Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.048527

ARTICLE

Improving the Segmentation of Arabic Handwriting Using Ligature Detection Technique

Husam Ahmad Al Hamad* **and Mohammad Shehab***

College of Computer Sciences and Informatics, Amman Arab University, Amman, 11953, Jordan

*Corresponding Authors: Husam Ahmad Al Hamad. Email: hhamad@aau.edu.jo; Mohammad Shehab. Email: moh.shehab12@gmail.com

Received: 11 December 2023 Accepted: 26 February 2024 Published: 15 May 2024

ABSTRACT

Recognizing handwritten characters remains a critical and formidable challenge within the realm of computer vision. Although considerable strides have been made in enhancing English handwritten character recognition through various techniques, deciphering Arabic handwritten characters is particularly intricate. This complexity arises from the diverse array of writing styles among individuals, coupled with the various shapes that a single character can take when positioned differently within document images, rendering the task more perplexing. In this study, a novel segmentation method for Arabic handwritten scripts is suggested. This work aims to locate the local minima of the vertical and diagonal word image densities to precisely identify the segmentation points between the cursive letters. The proposed method starts with pre-processing the word image without affecting its main features, then calculates the directions pixel density of the word image by scanning it vertically and from angles 30° to 90° to count the pixel density from all directions and address the problem of overlapping letters, which is a commonly attitude in writing Arabic texts by many people. Local minima and thresholds are also determined to identify the ideal segmentation area. The proposed technique is tested on samples obtained from two datasets: A self-curated image dataset and the IFN/ENIT dataset. The results demonstrate that the proposed method achieves a significant improvement in the proportions of cursive segmentation of 92.96% on our dataset, as well as 89.37% on the IFN/ENIT dataset.

KEYWORDS

Arabic handwritten; segmentation; image processing; ligature detection technique; intelligent recognition

1 Introduction

Despite significant technological progress, numerous individuals continue to rely on handwritten text for their work, making the development of automated text recognition a complex endeavour in computer vision [1]. Text-recognition systems employ automated processes to convert text found in images into digital formats, enabling them to identify typed and handwritten characters useful in various domains.

The primary challenges in handwriting recognition primarily pertain to issues related to distortions and variations in patterns. Therefore, the process of feature extraction plays a crucial role

[2]. Opting for manual feature selection might result in a lack of necessary information for precise character classification. On the other hand, a high number of features can often pose problems due to the increased complexity of handling multidimensional data.

The issue of handwriting recognition has been extensively explored through various approaches, including support vector machines (SVMs) [3], K-nearest neighbors (KNNs) [4], and more recently, convolutional neural networks (CNNs) [5]. Nevertheless, earlier examinations in this space have fundamentally concentrated on Latin script penmanship recognition [6]. The Arabic dialect is one of the foremost talked-about dialects in the world, with over 320 million speakers. It presents a challenge for computer frameworks due to their cursive nature in both printed and transcribed shapes, which change in estimate and fashion [7]. Arabic letters in words are associated with even lines called strokes. The Arabic dialect comprises twenty-eight essential letters, each of which can be composed in up to four shapes depending on its position inside the word. Also, dabs in Arabic are utilized to distinguish between characters and can be found beneath or over the letter [8]. Additionally, the Arabic dialect is characterized by the nearness of diacritics (i.e., damma, kasra, and fatha), which confer diverse implications to the same word. In this manner, understanding the features of the Arabic dialect is pivotal for accomplishing precise results within the recognition preparation. Transcribed Arabic script division is isolating the script into bunches of related characters or segments. This handle is fundamental, to begin with, steps in numerous exercises such as optical character recognition (OCR), manually written Arabic content recognition, record assessment, and other related applications. The complex interconnects between Arabic calligraphy's characters make it troublesome to fragment Arabic calligraphy precisely [9]. Since Arabic penmanship styles shift, ligatures can be utilized, and character shapes depend on the setting, it can be troublesome to partition Arabic penmanship. To get exact division discoveries, analysts as often as possible combine space mastery, machine learning calculations, and picture-preparing procedures [10].

There are many research papers published in the Arabic handwritten recognition area [11]. These studies encompass various techniques, such as deep learning and artificial intelligence [12,13]. However, researchers have not yet achieved satisfactory results compared to English, which is the most widely spoken language in the world [14]. Additionally, Arabic script is predominantly cursive, meaning that letters are connected in most cases. This makes it challenging to separate individual characters, as the boundaries between them are not always well-defined. Furthermore, some Arabic characters have similar shapes, differing only in the placement of dots or other diacritics. Thus, improving handwriting recognition will also support the automatic reading of Arabic manuscripts, especially considering that Arabic script has not changed much over time and that automated processes can considerably expand the amount of available content [15].

This paper proposes a new Arabic handwritten recognition technique based on validating segmentation points and enhancing a heuristic feature technique. This technique is instrumental in extracting the features of handwritten Arabic words. Previous studies have utilized segmentation techniques to confirm segmentation points and validate the initial segmentation point sets. Introducing the merging of multiple successive segmentation areas is a further technique aimed at improving performance. The segmentation of handwritten Arabic words is accomplished by combining intelligent and heuristic techniques. Consequently, a significant contribution is made towards improving Arabic handwriting recognition based on the Ligature Detection Technique. The proposed technique aids in:

- Preprocessing word images to enhance image quality within the database by reducing distortions or improving features to yield more accurate outcomes.

- Determine correct segmentation points between the overlapped letters using a diagonal histogram (i.e., apply different angles between 30° and 90°).
- Locate the local minima of the vertical and diagonal word image densities, which leads to accurately identifying the segmentation points between the cursive letters.

The rest sections in this work are organized as follows: Related works are in Section 2. The proposed technique is in Section 3. The experiments and analysis of the results are in Section 4. Discussion and limitations are in Section 5. Finally, Section 6 presents the research conclusion and future works.

2 Literature Review

Research endeavors in the realm of word segmentation and recognition constitute a captivating and intricate domain. This allure stems from the myriad of applications that can benefit from a proficient handwritten word recognition system. The expanse of research within word segmentation and recognition is vast, and replete with complexities, largely due to the numerous algorithms that can be harnessed to implement word and handwriting recognition systems [16]. The persisting challenge in this field is intrinsically tied to the refinement of segmentation techniques aimed at extracting outcomes of utmost precision. Thus, this section shows the related works of this issue, followed by a summary as illustrated in Table 1.

Table 1: Summary of related works

Ref.	Year	Tech	Dataset	Results
Ali et al. [17]	2023	CNN	HODA	Accuracy 99.50%
Almutairy et al. [18]	2023	MHSA and BGRU	KHATT	MAE 5.28%, WER 25.13%
Ahamed et al. [19]	2020	CNN	Private	Accuracy 99.76%
Alqurashi et al. [20]	2021	CNN	Hijja	Accuracy 97%
Al Hamad et al. [21]	2022	ILDT	Benchmark	–
Zouhaira et al. [22]	2021	CNN and BLSTM	KHATT	CER 8%, WER 20.1%
Maalej et al. [23]	2022	MDLSTM	IFN/ENIT	92.59%
Granell et al. [24]	2023	BLSTM	HisClima	–
Pareek et al. [25]	2020	CNN and MLP	Private	Accuracy 97.21%
Shuvo et al. [26]	2022	CNN	Benchmark	Accuracy 99.62%
Souibgui et al. [27]	2022	Few-shot learning	Borg	SER 24%
Alghizzawi et al. [28]	2023	Self-supervised	TKH	Accuracy 95%
Cao et al. [29]	2020	CNN	Benchmark	–
Meddeb et al. [30]	2022	Deep-learning	BRAD	RMSE 80%
Awni et al. [31]	2022	Deep-learning	IFN/ENIT	96.11%

Ali et al. [17] utilized a modified deep convolutional neural network for the identification of hand-written digits, specifically Parisian/Eastern Arabic numerals and Arabic-Indic numerals. Their proposed model comprises two main components. The initial part incorporates multiple CNN blocks, involving normalization, pooling, and dropout techniques to engineer relevant features. The subsequent segment employs a flattened layer and a fully connected layer for handling classification tasks. They tested their approach on the HODA dataset and achieved an impressive accuracy of 99.50%. The experimental results investigated optimization techniques revealed that Adam's process achieved better results compared to other methods.

