ISSN: 2710-2165

Improved Technique in Arabic Handwriting Recognition

Ammar A. Al-Hamadani [1], Maad Kamal Al-Anni [2], Gamil R. S. Qaid [3], Najran Nasser Hamood [4]

[1]Faculty of Engineering and Built Environment, University Kebangsaan Malaysia , Bangi, Malaysia
Email: ammar.aladin@ukm.edu.my

[2]Aix Marseille University, CNRS, ENSAM, Universit´e De Toulon, LIS UMR 7020, 13397 Marseille, France
Email: alannimaad@gmail.com

[3]Faculty of Computer Science and Engineering, Hodeidah University, Al Hudaydah, Yemen
Email: dr.g_qaid@hoduniv.net.ye

[4] Faculty of Computer Science and IT, Sana'a University, Sana'a, Yemen
Email: n.aldawla@su.edu.ye

Article History

Received: Mar. 02, 2025

Revised: May 27, 2025

Accepted: Jun 11, 2025

Abstract

Arabic handwriting recognition has significant applications in fields like postal sorting, handwritten text identification, and cheque processing. The process involves several steps: preprocessing, feature extraction, and classification. Preprocessing enhances image quality through noise removal, normalisation, and binarisation, which are essential for accurate segmentation. Feature extraction captures key information such as stroke direction and spatial relationships, which are crucial for distinguishing between different characters. Hybrid methods, statistical features, and structural features are typical feature extraction strategies. Next, classification methods such as K-nearest neighbour and Support Vector Machines are employed to categorise the extracted features into predefined classes. The effectiveness of Arabic handwriting recognition systems depends heavily on the quality of feature extraction, which directly impacts recognition accuracy. Researchers have explored various techniques, including structural and statistical feature extraction, to optimise these systems. Exceptional accuracy rates are achieved through the utilisation of the proposed SVM linear kernel and KNN classifier with 99.64% and 97%, respectively.

Keywords- Convolutional Neural Networks (CNN), Classification, Handwritten Text Identification, Feature Extraction, Recognition Challenges, Support Vector Machines (SVM).

I. INTRODUCTION

Recognising Arabic characters has many functional applications in fields such as reading postal addresses for the purpose of sorting out mails, identifying words on a handwritten text pages and recognition of cheques, etc. For realizing characters or handwritten words, there are a number of approaches in computational pattern recognition, such as statistical methods, as in K-Nearest Neighbor KNN and networks of artificial neural. Normally, handwriting is cursive stems from different elements which are the paper's quality, the writer's style and geometric elements operated by the writing prerequisite Its shape and tracing quality are quite erratic [1].

A suggested approach for Arabic handwriting recognition involves a combination of preprocessing, feature extraction, and classification techniques. In preprocessing, methods like noise removal, normalization, and binarization can help enhance the quality of handwritten images, reducing variability and improving segmentation accuracy. The goal of feature extraction is to retrieve pertinent data from the input images, such as stroke direction, curvature, and spatial relationships between strokes, to enable discrimination between different characters. Finally, classification algorithms, such as SVM or CNN, are employed to classify extracted features into predefined character classes [2].

Feature extraction plays a crucial role in Arabic handwriting recognition systems, as it determines the discriminative power of the extracted features and directly impacts recognition accuracy. Various feature extraction techniques have been proposed for Arabic script, including structural features, statistical features, and hybrid approaches combining multiple feature types. Experimentation with different feature sets and classification algorithms is essential to identify the most effective combination for accurate Arabic handwriting recognition [3].

This paper aimed to develop robust and accurate handwriting recognition systems tailored to Arabic language applications. A multistage system is developed that includes text scanning, preprossing, classification, and post processing. In the subsequent sections, we will explore Arabic Character Characteristics, delve into the related work on the Handwriting Recognition System, discuss Feature Extraction methodologies, and present Experimental Outcomes and Discussions to assess the effectiveness of the suggested strategy.

II. RELATED WORKS

A team of researchers works with the Arabic handwritten character recognition AHCR system and collects various results. For preprocessing, numerous investigators used image thinning as a sequence code and to distinguish the Arabic letters. Authors have, in [2], suggested a two-stage transfer learning method employing ResNet-18 for offline handwritten word recognition. The method produced an excellent accuracy of 96.11% with the IFN/ENIT dataset, outperforming baselines such as AlexNet (83%) and ResNet-18 trained from scratch with an accuracy of 92.7%. Transfer learning from the ImageNet and AlexU-W data improved word recognition for misclassified words up to 35.45%. The method, nevertheless, is dependent upon heavy fine-tuning and progressive resize, which makes training time-consuming and complicated. Despite this, the method solves effectively the problem of low-resource and unbalanced data in recognizing handwriting in Arabic. In [3], the authors suggested an Offline Isolated Arabic Handwriting Character Recognition (OIAHCR) model with Support Vector Machine (SVM) and several feature extraction algorithms. Their model obtained an excellent classification accuracy of 99.64% with an SVM linear kernel over their own dedicated dataset of 560 handwritten character images. The method incorporated structural, statistical, and global transform features such as HOG and DCT. However, the assessment of the system is dependent upon a relatively limited and balanced dataset, which could restrict its validity to diverse or real-world environments. Even so, the use of intensive preprocessing and feature engineering resulted in higher accuracy compared with earlier methods. Authors in [4] created an Arabic handwritten text recognition model with a hybrid ResNet-BiLSTM-CTC architecture trained with KHATT and AHTID/MW datasets. The model obtained a character error rate (CER) of 13.2% and word error rate (WER) of 27.31% on KHATT, and increased to 6.6% CER and 17.42% WER on AHTID/MW with 3 layers of BiLSTM. Visual features were successfully extracted with ResNet, contextual dependency was captured with BiLSTM, and with the addition of the 3-gram language model, refinement in prediction ensued. This model outperformed earlier models such as MDLSTM and Kaldi in both datasets. One limitation is that the model's training increased in complexity with high GPU memory usage. For obtaining these features, moment functions were used by as in [5], the authors suggested an integrated Arabic handwritten word recognition system (SWT-WRS) from the Stationary Wavelet Transform (SWT) to address translation invariance deficiencies of Discrete Wavelet Transform (DWT). On the IFN/ENIT dataset with Gaussian SVM, their system ranked at an optimum recognition rate of 94.81%, higher than the highest achieved with the DWT-based system at 91.61%. The SWT-WRS ranked higher than alternative methods with k-NN (92.60%) and ANN (89.54%) classifier methods. The method does, however, present computational redundancy owing to SWT's non-decimated property, raising feature dimensionality and processing time. This aside, the SWT-WRS showed high generalizability with different classifiers and wavelet families. However, other researchers in [6], proposed a column scheme based LDNP method for hand-written digit recognition with the help of derivative Gaussian and Kirsch masks. The method obtained a best precision of 96.64% on the CVL data set, which significantly outperformed a number of current techniques. Further recall increased to 95.84% when feature combinations obtained by uniform grid sampling were applied. The last combined feature set achieved the optimal recall value of 96.59% on a feature vector of 4,088 dimensions. SVM was applied for classifying 10 digit classes. Its downside lies in the computational burden of the approach, given the high dimensional feature vectors and multiple mask operations. Authors in [7] suggested a CNN model with VGG16 for handwritten Devanagari character classification with the use of transfer learning. A new dataset with 92,000 images of 46 characters was created. It obtained an excellent accuracy of 96.58%, which outperforms earlier methods such as slice-based classification (82.12%). This proves the power of deep features and the use of transfer learning. Its high computational requirement is one limitation. Whereas other researchers employ horizontal and vertical profile of the projection as in [8], the authors implemented a CNN model that incorporated batch normalization and dropout techniques. This approach was tested using the MNIST dataset, a standard benchmark for digit recognition tasks. The model achieved an impressive accuracy of 99.4%, reflecting its strong performance and ability to generalize well. These results highlight the effectiveness of combining deep learning with regularization strategies. However, one drawback is that the model might struggle to scale efficiently when applied to more complex datasets without further refinement.

