# Arabic Word Recognition: Causes and Solutions for Straight-Line Problems in Triangle Geometry

**Nur Atikah Arbain[1], Azah Kamilah MudaDraman[2], Mohd Sanusi Azmi[3], Nurul Izrin Md Saleh[4]**

**Faculty Teknologi Maklumat Dan Komunikasi (FTMK)**
**University Technical Malaysia Melaka**

**DOI:** https://dx.doi.org/10.47772/IJRISS.2025.909000325

**Received: 20 August 2025; Accepted: 26 August 2025; Published: 10 October 2025**

## ABSTRACT

Arabic word recognition is crucial in pattern recognition and digitizing handwritten documents. Accurate triangle shapes for feature extraction are often hindered by straight-line problems. This study identifies the causes of these problems and proposes a robust method to improve recognition accuracy using triangle geometry method. The straight-line issue disrupts the formation of triangles, essential for feature extraction. The proposed method rearranges coordinate points to form new triangles, addressing issues from collinear points and parallel lines. Four specific situations causing straight-line problems were analyzed, including proximity affecting angle extraction. The method improved classification performance: the Support Vector Machine (SVM) achieved 77.28% accuracy with features based on three side lengths, while the Random Forest classifier reached 64.22%. These results show that the method effectively enhances Arabic word recognition by solving geometric challenges in triangle feature extraction.

**Keywords:** Arabic Handwriting, Triangle Geometry, Straight-Line Problem, Support Vector Machine

## INTRODUCTION

Handwriting recognition is a critical technology that translates human handwriting into computer-readable algorithms. It plays a significant role in document analysis and recognition, supporting a wide range of applications. Notably, current research focuses on the complexities of recognizing handwritten Arabic words, a field that has garnered significant interest due to its unique challenges. The diversity in styles, patterns, and nuances of Arabic handwriting makes it particularly difficult to decipher. Arabic script, with its intricate letter formations and contextual variations, poses significant hurdles for recognition systems. Therefore, advancements in this area hold great potential for improving document analysis and expanding automated text processing capabilities for Arabic texts.

In a broader context, words are fundamental units made up of multiple characters, which form the essential elements of sentences. Understanding these structures involves decomposing words to analyze their role within sentences. During segmentation, words are isolated and processed by mapping from the page to the line, and then the word is extracted from the sentence in each line. Arabic words often consist of interconnected components, such as the term "taskheel" (تشكيل), which adds complexity to recognizing individual letters. Arabic letters vary in shape depending on their position at the beginning, middle, or end of words. The challenge in recognizing handwritten Arabic words is further compounded by the variability in letter forms, including initial, middle, and final configurations, as well as the presence of dots. This variability makes it difficult to achieve the precision and standardization typical of printed texts.

This paper proposes an enhanced method for Arabic handwritten word recognition using triangle geometry representation through coordinate repositioning. This proposed method is focusing on the repositioning of triangle coordinates to better capture the distinct features of Arabic words. We aim to improve the accuracy and efficiency of handwritten Arabic word recognition systems. These advancements hold significant potential for enhancing document analysis and expanding automated text processing capabilities for Arabic texts.

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue IX September 2025

Arabic is a well-known Semitic language that first appeared between the 1st and 4th centuries. Many ancient Arabic manuscripts have been discovered, but they are often in poor condition, with some damage due to the inferior parchment used at the time [1]. However, advanced technologies now enable researchers to extract crucial information from these ancient manuscripts, such as their originality, date of writing, provenance, and authorship.

Recognizing Arabic handwriting in these ancient manuscripts presents several challenges over time, particularly in terms of pattern, style, form, and handwriting type. The culture and socioeconomics of the manuscript's origin are also significant challenges. Different places and regions exhibit diverse handwriting styles, writing directions, vocabulary, and letter widths [2]. This diversity has contributed to the development of Arabic vocabulary and lexicons that aid in identifying the place of origin of Arabic handwriting. Arabic handwriting is analyzed through its letters, which helps researchers recognize Arabic letters from word handwriting. Some Arabic letters can change depending on the word formation. Therefore, studies on Arabic word recognition are widely explored.

