

CONTOUR-BASED DIACRITICS DETECTION FOR ENHANCED ARABIC TEXT IMAGE PROCESSING

TARIK ABDEL-KAREEM ABUAIN¹

¹College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Saudi Arabia

E-mail: ¹t.aboain@seu.edu.sa

ID 55286 Submission	Editorial Screening	Conditional Acceptance	Final Revision Acceptance
07-08-24	08-08-2024	14-09-2024	21-09-2024

ABSTRACT

Using diacritics in words or letters are not merely additional or optional elements of the language, which make them essential components in many scripts. Altering certain diacritics can significantly affect both the syntax and semantics, potentially changing the word's meaning into a completely different one. Detecting Arabic diacritics accurately remains a critical yet challenging task within the world of document image processing, especially in languages where diacritics significantly impact meaning and pronunciation. However, much research addresses the primary objects (letters) in text and neglects the secondary objects (diacritics) and considers them as noise. Therefore, this research presents a contour-based methodology for diacritics detection (segmentation) to improve the quality and efficiency of image-processing techniques applied to Arabic handwritten texts by exploiting the features that can be extracted from the detected diacritics in both machine written and handwritten text images. The proposed method involves converting text images to grayscale, applying adaptive thresholding to produce binary text images, and employing contour tracing method to isolate diacritic regions. This method was tested on a self-created dataset of over 50 Arabic machine printed and handwritten text images. The results were very promising, where the accuracy rate of the proposed method on the Arabic machine printed text images achieved between 0.97% to 0.98%, while for the Arabic handwritten text images the proposed method achieved between 0.92% to 0.99% accuracy rate depending on the used threshold values per area in the text.

Keywords: *Text Image Segmentation, Diacritics Detection, Contour-Based Methodology, Image Processing*

1. INTRODUCTION

Text image segmentation involves the identification and gathering together of features processing having comparable traits. This handle might incorporate factual clustering, thresholding, identifying edges, recognizing districts, or a combination of these methodologies [1]. Be that as it may, the Arabic dialect has 28 letters, each of which shifts in shape depending on its position inside a word. Most Arabic letters are composed in an unexpected way if they appear up at the starting, center, conclusion, or as a separated character inside the word. Segmentation of the Arabic handwritten written content is very challenging task due to the writing styles variation, the different shapes each character can take based on its position in a word, and the particular writing fashion [2]. Also, the utilize of diacritics includes advance complexity, as losing a single diacritic can change the meaning of a word. Moreover, the Arabic text contains two

fundamental components which are primary and secondary objects as appeared in Figure 1.

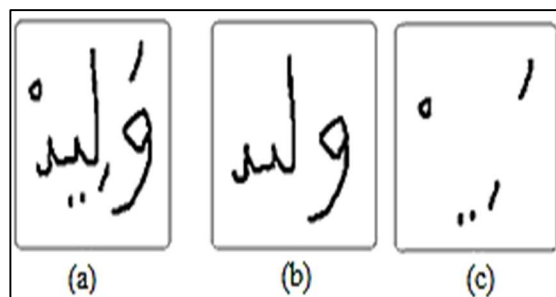


Figure 1: An Example Of Parts Of Arabic Text (A) Original Text (B) Primary Objects [Letters] (C) Secondary Objects [Diacritics].

Diacritics are small strokes added to the top or bottom of a script to extend its duration or attach a short vowel. The Arabic writing system uses eight diacritics to represent phonetic phenomena. Some of these diacritics are phonemic, while others are morphemic and syntactic, indicating the case or state of the word rather than its meaning.

Diacritics play a main role in comprehending the semantics of Arabic texts, especially for a novice Arabic reader who can't easily deduce the diacritics of a word from the context of the text. In the context of script phonology, diacritics are considered the main control item to tune the pronunciation process correctly and accurately. Moreover, diacritics' existence in texts is crucial for the success of many computational linguistic applications, such as automatic text machine translation, automatic document text analysis, automatic speech recognition, and automatic vocalization of written text.