In the classification stage, [9] introduced an architecture for handwritten character recognition in Arabic using the AHCD dataset of 16,800 images. The model's accuracy stands at 97.2% and jumps to 97.7% when data augmentation is employed. The architecture comprises three dense and six convolutional layers, trained with dropout and L2 regularization. The present architecture outperformed all earlier models when tested with the same dataset. Its modest improvement after employing data augmentation indicates possible overfitting or saturation. Similarly, in [10], the authors presented an improved GAN model, which is especially suitable for the challenging task of handwritten text recognition, especially for visually impaired people. Their approach combined style embeddings with transfer learning and CNNs, allowing for the generation of synthetic text that is very realistic. The model achieved an incredible accuracy of 99% and a validation loss of 0.01, outpacing the traditional systems to a great extent. Style-aware conditioning and dual-path discriminator architecture enhanced authenticity and contextual identification. Nevertheless, the principal limitation of the model is training instability and vulnerability to mode collapse, which are typical issues with GAN architectures.

III. METHODOLOGY

1. Arabic Character Characteristics: There are more than a billion speakers of Arabic worldwide in different aspects of their life, such as religious matters or on their daily basis's activities [11]. The Arabic language can be written in cursive, from right to left. It has twenty-eight basic letters and eight diacritics [11]. Each letter can have various shapes depending on its place in the word, as in Table 1. In addition, Various fonts are available that lead to make changing the shape of Arabic characters dramatically [12].

TABLE I. Arabic Characters.

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

Table 1 demonstrates the twenty-eight characters and their different forms. There are a number of forms for each character based on where it appears in the word. Every letter is penciled in an insulated shape when it is written separately and when it is written in conjunction with other characters in the word, it can take on up to three distinct forms. For example, the letter Ain has four different shapes: Separated form (ع) and Initial, Median, and Last shapes (ع ـعـ عـ), respectively starting from the right side [13].

2. Dataset: **t**he dataset utilized in this study was created with an organized data collection method aimed at capturing variant handwriting in the script of the Arabic language. A total of 560 handwritten character images, consisting of 20 instances for each of the 28 letters of the Arabic script, were acquired. The data was obtained from 20 individuals, aged from 18 to 45, with diverse educational attainment from high school to postgraduate. Both male and female participants and individuals with different handwriting habits and motor skills were included to ensure that the dataset would contain a broad range of natural script formation variability. The participants were all asked to write the letters in a black ballpoint pen on A4-sized white sheets of paper under controlled circumstances. Writing exercises were done in well-illuminated indoor areas to ensure consistent image quality. Participants sat at tables and were asked to write as they normally would without size or direction restrictions, so that the data were an accurate reflection of intra- and inter-writer variation. The sheets were scanned in their completed form at 300 DPI resolution with the use of a flatbed scanner, and characters were cropped manually and labeled. This method ensured that each image maintained essential features like stroke thickness, curvature, slant, and spacing. Figure 1 demonstrates the outcome results gathered from the experiment.

ISSN: 2710-2165

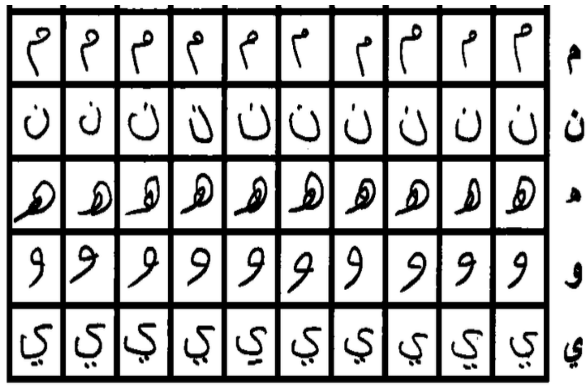

Figure 1. Samples from the suggested dataset.

3. Suggested Recognition System for Handwriting: The suggested approach for AHCR consists of several main steps. Each of these step's impact on performance, accuracy and recognition of the system. Firstly, the input images were converted into grayscale using many procedures as illustrated in Figure 2.

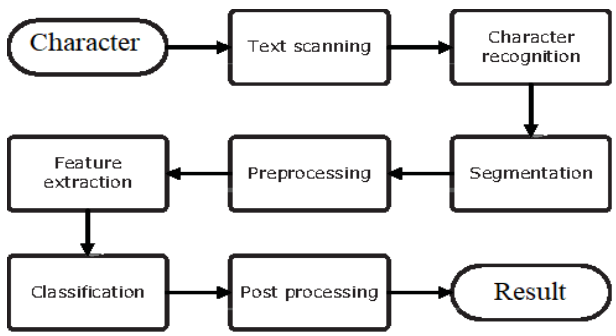

Figure 2. Flow diagram for the suggested system.

