An Optimized Approach for Handwritten Arabic Character Recognition based on the SVM Classifier

Biteur Kada
Department of Automatic and Electromechanical Engineering, Faculty of Science and Technology, Ghardaia University, Algeria | COSNA Laboratory, Tlemcen University, Algeria
biteur.kada@univ-ghardaia.dz (corresponding author)

Arif Mohammed
Department of Automatic and Electromechanical Engineering, Faculty of Science and Technology, Ghardaia University, Algeria | Faculty of Applied Sciences, University of Ouargla, Algeria
arifmed.univ.ghard@gmail.com

Benhammadi Abdelmajid
Department of Automatic and Electromechanical Engineering, Faculty of Science and Technology, Ghardaia University, Algeria
nouce.2020@gmail.com

Received: 15 October 2024 | Revised: 18 November 2024 | Accepted: 25 November 2024

 DOI: https://doi.org/10.48084/etasr.9292

ABSTRACT

Optical Character Recognition (OCR) is an essential technology, capable of addressing complex challenges while simplifying numerous human activities. Although it emerged in the 1970s with various solutions, these efforts primarily focused on Latin-based languages, leaving other writing systems, such as Arabic, largely underexplored. In this context, this study proposes an innovative offline Arabic handwriting recognition system based on a structural segmentation method combined with the use of Support Vector Machines (SVM) for character classification. An in-depth review of different character segmentation methods was followed by an in-depth analysis of the OCR field. This study examined the challenges associated with normalization, a recurring issue in the processing of handwritten scripts. Finally, after comparing the unique characteristics of Arabic handwritten characters with existing segmentation techniques, an approach was developed based on a segmentation algorithm to improve the accuracy and efficiency of the recognition process.

Keywords-OCR; segmentation; Arabic characters; SVM; post-processing

I. INTRODUCTION

Researchers specializing in Arabic handwritten character recognition are exploring a rapidly growing field that has gained importance over the past two decades to overcome the unique challenges of this script and broaden the scope of Arabic character recognition [1]. An effective handwriting recognition system should be capable of locating, recognizing, and interpreting any handwritten text or numbers, regardless of the medium or the document's quality, whether it be maps, forms, calendars, or ancient manuscripts [2]. The field has a wide range of practical applications, including postal services for recognizing postal codes and addresses, automated check processing in banking, electronic document flow management in administrative settings, indexing and retrieval in digital libraries, and biometric writer identification [3]. These applications highlight the potential of Arabic character recognition and the technical challenges to tackle [4].

II. BACKGROUND

A. Preprocessing Phase

Preprocessing involves preparing data obtained from sensors for the next phase of processing. Its primary objective is to reduce the noise overlaid onto the data, aiming to preserve only the essential information corresponding to the depicted shape. This noise can result from various acquisition conditions, such as lighting or incorrect document formatting [5], as well as the condition of the source document. Typical preprocessing techniques involve tasks such as thresholding, dilation, erosion, skeletonization, and normalization (see Figure 1) [4-6].

B. Segmentation Phase

Segmentation involves dividing a real scene into constituent parts or objects, effectively mapping the scene onto a plane. It serves as an initial step in shape recognition and must possess specific attributes that represent the areas to extract. These attributes are essential for the subsequent classification of individual points within segmented regions [7, 8].