Several advanced handwriting recognition methods have been developed and proposed to recognize Arabic word handwriting. These include multi-stage feature extraction [3], statistical geometric methods [4], neural network-based fusion [2], HOG and Gabor filter descriptor-based systems [5], a combination of convolutional neural networks and autoencoders (AECNN) [6], and multilevel pyramid models [7]. However, each researcher has different limitations in recognizing subword images due to the dataset used, the pattern and letters of the handwriting itself, and the writing order. The use of triangle geometry in pattern recognition has become widely accepted in various computer science fields. This geometric concept is employed in feature extraction techniques to obtain essential characteristics from images [5], [6], [8], [9], [10]. A triangle, defined by three interconnected edges and vertices [11], offers a versatile framework for extracting unique features. Triangles are generally categorized into three types: equilateral, isosceles, and scalene, each differentiated by the lengths of their sides. Despite the different types of triangles, one fundamental principle holds: the sum of the three angles always equals 180 degrees [12].

Furthermore, triangles can be classified based on their angles, resulting in types such as acute, right, and obtuse triangles. This dual classification system, based on side lengths and angles, enhances the utility of triangle geometry in pattern recognition. The application of triangle geometry features in handwriting recognition, specifically for digits [13], [14], calligraphy [15], [16], and document pages [17], [18], has shown multiple benefits. [13] constructed the triangle geometry shape using a binary representation method, allowing for the calculation of three points that define the triangle. [13] used properties derived from scalene triangles, including the ratios of sides, angles, and gradients, as proposed features for triangle geometry. These features effectively represented both digits and calligraphy characters.

The triangle geometry method has been adapted for recognizing Arabic words. In [13], the straight-line problem was addressed by adjusting the detected coordinate that caused the straight-line during triangle formation for digit recognition. The adjusting is performed by adding the value of 1 to the detected coordinate. However, the previous method described by [13] encountered issues with word recognition, as words consist of multiple characters, unlike digits, which are represented by a single character. The proposed method by [13] was applied to recognize Arabic words, but the straight-line problem persisted. This has led to a new investigation into the root causes of the straight-line problem in Arabic word recognition. Therefore, this study aims to propose an enhanced triangle geometry representation by repositioning the triangle coordinates to improve Arabic word recognition.

## PROPOSED METHOD

In this study, the proposed method is proposed to solve the straight-line problem that occur during constructing triangle shape. The process, as illustrated in Fig. 2, consists of three main steps, each detailed in the subsequent subsections. The three main steps are:

1. Extracting Triangle Points
2. Identifying Straight-Line Problem
3. Repositioning Selected Triangle Points
4. Extracting Triangle Features

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue IX September 2025

### Extracting Triangle Points

To extract features of triangle geometry, identifying the three points that form the triangle is crucial. In [13]'s work, the three points are referred to as Point A, Point B, and Point C, with Point C serving as the centroid. Point A is always positioned between Point C and the endpoint, while Point B is between the starting point and Point C. Before identifying Points A and B, it is essential to determine the centroid, denoted as Point C. The illustration of triangle shape is shown in Fig. 1 while the determination of the centroid relies on a set of rules established by [13], as illustrated in TABLE I.

TABLE I. Y-COORDINATE POSITION RULES

| Position | Rules |
|---|---|
| 1 | $yA \geq yC \geq yB$ |
| 2 | $yA \geq yB \geq yC$ |
| 3 | $yA \leq yC \leq yB$ |
| 4 | $yA \leq yB \leq yC$ |
| 5 | $yA \leq yB \leq yC$ |
| 6 | $yA \geq yC \leq yB$ |