The proposed approach consists of several processes that can be summed up as feature extraction, preprocessing, classification, and recognition. In addition, each one of these steps has its impact and benefits towards implementing the recognition process in an efficient way. Here, the suggested procedure is described below in detail:

3.1. Preprocessing: Preprocessing plays a significant role in the AHCR because of its effectiveness in influencing the recognition's outcome. Several actions should be involved in the phase of preprocessing that make the suggested approach achieve a high accuracy [14]. Figure 3 shows the primary preprocessing procedures.

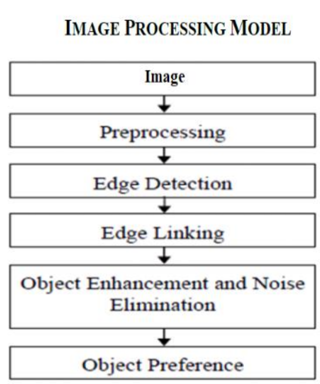

Figure 3. Main steps of the pre-processing.

3.2 Image Noise Reduction and Thresholding: For the AHCR, feedback comes in the form of RGB content images containing Arabic words. Converting these RGB images to grayscale is a crucial initial step before any processing begins. The suggested approach prioritises image thresholding as the first step. This process involves converting the grayscale input image into a binary format. By doing so, the image's dimensionality is minimised, leading to a reduction in processing time. This step is essential for optimising the efficiency of the recognition system, as it simplifies subsequent operations by focusing on the essential elements of the image. By converting the image to binary, the system can more effectively identify and analyse the distinct features of the Arabic characters, leading to more accurate recognition results. Overall, this preprocessing step plays a vital role in streamlining the recognition process and enhancing the system's overall performance.

Figure 4. Image Thresholding.

In this paper, the thresholding technique created depending on Fuzzy C-Means clustering (FCM) which was suggested in [15], was utilized to convert the input grayscale image into binary. Next, a median filter was used to eliminate unwanted components from the binary image, as shown in Figure 4.

3.3 Eliminate the Black Space: The next step of the proposed method is to remove the undesirable black space in the background of the image. The black space can be represented by the surroundings of the character, which can negatively influence the result of the feature extraction. The suggested method for eliminating the black space depends on applying the Bounding Box tool in MATLAB. Firstly, the edge value is calculated from all character directions until another character represented by (1) appears, as demonstrated in Figure 5.

Figure 5. Computing of (0) values.

After doing that, the resultant minimum distance of the outcomes is saved it for the all-borders directions. Secondly, it is a Bounding Box drawing around the saved distances points and cropping the image from these points [16]. The outcome of removing the black space technique is shown in Figure 6.

3.4 Image Thinning& Normalization: The procedure to minimize image’s size to a compact size and finding the medial axis which describes as a group of pixels S where these pixels having an equivalent distance from the boundary pixels around it, and the yield of this method is a skeleton for the word writing by hand. This method must protect the symmetry and the links amongst the letters and the place of the original letter [17, 18], focused on the border line of pixels, eliminating recursively obtaining into consideration maintaining the geometry, connections and the location. The thinning image is applied to extract the structural features in the suggested system. Figure 7 shows the findings from using the skeleton image technique suggested in [18]. The suggested Arabic dataset has different sizes of image. It is vital to produce every image in the database has a comparable size, thus the recognition process may be completed quickly. Following analysis of various sizes, the 64*64 yielded the highest recognition rate. The entire images from the collected data have been normalized into a size of 64*64. Figure 7 shows an example of this thinning and normalization.

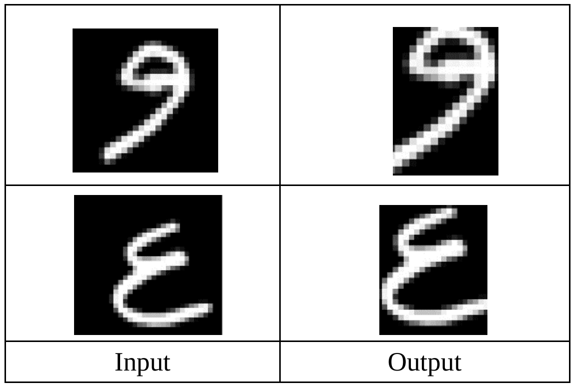

Figure 6. The result of the technique to eliminate dark spaces.

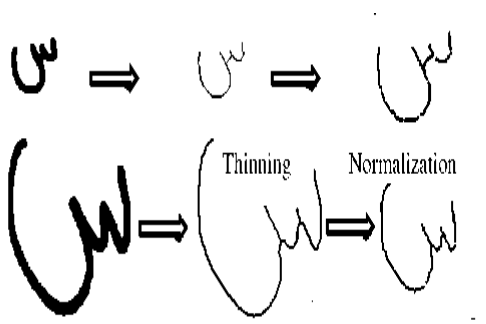

Figure 7. The result of eliminating the black space method.

3.4 Feature Extraction: The most crucial step in AHCR. A robust feature extraction procedure is necessary for the best recognition. Many of feature extraction processes have been suggested for recognition purposes, even though three primary categories of characteristics can be recognized that can be derived from the character images.

3.4.1 Structural Features: By elucidating a pattern's local and global aspects, structural features determine its topological and geometrical features. The structural characteristics depend on the type of pattern that needs to be classified [19]. The features of Arabic letter scan include dots, zigzags, loops, intersection points, end points and strokes in several directions. Preprocessing phases create three types of images. The thinned image is one of these kinds that is used to extract structural characteristics. In AHCR, a number of structural features have been obtained that can be summarized as: intersection points, loops, end points and dots as demonstrated in Figure 8.

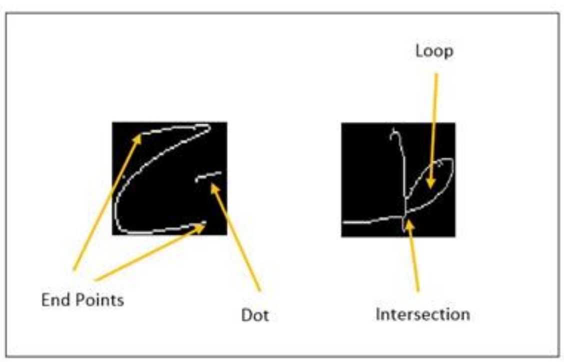

Figure 8. Structural Feature Extraction.