TABLE I. ARABIC ALPHABET IN ITS DIFFERENT FORMS

Name	Initial	Medial	Final	Separate	Pronunciation
alif*	ا	ـا	ـا	ا	See opposite
baa’	بـ	ـبـ	ـب	ب	b
taa’	تـ	ـتـ	ـت	ت	t
thaa’	ثـ	ـثـ	ـث	ث	th
jiim	جـ	ـجـ	ـج	ج	j
Haa’	حـ	ـحـ	ـح	ح	H
khaa’	خـ	ـخـ	ـخ	خ	kh
daal*	د	ـد	ـد	د	d
dhaal*	ذ	ـذ	ـذ	ذ	dh
raa’*	ر	ـر	ـر	ر	r
zaay*	ز	ـز	ـز	ز	z
siin	سـ	ـسـ	ـس	س	s
shin	شـ	ـشـ	ـش	ش	sh
Saad	صـ	ـصـ	ـص	ص	S
Daad	ضـ	ـضـ	ـض	ض	D
Taa’	ط	ـط	ـط	ط	T
DHaa’	ظ	ـظ	ـظ	ظ	DH
:ain	عـ	ـعـ	ـع	ع	:
ghain	غـ	ـغـ	ـغ	غ	gh
faa'	فـ	ـفـ	ـف	ف	f
qaaf	قـ	ـقـ	ـق	ق	g
kaaf	كـ	ـكـ	ـك	ك	k
laam	لـ	ـلـ	ـل	ل	l
miim	مـ	ـمـ	ـم	م	m
nuun	نـ	ـنـ	ـن	ن	n
haa’	هـ	ـهـ	ـه	ه	h
waaw	و	ـو	ـو	و	w
yaa’	يـ	ـيـ	ـي	ي	y
on alif	أ	ـأ	ـأ	أ	

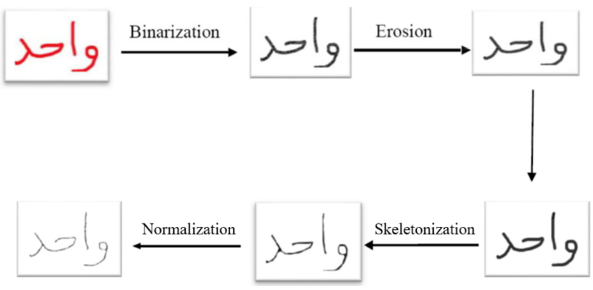

Fig. 1. Effects of certain pre-processing operations.

C. Segmentation Stages

1) Segmentation of Text into Lines

Methods for processing Arabic text often rely on horizontal projection to extract lines. However, the presence of diacritical marks complicates this process and can sometimes result in confusion between lines. This issue arises when the spacing is determined using a simple average of the various gaps. To address this challenge, some approaches first determine the different lines of text and then categorize the text blocks according to their closeness to the already identified writing lines. This approach improves the accuracy of line segmentation [9-10].

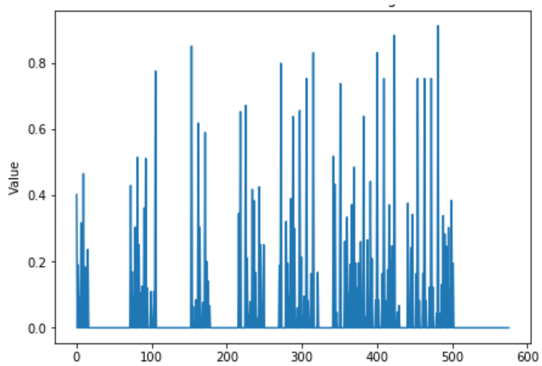

Fig. 2. Example of a horizontal histogram of a line of text.

In the context of Latin scripts, lines can also merge due to the ascenders and descenders of letters. When such fusion occurs, an empirical correction method involves first identifying the line with the highest concentration of black pixels [11]. The areas above and below this line are then examined based on the black pixel densities of the surrounding lines. For instance, if the fusion occurs in the upper section, the line with the lowest pixel density in that area is considered to represent the boundary between the fused lines [12].

2) Segmentation in Pseudo-Word (PAW: Piece of Arabic Word)

This process involves analyzing the histogram of the vertical projections of the different lines of text. However, this method is ineffective when the text lines are vertically overlapped. In such cases, alternative techniques are employed, such as contour determination, skeletonization, or the analysis of connected components. The choice of technique is often influenced by the specific analysis method being used.

Fig. 3. Example of overlapping PAWs respectively from right to left between "ثلاثـــون" and " ألـف ".

3) Segmentation of the Line of Text into Words

In Arabic OCR, segmentation primarily focuses on extracting PAWs (Partially Aligned Words), while word-level analysis takes place during the postprocessing phase, if necessary, to validate results or correct recognition errors. To tackle the challenge of inaccurate word segmentation, some approaches adopt a method that processes only one word at a time [13, 14].