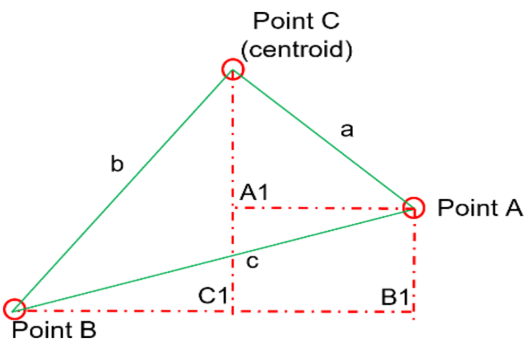


Fig. 1. Positions of point A, B and C

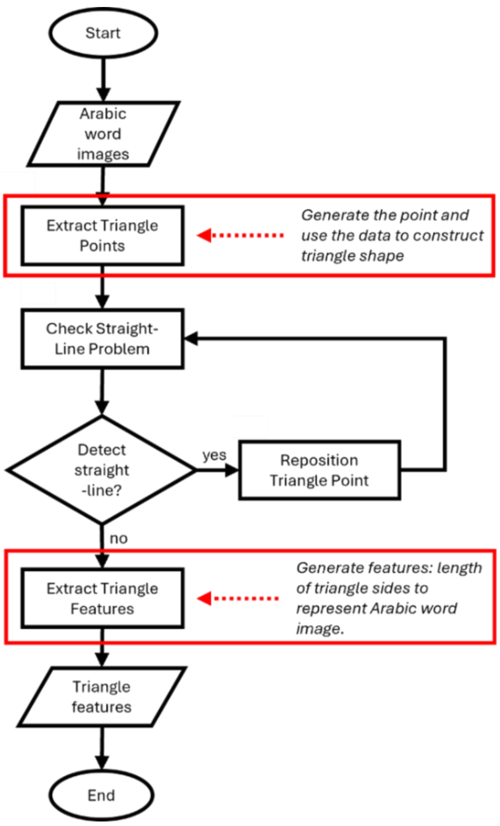


Fig. 2. Process flow of proposed method

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue IX September 2025

The formula for the centroid (Point C) based on [13] is shown in equations 1 and 2.

$$\bar{x} = \frac{1}{|R|} \cdot \sum(u, v)\epsilon R^{u} \quad (1)$$

$$\bar{y} = \frac{1}{|R|} \cdot \sum(u, v)\epsilon R^{v} \quad (2)$$

Where |R| represents a set of real numbers, u represents the width of binary image while v represents the height of binary image. The illustration of triangle shape on AHDB (Arabic handwritten word images) dataset was shown in Fig. 3.

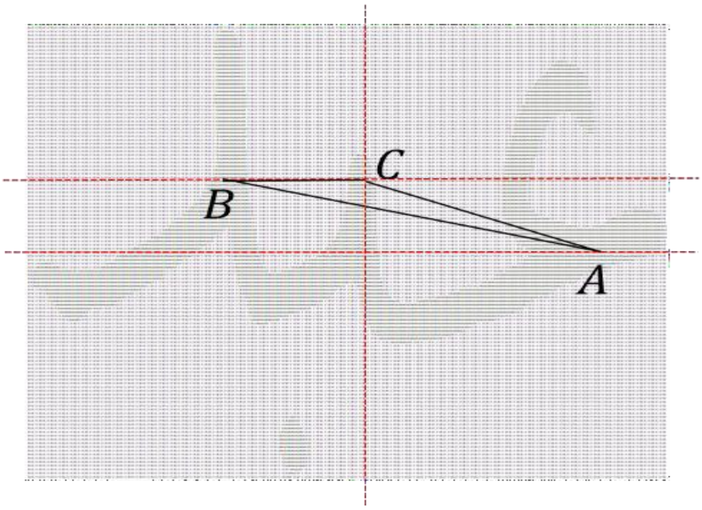


Fig. 3. Illustration triangle shape on AHDB image

**Identifying Straight-Line Problem**

The triangle geometry shape is constructed based on three interconnected points. However, a straight-line problem can occur during the formation of the triangle, preventing the shape from forming correctly. The formation process is disrupted when a specific triangle point produces a gradient value of 0. This problem is illustrated in Fig 4, which shows four situations where the straight-line problem was detected during the formation of triangles in the AHDB dataset.