3.4.2 Statistical Features: Statistical features are quantitative metrics computed over images or image areas. Examples of these measurements are pixel densities, Fourier descriptors, histograms of chain, moments, and code directions [20]. Statistical features are simple to be computed and texted independently. In the suggested method, two categories of statistical features have been utilized, namely:
A. The feature of connected components: Arabic characters come in a variety of shapes. Different shapes have different numbers of pixels (segments) that are joined together, as demonstrated in Figure 9. The concept behind connecting components is to perform a left-to-right scan of the whole image to discover the connected pixel groups (8 linked neighbors). Next, every group of connected

pixels will be assigned a label number. As a result, the number of linked components is the feature that is obtained through this approach. This process is beneficial in Arabic letters, since there are a number of letters that have various numbers of connected elements.

One- Segment Class (11 Character)
ا ح د ر س ص ع ل م ه و
Two- Segment Class (15 Character)
ب ت ث ج خ ذ ز س ض ط غ ف ق ك ن
Three- Segment Class (4 Character)
ت ظ ق ي
Four- Segment Class (2 Character)
ث ش

Figure 9. Categories of Arabic Characters.

The feature of connected elements obtained from the binary image extracted from the prior stage and figure 10 illustrates how different colors of rectangles are drawn across every component in the binary image.

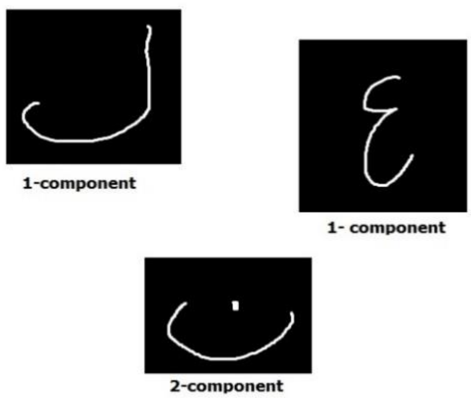

Figure 10. Linked Components Feature (zoom in to see details).

B. Zooming Features: In partitioning features, the image is split up into several regions and a specific feature is obtained from each region. Several characteristics discovered throughout this method improve recognition accuracy. An image from the previous step that features clever edge detection is used in this operation. Firstly, four zones are separated in the image as shown in Figure 11 then, the diagonal pixels summation for each zone was computed as a characteristic of that area. Secondly, the image splits into (16) horizontal and vertical blocks as shown in Figure 12, later on block features are determined by adding up all of the block's pixels.

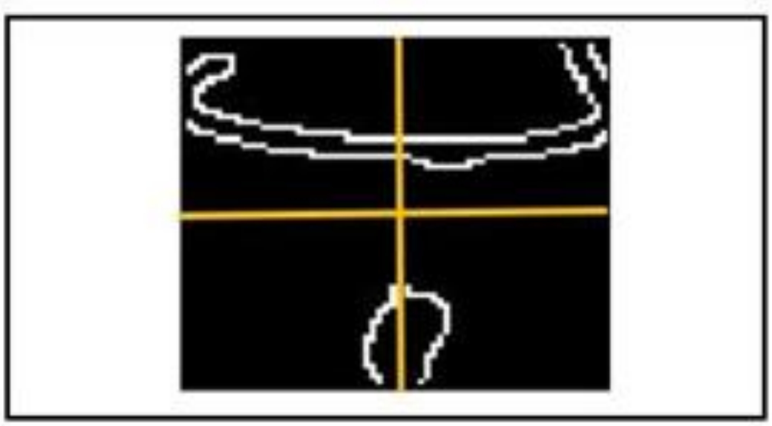

Figure 11. Split image into four zones.

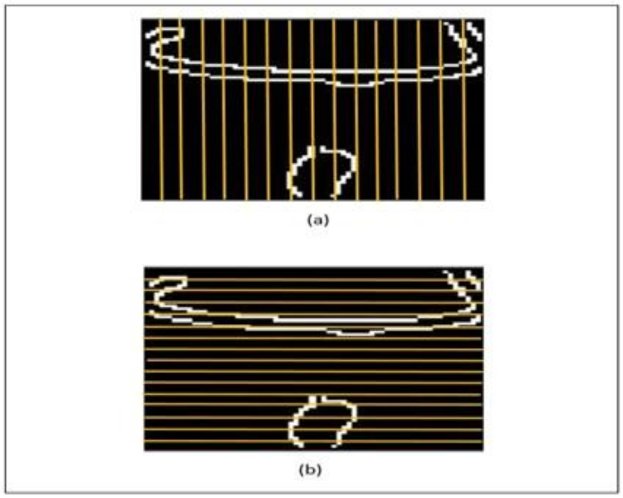

Figure 12. (a) 16-vertical zones; (b) 16-horizontal zones.

3.5 Global Transformation: The transforming schemes change the transformation of pattern's pixels to a further compacted form which decreases the features' dimensionality [21]. The proposed global transformation consisted of the following techniques:

3.5.1 Meshing technique to divide a region: In order to extract statistical characteristics from handwritten samples, a character picture is usually divided into smaller sections using the meshing approach, such as 8×8. This is achieved by partitioning the character image area with imaginary grids. Linear meshing is produced if the grids equally split the character pictures, as shown in Figure 13(a). However, due to significant variations in handwriting styles, linear meshing may not always yield optimal results. To address this issue, both local and global elastic meshing are applicable [22], as shown in Figures 13(b) and (c). These meshing techniques offer a distribution of black pixels throughout each row and column, either locally or globally. With elastic meshing producing a more uniform histogram. This uniformity helps shape and standardize the handwritten character images, making the method more resilient to local deformations and stretching caused by different handwriting styles.

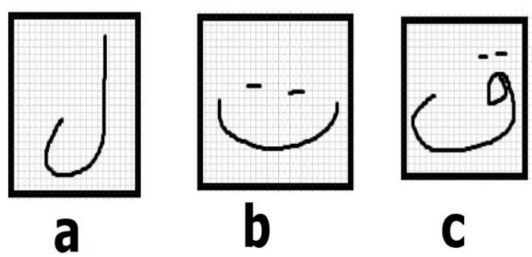

Figure 13. Global elastic meshing and linear meshing: (a) 8×8 linear meshing; (b) 8×8 global elastic meshing; (c) 8×8 local elastic meshing.