4) Segmentation of Words into Characters

Segmenting characters (or graphemes) is one of the most challenging issues in Arabic script recognition. The difficulties at this stage are akin to those encountered in Latin manuscript (cursive) recognition but are often more complex due to the wide variety of Arabic character forms, the minimal spacing between successive characters, the elongation of horizontal ligatures, and the existence of vertical ligatures [14, 15].

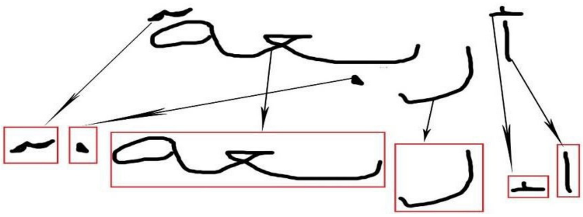

Fig. 4. Segmentation example.

D. Segmentation Approaches

The application of segmentation methods for cursive Latin scripts has been thoroughly researched. However, applying these algorithms to Arabic writing poses considerable challenges. There are five main approaches to segmenting Arabic words [13-15]:

- The first approach assumes that the input word is already segmented into individual characters.
- The second approach focuses on breaking down the input word into components that are smaller than a character.
- The third approach is dedicated to segmenting the input word into its constituent characters.
- The fourth approach treats the word as a whole, and segmentation is managed within a recognition module or as a submodule.
- The fifth approach processes the word without any segmentation.

The first approach assumes that the words were already segmented upon entry. The second one focuses on breaking words down into components smaller than characters, which is particularly suitable for handwritten manuscripts. The third segments words into individual characters for recognition, typically used for printed text. The fourth approach recognizes words without prior segmentation by utilizing morphological primitives and comparison models. However, this method has the drawback that the definition of these primitives depends on the character size and font, limiting the variety of fonts and sizes that could be processed. In the fifth approach, whole words are recognized without any segmentation, which restricts vocabulary flexibility.

E. Proposed Segmentation Approach

This study proposes a straightforward structural segmentation method that falls under the second approach. The principle is simple: scan the image of the source PAW (see Figure 3) in the reading direction (from right to left for Arabic) and perform an additional sweep from the bottom upwards according to the following guidelines:

- The first or last identified segment is referred to as the segment that carries the character.
- A division or combination with a segment labeled as a basic character carrier (level 1, see Figure 5) eliminates the label.
- A division or merge operation with a segment labeled as a composite character (level 2). The set of segments under processing is then evaluated as level 3 segments, leading to the identification of a character.

Figure 5 illustrates the sequence of these steps.

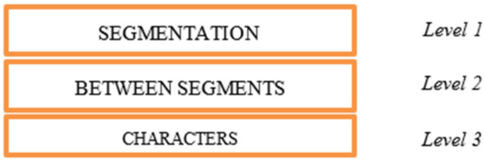

Fig. 5. Different levels of feature extraction.

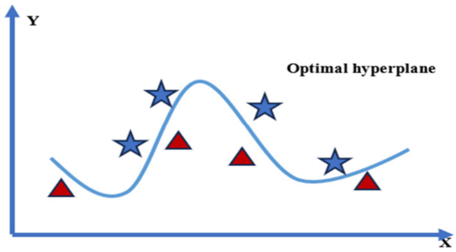

Fig. 6. Border determination problem rather far from the points of different classes.

III. METHOD

A. Mathematical Foundations

1) Learning Challenge

The focus is on a function f (which may be non-deterministic) that generates an output $y = f(x)$ based on a specific set of inputs x. The goal is to find the function f from the single observation of several input-output pairs $\{(x_i, y_i)$, $i = 1, \ldots, n\}$ to predict other events.

A pair (X, Y) of random variables is considered, with values in $X \times Y$. Only the case $Y = \{-1, 1\}$ (classification) is of interest here. The joint distribution of (X, Y) is unknown. Knowing a sample $S = \{(X_1, Y_1)(X_n, Y_n)\}$ of n copies independent of (X, Y), a function $h: X \rightarrow Y$ is needed such that $P(h(X)! = Y)$ is minimal [9].