The importance of writing scripts with diacritics lies in the fact that they are used to read and comprehend important texts such as the Qur'an, the Noble Hadith, and Arabic literature and poetry, to void both the misunderstanding of their meaning and the verbal errors. When Arabic script is written without diacritics, a single word may have several possible meanings. On the other hand, when the Arabic script is written with diacritics, the word has a clear meaning (Table 1 shows an example of the word "قَبِلَ" that came in different meanings based on the existing diacritics).

Table 1: An Example Of An Arabic Word "قَبِلَ" That Came In Different Meanings Based On Diacritics.

Arabic Word	Pronunciation	Meaning	Usage Example - English Translation
قَبِلَ	qabala	To accept, to receive	قَبِلَ المدير الطلب. The manager accepted the request.
قَبِلَ	qubila	Was accepted	قَبِلَ الطالب في الجامعة. The student was accepted into the university.
قَبِلَ	qabl	Before (time / place)	جاء أحمد قبل محمد. Ahmed came before Mohammed.
قَبِلَ	qabila	Accepted	قَبِلَ الطالب النصيحة. The student accepted the advice.
قَبِلَ	qibal	Direction, towards	جلست قِبَلَ الكعبة. I sat facing the Kaaba.
قَبِلَ	qubul	In front of, facing	كان القائد واقفاً قِبَلَ الجنود. The leader was standing in front of the soldiers.

Table 2 illustrates the Arabic diacritic system, including the marks used to indicate short vowels, lengthening, and other phonetic details in Arabic script. Three diacritics represent short vowels: Fatha (/a/) and Damma (/u/) are written above the letter, and Kasra (/i/) is written below it. Another three diacritics, known as nunation, indicate nominal indefiniteness in Modern Standard Arabic by adding

/n/ to the short vowel. A special diacritic called Shadda is used to indicate the doubling of a consonant sound and can be combined with any of the previously mentioned diacritics. Lastly, there is the Sukun, a diacritic used to mark the absence of a vowel.

Table 2: The Arabic diacritic system.

Diacritic	Name (Arabic)	Name (English)	Symbol	Pronunciation Example	Example Word (with Translation)
◌َ	فتحة	Fatha	a	Short "a" sound	كَتَبَ (kataba) – "he wrote"
◌ُ	ضمة	Damma	u	Short "u" sound	كُتُبَ (kutub) – "books"
◌ِ	كسرة	Kasra	i	Short "i" sound	كِتَابَ (kitab) – "book"
◌ْ	تنوين الفتح	Fathatan	an	Nasalized "an" sound	كِتَابًا (kitabān) – "a book"
◌ٌ	تنوين الضم	Dammat an	un	Nasalized "un" sound	كُتُبًا (kutubun) – "books"
◌ٍ	تنوين الكسر	Kasratan	in	Nasalized "in" sound	كِتَابِي (kitabīn) – "of a book"
◌◌	سكون	Sukun	silent	No vowel sound	كَتَبَ (katab) – end sound without vowel
◌◌◌	شدة	Shadda	double consonant	Double consonant sound	كَتَبَ (kataba) – "he made someone write"
ا	همزة القطع	Hamzat Al-Qat'	hamza	Always pronounced	أَخَ (akh) – "brother"
أ	همزة الوصل	Hamzat Al-Wasl	Connective Hamza	Pronounced at the beginning of speech but not in the middle	اسْمَ (ism) – "name"

The issue is that current research does not prioritize the secondary foreground marks (diacritics), despite its significant role in authenticating Arabic handwritten documents. Instead, the focus remains primarily on the primary foreground elements like letters and subwords. As a result, the lack of focus on diacritics has led to insufficient research and development in this area, impacting the accuracy of Arabic handwritten text recognition. Given the critical role diacritics play in conveying the correct meaning of words, addressing this research gap is essential for improving the overall effectiveness and precision of these recognition systems. In this paper, we will address this matter using contour-based technique to extract diacritics from Arabic handwritten text images and hence use these diacritics as features.

