Identification of paleographic curvature using skeletonization and key point detection

Fadhilatul Fitriyah[1], Dian Andriana[2], Muhammad Zulhaj Aliansyah[3], Lukman Hakim[1], Muhammad Faishol Amrulloh[1]
[1]Department of Informatics Engineering, Faculty of Engineering, Universitas Yudharta Pasuruan, Pasuruan, Indonesia
[2]Research Center for Artificial Intelligence and Cyber-Security, National Research and Innovation Agency (BRIN), Bandung, Indonesia
[3]Department of Data Science, Faculty of Computer Science, Universitas Pembangunan Nasional Veteran Jawa Timur, Surabaya, Indonesia

Article Info

Article history:

Received Aug 25, 2025
Revised Dec 27, 2025
Accepted Jan 30, 2026

Keywords:

Aksara Jawi
Connected component
Key points
Manuscript
Paleography
Segmentation
Skeletonization

ABSTRACT

Jawi script represents a vital component of the Islamic intellectual heritage of the Nusantara, preserved across numerous classical manuscripts. A primary challenge in digitizing these documents is character segmentation, particularly where handwritten characters connect without distinct boundaries. This research proposes a skeletonization-based segmentation method to address this issue, utilizing a dataset from 17 pages of the "Kitab Syair Perahu" manuscript containing 269 test characters. The pre-processing stage involves grayscale conversion, binarization, and noise removal through connected component analysis (CCA). The segmentation process then integrates skeleton structures, centroid positioning, intersection points, and loop detection. Evaluation results show the system successfully identified 187 out of 269 characters, achieving an accuracy of 0.801, a precision of 0.895, a recall of 86.38%, and an F1-score of 88.91%. While these results demonstrate the method's effectiveness, the small dataset from a single manuscript limits its generalizability. Nevertheless, this study establishes a foundational step toward an automated Jawi image-processing system and the digital preservation of Islamic Nusantara literacy, contributing a tailored skeletonization-based approach for Jawi script.

Corresponding Author:

Muhammad Faishol Amrulloh
Department of Informatics Engineering, Faculty of Engineering, Universitas Yudharta Pasuruan
Jl. Yudharta No.7, Kembang Kuning, Sengonagung, Purwosari, Pasuruan, East Java 67162, Indonesia
Email: faishol@yudharta.ac.id

1. INTRODUCTION

Turats Nusantara refers to the Islamic intellectual heritage preserved in manuscripts written in adapted Arabic scripts, notably Pegon and Jawi [1], [2]. Etymologically, turats (from the root wa-ra-tsa) denotes inherited knowledge transmitted across generations [3]. Beyond mere adoption, Jawi script represents a sophisticated phonological adjustment of Arabic letterforms to suit regional languages while maintaining complex cursive characteristics [4]-[7].

Despite its historical significance, the digitalization of Jawi manuscripts remains challenging [8]. A major hurdle lies in character segmentation within optical character recognition (OCR) pipelines, as Jawi is inherently cursive and context-sensitive [9], [10]. Unlike Latin text, Jawi characters change shape based on their position initial, medial, or final and connect through shared strokes, often lacking clear boundaries [11]-[13]. These traits, coupled with non-uniform diacritics, handwriting variability, and the scarcity of Jawi-specific datasets, make segmentation a complex yet vital task for the digital preservation of Malay-Islamic manuscripts [14]-[20].

Previous segmentation studies have primarily focused on Arabic script. Seam carving approaches can handle touching characters but are highly sensitive to noise and image quality [21]. Morphological and connected component-based methods perform well on structured text but struggle with handwritten documents due to their complexity and input dependency [22]. Hybrid deep learning approaches combining convolutional neural networks (CNNs) with thinning or segmentation hypothesis graphs achieve high accuracy but require large annotated datasets and involve significant computational complexity [23]. A summary of representative segmentation and recognition approaches, their datasets, strengths, limitations, and relevance to Jawi script is presented in Table 1.

Research explicitly targeting Jawi script remains limited. Existing works largely focus on isolated character recognition using Freeman chain code combined with rule-based classifiers or support vector machines (SVMs), reporting high accuracy under controlled conditions [24], [25]. However, these approaches do not address the segmentation of connected handwritten characters in authentic manuscript settings.