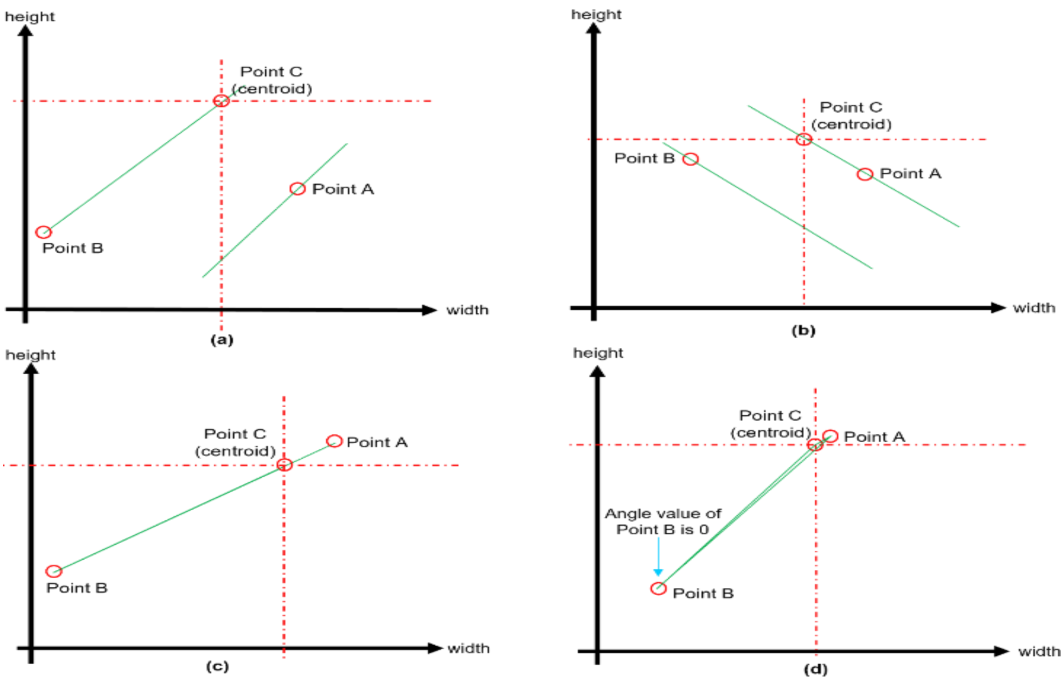


Fig. 4. Illustration of straight-line problems

In Fig. 4 (a), Points B and C are connected but Point A is not connected to Points B and C because the gradient value is 0. A similar issue occurs in Fig. 4 (b), but with a different triangle point. Fig. 4 (c) shows Points A, B, and C aligned, clearly causing a straight-line problem. In Fig. 4 (d), Points A, B, and C are connected, but Points A and C are too close to each other, making the triangle nearly a straight line and resulting in a gradient value of 0 for Point B. To address the issues observed in Fig. 4 (a-d), a benchmark using the angles within the triangle is applied. The angle values are crucial for identifying if a selected triangle point has an angle of 0, even when the triangle shape appears correctly formed. The formulas for calculating the angles at the triangle points are provided in Equations 3, 4, and 5, with the symbols a, b, and c corresponding to Fig. 3. The near-straight-line problem illustrated in Fig. 4 (d) typically arises when the coordinates of two triangle points are close to Point C. Specifically, this occurs when the x-coordinate ($\bar{x}$) or y-coordinate ($\bar{y}$) of any two triangle points differ by only one or two units.

$$\angle A = cos^{-1}\left(\frac{b^2+c^2-a^2}{2bc}\right) \quad (3)$$

$$\angle B = cos^{-1}\left(\frac{a^2+c^2-b^2}{2ac}\right) \quad (4)$$

$$\angle C = cos^{-1}\left(\frac{a^2+b^2-c^2}{2ab}\right) \quad (5)$$

**Repositioning Selected Triangle Points**

Repositioning triangle points is a proposed technique devised to improve prior solution in [13] for the straight-line problem encountered during digit recognition. This earlier solution aimed to resolve issues arising during the formation of triangle geometry shapes for digit images. Similarly, the straight-line problem also occurs in the formation of triangle geometry shapes for Arabic handwritten word images. According to the prior solution, a value of one was added to the coordinates of the triangle points causing the straight-line issue in digit images, suitable due to the small size and single-character nature of digits.

In contrast, for Arabic handwritten word images, which are larger and consist of multiple characters, an added value of ten is proposed to adjust the triangle points. The value of ten is determined appropriate which tested a range of values from 1 to 15 to increase the coordinate values without altering the triangle's midpoint or center. The maximum value of 15 is chosen to ensure the coordinate points do not fall outside the designated zone, avoiding outliers.