The authors in [18] introduced the Bidirectional Gated Recurrent Units (BGRU) and multi-head self-attention (MHSA) models in an attempt to mimic the functionality of our previous grapheme segmentation model (GSM). Within the recommended system, word inserting and the combination of BGRU and multi-head self-attention work together to assist in discovering control focuses (CPs) that are significant for written-by-hand content division. These CPs incorporate three critical geometric areas that portray the bounds of each grapheme: the endpoint (EP), the ligature valley point (LVP), and the starting point (SP). Our model's viability was demonstrated by exploratory comes about on the benchmark ADAB and online-KHATT datasets, where it accomplished 3.17% and 5.28% Cruel Outright Mistake and 12.25% and 25.13% Word Mistake Rate, separately.

CNN components were utilized by Ahamed et al. [19] to recognize written by hand Arabic numbers. Moreover, the authors included other morphological forms to an already-existing dataset of manually written Arabic numerals, bringing the whole number of pictures within the dataset up to 72,000. Through changes to a already proposed CNN show, they got a marvellous exactness rate of 98.91%. Also, their proposed engineering attained a momentous 99.76% accuracy, which puts it in line with the foremost later advancements within the field of transcribed Arabic numeral recognition.

Alqurashi et al. [20] proposed an interesting dataset called Hijja, utilized comprised written by hand Arabic letters by children aged 7 to 12. Furthermore, they displayed a show for programmed penmanship recognizable proof that produces utilize of a CNN. With its grid-like engineering, this demonstration can classify characters into a few categories and recognize characteristics by combining completely associated, pooling, and convolutional layers. Amazing precision of 97% on the Hijja dataset and 88% on the Arabic Written by Hand Character dataset were illustrated by the outcomes.

To broaden the range of areas that are not anticipated to constitute segmentation areas, Al Hamad et al. [21] raised the vertical pixel density. The technique calculates the density of the vertical pixel for detecting the local minimum points and determining the segmentation points accurately. By increasing the spacing between characters and limiting the number of local maxima and minima into which the word picture will split, the method specified the segmentation points. Then, following the neighborhood's centre, it develops a line-shaped structuring element and a block along the word image's vertical column.

In [22], the authors integrated CNN with Bidirectional Long Short-term Memory (BLSTM) techniques to recognize Arabic handwriting letters. The KHATT database was used to evaluate the efficiency of the proposed technique. In the preprocessing stage, they removed extra white regions and conducted binarization. CNNs were employed to capture features, while BLSTM and Connectionist Temporal Classification (CTC) were used for sequence modeling. The outcomes demonstrated a significant enhancement in performance, with an 8% Character Error Rate (CER) and a 20.1% Word Error Rat (WER) performance measure, outperforming other approaches.

In [23], the multi-dimensional Long Short-Term Memory (MDLSTM) design was inspected, and three new models' coordination Dropout, amended straight unit (ReLU), and Maxout were presented for offline Arabic penmanship recognition. In addition, to assess the execution of these models, the authors utilized an information increase method which included new operations utilized within the current IFN/ENIT dataset. The outcomes appeared that the proposed models accomplished the most elevated precision of 92.59%.

Granell et al. [24] displayed a database called HisClima combined with a perplexing collection of chronicled report images. The images in this database are similarly isolated between unthinkable and running content groups. Both age bunches have a combination of transcribed and pre-printed fabric. There are numerous acronyms and specialized terms utilized within the content that alludes to climate and vintage boats. HisClima may be a profitable asset for inquiring about innovation related to handling and analyzing photos of old reports. This covers the spaces of running and unthinkable content acknowledgment. The consider gives pattern comes about for Probabilistic Ordering, Content Recognition, and Record Format Examination. Even though these results show excellent performance, there is still room for improvement, and some guidelines are offered to help with that.

Pareek et al. [25] presented a novel method for handwritten letter recognition using artificial intelligence. CNN and Multilayer Perceptron (MLP) were used in this groundbreaking study. Their experimental experiments yielded noteworthy results in terms of consistency and precision. These fruitful outcomes subsequently offered a solid basis for augmenting the system's functionality even further. The focus on improving the dataset to increase the system's resistance to differences found in different handwriting styles was one important feature.

Shuvo et al. [26] introduced a new technique to improve the recognition of handwritten numbers. The scientists used a CNN to categorize the images of handwritten digits rather than traditional feature extraction approaches. Their method's primary contribution is its preprocessing of word pictures, which skips any explicit feature extraction steps. Their study's outcome was particularly impressive because, when compared to other researchers' approaches, their proposed strategy produced much greater recognition accuracy for all categories of handwritten numbers.

Souibgui et al. [27] employed few-shot learning to recognize handwriting. The authors used a limited collection of sample photos for every alphabet symbol to reduce the need for manual annotation. The procedure entails locating alphabet symbols inside a text line image and decoding them into a final sequence of transcribed symbols using similarity scores. During the model's pretraining phase, artificial line pictures created from a single alphabet are used as training data. The alphabet used here may not be the same as the one in the target domain. There is another training phase that is carried out to close the gap between the source and target data.

In [28], the authors focused on introducing a new method to deal with recognizing documents that contain music and text characters. The proposed method included extracting a set of features from documents. After that, training a neural network to improve the recognition procedures. The results illustrated that the proposed method achieved accurate scores. According to the study's findings, the proposed technique surpasses basic classification methods in terms of accuracy. Furthermore, as the size of the training dataset for classification increases, the results indicate that the supervised learning framework leads to even better performance.

Cao et al. [29] proposed a new method based on the hierarchical understanding of Chinese characters for zero-shot handwritten Chinese character recognition (HCCR). The authors started by determining the relationships between the letters and major components to get the tree structure (i.e., set the priorities). After that, HCCR has been applied to encrypt the tree structure. Then, CNN was used to learn the structure and radical letters. HCCR proved its efficiency compared with some methods in the literature.

Meddeb et al. [30] introduced a technique for recommending and predicting ratings for Arabic textual documents. Their method, a fusion of neural collaborative embedding and filtering, harnessed deep feature learning and deep interaction modeling to grasp both the broader context and word frequency details. They applied this approach to the Books Reviews in Arabic Dataset (BRAD), comprising 510,600 reviews, and their results outperformed existing methods, delivering an impressive RMSE of 0.800.

Awni et al. [31] discussed the performance of three randomly initialized deep convolutional neural networks in identifying Arabic handwritten words. Subsequently, the authors assessed how well the pre-trained ResNet18 model, initially trained on the ImageNet dataset, performed on the same task. In addition, introduced a novel approach that involves sequentially transferring mid-level word image representations through two phases using the ResNet18 model. This approach underwent evaluation through four sets of experiments conducted on IFN/ENIT datasets. The findings showed a 14% improvement in recognition accuracy for ten frequently misclassified words in the IFN/ENIT dataset when utilizing ImageNet as a source dataset.

3 Methodology

This section presents the technique of Arabic handwritten scripts, as well as introduces a new method to improve the segmentation results, Fig. 1 shows the architecture of the proposed method. It's worth mentioning that the proposed technique differs from other techniques. It removes the punctuation marks and dots, which aim to make it easier to locate the expected ligatures or segmentation points between the letters. In addition, it calculates the script image's pixel density by scanning the word image from 30° to 90° and counting the pixel density from all directions which is useful for recognizing Arabic handwritten characters, more details are shown in the following sub-section.

