabic, Russian, and East Asia languages). It accepts images, PDF files, and document folders and converts them into editable text. It supports Windows and Mac OS.

Table 8: The recognition rates for some Arabic OCR software (2013–2022)

Ref.	Year	Metric	Arabic dataset	Tesseract	Abbyy FineReader	Sakhr	Readiris
[30]	2013	WRR	Private	-	-	57.8%	84.0%
[139]	2014	WRR	Private (newspapers, journals, books)	-	-	92.96%	71.83%
[140]	2016	WER	Private (historical manuscript)	99.4%	100.0%	99%	
[141]	2016	WRR	Private (archives of Alahly journal)	-	80.0%	88.0%	70.0%
[71]	2017	CER	APTID/MF-250 images	93.0%	53.0%	76.0%	80.0%
[28]	2017	CRR	KAFD (240 images)	48.61	75.19	51.56%	
[124]	2017	WRR	Gold standard (Book: Ibn al-Faqīh's Kitāb al-Buldān)	-	65%-75%	-	65%-75%
[142]	2019	WRR	Private (195 words)	35.9%	95.9%	96.4%	65.1%
[143]	2021	CRR	Private (archives of Al-Abhath journal-10 pages)	-	85.52%	-	-
This paper	2022	WRR	APTI (1,000 images)	71.0%	74.8%	-	-

8 Performance Evaluation

Many evaluation metrics are used to evaluate the performance of the Arabic OCR system. Character Error Rate (CER) and Word Error Rate (WER) are the most used metrics. According to Eq. (3), CER is calculated where i_C, d_C and s_C denote the minimal number of character insertion, deletion, and substitution operations (Edit distance), respectively, to transform the OCR output to the ground truth script (see Fig. 30). n_C denotes the total number of characters in the text. WER is calculated according to Eq. (4), where i_w, d_w an s_w denote the minimal number of word insertion, deletion, and substitution operations, respectively, to transform the OCR output to the ground truth script. n_w indicates the total number of words in the text. The Character Recognition Rate (CRR) and Word Recognition Rate (WRR) are computed according to Eqs. (5) and (6), respectively [28].

Some dataset contains isolated characters for OCR evaluation. Therefore Eq. (7) is used [123], where c_C denotes the total number of correctly recognized characters, and n_C denotes the total number of tested characters. Eq. (8) is used to find the word accuracy rate, where w_C denotes the total number of correctly recognized words, and n_w denotes the total number of tested words. Text Recognition Rate (TRR) considers the whole image text as one unit. The image could contain one word, a line of words, or a paragraph. TRR measures the percentage of correctly recognized text images according to Eq. (9), where t_C is number of images that are correctly recognized, n_t indicates the total number of text images [71].

$$CER = \frac{i_C + d_C + s_C}{n_C} * 100\% \tag{3}$$

$$WER = \frac{i_w + d_w + s_w}{n_w} * 100\% \tag{4}$$

$$TER = \frac{i_t + d_t + s_t}{n_t} * 100\%$$

$$CRR = 100 - CER \tag{5}$$

$$WRR = 100 - WER \tag{6}$$

$$CA = \frac{c_C}{n_C} * 100\% \tag{7}$$

$$WA = \frac{w_C}{n_w} * 100\% \tag{8}$$

$$TRR = \frac{t_C}{n_t} \tag{9}$$

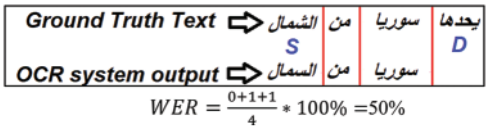


Figure 30: Example of computing WER

9 Conclusion and Future Work

Developing an accurate and fast Arabic Optical Character Recognition system will be helpful for many people in the Arab and Muslim regions. However, the accuracy of the existing printed-Arabic commercial OCR software does not exceed 75%, according to some studies, when tested on a page-level image. Additionally, most current approaches work offline and do not recognize the Arabic script in real-time speed. Recognizing Arabic text is a challenging task due to many reasons. Therefore, Arabic character recognition is still an open research area, and there is a range for enhancing and improving the existing systems. Many approaches are evaluated on a private dataset, a word, or a paragraph level, making it difficult to know their performance in real-world Arabic scripts. Convolutional Neural Networks have been used wildly during the last decade in Arabic optical recognition and showed significant results compared with handcrafted approaches. A comprehensive review of the latest advances during the previous decade in Arabic Optical character recognition is introduced in this paper. This paper reviews the following: the characteristics of the Arabic language;

different types of OCR systems; the main stages of the Arabic OCR system, the techniques used in each step, and the researchers' contributions; comparisons between the existing Arabic OCR methods, commercial and open-source software; the current datasets for Arabic OCR and their characteristics; evaluation metrics for the OCR system. Future works include the followings: customizing and training popular CNNs models such as DenseNet, AlexNet, VGGNet, MobileNet, SqueezeNet, ResNet, and GoogLeNet to develop a new Arabic OCR system; using a large dataset for training the CNNs model to achieve a higher recognition rate and using RNNs with CNNs for handwritten and printed text; evaluating the proposed approach on a well-known benchmark to measure the actual performance of Arabic OCR systems.