2) Real-valued Classification

The error is calculated with $P(h(X)! = Y) = P(Yf(X) \leq 0)$. This gives some idea of confidence in the classification. Ideally, $|Yf(X)|$ is proportional to $P(Y|X)$. $Yf(X)$ represents the margin of in (X, Y). The goal is to construct f and h. This can be achieved by the following [12].

a) Entries Transformation

It might be essential to convert the input data to facilitate processing. Let X represent a space of objects. These entries can be converted into vectors in a feature space using a function $\Phi: X \rightarrow F$, where F is not strictly finite but contains a scalar product (known as a Hilbert space). The Hilbert space acts as an extension to the Euclidean space and may possess an infinite count of dimensions. This transformation addresses nonlinearity, allowing us to choose a linear separation (a nonlinear problem can be reduced to a classical linear problem) [14-15]. Therefore, the goal is to select the optimal hyperplane that accurately classifies the data (where feasible) while maximizing the distance from all points to be classified.

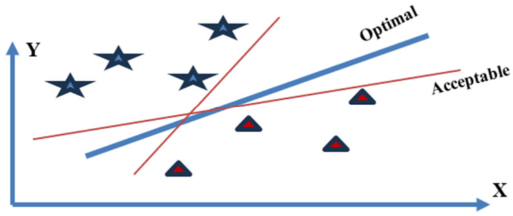

Fig. 7. Example of an optimal hyperplane search.

However, the selected separating hyperplane must maximize the gap separating the classes. This gap refers to the distance from the dividing hyperplane to the closest data points of both classes, ensuring that the classification is as robust as possible by maximizing this separation [16, 17].

b) Maximizing the Margin

The margin refers to the distance between the nearest data point and the hyperplane.

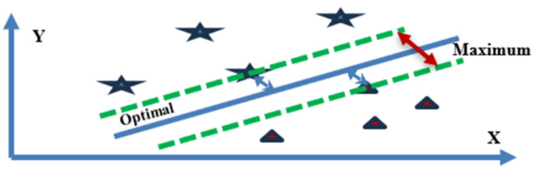

Fig. 8. The relation between margin, support vector points, and optimal hyperplane.

In a linear model, a separating hyperplane (decision boundary) has the equation $w.x + b = 0$. The distance from a point to the plane is given by $d(x) = | w.x + b \, \|w\|$. The optimal hyperplane is the one for which the distance to the nearest points (margin) is maximum. Let x_1 and x_2 be the points of different classes, such as ($f(x_1) = 1$ and $f(x_2) = -1$), $(w.x_1) + b = 1$ and $(w.x_2) + b = (x_1 - x_2) = 2$. Hence, $(w \, / \, \|w\|.(x_1 - x_2)) = 2 \, / \, \|w\|$. Thus it can be concluded that maximizing the margin involves minimizing $\|w\|$ subjected to specific constraints.

c) Primal Problem

A point (x, y) is well classified if and only if $f(x) > 0$. Since the pair (w, b) is defined with a multiplicative coefficient, $f(x) \geq 1$. From this, the minimization problem is derived under the following constraints, considering the previous discussion:

$$min\frac{1}{2}\|w\|^2, \;\; \forall i, y_i(x.x_i + b) \geq 1 \tag{1}$$

It may be more convenient to minimize $\|w\|^2$ instead of minimizing $\|w\|$ directly.

d) Dual Challenge

Lagrange factors were used to shift from the primal issue to the dual challenge for every restriction. An example of a learning constraint can be given as:

$$\begin{cases} max \sum_{i=1}^{n} \alpha - \frac{1}{2}\sum_{ij} \alpha_i \alpha_j \gamma_i \gamma_j x_i x_j \\ \qquad \forall i, 0 \leq \alpha_i \leq C \\ \qquad \sum_{i=1}^{n} \alpha_i \alpha_i = 0 \end{cases} \tag{2}$$