2. RELATED WORK

The process of enhancing image preprocessing and segmentation for Arabic handwritten documents requires meticulous attention to detail and innovative techniques. Studies have stressed the significance of addressing both the primary text image and the secondary foreground elements, such as diacritics, to achieve the best possible text image quality and processing results. One notable experiment conducted by [3] used region-based approach for diacritics segmentation, producing promising results in this domain. The Watershed algorithm, as employed in this experiment, showcased the potential of region-based techniques in successfully segmenting diacritics in Arabic handwritten documents.

Building on this foundation, a novel framework has been introduced to facilitate diacritics segmentation in Arabic handwritten documents through region-based segmentation techniques. This framework aims to address the unique challenges posed by Arabic script and enhance the accuracy of diacritics segmentation in this context [4].

The segmentation process plays a crucial role in dividing the text image into distinct parts, ranging from lines to individual words or Pieces of Arabic Word (PAW), ultimately contributing to improved text image analysis and processing outcomes [5]. By introducing a reliable segmentation technique tailored specifically for Arabic handwritten script, this research underscores the significance of factors like script height, character width, and pen thickness in optimizing the segmentation process and enhancing the overall quality of image preprocessing and segmentation in

Arabic handwritten documents [6]. Text image segmentation, as a pivotal aspect of this research, holds the potential to revolutionize the analysis and interpretation of Arabic handwritten documents by facilitating more precise and efficient processing methods.

Furthermore, in recent years, there has been a significant increase in interest in diacritics restoration. Researchers have utilized various methods, including Markov models, machine learning, and more recently, deep learning. Nevertheless, the outcome remains incomplete despite these efforts [7]. The Arabic written language contains diacritics symbols, which are crucial for indicating the exact pronunciation of the word [8]. The diacritics in Arabic serve as additional symbols within a letter to differentiate sounds or distinguish between words. These marks are extensively utilized in languages like Arabic, where semantic clarity is vital [4]. In standard Arabic writing, diacritics are not widely employed, and professional readers often infer the meaning of words from the context [9]. Utilizing diacritical marks aids in clarifying the linguistic ambiguity present in the text [10].

Arabic, like Hebrew, is a language abundant in morphology and typically written without diacritics, leading to considerable ambiguity [11]. In Arabic, diacritics serve the purpose of indicating both pronunciation and meanings [12]. Diacritics play a crucial role in determining the meaning of words, as altering diacritics can result in distinct meanings for the same word [13]. In addition, In Arabic, certain diacritics are not allowed to be placed on specific characters. For instance, phonetically, it is prohibited to place a Fatha on the letter "و" [14]. Figure. 1 sums up the importance of diacritic in Arabic language and how it affects the meaning of the word.

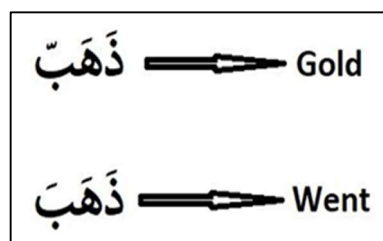


Figure 2: An Example Of Same Word But Different Diacritical Marks And Different Meaning.

However, in this paper, we will emphasize contour-based diacritics segmentation in Arabic handwritten, contour-based tracing technique is used to locate the diacritics and identify them in the text

image. On the other hand, in previous research conducted by [3], a region base technique is used to extract diacritics from Arabic handwritten text image. The region-based segmentation technique categorizes the text image into homogeneous regions according to pixel attributes [15]; however, in the context of diacritics segmentation, it can result in erroneous segmentation by conflating with neighboring characters, and it is also susceptible to noise when documents exhibit varying qualities. While contour-based segmentation directly identifies object boundaries through edge detection or contour tracing techniques. Moreover, in both contour and region methods, local decisions are made. In the contour method, this could involve identifying an edge at a pixel with a high gradient.