To avoid proposed coordinate values lies at an outlier of zone, a suitable range value is proposed. It is important to ensure the triangle geometry shape is formed within the zone so that the features can be extracted appropriately. Range of value is an indicator as the maximum value that can be used to propose suitable value for coordinate rearrangement without causing the proposed value of coordinate point to fall at outlier of zone. Therefore, range value is proposed based on height or width of selected triangle geometry point is as follows:

$$x\overline{pA}maxR = width\ of\ selected\ \overline{pA} \quad (6)$$

$$y\overline{pA}maxR = height\ of\ selected\ \overline{pA} \quad (7)$$

$$x\overline{pB}maxR = width\ of\ selected\ \overline{pB} \quad (8)$$

$$y\overline{pB}maxR = height\ of\ selected\ \overline{pB} \quad (9)$$

$\overline{pA}$ represents as Point A while $\overline{pB}$ represents as Point B. $x$ represents as $\bar{x}$-coordinate of Point while $y$ represents as $\bar{y}$-coordinate of Point. $maxR$ represents as maximum range of value. In this case, the coordinates of Point C remain unchanged due to Point C is a center point and any changes of coordinates Point C will affect division of zones process. The coordinate rearrangement is proposed after a suitable range of value is determined. The proposed coordinates can be either both coordinates of point, or any of coordinate of point depends on straight-line problem situation as shown in Fig. 4. The calculation of proposed coordinate rearrangement can be expressed as follows:

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue IX September 2025

$$cp\bar{S}_{cV} = cpS_{maxR}/2 \quad (10)$$

$$cp\bar{S}_{nc} = cp\bar{S}_{oc} + cp\bar{S}_{cV} \quad (11)$$

$$cp\bar{S}_{nc} = cp\bar{S}_{oc} - cp\bar{S}_{cV} \quad (12)$$

*cp* represents the selected coordinate of axis while $\bar{S}$ represents selected triangle geometry point and *maxR* represents maximum range of value. Next, *n* represents new proposed coordinate of rearrangement while *o* represents original coordinate and *V* represents proposed suitable value for coordinate rearrangement. The algorithm of proposed coordinate rearrangement is shown in Fig. 5.

1. Start
2. Read input:
2.1. Image: height and width,
2.2. Point $\overline{pA}$, Point $\overline{pB}$, Point $\overline{pC}$
3. Initialize:
3.1. $x\overline{pA}maxR$, $y\overline{pA}maxR$, $x\overline{pB}maxR$, $y\overline{pB}maxR$,
3.2. $x\overline{pA}cV = x\overline{pA}maxR/2$,
3.3. $y\overline{pA}cV = y\overline{pA}maxR/2$,
3.4. $x\overline{pB}cV = x\overline{pB}maxR/2$,
3.5. $y\overline{pB}cV = y\overline{pB}maxR/2$
4. If $\overline{pA}$ is close to $\overline{pC}$, then
4.1. Initialize $cp\bar{S}_{oc}$ = selected coordinate $\overline{pA}$
4.2. Initialize $cp\bar{S}_{cV}$ = $x\overline{pA}_{cV}$ or $y\overline{pA}_{cV}$
4.3. If $\overline{pB}$ is above, then
4.3.1. $cp\bar{S}_{nc} = cp\bar{S}_{oc} + cp\bar{S}_{cV}$
4.4. else if $\overline{pB}$ is bottom then
4.4.1. $cp\bar{S}_{nc} = c\overline{p\bar{S}}_{oc} - cp\bar{S}_{cV}$
5. else if $\overline{pB}$ is close to $\overline{pC}$, then
5.1. Initialize $cp\bar{S}_{oc}$ = selected coordinate $\overline{pB}$
5.2. Initialize $cp\bar{S}_{cV}$ = $x\overline{pB}_{cV}$ or $y\overline{pB}_{cV}$
5.3. If $\overline{pA}$ is above, then,
5.3.1. $cp\bar{S}_{nc} = cp\bar{S}_{oc} + cp\bar{S}_{cV}$
5.4. else if $\overline{pA}$ is bottom then,