This represents a quadratic optimization challenge of size n, where n refers to the total number of samples. A matrix is defined, known as the Hessian matrix, representing the product matrix of the inputs X. Using matrix notation facilitates a simpler computation of the problem [13]. It can be shown that if α_i^* are solutions to this problem, then:

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i \tag{3}$$

Only the α_i values corresponding to the nearest points are non-zero, which are referred to as support vectors. Consequently, the associated decision function is:

$$f(x) = \sum_{i=1}^{n,} \alpha_i^* x_i y_i . x + b \tag{4}$$

However, there are instances where the inputs cannot be classified linearly [11].

3) Nonlinearity (Non-Separable Case/Soft Margin)

We begin with the primal linear problem and introduce spring variables to relax the constraints:

$$\begin{cases} min\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \varepsilon_i \\ \forall i, y_i(x.x_i + b) \geq 1 - \varepsilon_i \end{cases} \tag{5}$$

This is penalized for any violations of the constraint. From this, the dual problem can be derived, which follows a similar structure as in the separable scenario:

$$\begin{aligned} & max \sum_{i=1}^{n} \alpha - \frac{1}{2}\sum_{ij} \alpha_i \alpha_j \gamma_i \gamma_j x_i x_j \\ & \quad \forall i, 0 \leq \alpha_i \leq C \\ & \quad \sum_{i=1}^{n} \alpha_i \alpha_i = 0 \end{aligned} \tag{6}$$

The sole distinction is the upper limit C on α.

a) Core Function (Kernel)

In the linear scenario, the data can be converted into a space where classification is simpler. In such instances, the most commonly utilized space for re-description is $\mathbb{R}$ (the set of real numbers). However, for nonlinear cases, this space proves insufficient for classifying the inputs. Consequently, a transition to a higher-dimensional space is needed, where $(F) > d$. The transition in $F = \mathbb{R}^3$ makes linear separation of data possible. Therefore, the following must be solved:

$$\begin{cases} max \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{ij} \alpha_i \alpha_j y_i y_j \varphi(x_i).\varphi(x_j) \\ \quad \forall j, \quad 0 \leq \alpha_i \leq C, \qquad \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{7}$$

The solution has the form:

$$f(x) = \sum_{i=1}^{n}. \alpha_i^* \, y_i \varphi(x_i).\varphi(x_j) + b \tag{8}$$

The problem and its solution rely solely on the scalar product $\varphi(x) \times \varphi(x')$. Instead of selecting the nonlinear transformation $\Phi: X \to F$, we opt for a function $K: X \times X \to \mathbb{R}$, known as the kernel function. This represents a scalar product in the intermediate representation space. Consequently, k is linear, which allows connecting it back to the linear case discussed in the previous paragraphs. Thus, this function expresses the distribution of the examples in this space as $k(x, x') = \varphi(x) \cdot \varphi(x')$. When k is appropriately chosen, it is not required to calculate the representation of the examples in this space to evaluate φ.

b) Mercer Condition

A symmetric function k is a kernel if $\left(k(x_i, x_j)\right)_{i,j}$ is a defined matrix positive. In this case, there is a space F and a function f such that $k(x, x') = F(x) \times f(x')$.

B. Computation Time and Convergence

The complexity (computation time) of the SVM algorithm [16] was assessed, which is not influenced by the number of inputs to be classified (d) but depends on the number of training samples (n). Its complexity is polynomial in n:

$$dn^2 \leq Complexity \leq dn^3$$

with the size of the hessian matrix being n^2.

All elements of the matrix should be processed along with all the entries. When dealing with a very large dataset, the computation time can become prohibitive. As a result, SVMs are most suitable for small classification problems [17].