3.5.2 8-directional feature extraction: The characteristic with eight directions [23] is frequently employed in HCCR due to its remarkable ability to represent stroke patterns. This method uses eight 2D directions to infer features from every point on the online trajectory, leading to the creation of eight corresponding pattern images. It can be expanded to sixteen directions or simplified to four, but eight provides a compromise between storage needs and accuracy. Extracting the 8-directional features involves multiple steps applied to the input handwritten character sample. These steps consist of linear normalization, the introduction of fictitious strokes, and a 64x64 normalized online character sample is the result of nonlinear shape normalization, equidistance resampling, and smoothing. For every location along the online trajectory, the eight-directional features are calculated, leading to the generation of eight directional pattern images. In the end, a 512-dimensional raw characteristic vector is produced. The steps to obtain the 8-directional features are outlined below:

1. Linear normalization: The initial character paths are resized to a constant 64x64 pixel size using linear mapping that preserves the aspect ratio.

2. Addition of invariant feature: Invariant techniques are applied to each consecutive pair of strokes.

3. Resampling: The points in the character are resampled into a series of points spaced equally apart.

4. Smoothing: A mean filter is applied to smooth the trajectory sequence.

5. The extraction of all directional vectors in which each trajectory point Pj is the key point and Vj is the direction vector, defined in Eq. 1:

$$\vec{V}_j = \begin{cases} \overrightarrow{P_j P_{j+1}}, \; if\ P_j\ is\ a\ start\ point \\ \overrightarrow{P_{j-1} P_{j+1}}, \; if\ P_j\ is\ a\ internal\ point \\ \overrightarrow{P_{j-1} P_j}, \; if\ P_j\ is\ a\ end\ point \end{cases} \quad (1)$$

6. Projection: An 8-D The direction code is created at each location along the trajectory by projecting the directional vector Vj onto two of the eight-directional axes.
7. Blurring: The 8x8 subblocks of the character image are created using either elastic or fixed meshing. A 512-dimensional feature is created within each subblock by applying a Gaussian filter to blur the 8-D direction codes.
8. Transformation: To improve the alignment of the derived feature vector's distribution with a Gaussian distribution, a variable transformation y=x0.5 is applied to each element.

3.5.3 Invariant Feature Extraction: One of the main features of the Scale-Invariant Feature Transformation (SIFT) technique is scale invariance [24]. In order to achieve scale invariance, SIFT uses a Laplacian pyramid, as shown in Eq. 2, which originates from the difference of Gaussian (DoG) function at different levels.

$$D(x, y, \delta) = (G(x, y, \delta k) - G(x, y, \delta)) * I(x, y) \quad (2)$$

Where:

$$G(x, y, \delta) = \frac{1}{2\,\delta^2} exp\left[-\frac{x^2 + y^2}{2\,\delta^2}\right] \quad (3)$$

Since features in an image primarily lie on these regions in Eq. 4, it is easy to retrieve high-frequency information about the image using the Laplacian pyramid L as depicted in Eq. 3.

$$L(x, y, \delta) = G(x, y, \delta) * I(x, y) \quad (4)$$

A comparable down sampling of the Gaussian image by a factor of two occurs after each octave. The number of octaves is essential for identifying key points across various scales, with the octave count and scale based on the original image's dimensions.
Every pixel value in I(x,y)I(x, y)I(x, y) is compared to eight nearby pixels at each DoG octave level in order to identify key locations. Additionally, their values at the retrieved spots are compared with neighboring pixels in the layers above and below. However, in the first and last scales, the lack of sufficient neighboring pixels limits the ability to identify local minima or maxima. Each relevant point's position, scale, and minimum or maximum value are recorded as a result of this examination.
Finding important spots in an image does not ensure that they will all be used in the way that has been suggested. It is necessary to discard unnecessary feature points, particularly those arising in regions with low contrast or imperfectly localized along borders. As a result, each key point's location, scale, and orientation are ascertained using the extremal points that were obtained via the DoG space search. In order to express orientation data, a key-point descriptor is constructed to guarantee a uniform orientation for every key point based on local image properties. Equation 5 provides a mathematical representation and assignment of the orientation of the identified important points. A region around each key point is selected, with the region's size represented within a circle centered at the key point, mathematically illustrated in Eqs. 5 and 6.

$$B(x, y) = \sqrt{(L(x+1, y) - L(x-1, y)^2) + (L(x, y+1) - L(x, y-1))^2)} \quad (5)$$
$$\Theta(x, y) = \tan^{-1}\big((L(x, y+1) - L(x, y-1)\big) / L(x+1, y) - L(x-1, y))) \quad (6)$$

The next stage is to define the image descriptor, which is an exclusive characteristic of the image that contains all the details about the retrieved important spots. Key points extracted from the binary image and the image mask at the same locations are analysed. Additionally, these key points must be situated within the extremal region. Invariant feature qualities are a distinguishing characteristic of high-quality images; they should not be impacted by outside disruptions. The external key points are important since they define the image's unique characteristics.

3.5.4 LDA-based feature reduction of dimensions: LDA searches the underlying space for the vectors that most effectively differentiate between classes [21]. More specifically, given several independent characteristics that characterize the data, In order to maximize the mean differences between the intended classes, LDA creates a linear mixture of these features. From a mathematical perspective, we establish two metrics for every sample in every class: (i) the scatter matrix within the class, as given by Eq. 7.

$$S_w = \sum_{j=1}^{c} \sum_{i=1}^{N_j} (x_i^j - \mu_j)\ (x_i^j - \mu_j)^T \qquad (7)$$

Where x^j denotes the class j's i^{th} sample, the mean for class j is μj, Class count is represented by c, and Nj represents the sample size for sin class j: and (ii) a scatter matrix that is between classes.

$$S_b = \sum_{j=1}^{c} (\mu_j - \mu)\ (\mu_j - \mu)^T \qquad (8)$$

Where the mean of all classes is represented by μ.

LDA aims to maximize the between-class measure and minimize the within-class measure. Maximizing the ratio detS_b/detS_w is one approach to do this. If $Sw$is an indistinct matrix, consequently, the ratios reached their maximum when the projection matrix's column vectors $Wlda$ are the −1 eigen vectors of $SwSb$.