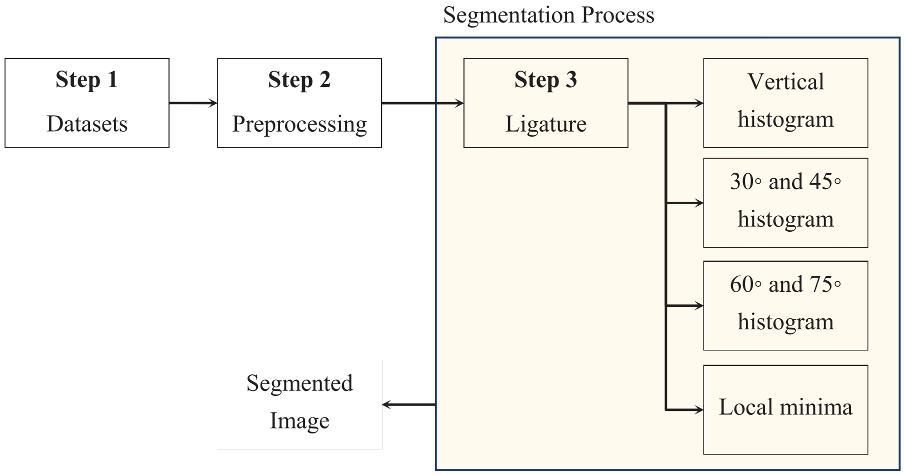


Figure 1: Architecture diagram of the proposed method

3.1 Step 1: Datasets

Two datasets were used to evaluate the proposed technique. Our initial self-curated image dataset was assembled from the contributions of 10 distinct individuals. It consisted of 500 random words extracted from two paragraphs encompassing all conceivable Arabic letter shapes handwritten by those individuals, as there is not a universally accepted Arabic benchmark dataset for Arabic letter discrimination [32]. The selection of this dataset ensures that our findings can be benchmarked against those of other researchers who utilize the same dataset. Subsequently, all Word images were converted to the (jpg) format. The data utilized in this study comprises a diverse assortment of Arabic handwritten word samples, as depicted in Fig. 2.

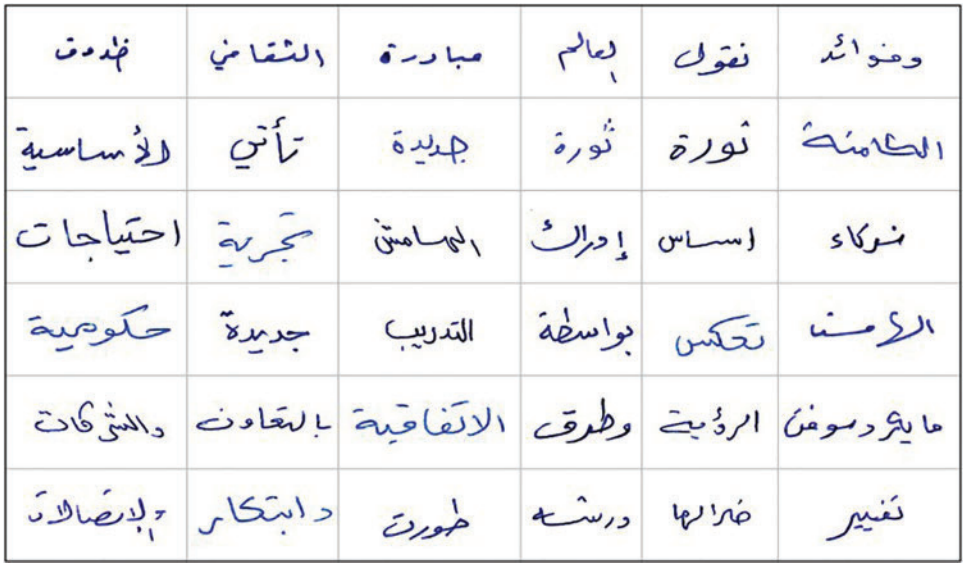


Figure 2: Samples of handwritten words benchmark dataset written by ten different persons

The second well-known dataset, namely, the IFN/ENIT, contains 32,492 handwritten Arabic words representing the names of Tunisian cities alongside their postal codes, with contributions from over a thousand unique individuals [33]. These city names were scanned at a resolution of 300 dpi and transformed into binary images. The dataset is partitioned into five distinct sets (a, b, c, d, e), enabling researchers to utilize some for-training purposes and others for testing. Fig. 3 shows a sample of the mane of four cities.

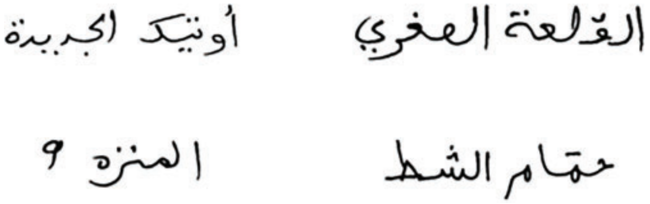


Figure 3: Samples of handwritten words for four cities from IFN/ENIT dataset

3.2 Step 2: Preprocessing

Prior to further processing, as in the majority of handwriting segmentation and recognition systems, the words image that was obtained were in an RGB-scale format changed from RGB-level images to grey-level to utilize binary format. Nevertheless, concisely, it entails converting a greyscale image (0–255) into a binary image (0–1) causing some features to be lost which can have little effect on the accuracy of the results.

There are many advantages to storing word images in this format, for example removing color levels makes it simpler to manipulate images. In addition, Additional processing will be quicker, and less expensive in time and calculations, and enable more compact storage. There are of course a number of disadvantages such as the loss of some of the information from the original image or the introduction of anomalies or noise. Most researchers concluded that current techniques are still not enough for the conversion of difficult images. However, they noted that the best results that were attained employed local conversion techniques. An important part of the image involves determining the filter at which pixels in the images become white or dark. Local conversion techniques calculate a threshold based on neighboring pixels. While global techniques determine one style for the whole image.

The technique described here includes a last preprocessing step at this level. The technique removes the little foreground objects that were not part of the writing by implementing a series of rules to detect and remove noise elements that were either prominent in the original word image or that were added during filling the circles. When the components of the word matrix were found, it was feasible to carry out a number of helpful operations. Prior to further processing, it was possible to convert a word image to binary format and then extract the features, and connected components that were considered noise in the word image.

Another aspect of preprocessing is eliminating dots and punctuation marks, removing these shapes strokes have been used to enhance the performance of segmentation, dots form obstacles for the smooth segmentation of handwritten letters for many reasons. There are three advantages to eliminating punctuating marks. First, to enhance the results of the ligature baseline detection technique. Second, to increase over-segmentation accuracy using the segmentation technique. It uses a vertical histogram by calculating the distance between the top and bottom foreground pixels of a word image; this leads to reduced training errors and times of training/testing neural network and increases the performance system. There are two disadvantages of removing the dots. Some dots are critical to be recognized for many reasons such as dote size and merging with other letters. This makes a hardness to remove it. Also, some letters such as ‘ر‘, د’ are detected as dots, this means there is a possibility that the algorithm may delete them.

As mentioned above, the preprocessing of the initial and crucial stage in developing an OCR model focuses on enhancing image quality within the database by either reducing distortions or improving features to yield more accurate outcomes. Despite the fact that the IFN/ENIT images had already undergone extraction and binarization, we implemented a preprocessing step to resize the input images to the 32 × 128 dimensions without introducing any distortion. Consequently, the main steps of the pre-processing for the word image are shown below.