IV. RESULTS AND DISCUSSION

Handwritten words (numbers) were segmented using the proposed approach. The parameters were extracted automatically, emphasizing the static characteristics of each stroke within the word, in line with the results presented and the discussed algorithm (SVM). Experiments were carried out to determine the optimal settings for recognition. The dataset consisted of 1564 unique images for training, categorized into 24 classes using a batch size of six. The False Acceptance Rate (FAR) is the ratio of incorrectly accepted symbols to the total number of attempts, while the False Rejection Rate (FRR) is the ratio of genuine samples mistakenly rejected to the total number of legitimate attempts. By adjusting the decision threshold, FAR and FRR were examined to identify the optimal SVM parameters to determine the Equal Error Rate (EER) where FAR and FRR intersect.

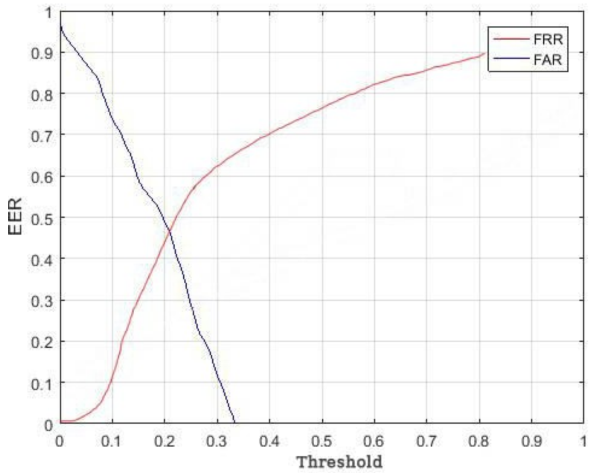

Fig. 9. FRR and FAR curves for Parameters I.

- Parameters I: Figure 9 shows the FAR and FRR curves using an RBF kernel with $\gamma = 0.001$ and $C = 10$, where C balances the fit to the training data and the simplicity of the decision boundary, and that γ determines the width of the kernel function. The results indicate that the first parameter has minimal influence on this method, resulting in an EER of 0.473 at a threshold of 0.21. Theoretically, a zero EER indicates perfect recognition, highlighting the importance of further parameter optimization to improve model accuracy, especially since the value of 0.473 is relatively high.
- Parameters II: Figure 10 shows the FAR and FRR curves utilizing a linear kernel with $C = 1000$ and $\gamma = 0.001$. This setup demonstrates a significant improvement in accuracy, resulting in an EER of 0.224 at a threshold of 0.258.

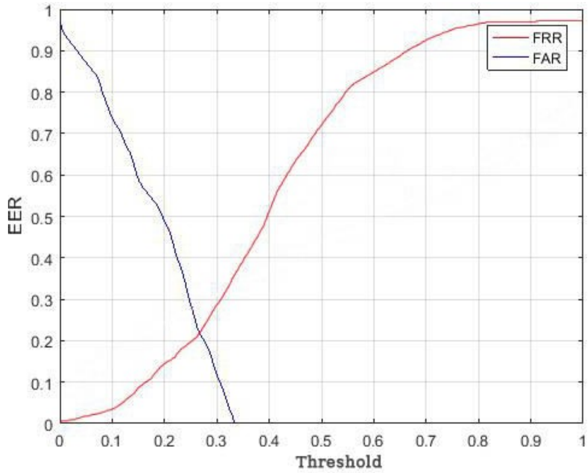

Fig. 10. FRR and FAR curves for Parameters II.

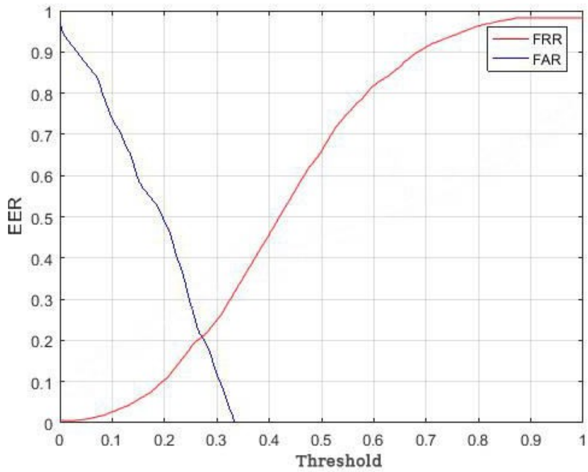

Fig. 11. FRR and FAR curves for a combination of Parameters I & II.