Funding Statement: The author received no specific funding for this study.

Conflicts of Interest: The author declares that they have no conflicts of interest to report regarding the present study.

References

1. Adriano, J. E. M., Calma, K. A. S., Lopez, N. T., Parado, J. A., Rabago, L. W. et al. (2019). Digital conversion model for hand-filled forms using optical character recognition (OCR). *IOP Conference Series: Materials Science and Engineering, 482(1),* 012049. DOI 10.1088/1757-899X/482/1/012049.
2. Alghyaline, S. (2022). Real-time Jordanian license plate recognition using deep learning. *Journal of King Saud University-Computer and Information Sciences, 34(6),* 2601–2609. DOI 10.1016/j.jksuci.2020.09.018.
3. Radha, R., Aparna, R. (2013). Review of OCR techniques used in automatic mail sorting of postal envelopes. *An International Journal of Signal & Image Processing, 4(5),* 45–60. DOI 10.5121/sipij.2013.4504.
4. Agrawal, P., Chaudhary, D., Madaan, V., Zabrovskiy, A., Prodan, R. et al. (2021). Automated bank cheque verification using image processing and deep learning methods. *Multimedia Tools and Applications, 80(4),* 5319–5350. DOI 10.1007/s11042-020-09818-1.
5. Bassam, R., Samann, F. (2020). Smart parking system based on improved OCR model. *IOP Conference Series: Materials Science and Engineering*, 012007.
6. Larsson, A., Segerås, T. (2016). *Automated invoice handling with machine learning and OCR. KTH Royal Institute of Technology*.
7. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788. Las Vegas, NV, USA.
8. Ren, S., He, K., Girshick, R., Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6),* 1137–1149. DOI 10.1109/TPAMI.2016.2577031.
9. Ahmed, S., Naz, S., Razzak, M., Yousaf, R. (2017). Deep learning based isolated Arabic scene character recognition. *1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, pp. 46–51. Nancy, France.
10. Butt, H., Raza, M. R., Ramzan, M. J., Ali, M. J., Haris, M. (2021). Attention-based CNN-RNN Arabic text recognition from natural scene images. *Forecasting, 3(3),* 520–540. DOI 10.3390/forecast3030033.
11. Sabbour, N., Shafait, F. (2013). A Segmentation-free approach to Arabic and Urdu OCR. *Document Recognition and Retrieval XX, 8658,* 215–226. DOI 10.1117/12.2003731.
12. Yamina, O. J., El Mamoun, M., Kaddour, S. (2017). Printed Arabic optical character recognition using support vector machine. *Proceedings of the International Conference on Mathematics and Information Technology*, pp. 134–140. Adrar, Algeria.