1. Convert RGB image to binary image

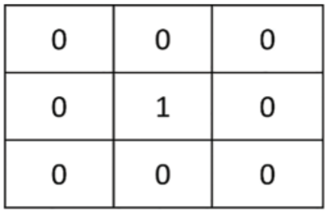
The technique converts RGB images to grayscale by eliminating the saturation and hue information and keeping the luminance. A binary image is then produced by assigning 0 s to all values under a globally defined 1 s threshold, then the technique employs a histogram threshold using Otsu’s method, which chooses the threshold value to minimize the black and white pixels of class variation using a 256-picture histogram.

2. Remove dotes and isolated pixels

The technique removes isolated foreground pixels 1 s that is surrounded by background pixels 0 s taking into account the size and dimensions of the pattern in length and width, as shown in Fig. 4.

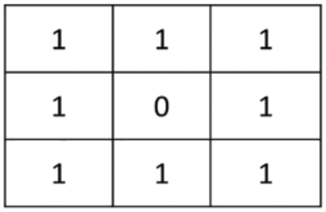
3. Fill close and open (missing circle) letters

Examples of close letters In Arabic scripts are: 'ق', 'ص', 'ه', 'هـ', 'ضـ', 'ط', and missing circle letters (open litters) examples are: 'عـ‘, 'جـ', 'ع', 'ـعـ', 'غ', 'ح'. The technique fills the interior background pixels 0 s surrounded by foreground pixels 1 s that take into account the size and dimensions of the pattern in length and width, as shown in Fig. 5.



0	0	0
0	1	0
0	0	0

Figure 4: Remove dotes and isolated pixels



1	1	1
1	0	1
1	1	1

Figure 5: Fill close and open (missing circle) letters

The technique includes the background pixels 0 s in the morphological computation without foreground pixels 1 s. The interior pixel of the pattern identifies the pixel in the image and creates a flat structure element for the binary images to remove select punctuation marks and then delete them.

3.3 Step 3: Ligature Detection Method

This section describes the proposed technique. Regarding the complicated Arabic handwritten characters (i.e., the connected characters, unconnected characters, and the curved shape of some characters), the ligature detection technique is divided into sub-methods to recognize the different shapes of Arabic handwritten characters.

3.3.1 Vertical Histogram

In Arabic handwriting, a ligature is a brief stroke that joins two letters together. The central portion of handwritten words typically contains the main component of ligatures. A ligature detection method can be used to locate potential ligatures by locating the middle region and creating a vertical histogram for the foreground pixels of the handwriting. If the pixel density of a vertical column in the image were zero, this would instantly suggest a white space (foreground pixels). Thus, this would be indicated as a segmentation point without additional investigation. However, the pixel density would then be determined if foreground pixels were found while scanning the vertical column. The density of vertical column pixels had to be lower than a specific threshold in order to determine if the word image should be segmented at that specific location.

Regarding the segmentation areas of non-cursive letters, Fig. 6 provides an illustration of how to define distinct segmentation areas of the word image containing separate letters or sub-words words the vertical histogram technique is used to determine these areas by calculating the total background pixels, and then placing them in the middle of the distance between the two letters, as shown in Fig. 6c. The local minima of the cursive letters are also calculated as will be explained in detail later. As shown in Fig. 6d, the segmentation point between the two cursive letters is determined by calculating the local minima histogram of vertical pixel density and placing reliable segmentation points at the correct position when the density column is lower than the specific threshold after calculating the median of the local minima between the two letters.

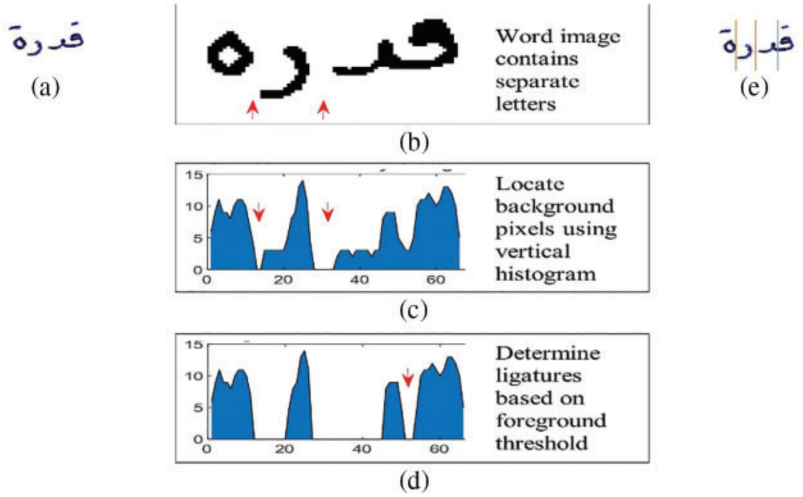


Figure 6: Determine segmentation points using vertical histogram (a) word image before preprocessing (b) word image after preprocessing, (c) vertical histogram locates white segmentation points, (d) vertical histogram after determining ligature segmentation points, and (e) the word image after segmentation

3.3.2 30° and 45° Histogram

This section shows the mechanism of dealing with a word that contains vertically overlapped letters but is not connected. The technique has improved the accuracy of the correct segmentation points by calculating the diagonal histogram between angles 30° to 45°. It can be noticed that Figs. 7a and 7b show an example of the letter (wa 'و') overlapping with the letter (ra 'ر') and (fa 'ف'), this overlap hinders finding a correct segmentation point using the previous method. Therefore, the technique finds the diagonal histogram on all angles and thus accurately determines the overlapping segmentation points that will increase the segmentation performance. Figs. 7c and 7d illustrate a segmentation histogram of angles 30° and 45°, respectively. In both cases, the correct segmentation points were placed accurately.

3.3.3 60° and 75° Histogram

If two angles achieve the white segmentation area of the overlapping between letters as shown in Fig. 8a, in this case, and for more efficiency, the method will ignore the second angle as where a correct segmentation point is placed by the previous one. Fig. 8c shows the segmentation histogram of degrees 60° where a correct segmentation point has been successfully placed, and Figs. 8b and 8d show the segmentation histogram of degrees 60° and 75°, where a correct segmentation point has also been successfully placed.

3.3.4 Local Minima

Returning to segmentation cursive letters, the technique of local minima between two characters is used, where the technique determines the correct segmentation points if the calculated local minima are less than a certain threshold calculated based on the stroke width of the letter. To be clear, Fig. 9 shows an example explaining all the steps, and Fig. 9b shows the word image after preprocessing according to the previously explained technique. Fig. 9c appears the vertical histogram that calculates the local minima between the two letters, thereafter and accordingly determines a correct segmentation point that achieves the calculated threshold. Fig. 9d shows the locations of all segmentation points after applying the technique of removing the background pixels in place of the points. The overall results appear in Fig. 9e by placing the segmentation points correctly between all letters of the cursive word.