In the region method, it might involve merging or splitting regions based on a local, short-term strategy. Region-based techniques are more suitable for defining a global objective function [16]. The contour extraction method offers numerous benefits compared to alternative segmentation techniques, including providing a precise characterization of character shapes and capturing intricate details, especially for small fonts [17]. Contour tracing is susceptible to noise and interruptions in characters. Moreover, it is prone to over-segmentation, particularly when characters consist of multiple components [18]. Machine learning methods may enhance accuracy in diacritic detection; however, they necessitate extensive datasets for model training and prolonged execution times. Furthermore, techniques such as "Template Matching" may be vulnerable to noise and minor discrepancies in diacritic forms.

Therefore, in this research, we will utilize this technique to extract diacritics from machine printed and Arabic handwritten text images, and further investigate its potential application as features in the segmentation and classification processes. This paper aims to address the central research question: "How can contour-based segmentation methods be utilized for precise diacritic detection in Arabic text images?"

3. PROPOSED METHOD

The proposed methodology for detecting diacritics is illustrated here. The method consists of six stages namely as convert to grayscale image, binary image, trace the contours to detect diacritics, find the minimum and maximum area threshold, and finally extract the diacritics as in Figure 3.

The following pseudocode elaborates the process of the proposed method.

1. Image preparation

1.1. Import libraries OpenCV and NumPy

1.2. Load image

2. Image preprocessing

2.1. Convert image to Grayscale

2.2. Convert the image to Binary

2.3. Eliminate noise in image

3. Contours tracing

3.1. Create a blank canvas with the same image size

3.2. Trace text contours in the binary image

3.3. Set the `min_area_threshold` and the `max_area_threshold`

3.4. Loop over the detected contours

3.5. Output the canvas with diacritics

However, the process encompasses preprocessing image such converting image to grayscale and binary to reduce noise and to make it easy for contour to be detected. Moreover, the contours will be filtered based on the maximum and minimum area thresholds to distinguish between small diacritics marks from larger primary objects.

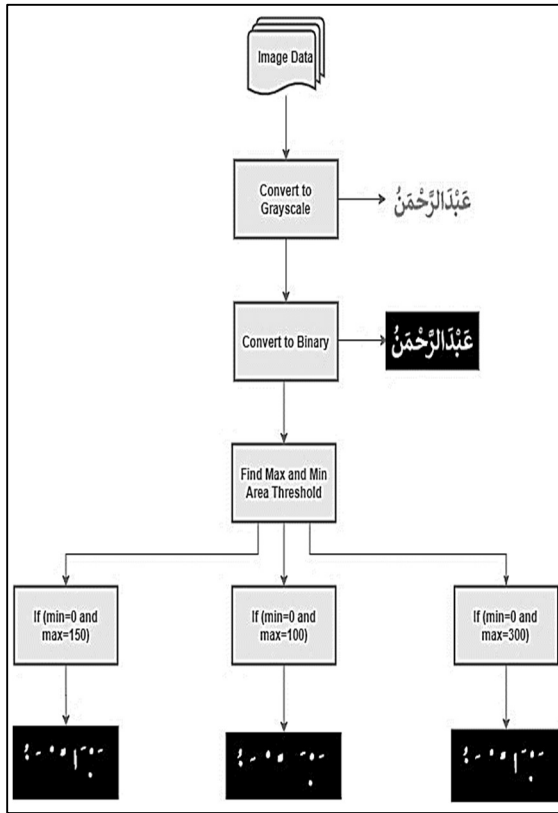


Figure 3: Workflow Of The Proposed Method.

3.1. Image Data

The datasets utilized in this study consist of self-created Arabic machine written and handwritten text images as presented in Figure 4, showcasing diverse writing styles, and containing a wide range of diacritics so that we can conduct the contour-based diacritics detection efficiently.

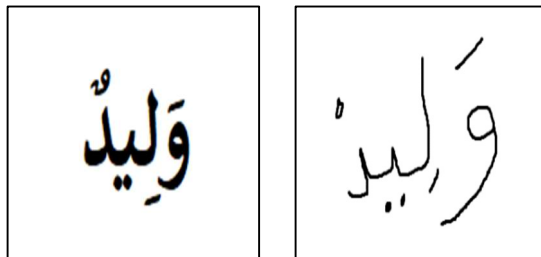


Figure 4: (left) Machine Printed Image, (right) Handwritten Text Image.