13. Talaat, A., Refaat, M. M., Sallam, A. A. (2014). A proposed OCR algorithm for the recognition of handwritten Arabic characters. *Journal of Pattern Recognition and Intelligent Systems, 2(1),* 90–114.
14. Rashad, M., Amin, K., Hadhoud, M., Elkilani, W. (2012). Arabic character recognition using statistical and geometric moment features. *Proceedings of the 2012 Japan-Egypt Conference on Electronics, Communications and Computers*, pp. 68–72. Alexandria, Egypt.
15. Rosenberg, A., Nachum, D. (2012). *Using SIFT descriptors for OCR of Printed Arabic*. Israel: Tel Aviv University.
16. Bay, H., Tuytelaars, T., Gool, L. V. (2006). SURF: Speeded up robust features. *Proceedings of the 9th European Conference on Computer Vision*, pp. 404–417.
17. Jebril, N. A., Al-Zoubi, H. R., Abu Al-Haija, Q. (2018). Recognition of handwritten Arabic characters using histograms of oriented gradient (HOG). *Pattern Recognition and Image Analysis, 28(2),* 321–345. DOI 10.1134/S1054661818020141.
18. Manjunath Aradhya, V. N., Hemantha Kumar, G., Noushath, S. (2008). Multilingual OCR system for South Indian scripts and English documents: An approach based on Fourier transform and principal component analysis. *Engineering Applications of Artificial Intelligence, 21(4),* 658–668. DOI 10.1016/j.engappai.2007.05.009.
19. Mansouri, S., Charhad, M., Zrigui, M. (2017). Arabic text detection in news video based on line segment detector. *Research in Computing Science, 132(1),* 97–106. DOI 10.13053/rcs-132-1-9.
20. Qaroush, A., Jaber, B., Mohammad, K., Washaha, M., Maali, E. et al. (2022). An efficient, font independent word and character segmentation algorithm for printed Arabic text. *Journal of King Saud University-Computer and Information Sciences, 34(1),* 1330–1344. DOI 10.1016/j.jksuci.2019.08.013.
21. Balaha, H. M., Ali, H. A., Youssef, E. K., Elsayed, A. E., Samak, R. A. et al. (2021). Recognizing Arabic handwritten characters using deep learning and genetic algorithms. *Multimedia Tools and Applications, 80(21–23),* 32473–32509. DOI 10.1007/s11042-021-11185-4.
22. Tian, Z., Huang, W., He, T., He, P., Qiao, Y. (2016). Detecting text in natural image with connectionist text proposal network. *European Conference on Computer Vision*, pp. 56–72. Amsterdam, The Netherlands.
23. Ye, J., Chen, Z., Liu, J., Du, B. (2020). Textfusenet: Scene text detection with richer fused features. *IJCAI International Joint Conference on Artificial Intelligence*, vol. 20, pp. 516–522. DOI 10.24963/ijcai.2020.
24. Mackay, R. S., Percival, I. C. (2021). R-YOLO: A real-time text detector for natural scenes with. *Sensors, 21(3),* 888. DOI 10.3390/s21030888.
25. Boudelaa, S., Pulvermüller, F., Hauk, O., Shtyrov, Y., Marslen-Wilson, W. (2010). Arabic morphology in the neural language system. *Journal of Cognitive Neuroscience, 22(5),* 998–1010. DOI 10.1162/jocn.2009.21273.
26. Farghaly, A., Shaalan, K. (2009). Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing, 8(4),* 1–22. DOI 10.1145/1644879.1644881.
27. Lu, Z. A., Bazzi, I., Kornai, A., Makhoul, J., Natarajan, P. S. et al. (1999). Robust language-independent OCR system. *27th AIPR Workshop: Advances in Computer-Assisted Recognition*, pp. 96–104. Washington, USA.
28. Alghamdi, M., Teahan, W. (2017). Experimental evaluation of Arabic OCR systems. *PSU Research Review, 1(3),* 229–241. DOI 10.1108/PRR-05-2017-0026.
29. Hegghammer, T. (2022). OCR with tesseract, Amazon textract, and google document AI: A benchmarking experiment. *Journal of Computational Social Science, 5(1),* 861–882. DOI 10.1007/s42001-021-00149-1.
30. Alginahi, Y. M. (2013). A survey on Arabic character segmentation. *International Journal on Document Analysis and Recognition, 16(2),* 105–126. DOI 10.1007/s10032-012-0188-6.
31. Naz, S., Umar, A. I., Shirazi, S. H., Ahmed, S. B., Razzak, M. I. et al. (2016). Segmentation techniques for recognition of Arabic-like scripts: A comprehensive survey. *Education and Information Technologies, 21(5),* 1225–1241. DOI 10.1007/s10639-015-9377-5.