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue IX September 2025

5.4.1. $cp\bar{S}_{nc} = cp\bar{S}_{oc} - cp\bar{S}_{cV}$

6. End If

7. End

Proposed method algorithm

**Extracting Triangle Features**

In this study, the extracted features were obtained using the triangle geometry method as described by [19]. This method represents features by the lengths of the triangle sides. The formula used by [19] for calculating the lengths of the triangle sides is as follows:

$$LS_a = \sqrt{(A_x - C_x)^2 + (A_y - C_y)^2} \qquad (13)$$

$$LS_b = \sqrt{(B_x - C_x)^2 + (B_y - C_y)^2} \qquad (14)$$

$$LS_c = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2} \qquad (15)$$

The formulas in Equations 13 to 15 have been modified according to the illustrated triangle shape in Fig. 1. By using Fig. 1, $LSa$ represents length of triangle side a, $LSb$ represents length of triangle side b and $LSc$ represents length of triangle side c. Meanwhile, A represents Point A, B represents Points B and C represents Point C. Using the feature extraction method described by [19], 99 features were extracted. To improve classification accuracy, [13] proposed a zoning method for digit recognition. The zoning method included Cartesian Plane Zone, Horizontal Plane Zone, Vertical Plane Zone, and 45º Zoning as shown in TABLE II.

TABLE II. SUMMARY OF ZONING METHOD BY [13]

| No | Zoning Types | Number of zones |
|---|---|---|
| 1 | Cartesian Plane Zone | 5 |
| 2 | Horizontal Plane Zone | 6 |
| 3 | Vertical Plane Zone | 14 |
| 4 | Zoning based 45º | 8 |
| **Total zones** | | **33** |

## DISCUSSION AND RESULT

The experimental evaluation conducted on the AHDB dataset [1] demonstrates the effectiveness of the proposed feature representations in Arabic handwritten word recognition. The AHDB dataset, introduced by [20], serves as a valuable benchmark for Arabic handwriting recognition research. By partitioning the dataset into 70% training and 30% testing subsets, the classifiers were evaluated under conditions that reflect their ability to generalize across different handwriting styles contributed by multiple writers. Two machine learning algorithms which are Support Vector Machine (SVM) and Random Forest (RF) were employed to assess the discriminative capability of the selected features. For the SVM classifier, parameter optimization was performed using a grid search approach on the cost (C) and kernel parameter (γ), which are crucial in determining classification performance. The best recognition accuracy was obtained with parameter settings of C = 32.0 and γ = 0.03125, indicating that careful tuning of these parameters is necessary to achieve optimal performance. This finding aligns with previous studies where SVM consistently outperformed other classifiers in handwriting recognition tasks when properly optimized.

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue IX September 2025

The comparative results in TABLE III highlight two important observations. First, the feature set consisting of three lengths of sides produced consistently higher recognition accuracy for both classifiers. This feature representation achieved 77.28% accuracy with SVM and 64.22% with RF, compared to 76.46% and 61.63%, respectively, for the feature set based on two sides and one angle. Although the difference in performance between the two feature sets is relatively small, the superiority of the three-sides feature suggests that it captures more distinctive structural information of handwritten words, thereby contributing to better class separability. Second, a clear performance gap exists between the two classifiers. SVM achieved substantially higher recognition rates than RF for both feature sets, outperforming it by approximately 13 to 15%. This result underscores the suitability of SVM for handwriting recognition tasks, particularly due to its ability to construct optimal decision boundaries in high-dimensional feature spaces. By contrast, RF, despite its robustness and capability in handling non-linear problems, appears less effective in capturing the subtle structural variations of Arabic handwritten words. This may be attributed to RF's reliance on majority voting among decision trees, which can sometimes oversimplify fine-grained distinctions present in handwriting data.

TABLE III. RESULT BASED ON SVM AND RF MEASUREMENT

| Features | SVM | RF |
|---|---|---|
| Three lengths of sides | 77.28% | 64.22% |
| Two sides and one angle | 76.468% | 61.63% |

Taken together, the results provide evidence that the proposed features, especially the three-sides length representation, are effective in addressing the straight-line problem in Arabic handwriting image recognition. The findings also reinforce the critical role of classifier selection and parameter optimization in improving recognition performance. These outcomes open several avenues for future research.

First, while the three-sides length features performed best in this study, combining them with angular and curvature-based features may further improve recognition accuracy by incorporating complementary shape information. Second, further refinement of classification techniques and parameter optimization methods could be explored to enhance performance and stability. Finally, testing on larger and more diverse Arabic handwriting datasets would strengthen the generalizability of the findings and validate the robustness of the proposed approach across broader writing variations.