3.2. Grayscale Image

Arabic s come with different colors, thus there is a need for conversion to grayscale image to help smooth out the image and reduce the noise. Colorization of images involves the estimation of RGB colors for grayscale images [19] to improve the

quality of the image. Grayscale images are commonly employed as covert media when categorizing the available works into various groups [20] using Eq. 1. Figure 5 shows a sample of the resulting grayscale image.

$$Grayscale = 0.299R + 0.587G + 0.114B... (1)$$



Figure 5: (left) Original Text Image, (right) Grayscale Text Image.

3.3. Binary Image

After the image is converted to grayscale, another preprocessing step will take place which is binarization. Binarization of an image involves transforming an image with single-color qualities into a binary form, where each pixel can only exhibit one of two colors [21] using Eq. 2. This process is significant as it facilitates feature selection, aiding in the identification and extraction of relevant features for various applications [22]. Figure 6 shows a sample of the resulted binary image.

$$Binary(x, y) = 255 * (Grayscale(x, y) > T) \dots (2)$$



Figure 6: (left) grayscale Text Image, (right) Binary Text Image.

3.4. Contour-Based Diacritics Detection and Extraction

In this paper, contour-based method is used to detect diacritics from Arabic handwritten text images, contour-based approach relies on tracing from a starting endpoint and continuously following the character's boundary until reaching the touching point [23]. This step is based on Eq. 3. In addition, we found that contour-based method is also useful

for this research to differentiate primary and secondary objects from Arabic machine printed and handwritten text images. More details about this approach can be found in the result section.

$$findContours (Canny (GaussianBlur (Grayscale (Image)))) \dots(3)$$

Following the application of the contour-based methodology for detection purposes, this section focuses on extracting diacritics from Arabic handwritten text images, thereby achieving the research objective. Figure 7 depicts both the original and extracted diacritics images, showcasing the effectiveness of the segmentation process.



Figure 7: (left) A Sample of Original Handwritten Text Image, (right) Extracted Diacritics.

4. Results And Discussion

In this paper, the proposed method was tested on self-created Arabic machine printed and handwritten text images that have words with diacritics. In addition, more than 50 images were tested by our proposed method. The proposed method has been Implemented using python programming language, also OpenCV functions were used with python code to implement the preprocessing steps and as well as contour-based to detect diacritics in images. Furthermore, the workflow of the proposed method successfully converted the original image to grayscale and applied Gaussian blur to reduce noise.

The adaptive thresholding method effectively generated a binary image suitable for contour detection. By implementing that, diacritic contours were identified and filtered based on area thresholds. For instance, the minimum and maximum thresholding implemented in this experiment may vary between (min=0) and (max=300), depending on the text image.

The proposed method performs as the higher the thresholding area the most likely it is a primary object is detected which is not needed in this research. Conversely, the lower the thresholding

area the most likely diacritics are detected which is the main idea of this paper. Moreover, in this study we will emphasize on three areas threshold which are (0-300), (0-100) and (0-150), and the reason why we choose those areas is because larger than (0-300) are highly likely prone to detect primary objects such as letters.

The accuracy rates for diacritics detection vary between machine printed and handwritten text images across different area thresholds is shown in Table 3. However, for machine printed images, the precision is somehow high, with rates of 0.97 for regions 0–300 and 0–150 and slightly higher at 0.98 for 0–100, accuracy is consistently outstanding and shows strong performance. Handwritten text images, on the other flip, show greater unpredictability: the precision is 0.93 for regions 0 – 300, decreases to 0.92 for 0 – 100, and spikes at 0.99 for 0 – 150, indicating that this threshold best strikes a compromise between complexity and detection accuracy.

All in all, handwritten text has benefit from a specific area range, reflecting the difficulty of handwritten diacritical marks, while machine printed text presents steady precision compared to handwritten. Eq. 4 used for obtaining the precision rates are as follows:

$$Precision = \frac{TP}{TP+F} \dots(4)$$

Table 3: Accuracy rate of the proposed method per area threshold.