32. Malik, S., Sajid, A., Ahmad, A., Almogren, A., Hayat, B. et al. (2020). An efficient skewed line segmentation technique for cursive script OCR. *Scientific Programming, 2020,* 1–12. DOI 10.1155/2020/8866041.
33. Shamim, S. M., Miah, M. B. A., Sarker, A., Rana, M., Jobair, A. A. (2018). Handwritten digit recognition using machine learning algorithms. *Indonesian Journal of Science and Technology, 3(1),* 29–39. DOI 10.17509/ijost.v3i1.10795.
34. Kadi, M. (2019). Isolated Arabic characters recognition using a robust method against noise and scaling based on the «Hough transform». *International Journal of Information Science and Technology, 3(4),* 34–43.
35. Dalbir, S., Singh, S. K. (2015). Review of online and offline character recognition. *International Journal of Engineering and Computer Science, 4(5),* 11729–11732.
36. Zayene, O., Seuret, M., Touj, S. M., Hennebert, J., Ingold, R. et al. (2016). Text detection in Arabic news video based on SWT operator and convolutional auto-encoders. *12th IAPR International Workshop on Document Analysis Systems*, pp. 13–18. Santorini, Greece.
37. The World Bank (2022). Arab World Population.
38. Tafti, A. P., Baghaie, A., Assefi, M., Arabnia, H. R., Yu, Z. et al. (2016). OCR as a service: An experimental evaluation of Google Docs OCR, Tesseract, ABBYY Finereader, and Transym. In: *Lecture notes in computer science*, vol. 10072, pp. 735–746. DOI 10.1007/978-3-319-50835-1.
39. Shen, M., Hansheng, L. (2015). Improving OCR performance with background image elimination. *12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 1566–1570. Zhangjiajie, China.
40. Ahmad, R., Afzal, M. Z., Rashid, S. F., Liwicki, M., Breuel, T. et al. (2016). A novel skew detection and correction approach for scanned documents. *IAPR International Workshop on Document Analysis Systems*, pp. 1–3. Santorini, Greece.
41. Alghamdi, M. A., Teahan, W. J. (2017). A new thinning algorithm for Arabic script. *International Journal of Computer Science and Information Security, 15(1),* 204–211.
42. Michalak, H., Okarma, K. (2019). Fast binarization of unevenly illuminated document images based on background estimation for optical character recognition purposes. *Journal of Universal Computer Science, 25(6),* 627–646.
43. Brisinello, M., Grbic, R., Pul, M., Andelic, T. (2017). Improving optical character recognition performance for low quality images. *59th International Symposium ELMAR*, pp. 167–171. Zadar, Croatia.
44. Nosseir, A., Adel, O. (2018). Automatic extraction of Arabic number from Egyptian ID cards. *ACM International Conference Proceeding Series*, pp. 56–61. Cairo, Egypt.
45. Qaroush, A., Awad, A., Modallal, M., Ziq, M. (2020). Segmentation-based, omnifont printed Arabic character recognition without font identification. *Journal of King Saud University-Computer and Information Sciences, 34,* 3025–3039.
46. Hamid, A., Haraty, R. (2001). A Neuro-heuristic approach for segmenting handwritten Arabic text. *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications*, pp. 110–113. Beirut, Lebanon.
47. Elarian, Y., Ahmad, I., Awaida, S., Al-khatib, W., Zidouri, A. (2015). Arabic ligatures: Analysis and application in text recognition. *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 896–900. Tunis, Tunisia.
48. Essa, N., El-Daydamony, E., Mohamed, A. A. (2018). Enhanced technique for Arabic handwriting recognition using deep belief network and a morphological algorithm for solving ligature segmentation. *ETRI Journal, 40(6),* 774–787. DOI 10.4218/etrij.2017-0248.
49. Lawgali, A., Bouridane, A., Angelova, M., Zabih, G. (2011). Automatic segmentation for Arabic characters in handwriting documents. *18th IEEE International Conference on Image Processing*, pp. 3529–3532. Brussels, Belgium.

50. Kiaei, P., Javaheripi, M., Mohammadzade, H. (2019). High accuracy farsi language character segmentation and recognition. *27th Iranian Conference on Electrical Engineering*, pp. 1692–1698. Yazd, Iran.
51. Cesare, S., Xiang, Y. (2012). Feature extraction. In: *Springer briefs in computer science*, pp. 57–61. London: Springer.
52. Nabi, G., Shaikh, N. A., Rajper, R. A., Shaikh, R. A. (2021). Thinning for segmentation-based and segmentation-freefor Arabic scriptadopting languages. *Sindh University Research Journal, 53(3),* 271–274.
53. Osman, Y. (2013). Segmentation algorithm for Arabic handwritten text based on contour analysis. *2013 International Conference on Computer, Electrical and Electronics Engineering: Research Makes a Difference*, pp. 447–452. Khartoum, Sudan.
54. Elkhayati, M., Elkettani, Y., Mourchid, M. (2022). Segmentation of handwritten Arabic graphemes using a directed Convolutional Neural Network and mathematical morphology operations. *Pattern Recognition, 122,* 108288. DOI 10.1016/j.patcog.2021.108288.
55. Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pp. 1–8. Kerkyra, Greece.
56. Chergui, L., Kef, M. (2015). SIFT descriptors for Arabic handwriting recognition. *International Journal of Computational Vision and Robotics, 5(4),* 441–461. DOI 10.1504/IJCVR.2015.072193.
57. Zahedi, M., Eslami, S. (2011). Farsi/Arabic optical font recognition using SIFT features. *Procedia Computer Science, 3,* 1055–1059. DOI 10.1016/j.procs.2010.12.173.
58. Stolyarenko, A., Dershowitz, N. (2011). OCR for Arabic using SIFT descriptors with online failure prediction. *Imaging, 3(1),* 1–10.
59. Hassan, A. K. A., Mahdi, B. S., Mohammed, A. A. (2019). Arabic handwriting word recognition based on scale invariant feature transform and support vector machine. *Iraqi Journal of Science, 60(2),* 381–387.
60. Dalal, N., Triggs, B. (2010). Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp. 886–893. San Diego, CA, USA.
61. Khaissidi, G., Elfakir, Y., Mrabti, M., Yacoubi, M. E., Chenouni, D. et al. (2016). Segmentation-free word spotting for handwritten Arabic documents. *International Journal of Interactive Multimedia and Artificial Intelligence, 4(1)*. DOI 10.9781/ijimai.2016.411.
62. Bay, H., Ess, A., Tuytelaars, T., van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding, 110(3),* 346–359. DOI 10.1016/j.cviu.2007.09.014.
63. Alsimry, A., Ali, K. H., Abood, E. W. (2021). A new approach for finding duplicated words in scanned Arabic documents based on OCR and SURF. *Journal of Basrah Researches (Sciences), 47(1),* 201–215.
64. Bagasi, B., Elrefaei, L. A. (2018). Arabic manuscript content based image retrieval: A comparison between surf and brisk local features. *International Journal of Computing and Digital Systems, 7(6),* 355–364. DOI 10.12785/ijcds/070604.
65. Leutenegger, S., Chli, M., Siegwart, R. Y. (2011). BRISK: Binary robust invariant scalable keypoints. *2011 International Conference on Computer Vision*, pp. 2548–2555. Barcelona, Spain.
66. Torki, M., Hussein, M. E., Elsallamy, A., Fayyaz, M., Yaser, S. (2014). Window-based descriptors for Arabic handwritten alphabet recognition: A comparative study on a novel dataset. DOI 10.48550/arXiv.1411.3519.
67. Nashwan, F. M. A., Rashwan, M. A. A., Al-Barhamtoshy, H. M., Abdou, S. M., Moussa, A. M. (2018). A holistic technique for an Arabic OCR system. *Journal of Imaging, 4(1),* 1–11.
68. Naz, S., Umar, A. I., Ahmed, S. B., Ahmad, R. (2018). Statistical features extraction for character recognition using Recurrent Neural Network. *Pakistan Journal of Statistics, 34(1),* 47–53.
69. Jaiem, F. K., Kanoun, S., Khemakhem, M., El Abed, H., Kardoun, J. (2013). Database for Arabic printed text recognition research. In: *Lecture notes in computer science*, vol. 8156, pp. 251–259. DOI 10.1007/978-3-642-41181-6.