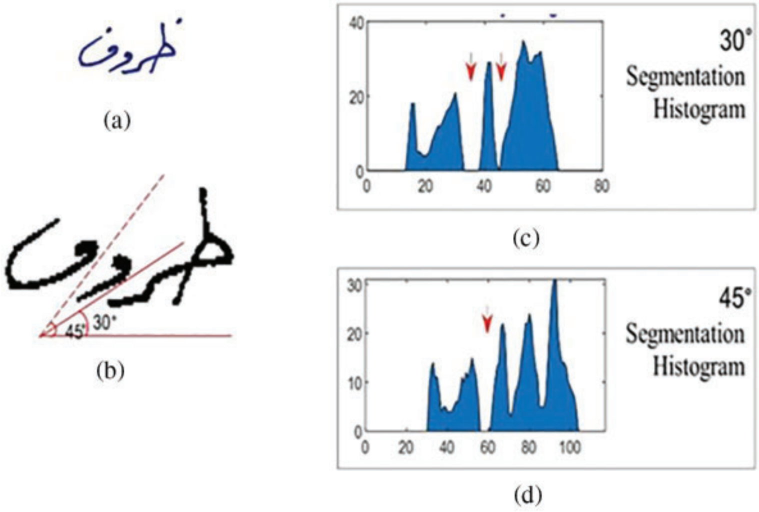


Figure 7: Determine segmentation points using angles 30° and 45° histogram (a) overlapped word image before preprocessing (b) overlapped word image after preprocessing, (c) 30° histogram diagonal locates white segmentation points, (d) 45° histogram locates white segmentation points

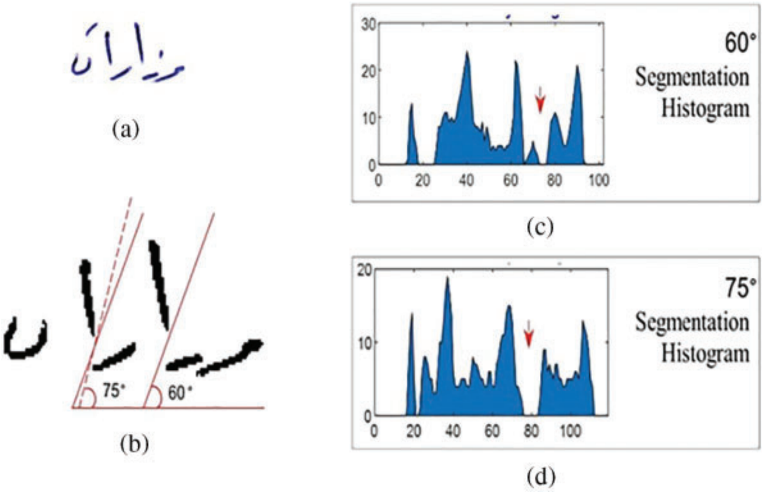


Figure 8: Determine segmentation points using angles 60° and 75° histogram. (a) overlapped word image before preprocessing, (b) overlapped word image after preprocessing, (c) 60° histogram diagonal locates white segmentation points, (d) 75° histogram locates white segmentation points

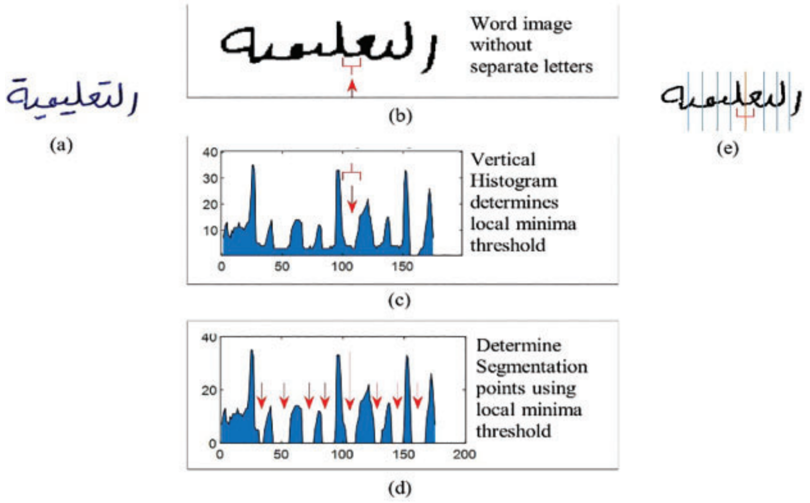
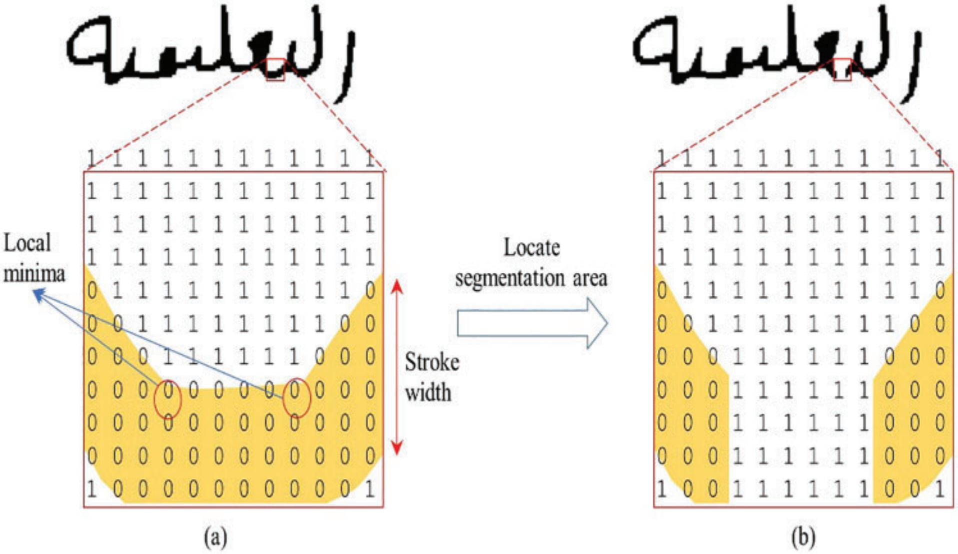


Figure 9: Determine segmentation points using local minima threshold (a) word image before preprocessing (b) word image after preprocessing, (c) vertical histogram of the image, (d) vertical histogram after determining ligature segmentation points, and (e) the word image after segmentation

By locating local minima between the cursive letters and calculating the stroke width of the handwritten word, the technique placed the appropriate segmentation area when the vertical density of stroke background pixels between two local minima is smaller than the maximum density of the stroke in any part of the word. Fig. 10 illustrates the method for determining the segmentation area. Fig. 10a shows the first step that locates the local minima between two letters. When the ligature calculations match the specific threshold, the segmentation area is placed as shown in Fig. 10b.



```
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 1 1 0
0 0 1 1 1 1 1 1 1 1 0 0
0 0 0 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 1
```

```
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 1 1 0
0 0 1 1 1 1 1 1 1 1 0 0
0 0 0 1 1 1 1 1 1 0 0 0
0 0 0 1 1 1 1 1 1 0 0 0
0 0 0 1 1 1 1 1 1 0 0 0
0 0 0 1 1 1 1 1 1 0 0 0
1 0 0 1 1 1 1 1 1 0 0 1
```

Figure 10: Locating segmentation points using local minima and threshold (a) local minimum of segmentation area (b) segmentation area after locating

4 Experimental Results and Analysis

The primary measure for assessing a model classifier's performance is accuracy, reflecting the percentage of test samples correctly predicted by the classifier on new or unseen data. Additionally, our evaluation involved three criteria: (i) instances of over-segmentation, (ii) missed segmentations, and (iii) bad segmentations. Over-segmentations occur when a character is split into more than three parts [34]. Missed segmentations refer to correct segmentation points that were not identified. Finally, bad segmentations happen when a segmentation point fails to perfectly separate two characters, resulting in the point dissecting a piece of an adjacent character or a similar irregularity.

4.1 Using Our Initial Self-Curated Image Dataset

This section shows the procedures of ligature detection technique on Our initial self-curated image dataset. 3,097 segmentation points (SP) were tested across the 500-word images extracted from the database, where the number of segmentation errors found are for letters and not for words. As shown in Table 2, the criteria were used to demonstrate the technique's precision. For instance, the first criterion of error rate is missed segmentations, which refers to the possibility of two adjacent letters not being separated at all. Bad segmentations, or those that are neither correct nor missed, make up the second criterion. A bad segmentation occurs when certain letter components have been improperly separated or when two adjacent letters have been split in a way that either one or both of the letters have been deformed. The overall technique results performed extremely well with 92.96% accuracy, error rates for missed segmentation points were very low at 0.16%, and the bad segmentation point achieved 6.88% which is also promising compared to the results of other studies.