Average Precision Rates			
Image Type	Area (0 – 300)	Area (0 – 100)	Area (0 – 150)
Machine Printed	0.97	0.98	0.97
Handwritten	0.93	0.92	0.99

5. ADVANCEMENTS OF THE PROPOSED METHOD

Considering the recent studies, this research contributes to an enhanced contour-based technique for diacritics segmentation. This method's

best benefit is that it uses less resources and is computationally efficient. When compared to recent techniques, the contour-based method offers better diacritics detection by emphasizing on shape and boundary detection, unlike region-based methods, which focus on identifying color or pixel intensity within specific areas, which might cause inaccuracies when diacritics are overlap with main text.

Through the application of the contour-based technique, this study advances the recent findings of Arabic text diacritics segmentation. The main advantage of this approach is that it is computationally efficient and requires fewer resources. Contrary to region-based methods, which focus on finding color or pixel intensity within certain areas and can make mistakes when diacritical marks overlap with main text, contour-based methods focus on finding shape and boundaries, which makes them more accurate than newer methods.

6. LIMITATIONS OF THE PROPOSED METHOD

Contour-based methods, although it is widely used in image processing areas, yet it has limitations when it comes to diacritics detection due to several factors. One major drawback is its sensitivity to noise and variation in diacritic sizes and shapes which leads to inconsistent results.

Arabic diacritics usually appear as a small feature which can be difficult to isolate them precisely using contour-based, especially in handwritten text where strokes overlap. Furthermore, contour-based approach normally relies on predetermined thresholds values, making them less flexible to the diverse characteristics of diacritics across different writing styles. Table 4 shows sample results of the proposed method tested on both machine printed and handwritten text images. For instance, there are only three thresholds' values and based on these predefined values diacritics will be extracted.

On top of that, going beyond these thresholds' areas like (0-400) might lead to crossing the contours and big Arabic letters will be detected. All in all, despite the fact contour-based technique present flexibility in image processing tasks, its drawbacks make it less appropriate for accurate and solid diacritics detection in both machine printed and handwritten text images.

Thus, fine-tuning these parameters could improve performance for different image sets.

Moreover, the quality and resolution of the input images significantly affect the outcomes. Higher resolution images may require different processing parameters.

7. CONCLUSION AND FUTURE WORK

In this paper, a contour-based method for diacritics detection for Arabic text images is implemented. Moreover, this method is tested on several images that contain many diacritics marks and it has successfully achieved the objectives of this paper by extracting diacritics features. Furthermore, the proposed method has achieved a very promising result when it comes to accuracy rates, where machine printed text images have achieved between 0.97% to 0.98%, whereas for the Arabic handwritten text images have achieved between 0.92% to 0.99% accuracy rates.

Although the proposed method provides significant improvement, several issues remain unsolved. Firstly, the technique is still somehow sensitive to extreme noise and overlap handwritten documents. In addition, the reliance of handcrafted area thresholding may limit the method's scalability when it comes to larger and more diverse datasets.

To conclude, despite the fact contour-based has limitations and inaccuracies in terms of detecting diacritics in both machine written and handwritten text images due the predetermined threshold values, yet it has better opportunity in the future to be enhanced for further research and experiments to become more precise.



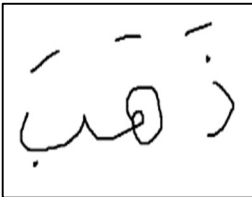


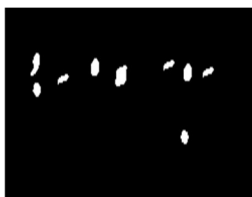
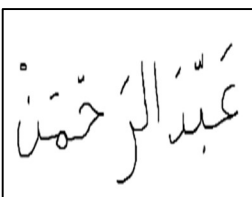