70. Nemouchi, S., Meslati, L. S., Farah, N. (2012). Classifiers combination for Arabic words recognition. *International Conference on Image and Signal Processing*, pp. 562–570. Berlin, Heidelberg, Springer.
71. Alkhateeb, F., Abu Doush, I., Albsoul, A. (2017). Arabic optical character recognition software: A review. *Pattern Recognition and Image Analysis, 27(4),* 763–776. DOI 10.1134/S105466181704006X.
72. Huang, J., Ul Haq, I., Dai, C., Khan, S., Nazir, S. et al. (2021). Isolated handwritten pashto character recognition using a K-NN classification tool based on zoning and hog feature extraction techniques. *Complexity, 2021,* 1–8.
73. Boufenar, C., Batouche, M., Schoenauer, M. (2018). An artificial immune system for offline isolated handwritten Arabic character recognition. *Evolving Systems, 9(1),* 25–41. DOI 10.1007/s12530-016-9169-1.
74. Redmon, J., Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv:1804.02767.
75. Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448. Santiago, Chile.
76. Alghyaline, S. (2019). A real-time street actions detection. *International Journal of Advanced Computer Science and Applications, 10(2),* 322–329. DOI 10.14569/IJACSA.2019.0100243.
77. Alghyaline, S., Hsieh, J. W., Chuang, C. H. (2017). Video action classification using symmelets and deep learning. *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 414–419. Banff, Canada.
78. Radwan, M. A., Khalil, M. I., Abbas, H. M. (2018). Neural networks pipeline for offline machine printed Arabic OCR. *Neural Processing Letters, 48(2),* 769–787. DOI 10.1007/s11063-017-9727-y.
79. Saidane, Z., Garcia, C. (2007). Automatic scene text recognition using a Convolutional Neural Network. *Proceedings of the 2nd International Workshop on Camera-Based Document Analysis and Recognition*, pp. 100–106. Curitiba, Brazil.
80. Elleuch, M., Maalej, R., Kherallah, M. (2016). A new design based-SVM of the CNN classifier architecture with dropout for offline Arabic handwritten recognition. *European Workshop on Visual Information Processing (EUVIP)*, pp. 1712–1723. Elsevier Masson SAS.
81. Mustafa, M. E., Elbashir, M. K. (2020). A deep learning approach for handwritten Arabic names recognition. *International Journal of Advanced Computer Science and Applications, 11(1),* 678–682. DOI 10.14569/issn.2156-5570.
82. Naz, S., Khan, N. H., Zahoor, S., Razzak, M. I. (2020). Deep OCR for Arabic script-based language like pastho. *Expert Systems, 37(5),* 1–11. DOI 10.1111/exsy.12565.
83. Sokar, G., Hemayed, E. E., Rehan, M. (2019). A generic OCR using deep Siamese Convolution Neural Networks. *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference*, pp. 1238–1244. Columbia, Canada.
84. Ashiquzzaman, A., Tushar, A. K. (2017). Handwritten Arabic numeral recognition using deep learning neural networks. *2017 IEEE International Conference on Imaging, Vision and Pattern Recognition*, pp. 1–4. Dhaka, Bangladesh.
85. Abdi, M. N., Khemakhem, M. (2012). Arabic writer identification and verification using template matching analysis of texture. *Proceedings-2012 IEEE 12th International Conference on Computer and Information Technology*, pp. 592–597. Sichuan, China.
86. Journal, I., Network, C., Security, I., El, M., Charaf, H. et al. (2016). Template matching for recognition of handwritten Arabic characters using structural characteristics and freeman code. *International Journal of Computer Science and Information Security, 14(12),* 31–40.
87. Maghrabi, S. M. (2017). An offline Arabic handwritten character recognition system using template matching. *International Journal of Computer Technology & Applications, 8(5),* 602–608.
88. Farhat, A., Al-Zawqari, A., Al-Qahtani, A., Hommos, O., Bensaali, F. et al. (2016). OCR based feature extraction and template matching algorithms for qatari number plate. *2016 International Conference on Industrial Informatics and Computer Systems*, Futuroscope-Poitiers, France.