- Combination of Parameters I and II: Figure 11 shows the results utilizing a linear kernel with $C = 100$ and $\gamma = 0.001$. This configuration improved the system's accuracy, achieving an EER of 0.215 at a threshold of 0.261.

- Parameters III: Figure 12 shows the results using different kernels, linear and RBF, respectively, with a $\gamma = 1000$ and $C = 100$. These parameters led to an improvement in accuracy, achieving an EER of 0.113 for the RBF kernel at a threshold of 0.3 and an EER of 0.18 for the linear kernel at a threshold of 0.28. These results show that the error rate with an optimized combination of the three parameters (kernel type, gamma rate, and regularization parameter C) was reduced to 11.3%, which corresponds to a recognition rate of 88.7%. This is an improvement compared to the previous combinations. This enhancement demonstrates the significant impact of parameter selection on the performance of the classification model.

(a)

(b)

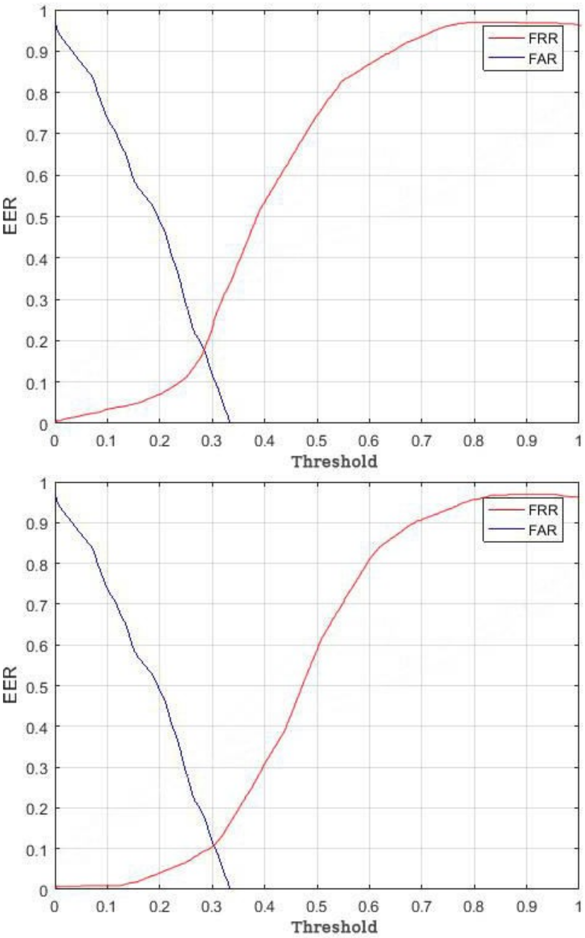

Fig. 12. Results for parameters III: (a) linear, (b) RBF.

Finally, based on the results obtained for different parameters, a significant improvement in the accuracy of the proposed system was achieved. EER gradually decreases as the threshold increases.

This study explored various SVM parameters that contribute to a high accuracy rate in Arabic handwriting recognition, utilizing two distinctly different kernels. The SVM classification results can be further enhanced by optimizing the following factors:

- The C and γ parameters.
- The size of the dataset.
- The appropriate kernel types.

V. CONCLUSION

Arabic OCR remains a challenging field due to the inherent complexities of the Arabic script, such as character shape variability, ligatures, and diacritical marks. Several approaches, including segmentation-free methods, neural networks, and hybrid techniques, have been developed to address these challenges. Among these, the proposed method, based on structural segmentation combined with SVM, was proven particularly effective. With a recognition rate of 88.7% and an EER of 0.113, it offers a competitive and balanced solution between accuracy and complexity. Unlike neural network-based methods, this approach is less resource-intensive while remaining robust against stylistic and structural variations in Arabic characters. Nonetheless, this method presents opportunities for further improvement, particularly through the integration of hybrid techniques and the optimization of SVM parameters. It also holds significant potential for practical applications, such as digitizing ancient manuscripts and automating document management systems in Arabic.