Table 1. Related research

Reference	Method	Dataset	Strengths	Limitations	Relevance to Jawi
[21]	Seam carving	Islamic Educational, Scientific and Cultural Organization – Arabic Database (IESK-ArDB, Institut für Nachrichtentechnik / École Nationale d'Ingénieurs de Tunis (Arabic Handwritten Word Database) (IFN/ENIT)	Handles touching characters	Noise-sensitive; energy-dependent	Arabic-only
[22]	Morphological + CC analysis	Arabic handwritten docs	Effective for structured text	Quality-sensitive; multi-stage	Arabic-only
[23]	CNN + SHG + thinning	Arabic handwritten	High accuracy; confidence-based	Complex; data-dependent	Arabic-only
[24]	FCC + regular expression + Decision tree	Isolated Jawi (live writing, 10 users))	Real-time; reliable (α = 0.839)	Isolated only; Android-only	Jawi-specific (isolated)
[25]	FCC + SVM + decision rules	Isolated handwritten Jawi	High accuracy (92.86%)	Similar-shape confusion	Jawi-specific (isolated)
This work	Skeletonization + keypoint detection	Jawi manuscript "Syair Perahu"	Lightweight; interpretable; robust	Errors on complex ligatures	Jawi manuscript

However, most of these studies remain limited to the standard Arabic script and have not been specifically applied to Jawi, which exhibits distinct structural complexities and character variations. One of primary challenges in Jawi script segmentation is the process of segments of connected characters, whose ligature structures are often irregular and highly variable. Therefore, there is a need for an approach capable of accurately identifying character boundaries even under complex connection patterns.

Compared with existing approaches, the proposed method offers a lightweight and interpretable alternative for Jawi character segmentation. Morphology-based methods perform well on printed text but struggle with irregular handwritten ligatures [22], while seam carving approaches are highly sensitive to noise and image degradation [21]. CNN-based methods require large annotated datasets and provide limited interpretability, which is impractical for Jawi manuscripts [23], [26]. In contrast, this study employs skeletonization and keypoint detection using intersection, loop, and centroid cues to robustly segment characters in small and noisy manuscript datasets.

Therefore, an alternative segmentation strategy is required that is lightweight, interpretable, and effective on small, noisy datasets. This study proposes a skeletonization-based segmentation method that integrates connected component analysis (CCA) with keypoint detection, including centroids, intersection points, and loops, to identify character boundaries within complex ligatures. Beyond improving segmentation accuracy, the method emphasizes palaeographic curvature characteristics of Jawi script as structural cues for character isolation, supporting manuscript retrieval and digital preservation of Islamic Nusantara heritage.

2. METHOD

In this research, a Jawi script letter segmentation method is proposed that integrated skeletonization technique to extract the basic structure of character strokes CCA to identified connected area in an image, and bounding box as the spatial boundaries of each component analysed [27]. This research is composed of several main steps, include pre-processing, skeletonization, CCA, bounding box definition, and character segmentation based on the key points identification. The overall flow of the character segmentation process in this research is represented in Figure 1.

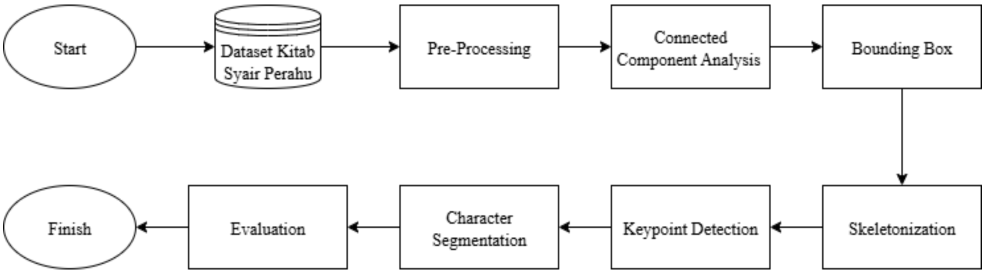

Figure 1. Flow of research

2.1. Dataset

The dataset consists of scanned pages of the “Kitab Syair Perahu” manuscript written in Jawi script [28]. The scanning process was conducted by the Research Center for Artificial Intelligence and Cybersecurity, National Research and Innovation Agency (BRIN). The manuscript comprises 17 pages in PNG format. Ground truth labeling was performed independently by four santri proficient in Jawi reading. Disagreements among annotations were resolved through consensus discussions facilitated by the Turats Division of Universitas Yudharta Pasuruan, which also conducted final validation. This process ensured both technical accuracy and consistency with turats scholarship. A sample line image is shown in Figure 2.