89. Hairuman, I. F. B., Foong, O. M. (2011). OCR signage recognition with skew & slant correction for visually impaired people. *Proceedings of the 2011 11th International Conference on Hybrid Intelligent Systems*, pp. 306–310. Melacca, Malaysia.
90. Almustafa, K., Zantout, R. N., Obeid, H. R. (2011). Peak position recognizing characters in Saudi license plates. *2011 IEEE GCC Conference and Exhibition*, 11962336. DOI 10.1109/IEEEGCC.2011.5752479.
91. Abdalkafor, A. S., Alheeti, K. M. A., Al-Jobouri, L. (2021). A feature extraction method for Arabic offline handwritten recognition system using naïve Bayes classifier. *2021 International Conference on Computing and Communications Applications and Technologies*, pp. 82–87. London, UK.
92. Saeed, U., Tahir, M., AlGhamdi, A. S., Alkatheiri, M. S. (2020). Automatic recognition of handwritten Arabic using maximally stable extremal region features. *Optical Engineering, 59(5),* 1–19. DOI 10.1117/1.OE.59.5.051405.
93. Jayech, K., Mahjoub, M. A., Ben Amara, N. (2016). Arabic handwritten word recognition based on dynamic Bayesian network. *International Arab Journal of Information Technology, 13(3),* 276–283.
94. Chang, C. C., Lin, C. J. (2011). LIBSVM. *ACM Transactions on Intelligent Systems and Technology, 2(3),* 1–27. DOI 10.1145/1961189.1961199.
95. Elleuch, M., Lahiani, H., Kherallah, M. (2016). Recognizing Arabic handwritten script using support vector machine classifier. *International Conference on Intelligent Systems Design and Applications*, pp. 551–556. Porto, Portugal.
96. Elzobi, M., Al-Hamadi, A., Saeed, A., Dings, L. (2012). Arabic handwriting recognition using gabor wavelet transform and SVM. *International Conference on Signal Processing Proceedings*, pp. 2154–2158. Gold Coast, QLD, Australia.
97. Alsaeedi, A., Al Mutawa, H., Natheer, S., Al Subhi, W., Snoussi, S. et al. (2018). Arabic words recognition using CNN and TNN on a smartphone. *2nd IEEE International Workshop on Arabic and Derived Script Analysis and Recognition*, pp. 57–61. London, UK.
98. Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, pp. 257–286. New York, USA.
99. Gayathri, P., Ayyappan, S. (2014). Off-line handwritten character recognition using hidden markov model. *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics*, pp. 518–523. Delhi, India.
100. Prasad, R., Saleem, S., Kamali, M., Meermeier, R., Natarajan, P. (2008). Improvements in hidden markov model based Arabic OCR. *Proceedings-International Conference on Pattern Recognition*, pp. 1–4. Tampa, FL, USA.
101. Ahmad, I., Fink, G. A. (2015). Multi-stage HMM based Arabic text recognition with rescoring. *13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 751–755. Tunis, Tunisia.
102. Krayem, A., Sherkat, N., Evett, L., Osman, T. (2013). Holistic Arabic whole word recognition using HMM and block-based DCT. *12th International Conference on Document Analysis and Recognition*, pp. 1120–1124. Washington DC, USA.
103. Pechwitz, M., El Abed, H., Märgner, V. (2012). Handwritten Arabic word recognition using the IFN/ENIT-database. In: *Guide to OCR for Arabic scripts*, pp. 169–213. London: Springer.
104. Darwish, S. M., Elzoghaly, K. O. (2020). An enhanced offline printed Arabic OCR model based on Bio-inspired fuzzy classifier. *IEEE Access, 8,* 117770–117781. DOI 10.1109/Access.6287639.
105. Kessab, B. E. L., Daoui, C., Bouikhalene, B., Salouan, R. (2015). Isolated handwritten Arabic numerals recognition using the K-Nearest Neighbor and the hidden markov model classifiers. *Facta Universitatis. Series Mathematics and Informatics, 30(5),* 731–740.
106. Wu, X., Gao, Y., Jiao, D. (2019). Multi-label classification based on random forest algorithm for non-intrusive load monitoring system. *Processes, 7(6),* 1–14. DOI 10.3390/pr7060337.