Table 2: Performance of the segmentation technique

SP	Correct segmentation	Segmentation error rates		
		Missed	Bad	Total
3,097	2,879 92.96%	5 0.16%	213 6.88%	218 7.04%

Table 3 presents the segmentation performance results for ten writers. These writers wrote some Arabic paragraphs containing many different words in completely different ways under different circumstances. Thus, it may affect the level of accuracy of the results. The results illustrate that the segmentation of the written letters of people (3–5, and 7–9) was excellent with missing errors of 0.00%, and this indicates the clarity and consistency of writing and character. In addition, the calculation CPU time for segmentation for each experiment was taken into consideration, to show the efficiency of the technique. It is worth mentioning that we used a PC Windows 11 operation system with a Core i7 processor and 16 GB RAM memory for all experiments. It can be realized that the difference in the style and method of writing scripts affects the processing speed. The width of the line strokes needs more processing, and therefore the increase in processing time for some writers despite the small number of segmentation points is due to the quality of writing words. For instance, the writing style of the writer number seven was clearly overlapping. In other words, the overlapping between the words' letters without enough spaces by the writer is obviously at the end of the first sub-string of the word's shape overlap vertically with the second sub-string of the word's shape, as shown in Fig. 11.

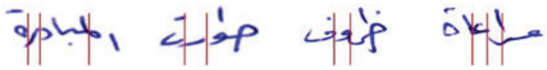


Figure 11: Samples of overlapping characters

Table 3: Segmentation performance for 10 persons

Writers	SP	CPU time	Correct segmentation	Segmentation error rates		
				Missed	Bad	Total
Writer (1)	295	0.0312	285	1	9	10
			96.61%	0.34%	3.05%	3.39%
Writer (2)	303	0.1406	276	1	26	27
			91.09%	0.33%	8.58%	8.91%
Writer (3)	354	0.4844	326	0	28	28
			92.09%	0.00%	7.91%	7.91%
Writer (4)	293	0.1719	281	0	12	12
			95.90%	0.00%	4.10%	4.10%
Writer (5)	294	0.1406	265	0	29	29
			90.14%	0.00%	9.86%	9.86%
Writer (6)	343	0.2812	317	1	25	26
			92.42%	0.29%	7.29%	7.58%
Writer (7)	305	0.5000	279	0	26	26
			91.48%	0.00%	8.52%	8.52%
Writer (8)	323	0.4688	301	0	22	22
			93.19%	0.00%	6.81%	6.81%
Writer (9)	310	0.3594	286	0	24	24
			92.26%	0.00%	7.74%	7.74%
Writer (10)	277	0.3281	263	2	12	14
			94.95%	0.72%	4.33%	5.05%

Fig. 12 shows the differences between the ten writers based on segmentation rates. Each column includes the rate of correct segmentation, missed (if any), and bad for each writer. The accuracy of the results varies due to the style and quality of writing as mentioned previously. The dash line indicates the performance of the technique for each writer (i.e., CPU time), relatively. It can be noticed that there is an inverse relationship between the number of PS and CPU time (i.e., when the determines a few segmentation points, the processor spends more time).

4.2 IFN/ENIT Dataset

This section shows the efficiency of the ligature detection technique using the IFN/ENIT dataset. It is worth mentioning that 569-word images were extracted from the IFN/ENIT dataset to test 4585 segmentation points (SP). The extraction aims to focus on the number of segmentation errors found for letters and not for words. Table 4 illustrates that the ligature detection technique achieves the correct segmentation with 89.37%, Error rates for missed segmentation points were very low at 2.34%, and the bad segmentation point achieved 8.29%. Fig. 13 shows an example of the ligature detection technique results on the IFN/ENIT database.

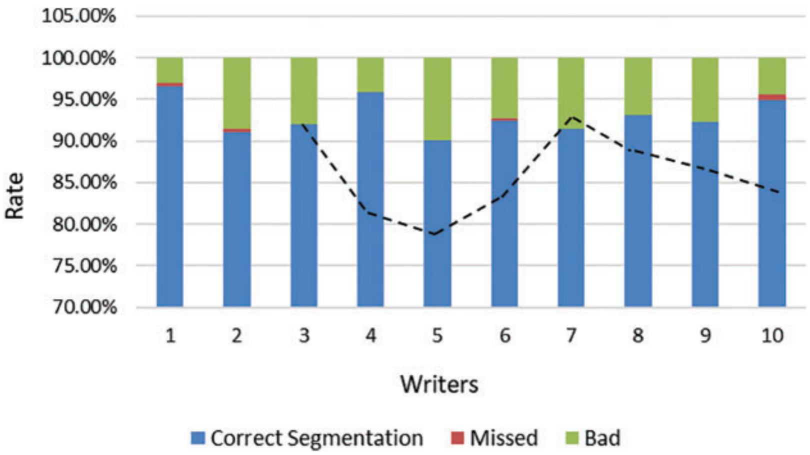


Figure 12: Segmentation rates for ten writers

Table 4: Performance of the segmentation technique of IFN/ENIT-database

SP	Correct segmentation	Segmentation error rates		
		Missed	Bad	Total
4,585	4,098 89.37%	107 2.34%	380 8.29%	487 10.63%

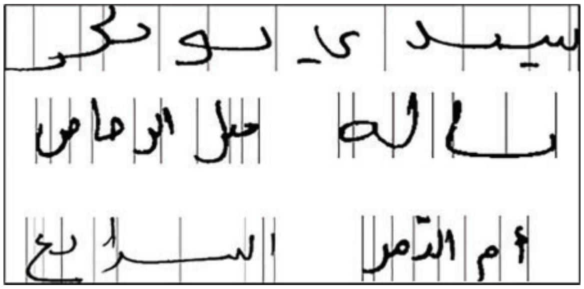


Figure 13: Samples of overlapping characters

5 Discussion and Limitations

The primary objective of this research was to create a precise segmentation technique for accurately processing handwritten Arabic scripts. This section will discuss the significance of the ligature detection technique in terms the performance and mechanism compared with the other techniques, as well as the challenges we faced in this work.

The ligature detection technique employed vertical histograms to segment the word and some basic important features in Arabic handwritten script were identified, such as average letter width, middle region detection based on vertical density histograms, local minima of a vertical histogram, and circular stroke location. The position of circles and semicircles, such as the letters 'هـ', 'و', 'ط', 'حـ ع" etc., or horizontal parallel lines, is another feature that is checked to identify potential segmentation points to see if there is a gap. The image of the word is scanned from top to bottom (vertically) and at angles between 30° to 90° that locate the foreground pixels between letters.

The technique proceeds to locate the vertical areas of segmentation between the letters of the word to identify areas of lower pixel density. Determining the low pixel density in the vertical direction using vertical histograms of word images would indicate prospective segmentation points. If the pixel density was zero upon inspection of a vertical column or a diagonal in the image from angle 30° to before angle 90°, this would indicate a white space and automatically warrant a segmentation point without further investigation. However, the pixel density would be determined later if a foreground pixel was discovered while scanning the vertical column. The vertical pixel density for that column had to be lower than a specific threshold in order to determine whether the word should be segmented at that specific threshold. The significance of this technique is that the segmentation point is given a confidence value based on where the feature is located. All segmentation points are then examined, and concentrations of segmentation points are eliminated to leave the segmentation points that are most indicative of a given region. The last step is to look over every segmentation point and remove any that have a low foreground pixel density.

The results of the ligature detection technique using two different datasets were satisfactory, as shown in Fig. 14. It can be noticed that the results were achieved using the IFN/ENIT dataset at 89.37%. Therefore, we plan to increase its efficiency to deal with other datasets, accurately.