The performance of LDA is enhanced by modified linear discriminant analysis, which corrects for errors in Sw estimate. If class j's covariance matrix is represented by the notation

$$\sum j = \frac{1}{N_j} \sum_{i=1}^{N_j} (x_i^j - \mu_j)\ (x_i^j - \mu_j)^T \qquad (9)$$

Then Sw is expressed with Σjas

$$S_w = \sum_{j=1}^{c} N_j \sum j \qquad (10)$$

Taking the eigen decomposition, we can rewrite Σ$j$as

$$\sum j = B_j \Lambda_j B_j^T \qquad (11)$$

Where $\Lambda_j = diag(\lambda_{j1},...,\lambda_{jd})$, and eigen values (sorted in descending or der) of Σj are represented by $\lambda_{jk}(k=1,2,\ldots,d)$, and $B_j = [\beta_{j1},...,\beta_{jd}]$, where the order of eigenvectors are $\beta_{jk}(k= 1,2,\ldots,d)$. Since B_j is orthonormal(unitary), ${}^T B_j B_j = I$. In order to offset the Σj estimate error with limited sample sizes, we employ to swap out Λ_j and rebuild Σj as ^{T}. where Λ_j $\Sigma_j = B_j \Lambda_j B_j \Lambda_j = diag(\lambda_{j1},...\lambda_{jm},\delta_j,...\delta_j)$.

$$\delta_j = \frac{1}{d-m} \sum_{k=m+1}^{d} \lambda_{jk} \qquad (12)$$

After that, the within-class scatter matrix that has been rebuilt becomes

$$S_{\tilde{w}} = \sum_{j=1}^{c} N_j \Sigma_{\tilde{j}} \qquad (13)$$

Applications involving character recognition typically use $S_w^{\sim}$ as a non-single matrix. Therefore, the ratio that is required is maximized when column 1 of the projection matrix's vectors W_{mlda} are eigen vectors of $S^{\sim -1}{}_w \ S_b$.

3.5.5 Features Normalization: A vital step designed to make sure that the mathematical computing is as easy and quick as possible. The normalization namely, scaling has been adopted in this step to create the feature ranges [0 1], using the formula below:

$$A1' = \frac{A1 - min(A1)}{\max(A1) - min(A1)} \qquad (14)$$

A1' is the normalized value, and A1 is the original value.

3.6 Recognition and Classification: The main task here is to take the appropriate decision regarding classifying a character to the right class the character belongs to. There are a variety of classifiers that could be used in recognition. Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) were selected for classifications and performance will be compared

3.6.1 SVM Classifier: In the late 1990s, Vapnik and Cortes [25, 26] invented SVMs as a type of probabilistic machine that could learn. SVMs quickly rose to the top of the classification systems used most frequently in pattern recognition and data mining applications, due to the elevated percentage of classification. SVMs have been successfully used by researchers in a variety of modern learning applications, including document analysis, bioinformatics, optical character recognition (OCR), and image classification. Sigmoid kernels, polynomials, RBF, and linear models are frequently employed with SVM. With several kernels of 1) linear, 2) polynomial, 3) RBF, and 4) sigmoid, a multiple class SVM classification (libsvm) has been employed in the proposed system [27] and this classification achieves a very high-level of effectiveness in the recognition process. The recognition is the last step in this process at which a matching process is performed on the nominated class using the ASCII character by the SVM and discovers the necessary character in the Arabic dialect. The overall recommended system is demonstrated in Figure 14.

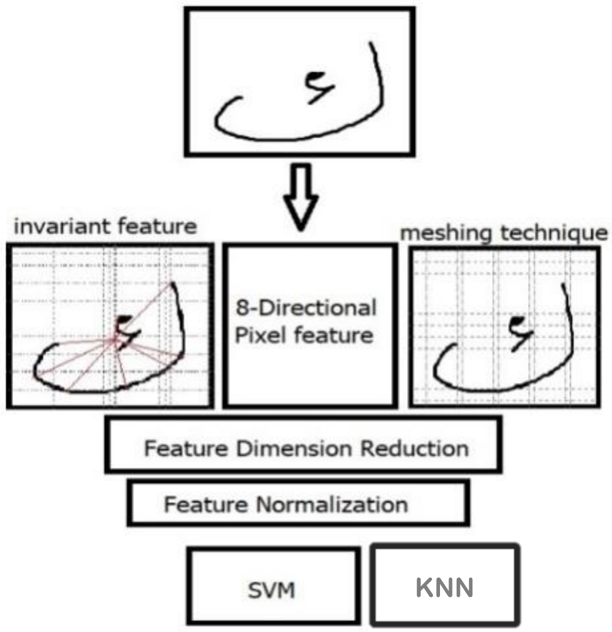

Figure 14. Proposed AHRC System

6.3.2 K-Nearest Neighbor (KNN): A K-NN classifier basically works, as far as the classification obstacle is involved, based on the assumption that all features are of identical value [28] [29]. When inappropriate and disturbing features influence the neighborhood investigation to the equivalent degree as highly pertinent features, the precision of the model is possible to fail. Feature assessment is a procedure used to estimate the optimum degree of impact of individual features utilizing a training set. When fruitfully applying pertinent features are ascribed a high value, while inappropriate features are granted a weight value around or approaching to zero. In addition, the suggested system utilizes Euclidian distance calculation in KNN to classify the images of the input character and the number of K (neighbors) is 3. By employing KNN classifier the system reached 97% recognition in terms of accuracy which is less in comparison with what SVM achieving.

IV. RESULTS AND DISCUSSION

The recommended technique is executed applying MATLAB(R2023b), with the specification: Windows 10 64-bit OS, Core (TM),i7-2328M, CPU@2.20GHz,2.20 GHz and the RAM is 8GBThe results were achieved in an effective way. The recommended dataset contains 560 images of handwritten characters. Each character was represented by twenty images transcribed in various styles. In the AHCR method fifty percent of the dataset was applied for training objective and other fifty percent for analysis and it accomplished a precision of 99.64%. By examining the 50% checking dataset, the images of the character offered 100% accuracy with one exception occurred with character (و) which offered 99.00% accuracy as demonstrated in Table 2.

TABLE II. ACCURATE RECOGNITION FOR THE AHCR SYSTEM.

No.	Character	Accuracy
1	أ, ب, ت, ث, ج, ح, خ, د, ذ, ر, ز, س, ش, ص, ض, ط, ظ, ع, غ, ف, ق, ك, ل, م, ن, ه, ي	100%
2	و	99%

ISSN: 2710-2165

Even though, with KNN the entire characters have been identified accurately with accuracy 100% apart from the characters (ب , ج , و , ه), when ((ج) %98,%97 (و) ,%98 (ه), and %99 (ب)). In the recommended method SVM categorization task with several kernels and every kernel complements various accuracy. Moreover, there is an essential factor that causes the SVM to work more efficiently. The most valuable factors in SVM are gamma (γ) and cost(c). It was concluded after many examinations to system that the greatest values for the factors were c = 4.0 and $\gamma = 0.25$. Moreover, various SVM kernels were examined and the best accomplishment was achieved by applying the SVM kernel. Figure 15 compares the performance in the form of average accuracy for both SVM and KNN.