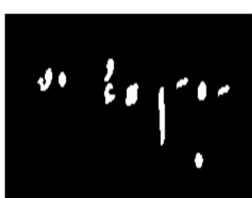
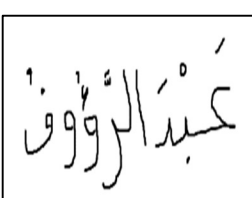
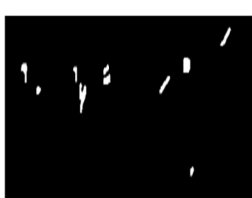
In the future, an investigation of the integration of contour-based and machine learning models will be conducted to develop a hybrid approach that merges the efficiency of current methods with the dependability of deep learning. For instance, they could employ Convolutional Neural Networks (CNNs) to enhance contour-based diacritic detection, thereby improving its performance with more intricate document variations.

REFERENCES

- [1] A. A. Sheikh, M. S. Azmi, W. A. K. Abuain, and M. A. Aziz, "Segmentation techniques for Arabic handwritten: a review," *Int. J. Electr. Comput. Eng.*, vol. 14, no. 2, pp. 1834–1841, 2024, doi: 10.11591/ijece.v14i2.pp1834-1841.
- [2] A. Ali, A. Ali, and M. Suresha, Survey on Segmentation and Recognition of Handwritten Arabic Script, *Springer Singapore*, July 2020.
- [3] A. A. Sheikh, M. S. Azmi, M. A. Aziz, M. N. Al-Mhiqani, and S. S. Bafjaish, "Diacritic