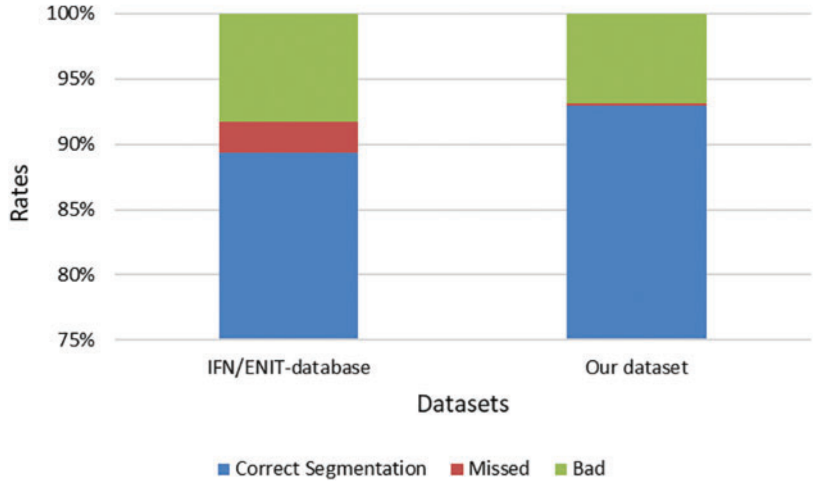


Figure 14: Performance of the segmentation technique

Due to the different datasets used to test the algorithms and techniques, the segmentation results may be difficult to compare to those of other researchers in the literature. So, in Table 5, we tried to summarize the comparison results obtained by many researchers in comparison to the proposed segmentation algorithms, as much as can. The results illustrate that the proposed technique has a higher level of word segmentation accuracy compared to the other techniques.

Table 5: Comparison with other segmentation research results in the literature

Ref.	Year	Technique	Results
Balaha et al. [35]	2021	CNN	90.70%
Rabi et al. [36]	2018	Hidden markov	87.93%
Ghanim et al. [37]	2020	Matching & ranking	91.60%
Maalej et al. [23]	2022	MDLSTM	92.59%
Daoud et al. [38]	2017	MCDNN	91.50%
Proposed technique			92.69%

In this work, we faced several limitations during the model training process. These included dealing with the diverse formation of ligatures caused by variations in handwriting styles and individual preferences, which makes it challenging to create a universal ligature detection model. Another issue was the scarcity of labeled training data, as building a robust model requires a significant number of annotated examples. Unfortunately, there is a shortage of extensive, annotated Arabic ligature datasets, which can impede the development and performance of such models. Furthermore, Arabic script exhibits a multitude of regional variations and calligraphic styles, making it difficult for ligature detection models to generalize across these diverse styles, often necessitating adaptation for specific variants. The following points show some cases that the proposed technique might fail to deal with them:

- When there are letters that overlap significantly vertically. So, the solution is to identify the separate (unconnected) elements and consider them as blocks, which are distinguished on their own and their location is determined according to the other letters.
- When unable to remove large cutting points. Thus, the solution is to determine possible shapes of punctuation points and remove them accordingly and not rely solely on their size compared to written letters.
- When removing letters as punctuation points. In this case, the solution is to identify punctuation shapes that can be selected. This helps not to remove shapes that are likely to be letters rather than punctuation.

6 Conclusion and Future Work

The ligature detection technique is proposed to improve the segmentation accuracy of Arabic handwriting. The methodology of the proposed technique is divided into four levels (i.e., Vertical histogram, 30° and 45° histogram, 60° and 75° histogram, and Local minima) to deal with all cases of the Arabic cursive letters. The initial self-curated image dataset and IFN/ENIT dataset were used to evaluate the efficiency of the proposed technique. The experiment results showed outperformed the proposed technique compared with other similar techniques in the literature. In the future, we plan to investigate the possibilities of using pre-trained transformer models for fine-tuning and limited-shot learning to address individualized requirements for recognizing digits. Additionally, we see promising potential in leveraging artificial intelligence techniques to replicate human-like text recognition abilities, especially when dealing with text in diverse colors, varied sizes, styles, and complex formats. Looking ahead, a promising avenue for future research could involve utilizing OCR methods to extract data from historical text images, particularly in domains like heritage culture and preservation.

Acknowledgement: None.

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

Author Contributions: Conceptualization: Husam Ahmad Al Hamad and Mohammad Shehab; Methodology: Husam Ahmad Al Hamad; Formal analysis: Mohammad Shehab; Original draft preparation: Husam Ahmad Al Hamad; Review and editing: Husam Ahmad Al Hamad and Mohammad Shehab; Visualization: Mohammad Shehab; Project administration: Mohammad Shehab; Husam Ahmad Al Hamad and Mohammad Shehab reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

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

References

[1] N. Altwaijry and I. Al-Turaiki, "Arabic handwriting recognition system using convolutional neural network," *Neur. Comput. App.*, vol. 33, no. 7, pp. 2249–2261, 2021. doi: 10.1007/s00521-020-05070-8.

[2] H. A. Al-Hamad and R. A. Zitar, "Development of an efficient neural-based segmentation technique for arabic handwriting recognition," *Pattern Recogn.*, vol. 43, no. 8, pp. 2773–2798, 2010. doi: 10.1016/j.patcog.2010.03.005.

[3] A. A. A. Ali and S. Mallaiah, "Intelligent handwritten recognition using hybrid CNN architectures based-SVM classifier with dropout," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3294–3300, 2022. doi: 10.1016/j.jksuci.2021.01.012.

[4] Y. Chychkarov, A. Serhiienko, I. Syrmamiikh, and A. Kargin, "Handwritten digits recognition using SVM, KNN, RF and deep learning neural networks," *CMIS*, vol. 2864, pp. 496–509, 2021. doi: 10.32782/cmis/2864-44.

[5] D. S. Prashanth, R. V. K. Mehta, K. Ramana, and V. Bhaskar, "Handwritten devanagari character recognition using modified lenet and alexnet convolution neural networks," *Wirel. Pers. Commun.*, vol. 122, no. 1, pp. 349–378, 2022. doi: 10.1007/s11277-021-08903-4.

[6] A. Al-Mansoori *et al.*, "Technology enhanced learning through learning management system and virtual reality googles: A critical review," in *From Industry 4.0 to Industry 5.0*, 2023, vol. 470, pp. 557–564. doi: 10.1007/978-3-031-28314-7_48.

[7] M. S. Daoud, M. Shehab, L. Abualigah, and C. L. Thanh, "Hybrid modified chimp optimization algorithm and reinforcement learning for global numeric optimization," *J. Bionic. Eng.*, vol. 20, no. 6, pp. 1–20, 2023. doi: 10.1007/s42235-023-00394-2.

[8] N. Alrobah and S. Albahli, "Arabic handwritten recognition using deep learning: A survey," *Arab J. Sci. Eng.*, vol. 47, no. 8, pp. 9943–9963, 2022. doi: 10.1007/s13369-021-06363-3.

[9] A. A. H. Al-Radaideh, M. S. Bin Mohd Rahim, W. Ghaban, M. Bsoul, S. Kamal and N. Abbas, "Arabic words extraction and character recognition from picturesque image macros with enhanced VGG-16 based model functionality using neural networks," *KSII Trans. Internet Inf. Syst.*, vol. 17, no. 7, pp. 1807, 2023. doi: 10.3837/tiis.2023.07.004

[10] S. Benbakreti, M. Benouis, A. Roumane, and S. Benbakreti, "Stacked autoencoder for arabic hand-writing word recognition," *Int. J. Comput. Sci. Eng.*, vol. 24, no. 6, pp. 629–638, 2021. doi: 10.1504/IJCSE.2021.119988.