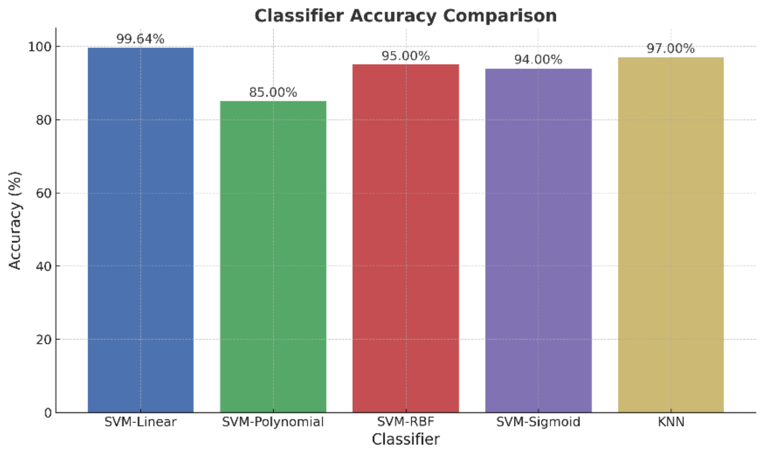

Figure 15. Comparing Various Recognition Accuracy.

It is worth mentioning at the end that multiple researchers' efforts on AHCR method and they attained better accuracy results. Though in comparison the suggested system with the previous systems, it is obvious that the recommended system provides the best accuracy results among all other previous systems, as seen in Figure 16 and Table 3, respectively.

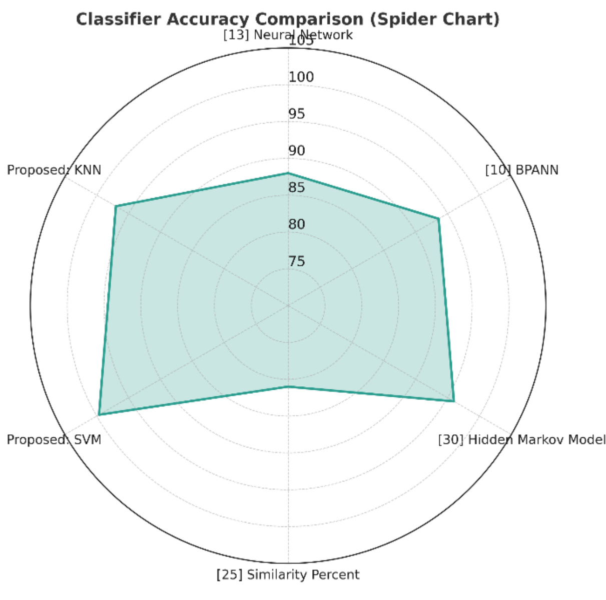

Figure 16. Accuracy Comparison Between Suggested and Current Systems.

The comparative study expressed through both the bar chart and the spider chart yields valuable information about the efficacy of the classification algorithms utilized in the published literature compared to the proposed model. Of significance is the performance of the proposed SVM classifier with an accuracy of 99.64%, closely followed by the proposed KNN classifier with an accuracy of 97%. These are remarkably high compared to many of the models documented in earlier research. For example, the Neural Network model in [13] attains an accuracy of 88.00%, whereas the BPANN model of [10] attains a relatively higher accuracy of 93.60%. The Hidden Markov Model (HMM) of [30] achieves an interval of 94% to 98%, which may compare favorably but still lags slightly behind the proposed SVM classifier at its best. Besides, the similarity method in [25] achieves the lowest accuracy among them all at 81%, which shows minimal robustness and reliability in challenging classification tasks.

TABLE III. COMPARISON OF ACCURACY BETWEEN THE SUGGESTED AND STAT OF THE ART.

Reference	Classifiers	Accuracy
[13]	Neural Network	88.00%
[10]	BPANN	93.60%
[30]	Hidden Markov Model	Varies from 94% to 98%
[25]	similarity percent	81%
Proposed System	**SVM**	**99.64%**
Proposed System	**KNN**	**97%**

The degree of improvement achieved by the proposed system, particularly the SVM model, highlights the robustness of the selected feature extraction and model optimization methods. This performance suggests that the proposed system can be more applicable for actual applications where high reliability of classification is required, for example, in medical diagnosis, biometric identification, or detecting fraudulent activity. These findings are in agreement with more current work in machine learning, where support vector machines (SVMs) are frequently championed for treating high-dimensional data and yielding excellent generalization performance, especially when linear separability or kernel transforms are utilized effectively. In the same manner, while the K-nearest neighbors algorithm, though straightforward, is still competitive when utilized with good normalization and distance measures, as attested to in its 97% accuracy here. By contrast, the comparably poor performance of similar methods and traditional neural networks can be attributed to hyperparameter sensitivity, data unbalancedness, or the lack of sophisticated preprocessing methods in past work. Although HMMs demonstrate relatively robust performance, they are limited in their use with variable problem spaces due to their use of sequential data structures.

V. CONCLUSION

Our article proposes a highly precise Offline Isolated system for recognizing Arabic characters, accompanied by a dedicated handwriting dataset. The system employs a dataset split, allocating 50% for training and 50% for testing. Exceptional accuracy rates are achieved through the utilization of an SVM linear kernel and a KNN classifier. The success of this system stems from a series of effective steps, beginning with robust preprocessing techniques such as FCM (Fuzzy C-Means) clustering. This is followed by meticulous feature extraction, extracting discriminative information from the character images. Finally, the system employs accurate recognition classifiers to achieve precise results. The effectiveness of the proposed system is underscored by its outperformance of existing systems in terms of recognition accuracy. By leveraging sophisticated preprocessing, feature extraction, and classification methodologies, the system demonstrates superior performance in recognizing Arabic characters. Notably, the integration of an SVM linear kernel and KNN classifier contributes significantly to the system's success, showcasing the importance of employing appropriate algorithms in character recognition tasks. Overall, the proposed system represents a significant advancement in Arabic character recognition, offering highly accurate results and paving the way for enhanced applications in various fields.