107. Hassanien, A. E., Tolba, M. F., Azar, A. T. (2014). Isolated printed Arabic character recognition using KNN and random forest tree classifiers. In: Hassanien, A. E., Tolba, M. F., Azar, A. T. (Eds.), *Communications in computer and information science*, pp. 10–17. Cham: Springer International Publishing.
108. Sahlol, A., AbdElfattah, M., Suen, C. Y., Hassanien, A. (2016). Particle swarm optimization with random forests for handwritten Arabic recognition system. *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics (AISI 2016)*, pp. 437–446. Cairo, Egypt. DOI 10.1007/978-3-319-48308-5.
109. Bassil, Y., Alwani, M. (2012). OCR Post-processing error correction algorithm using google online spelling suggestion. arXiv:1204.0191.
110. Doush, I. A., Alkhateeb, F., Gharaibeh, A. H. (2018). A novel Arabic OCR post-processing using rule-based and word context techniques. *International Journal on Document Analysis and Recognition, 21(1–2),* 77–89. DOI 10.1007/s10032-018-0297-y.
111. Doush, I. A., Al-Trad, A. M. (2016). Improving post-processing optical character recognition documents with Arabic language using spelling error detection and correction. *International Journal of Reasoning-Based Intelligent Systems, 8(3–4),* 91–103. DOI 10.1504/IJRIS.2016.082957.
112. Slimane, F., Ingold, R., Kanoun, S., Alimi, A. M., Hennebert, J. (2009). A new Arabic printed text image database and evaluation protocols. *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 946–950. Barcelona, Spain.
113. AbdelRaouf, A., Higgins, C. A., Pridmore, T., Khalil, M. (2010). Building a multi-modal Arabic corpus (MMAC). *International Journal on Document Analysis and Recognition, 13(4),* 285–302. DOI 10.1007/s10032-010-0128-2.
114. Lawgali, A., Angelova, M., Bouridane, A. (2013). HACDB: Handwritten Arabic characters database for automatic character recognition. *IEEE European Workshop on Visual Information Processing*, pp. 255–259. Paris, France.
115. Zayene, O., Hennebert, J., Masmoudi Touj, S., Ingold, R., Essoukri Ben Amara, N. (2015). A dataset for Arabic text detection, tracking and recognition in news videos-acTiV. *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 996–1000. Tunis, Tunisia.
116. Altwaijry, N., Al-Turaiki, I. (2021). Arabic handwriting recognition system using Convolutional Neural Network. *Neural Computing and Applications, 33(7),* 2249–2261. DOI 10.1007/s00521-020-05070-8.
117. Luqman, H., Mahmoud, S. A., Awaida, S. (2014). KAFD Arabic font database. *Pattern Recognition, 47(6),* 2231–2240. DOI 10.1016/j.patcog.2013.12.012.
118. Mahmoud, S. A., Ahmad, I., Alshayeb, M., Al-Khatib, W. G., Parvez, M. T. et al. (2012). KHATT: Arabic offline handwritten text database. *International Workshop on Frontiers in Handwriting Recognition, IWFHR*, pp. 449–454. Bari, Italy.
119. Mario, P., Samia, M., Volker, M., Ellouze, N., Hamid, A. (2002). IFN/ENIT-database of handwritten Arabic words. *Proceedings of CIFED 2002*, pp. 127–136. Hammamet, Tunisia.
120. Doush, I. A., Aikhateeb, F., Gharibeh, A. H. (2018). Yarmouk Arabic OCR dataset. *8th International Conference on Computer Science and Information Technology*, pp. 150–154. Amman, Jordan.
121. Amara, N. E. B., Mazhoud, O., Bouzrara, N., Ellouze, N. (2005). ARABASE: A relational database for Arabic OCR systems. *International Arab Journal of Information Technology, 2(4),* 259–266.
122. Hafiz, A., Bhat, G. (2016). Arabic OCR using a novel hybrid classification scheme. *Journal of Pattern Recognition Research, 11(1),* 55–60. DOI 10.13176/11.711.
123. El-Sawy, A., Mohamed Loey, H. E. B. (2017). Arabic handwritten characters recognition using Convolutional Neural Network. *WSEAS Transactions on Computer Research, 5(1),* 11–19.
124. Romanov, M., Miller, M. T., Savant, S. B., Kiessling, B. (2017). Important new developments in Arabographic Optical Character Recognition (OCR). arXiv:1703.09550.