- segmentation technique for Arabic handwritten using region-based,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 18, no. 1, pp. 778–784, 2020, doi: 10.11591/ijeecs.v18.i1.pp478-484.
- [4] A. A. Shiekh, M. S. Azmi, M. A. Aziz, M. N. Al-Mhiqani, and S. S. Bafjaish, “Framework of diacritic segmentation for Arabic handwritten document,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 24, no. 2, p. 1001, 2021, doi: 10.11591/ijeecs.v24.i2.pp1001-1008.
- [5] C. N. Network et al., “*Journal Pre-proof*,” 2021.
- [6] M. A. Fadeel, “An efficient segmentation algorithm for Arabic handwritten characters recognition system,” *Proc. - 2016 3rd Int. Conf. Math. Comput. Sci. Ind. MCSI 2016*, pp. 172–177, 2017, doi: 10.1109/MCSI.2016.28.
- [7] M. A. H. Madhfar and A. M. Qamar, “Effective Deep Learning Models for Automatic Diacritization of Arabic Text,” *IEEE Access*, vol. 9, pp. 273–288, 2021, doi: 10.1109/ACCESS.2020.3041676.
- [8] Y. Alalawi, D. M. Chandler, and N. R. Caluya, “A CNN-Based Arabic Diacritic Symbol Recognition System Using Domain Adaptation,” *ACM Int. Conf. Proceeding Ser.*, pp. 23–32, 2023, doi: 10.1145/3626641.3627212.
- [9] A. A. A. Ali, M. Suresha, and H. A. M. Ahmed, “A Survey on Arabic Handwritten Character Recognition,” *SN Comput. Sci.*, vol. 1, no. 3, 2020, doi: 10.1007/s42979-020-00168-1.
- [10] S. S. Bafjaish, A. Ramzani, M. Nasser, M. Sanusi, and H. Mahdin, “Skew Detection and Correction of Mushaf Al-Quran Script using Hough Transform,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 8, pp. 402–409, 2018, doi: 10.14569/ijacsa.2018.090852.
- [11] A. Shmidman, J. Guedalia, S. Shmidman, M. Koppel, and R. Tsarfaty, “A novel challenge set for Hebrew morphological disambiguation and diacritics restoration,” *Assoc. Comput. Linguist. ACL EMNLP 2020*, pp. 3316–3326, 2020, doi: 10.18653/v1/2020.findings-emnlp.297.
- [12] S. Alqahtani, A. Mishra, and M. Diab, “A multitask learning approach for diacritic restoration,” *Proc. Annu. Meet. Assoc. Comput. Linguist.*, pp. 8238–8247, 2020, doi: 10.18653/v1/2020.acl-main.732.
- [13] A. Qaroush, A. Awad, A. Hanani, K. Mohammad, B. Jaber, and A. Hasheesh, “Learning-free, divide and conquer text-line extraction algorithm for printed Arabic text with diacritics,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 9, pp. 7699–7709, 2022, doi: 10.1016/j.jksuci.2022.04.021.
- [14] M. A. A. Rashwan, A. A. Al Sallab, H. M. Raafat, and A. Rafea, “Automatic Arabic diacritics restoration based on deep nets,” *ANLP 2014 - EMNLP 2014 Work. Arab. Nat. Lang. Process. Proc.*, no. October, pp. 65–72, 2014, doi: 10.3115/v1/w14-3608.
- [15] A. Z. Atiyah and K. H. Ali, “Brain MRI Images Segmentation Based on U-Net Architecture,” *Iraqi J. Electr. Electron. Eng.*, vol. 18, no. 1, pp. 21–27, 2022, doi: 10.37917/ijeec.18.1.3.
- [16] J. Malik, S. Belongie, T. Leung, and J. Shi, “Contour and texture analysis for image segmentation,” *Int. J. Comput. Vis.*, vol. 43, no. 1, pp. 7–27, 2001, doi: 10.1023/A:1011174803800.
- [17] K. Mohammad, A. Qaroush, M. Washha, and S. Agaian, “Contour-based character segmentation for printed Arabic text with diacritics”, August 2019, doi: 10.1117/1.JEI.28.4.043030.
- [18] A. Qaroush, A. Awad, M. Modallal, and M. Ziq, “Segmentation-based , omnifont printed Arabic character recognition without font identification,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3025–3039, 2022, doi: 10.1016/j.jksuci.2020.10.001.
- [19] S. Anwar, M. Tahir, C. Li, A. Mian, F. S. Khan, and A. W. Muzaffar, “Image Colorization: A Survey and Dataset,” pp. 1–20, 2020, [Online]. Available: <http://arxiv.org/abs/2008.10774>.
- [20] N. Subramanian, O. Elharrouss, S. Al-Maadeed, and A. Bouridane, “Image Steganography: A Review of the Recent Advances,” *IEEE Access*, vol. 9, pp. 23409–23423, 2021, doi: 10.1109/ACCESS.2021.3053998.
- [21] G. Ramesh, J. Logeshwaran, and K. Rajkumar, “The smart construction for image pre-processing of mobile robotic systems using neuro fuzzy logical system approach,” *NeuroQuantology*, vol. 20, no. 10, pp. 6354–6367, 2022, doi: 10.14704/nq.2022.20.10.NQ555629.
- [22] E. S. M. El-Kenawy, M. M. Eid, M. Saber, and A. Ibrahim, “MbGWO-SFS: Modified Binary Grey Wolf Optimizer Based on Stochastic Fractal Search for Feature Selection,” *IEEE Access*, vol. 8, pp. 107635–107649, 2020, doi: 10.1109/ACCESS.2020.3001151.
- [23] I. Ullah, M. S. Azmi, M. I. Desa, and Y. M. Alomari, “Segmentation of touching Arabic characters in Handwritten documents by overlapping set theory and contour tracing,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 5, pp. 155–160, 2019, doi: 10.14569/ijacsa.2019.0100519.

Table 4: Samples Of Arabic Machine Printed And Handwritten Text Images Before And After The Diacritics Detection

Image Type	Original Image	Area Threshold	Extracted Diacritics	Description
Machine Printed		(Min=0, Max=300)		5 diacritics in this image are fully extracted
Handwritten				4 diacritics in this image are extracted, with 1 primary object detected which is “ذ”
Machine Printed		(Min=0, Max=100)		9 diacritics in this image are fully extracted
Handwritten				9 diacritics in this image are extracted, with 1 primary object detected which is “ر”
Machine Printed		(Min=0, Max=150)		8 diacritics in this image are extracted, with 2 non-diacritic objects detected which are “ء” “ا”
Handwritten				9 diacritics in this image are extracted, with 1 non-diacritic object detected which is “ء”