REFERENCES

[1] S. B. Ahmed, M. I. Razzak, and R. Yusof, Cursive Script Text Recognition in Natural Scene Images. Singapore: Springer Singapore, 2020.
[2] M. Awni, M. I. Khalil, and H. M. Abbas, "Offline Arabic handwritten word recognition: A transfer learning approach," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 10, pp. 9654-9661, 2022.
[3] M. Salam and A. A. Hassan, "Offline isolated Arabic handwriting character recognition system based on SVM," Int. Arab J. Inf. Technol., vol. 16, no. 3, pp. 467-472, 2019.
[4] A. M. Mutawa, M. Y. Allaho, and M. Al-Hajeri, “Machine Learning Approach for Arabic Handwritten Recognition,” Applied Sciences, vol. 14, no. 19, p. 9020, Oct. 2024.

[5] A. M. Al-Shatnawi, F. Al-Saqqar, and A. Souri, "Arabic handwritten word recognition based on stationary wavelet transform technique using machine learning," Transactions on Asian and Low-Resource Language Information Processing, vol. 21, no. 3, pp. 1-21, 2021.
[6] M. Aouine, Abdeljalil Gattal, Chawki Djeddi, and F. Abbas, "Handwritten digit recognition using a column scheme-based local directional number pattern," Bulletin of Electrical Engineering and Informatics, vol. 13, no. 6, pp. 4157–4167, Sep. 2024.
[7] C. Sharma, S. Sharma, None Sakshi, and H.-Y. Chen, "Advancements in handwritten Devanagari character recognition: a study on transfer learning and VGG16 algorithm," Deleted Journal, vol. 6, no. 12, Nov. 2024.
[8] S. Aabed and A. Khairaldin, "An End-to-End, Segmentation-Free, Arabic Handwritten Recognition Model on KHATT," arXiv.org, 2024.
[9] M. N. AlJarrah, M. Z. Mo'ath, and R. Duwairi, "Arabic handwritten characters recognition using convolutional neural network," in 2021 12th International Conference on Information and Communication Systems (ICICS), IEEE, 2021.
[10] Manel Ayadi, Nesrine Masmoudi, Latifa Almuqren, Raneem Oudah Aljohani, and Hadeel Saeed Alshahrani, "Empowering Accessibility in Handwritten Arabic Text Recognition for Visually Impaired Individuals through Optimized Generative Adversarial Network (GAN) model," Journal of disability research, vol. 4, no. 1, Jan. 2025
[11] H. Al-Yousefi and S. S. Upda, "Recognition of Arabic characters," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 14, no. 08, pp. 853-857, 1992.
[12] F. Salameh, ""Young Phoenicians" and the quest for a Lebanese language: between Lebanonism, Phoenicianism, and Arabism," in Arabic and Its Alternatives, Brill, 2020, pp. 111-129.
[13] Y. Hamdi, H. Boubaker, and A. M. Alimi, "Online Arabic handwriting recognition using graphemes segmentation and deep learning recurrent neural networks," in Enabling Machine Learning Applications in Data Science: Proceedings of Arab Conference for Emerging Technologies 2020, Singapore: Springer Singapore, 2021.
[14] Y. B. Hamdan and A. Sathesh, "Construction of statistical SVM based recognition model for handwritten character recognition," Journal of Information Technology, vol. 3, no. 2, pp. 92-107, 2021.
[15] W. Alomoush et al., "Fully automatic grayscale image segmentation based fuzzy C-means with firefly mate algorithm," Journal of Ambient Intelligence and Humanized Computing, vol. 13, no. 9, pp. 4519-4541, 2022.
[16] W.-C. Cheng et al., "Image haze removal using dark channel prior technology with adaptive mask size," Sensors & Materials, vol. 32, 2020.
[17] R. A. Lotufo et al., "Morphological image processing," in Microscope Image Processing, Academic Press, 2023, pp. 75-117.
[18] S. Djaghbellou, A. Attia, and A. Bouziane, "A survey on text-line segmentation process in historical Arab manuscripts," IAM, 2023.
[19] S. Hamida et al., "Cursive Arabic handwritten word recognition system using majority voting and k-NN for feature descriptor selection," Multimedia Tools and Applications, vol. 82, no. 26, pp. 40657-40681, 2023.
[20] H. M. Balaha, H. A. Ali, and M. Badawy, "Automatic recognition of handwritten Arabic characters: A comprehensive review," Neural Computing and Applications, vol. 33, pp. 3011-3034, 2021.
[21] A. Y. Muaad et al., "Arabic document classification: Performance investigation of preprocessing and representation techniques," Mathematical Problems in Engineering, vol. 2022, no. 1, p. 3720358, 2022.
[22] S. K. Jemni, Y. Kessentini, and S. Kanoun, "Improving recurrent neural networks for offline Arabic handwriting recognition by combining different language models," International Journal of Pattern Recognition and Artificial Intelligence, vol. 34, no. 12, p. 2052007, 2020.
[23] J. Memon et al., "Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR)," IEEE Access, vol. 8, pp. 142642-142668, 2020.
[24] S. B. Ahmed, M. I. Razzak, and R. Yusof, Cursive Script Text Recognition in Natural Scene Images. Singapore: Springer Singapore, 2020.
[25] A. Ganguly, R. Mitra, and J. Zhou, "Infinite-dimensional optimization and Bayesian nonparametric learning of stochastic differential equations," Journal of Machine Learning Research, vol. 24, no. 159, pp. 1-39, 2023.
[26] J. W. Shavlik and T. G. Dietterich, Eds., Readings in Machine Learning. Morgan Kaufmann, 1990.
[27] N. N. Abboud, "A mixed finite element formulation for the transient and harmonic exterior fluid-structure interaction problem," Stanford University, 1990.
[28] M. Steinbach and P.-N. Tan, "kNN: k-nearest neighbors," in The Top Ten Algorithms in Data Mining. Chapman and Hall/CRC, pp. 165-176, 2009.
[29] H. waleed Hamza, A. A. Al-Hamadani, "A Review on Artificial Intelligence methods and Signal Processing for EEG-Based lie and Truth Identification," Al-Iraqia Journal of Scientific Engineering Research, vol. 3, no. 2, Jun. 2024
[30] S. Mohammed Tariq, "Spatial Analysis of Local Statistics for Handwritten Signature Recognition", Al-Iraqia Journal of Scientific Engineering Research, vol. 3, no. 2, pp. 10–20, Jun. 2024.