[11] L. M. Lorigo and V. Govindaraju, "Offline arabic handwriting recognition: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 712–724, 2006. doi: 10.1109/TPAMI.2006.102.

[12] M. Alghizzawi, M. Habes, A. Al Assuli, and A. A. R. Ezmigna, "Digital marketing and sustainable businesses: As mobile apps in tourism," in *Artificial Intelligence and Transforming Digital Marketing*, 2023, pp. 3–13.

[13] M. Mudhsh and R. Almodfer, "Arabic handwritten alphanumeric character recognition using very deep neural network," *Inf.*, vol. 8, no. 3, pp. 105, 2017. doi: 10.3390/info8030105.

[14] H. Menhour *et al.*, "Searchable turkish ocred historical newspaper collection 1928–1942," *J. Inf. Sci.*, vol. 49, no. 2, pp. 335–347, 2023. doi: 10.1177/01655515211000642.

[15] H. A. Al Hamad, "Over-segmentation of handwriting arabic scripts using an efficient heuristic technique," in *2012 Int. Conf. Wavelet Analy. Pattern Recogn.*, Xian, China, IEEE, 2012, pp. 180–185.

[16] M. Shehab and L. Abualigah, "Opposition-based learning multi-verse optimizer with disruption operator for optimization problems," *Soft. Comput.*, vol. 26, no. 21, pp. 11669–11693, 2022. doi: 10.1007/s00500-022-07470-5.

[17] S. Ali *et al.*, "A recognition model for handwritten persian/arabic numbers based on optimized deep convolutional neural network," *Multimed. Tools App.*, vol. 82, no. 10, pp. 14557–14580, 2023. doi: 10.1007/s11042-022-13831-x.

[18] N. M. Almutairy, K. H. Al-Shqeerat, and H. A. Al Hamad, "A taxonomy of virtualization security issues in cloud computing environments," *Indian J. Sci. Tech.*, vol. 12, no. 3, pp. 1–19, 2019. doi: 10.17485/ijst/2019/v12i3/139557.

[19] P. Ahamed, S. Kundu, T. Khan, V. Bhateja, R. Sarkar, and A. F. Mollah, "Handwritten arabic numerals recognition using convolutional neural network," *J Ambient Intell. Human Comput.*, vol. 11, no. 11, pp. 5445–5457, 2020. doi: 10.1007/s12652-020-01901-7.

[20] D. R. Alqurashi, M. Alkhaffaf, M. K. Daoud, J. A. Al-Gasawneh, and M. Alghizzawi, "Exploring the impact of artificial intelligence in personalized content marketing: A contemporary digital marketing," *Migrat. Lett.*, vol. 20, no. S8, pp. 548–560, 2023.

[21] H. A. Al Hamad, L. Abualigah, M. Shehab, K. H. Al-Shqeerat, and M. Otair, "Improved linear density technique for segmentation in arabic handwritten text recognition," *Multimed. Tools App.*, vol. 81, no. 20, pp. 28531–28558, 2022. doi: 10.1007/s11042-022-12717-2.

[22] N. Zouhaira, M. Anis, and K. Monji, "Contribution on Arabic handwriting recognition using deep neural network," in *Hybrid Intell. Syst.: 19th Int. Conf. Hybrid Intell. Syst. (HIS 2019)*, Bhopal, India, Springer, 2020, pp. 123–133.

[23] R. Maalej and M. Kherallah, "New MDLSTM-based designs with data augmentation for offline arabic handwriting recognition," *Multimed. Tools App.*, vol. 81, no. 7, pp. 10243–10260, 2022. doi: 10.1007/s11042-022-12339-8.

[24] E. Granell *et al.*, "Processing a large collection of historical tabular images," *Pattern Recogn. Lett.*, vol. 170, no. 5, pp. 9–16, 2023. doi: 10.1016/j.patrec.2023.04.007.

[25] J. Pareek, D. Singhania, R. R. Kumari, and S. Purohit, "Gujarati handwritten character recognition from text images," *Procedia Comput. Sci.*, vol. 171, no. 1, pp. 514–523, 2020. doi: 10.1016/j.procs.2020.04.055.

[26] M. Shuvo, M. Akhand, and N. Siddique, "Handwritten numeral recognition through superimposition onto printed form," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 9, pp. 7751–7764, 2022. doi: 10.1016/j.jksuci.2022.06.019.

[27] M. A. Souibgui, A. Fornés, Y. Kessentini, and B. Megyesi, "Few shots are all you need: A progressive learning approach for low resource handwritten text recognition," *Pattern Recogn. Lett.*, vol. 160, no. 6, pp. 43–49, 2022. doi: 10.1016/j.patrec.2022.06.003.

[28] M. Alghizzawi, H. Alhanatleh, Z. M. Alhawamdeh, E. Ahmed, and J. A. Al-Gasawneh, "The intersection of digital marketing and business performance," *Migrat. Lett.*, vol. 20, no. 8, pp. 1202–1214, 2023.

[29] Z. Cao, J. Lu, S. Cui, and C. Zhang, "Zero-shot handwritten chinese character recognition with hierarchical decomposition embedding," *Pattern Recogn.*, vol. 107, no. 1, pp. 107488, 2020. doi: 10.1016/j.patcog.2020.107488.

[30] O. Meddeb, M. Maraoui, and M. Zrigui, "Arabic text documents recommendation using joint deep representations learning," *Procedia Comput. Sci.*, vol. 192, no. 1, pp. 812–821, 2021. doi: 10.1016/j.procs.2021.08.084.

[31] M. Awni, M. I. Khalil, and H. M. Abbas, "Offline arabic handwritten word recognition: A transfer learning approach," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9654–9661, 2022. doi: 10.1016/j.jksuci.2021.11.018.

[32] H. A. Al Hamad, "Effect of the classifier training set size on accuracy of pattern recognition," *Int. J. Eng. Res. Dev.*, vol. 12, no. 11, pp. 24–31, 2016.

[33] M. Shehab *et al.*, "A comprehensive review of bat inspired algorithm: Variants, applications, and hybridization," *Arch. Comput. Method Eng.*, vol. 30, no. 2, pp. 765–797, 2023. doi: 10.1007/s11831-022-09817-5.

[34] M. Elkhayati, Y. Elkettani, and M. Mourchid, "Segmentation of handwritten arabic graphemes using a directed convolutional neural network and mathematical morphology operations," *Pattern Recogn.*, vol. 122, no. 3, pp. 108288, 2022. doi: 10.1016/j.patcog.2021.108288.

[35] H. M. Balaha, H. A. Ali, M. Saraya, and M. Badawy, "A new arabic handwritten character recognition deep learning system (AHCR-DLS)," *Neur. Comput. App.*, vol. 33, no. 11, pp. 6325–6367, 2021. doi: 10.1007/s00521-020-05397-2.

[36] M. Rabi, M. Amrouch, and Z. Mahani, "Recognition of cursive arabic handwritten text using embedded training based on hidden markov models," *Int. J. Patt. Recogn. Artif. Intell.*, vol. 32, no. 1, pp. 1860007, 2018. doi: 10.1142/S0218001418600078.

[37] T. M. Ghanim, M. I. Khalil, and H. M. Abbas, "Comparative study on deep convolution neural networks DCNN-based offline arabic handwriting recognition," *IEEE Access*, vol. 8, pp. 95465–95482, 2020. doi: 10.1109/ACCESS.2020.2994290.

[38] M. S. Daoud, M. Shehab, H. M. Al-Mimi, L. Abualigah, R. A. Zitar and M. K. Y. Shambour, "Gradient-based optimizer (GBO): A review, theory, variants, and applications," *Arch. Comput. Method Eng.*, vol. 30, no. 4, pp. 2431–2449, 2023. doi: 10.1007/s11831-022-09872-y.