This can be concluded the study demonstrates that SVM, with optimized parameters, provides a superior classification framework compared to RF for Arabic handwritten word recognition. Moreover, the effectiveness of the three-sides length features highlights the potential of geometric-based descriptors in solving fundamental challenges such as the straight-line problem in handwriting analysis

## CONCLUSION

In this study, the focus is on addressing the straight-line problem that arises during the construction of triangle geometric features for Arabic handwritten word recognition. This problem occurs when the extracted coordinate points used to form triangles do not generate valid geometric shapes, thereby hindering the effectiveness of feature extraction. Building on prior research, an enhanced method is proposed as a value-added solution by rearranging coordinate points whenever a straight-line problem is detected. The aim is to ensure that meaningful and discriminative triangular features can still be constructed, even under problematic configurations.

Through detailed analysis, four distinct types of straight-line problems were identified. The first situation occurs when all three coordinate points are collinear, resulting in a degenerate triangle with zero area. The second and third situations arise when two sides become parallel, leading to overlapping or indistinguishable angles that reduce feature distinctiveness. The fourth situation represents a special case in which one of the points (either A or B) lies very close to point C, which acts as the midpoint. This proximity prevents the proper extraction of angular features, as the triangle becomes too narrow or collapses into a line-like structure. Among these, the fourth case required further investigation, as it uniquely impacts the stability of angle-based descriptors.

To overcome these challenges, the method dynamically rearranges the coordinate points to form a new, valid triangle whenever one of these problematic situations is detected. This adaptive rearrangement ensures that degenerate configurations are corrected, and meaningful geometric features can still be extracted. By addressing each of the four problem types, the proposed technique guarantees robustness in triangle formation and preserves the discriminative power of the features. The effectiveness of the method was validated using Support Vector Machine (SVM) and Random Forest (RF) classifiers. The experimental results showed that solving the straight-line problem not only improved the reliability of triangle-based feature extraction but also enhanced the overall recognition accuracy of Arabic handwritten word images.

In particular, SVM, when optimized with appropriate cost and gamma parameters, outperformed RF, indicating that the proposed geometric correction is most beneficial when combined with classifiers capable of handling high-dimensional feature spaces. The contribution of this study is twofold. First, it provides a clear categorization of straight-line problems in geometric feature construction, offering deeper insight into the causes behind degenerate triangle formations. Second, it demonstrates a practical corrective solution that ensures valid triangle shapes can always be constructed, thereby improving the robustness of handwriting recognition systems. This is particularly important in the context of Arabic handwriting, where variability in stroke placement and writing styles can frequently lead to problematic point configurations.

Overall, the proposed method highlights the importance of addressing geometric inconsistencies in feature extraction. By eliminating the limitations posed by straight-line problems, the approach contributes to more accurate, consistent, and reliable recognition of Arabic handwritten words, paving the way for further enhancements in feature design and classification techniques.

## ACKNOWLEDGMENT

The authors would also like to thank University Technical Malaysia Melaka for the support and research facilities provided.

## REFERENCES

1. Y. Chherawala and M. Cheriet, “Arabic word descriptor for handwritten word indexing and lexicon reduction,” Pattern Recognit, vol. 47, no. 10, pp. 3477–3486, 2014, doi: 10.1016/j.patcog.2014.04.025.
2. S. A. A. A. Arani, E. Kabir, and R. Ebrahimpour, “Handwritten Farsi Word Recognition Using NN-Based Fusion of HMM Classifiers with Different Types of Features,” Int J Image Graph, vol. 19, no. 1, pp. 1–21, 2019, doi: 10.1142/S0219467819500013.
3. K. Ubul, A. Adler, and M. Yasi, “Multi-Stage Based Feature Extraction Methods for Uyghur Handwriting Based Writer Identification,” Genetic Algorithms in Applications, 2012, doi: 10.5772/36623.
4. R. Tavoli, M. Keyvanpour, and S. Mozaffari, “Statistical geometric components of straight lines (SGCSL) feature extraction method for offline Arabic/Persian handwritten words recognition,” IET Image Process, vol. 12, no. 9, pp. 1606–1616, 2018, doi: 10.1049/iet- ipr.2017.0839.
5. S. Hamida, B. Cherradi, and H. Ouajji, “Handwritten Arabic Words Recognition System Based on HOG and Gabor Filter Descriptors,” 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology, IRASET 2020, pp. 1–4, 2020, doi: 10.1109/IRASET48871.2020.9092067.
6. S. Khosravi and A. Chalechale, “Recognition of Persian/Arabic Handwritten Words Using a Combination of Convolutional Neural Networks and Autoencoder (AECNN),” Math Probl Eng, vol. 2022, 2022, doi: 10.1155/2022/4241016.
7. A. Korichi, S. Slatnia, O. Aiadi, and B. Khaldi, “A generic feature- independent pyramid multilevel model for Arabic handwriting recognition,” Multimed Tools Appl, vol. 81, no. 15, pp. 20719–20739, Jun. 2022, doi: 10.1007/s11042-022-11979-0.
8. K. Okada, M. Kudo, and H. Yamana, “Estimating Answer Strategies using Online Handwritten Data: A Study using Geometry Problems,” in ACM International Conference Proceeding Series, Association for Computing Machinery, Sep. 2023, pp. 308–314. doi: 10.1145/3629296.3629347.