In conclusion, this study significantly contributes to Arabic OCR and provides a robust foundation for the development of efficient and accurate systems tailored to Arabic handwriting. By addressing current limitations and adopting innovative methods, this research opens up new avenues for breakthroughs in the automation of Arabic text recognition using artificial intelligence, with promising prospects for both academic research and practical applications.

REFERENCES

[1] M. Bennamoun and G. J. Mamic, "Optical Character Recognition," in *Object Recognition: Fundamentals and Case Studies*, M. Bennamoun and G. J. Mamic, Eds. London: Springer, 2002, pp. 199–220.

[2] S. Faizullah, M. S. Ayub, S. Hussain, and M. A. Khan, "A Survey of OCR in Arabic Language: Applications, Techniques, and Challenges," *Applied Sciences*, vol. 13, no. 7, Jan. 2023, Art. no. 4584, https://doi.org/10.3390/app13074584.

[3] M. S. Khorsheed, "Off-Line Arabic Character Recognition – A Review," *Pattern Analysis & Applications*, vol. 5, no. 1, pp. 31–45, May 2002, https://doi.org/10.1007/s100440200004.

[4] A. A. Alasadi, T. H. Aldhayni, R. R. Deshmukh, A. H. Alahmadi, and A. S. Alshebami, "Efficient Feature Extraction Algorithms to Develop an Arabic Speech Recognition System," *Engineering, Technology & Applied Science Research*, vol. 10, no. 2, pp. 5547–5553, Apr. 2020, https://doi.org/10.48084/etasr.3465.

[5] C. Bahlmann and H. Burkhardt, "The writer independent online handwriting recognition system frog on hand and cluster generative statistical dynamic time warping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 299–310, Mar. 2004, https://doi.org/10.1109/TPAMI.2004.1262308.

[6] M. Cheriet, N. Kharma, C. L. Liu, and C. Suen, *Character Recognition Systems: A Guide for Students and Practitioners*. John Wiley & Sons, 2007.

[7] L. A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 910–918, Aug. 1988, https://doi.org/10.1109/34.9112.

[8] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 1, pp. 100–132, Jan. 1985, https://doi.org/10.1016/S0734-189X(85)90153-7.

[9] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, Sep. 1993, https://doi.org/10.1016/0031-3203(93)90135-J.

[10] N. Altwaijry and I. Al-Turaiki, "Arabic handwriting recognition system using convolutional neural network," *Neural Computing and Applications*, vol. 33, no. 7, pp. 2249–2261, Apr. 2021, https://doi.org/10.1007/s00521-020-05070-8.

[11] N. Sabbour and F. Shafait, "A segmentation-free approach to Arabic and Urdu OCR," in *Document Recognition and Retrieval XX*, Feb. 2013, vol. 8658, pp. 215–226, https://doi.org/10.1117/12.2003731.

[12] L. M. Lorigo and V. Govindaraju, "Offline Arabic handwriting recognition: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 712–724, Feb. 2006, https://doi.org/10.1109/TPAMI.2006.102.

[13] H. A. Abdo, A. Abdu, R. Manza, and S. Bawiskar, "An approach to analysis of arabic text documents into text lines, words, and characters," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no. 2, May 2022, Art. no. 754, https://doi.org/10.11591/ijeecs.v26.i2.pp754-763.

[14] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[15] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, Sep. 2020, https://doi.org/10.1016/j.neucom.2019.10.118.

[16] X. Zhou, J. Li, C. Yang, and J. Hao, "Study on Handwritten Digit Recognition using Support vector machine," *IOP Conference Series: Materials Science and Engineering*, vol. 452, no. 4, Sep. 2018, Art. no. 042194, https://doi.org/10.1088/1757-899X/452/4/042194.

[17] B. Kiessling, G. Kurin, M. T. Miller, and K. Smail, "Advances and Limitations in Open Source Arabic-Script OCR: A Case Study," *Digital Studies / Le champ numérique*, vol. 11, no. 1, Dec. 2021, https://doi.org/10.16995/dscn.8094.