125. Mudhsh, M. A., Almodfer, R. (2017). Arabic handwritten alphanumeric character recognition using very Deep Neural Network. *Information, 8(3),* 1–14. DOI 10.3390/info8030105.
126. Fasha, M., Hammo, B., Obeid, N., Al Widian, J. (2020). A hybrid deep learning model for Arabic text recognition. *International Journal of Advanced Computer Science and Applications, 11(8),* 122–130. DOI 10.14569/issn.2156-5570.
127. Shams, M., Elsonbaty, A. A., El Sawy, W. Z. (2020). Arabic handwritten character recognition based on Convolution Neural Networks and support vector machine. *International Journal of Advanced Computer Science and Applications, 11(8),* 144–149. DOI 10.14569/issn.2156-5570.
128. Ahmed, R., Gogate, M., Tahir, A., Dashtipour, K., Al-Tamimi, B. et al. (2021). Deep Neural Network-based contextual recognition of Arabic handwritten scripts. *Entropy, 23(3),* 4–6. DOI 10.3390/e23030340.
129. Jbrail, M. W., Tenekeci, M. E. (2022). Character recognition of Arabic handwritten characters using deep learning. *Journal of Studies in Science and Engineering, 2(1),* 32–40. DOI 10.53898/josse2022213.
130. ABBYY Software (2022). ABBYY FineReader engine. https://www.abbyy.com/ocr-sdk.
131. Smith, R. (2007). An overview of the tesseract OCR engine. *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 629–633. Washington, USA.
132. Yousfi, S., Berrani, S. A., Garcia, C. (2015). ALIF: A dataset for Arabic embedded text recognition in TV broadcast. *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 1221–1225. Tunis, Tunisia.
133. Chabchoub, F., Kessentini, Y., Kanoun, S., Eglin, V., Lebourgeois, F. (2016). SmartATID: A mobile captured Arabic text images dataset for multi-purpose recognition tasks. *Proceedings of International Conference on Frontiers in Handwriting Recognition*, pp. 120–125. Shenzhen, China.
134. Saad, R. S. M., Elanwar, R. I., Kader, N. S. A., Mashali, S., Betke, M. (2016). BCE-Arabic-v1 dataset: Towards interpreting Arabic document images for people with visual impairments Rana. *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, pp. 1–8. New York, NY, USA.
135. Ahmed, S., Bin Naz, S., Razzak, M. I., Yusof, R. B. (2019). A novel dataset for English-Arabic scene text recognition (EASTR)-42 K and Its evaluation using invariant feature extraction on detected extremal regions. *IEEE Access, 7,* 19801–19820. DOI 10.1109/ACCESS.2019.2895876.
136. Balaha, H. M., Ali, H. A., Saraya, M., Badawy, M. (2021). A new Arabic handwritten character recognition deep learning system (AHCR-DLS). *Neural Computing and Applications, 33(11),* 6325–6367. DOI 10.1007/s00521-020-05397-2.
137. Sakhr Software (2022). Sakhr OCR. http://www.sakhr.com/index.php/en/solutions/ocr.
138. Readiris Software (2022). Readiris OCR. https://www.irislink.com/.
139. Saber, S., Ahmed, A., Hadhoud, M. (2014). Robust metrics for evaluating Arabic OCR systems. *International Image Processing, Applications and Systems Conference*, pp. 1–6. Hammamet, Tunisia.
140. Stahlberg, F., Vogel, S. (2016). QATIP–An optical character recognition system for Arabic heritage collections in libraries. *12th International Workshop on Document Analysis Systems*, pp. 168–173. Wuhan, China.
141. Saber, S., Ahmed, A., Elsisi, A., Hadhoud, M. (2016). Performance evaluation of Arabic optical character recognition engines for noisy inputs. *The 1st International Conference on Advanced Intelligent System and Informatics*, pp. 449–459. Beni Suef, Egypt.
142. Ali, A. N. (2019). Optical character recognition software: A comparative evaluation study for information retrieval. *International Journal of Library and Information Sciences, 6(4),* 142–170.
143. Kiessling, B., Kurin, G., Miller, M., Smail, K. (2021). Advances and limitations in open source Arabic-script OCR: A case study. *Digital Studies/Le champ numerique, 11(1),* 1–30. DOI 10.16995/dscn.8094.