9. H. Kaur and M. Kumar, “On the recognition of offline handwritten word using holistic approach and AdaBoost methodology,” Multimed Tools Appl, vol. 80, no. 7, pp. 11155–11175, Mar. 2021, doi: 10.1007/s11042-020-10297-7.
10. S. Boroojerdi and G. Rudolph, “Handwritten Multi-Digit Recognition With Machine Learning,” in 2022 Intermountain Engineering, Technology and Computing, IETC 2022, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/IETC54973.2022.9796722.
11. A. C. Robin, “93.39 The Most Scalene Triangle.,” The Mathematical Gazette, vol. 93, pp. 331–338, 2009, doi: 10.1080/00029890.1926.11986542.
12. M. A. Ronan, “Triangle geometries,” J Comb Theory Ser A, vol. 37, no. 3, pp. 294–319, 1984, doi: 10.1016/0097-3165(84)90051-7.
13. Mohd Sanusi Azmi, “Fitur Baharu Dari Kombinasi Geometri Segitiga Dan Pengezonan Untuk Paleografi Jawi Digital,” Doctoral dissertation, Universiti Kebangsaan Malaysia, 2013.
14. S. A. Miswan, M. S. Azmi, N. A. Arbain, A. Tahir, and A. R. Radzid, “Rearrangement of Coordinate Selection for Triangle Features Improvement in Digit Recognition,” Journal of Telecommunication, Electronic and Computer Engineering, vol. 10, no. 2–7, pp. 115–119, 2018.
15. M. S. Azmi, K. Omar, M. F. Nasrudin, A. K. Muda, and A. Abdullah, “Arabic calligraphy classification using triangle model for Digital Jawi Paleography analysis,” Proceedings of the 2011 11th International Conference on Hybrid Intelligent Systems, HIS 2011, no. July, pp. 704–708, 2011, doi: 10.1109/HIS.2011.6122194.
16. M. S. Azmi, K. Omar, M. F. Nasrudin, and A. K. Muda, “Features Extraction of Arabic Calligraphy using extended Triangle Model for Digital Jawi Paleography Analysis,” International Journal of Computer Information Systems and Industrial Management Applications, vol. 5, pp. 696–703, 2013.
17. A. Tahir, M. S. Azmi, N. Ahmad, N. A. Arbain, and A. R. Radzid, “Document Feature Extraction Based on Unoccupied Space Using Triangle Model : A Preliminary Work,” Journal of Telecommunication, Electronic and Computer Engineering, vol. 10, no. 2–7, pp. 97–101, 2018.
18. A. R. Radzid, M. S. Azmi, I. E. A. Jalil, R. Nordin, and N. Atikah, “Multiform Frame Shape Extraction Using Neighbouring Pixel Be- Haviors for Mushaf Al-Quran,” International Journal of Advanced Trends in Computer Science and Engineering, vol. 8, no. 1, pp. 251– 256, 2019.
19. N. A. Arbain, M. S. Azmi, A. K. Muda, A. R. Radzid, and A. Tahir, “Triangle Shape Feature based on Selected Centroid for Arabic Subword Handwriting,” International Journal of Advanced Computer Science and Applications, vol. 9, no. 12, pp. 169–173, 2018.
20. S. Al-Ma'adeed, D. Elliman and C. A. Higgins, "A data base for Arabic handwritten text recognition research," Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition, Niagra-on-the-Lake, ON, Canada, 2002, pp. 485-489, doi: 10.1109/IWFHR.2002.1030957.