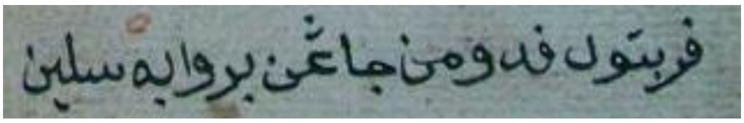

Figure 2. Line excerpt from “Kitab Syair Perahu” page 04 line 15

2.2. Pre-processing

Pre-processing begins with RGB-to-grayscale conversion to reduce image complexity, followed by binarization using Otsu’s thresholding method [29]-[33]. Otsu’s method automatically determines the optimal threshold by maximizing between-class variance, enabling clear separation between foreground text and background. The binarization process is defined as:

$$g(x,y) = \begin{cases} 1\ (white), if\ f(x,y) \geq T \\ 0\ (black), if\ f(x,y) < T \end{cases} \tag{1}$$

$g(x,y)$ is binarization result at pixel coordinates (x,y), $f(x,y)$ intensity value at the pixel position (x,y) in the grayscale image, and T is the threshold computed using Otsu’s criterion.

2.3. CCA

After binarization, CCA is applied to remove noise and identify relevant character components based on 8-neighborhood connectivity. Components are classified according to area thresholds: components larger than 50 pixels are considered main strokes, while components between 5 and 50 pixels are retained as potential diacritics if located near the main stroke. Smaller or distant components are discarded as noise. This filtering step preserves essential character structures for further analysis. The component labeling rule can be expressed as in (2).

$$Label_a = \begin{cases} Main\ stroke,\ \ if\ \ T_a \leq A \\ Diacritics,\ \ \ \ \ if\ \ T_a \geq A \end{cases} \tag{2}$$

$Label_a$ is the label category for component a, A represents the component area, T_a is the area threshold, e.g. 50. A component is considered a main stroke if it is assumed to be part of the character, while diacritics refer to small components such as dots or harakat.

2.4. Bounding box

In this step, the system identified character components using bounding box analysis derived from the CCA results, which is used to determine the boundaries of each object in the image [34], [35]. Large components or those with a vertical aspect ratio are assumed to be main strokes, while smaller components that do not meet these criteria are considered as diacritic candidates [36].

To associate diacritics with their corresponding main characters, the Euclidean distance between the centroid of a small component and the centroid of the nearest main stroke is calculated. The distance is defined as in (3):

$$e = \sqrt{(C_d - C_m)^2 + (C_d - C_m)^2} \tag{3}$$

e is the Euclidean distance between diacritic component and main stroke, (C_d, C_d) are the centroid coordinates of diacritic component, and (C_m, C_m) are the centroid coordinates of main stroke component. A small component is associated with a main stroke as a diacritic if its centroid lies within 60 pixels of the main stroke's centroid. Furthermore, the vertical position of the diacritic centroid relative to the main stroke centroid is used to determine whether it is located above or below main stroke [37]. This classification can be as in (4):

$$Label_u = \begin{cases} Above\, if\ v_d < v_m \\ Below\, if\ v_d > v_m \end{cases} \tag{4}$$

$Label_u$ is the label category for component u, v_d is the vertical coordinate of the diacritic centroid, v_m is the vertical coordinate of the main stroke centroid. If $(v_d < v_m)$, the diacritic is located above the main stroke, and $(v_d > v_m)$, the diacritic is located below the main stroke. It is important to note that in the digital image coordinate system, the origin point (0, 0) is located at the top-left corner, so smaller y values indicate higher positions in the image. This makes centroid comparison a reliable method for distinguishing diacritics positioned above or below the main stroke.

2.5. Skeletonization

Skeletonization is applied to the main stroke components to reduce stroke thickness to a one-pixel-wide representation while preserving topological structure [38]-[41]. This step simplifies character shapes and facilitates keypoint detection without losing essential palaeographic features [42]-[44]. Two skeletonization algorithms Zhang Suen and Lee were evaluated. The Zhang Suen algorithm, a two-subiteration thinning method, was selected due to its efficiency in preserving connectivity and minimizing spurious branches [45], [46]. The skeletonization process can be mathematically expressed as:

$$S(B) = \{p \in B \mid p \in \text{medial axis}(B) \wedge (B \setminus \{p\}) \text{ remains connected}\} \tag{5}$$

$S(B)$ is the binary object, p represents a pixel in object B, medial axis (B) denotes the medial axis of object B. $S(B)$ is the set of pixels in object B that lie on the medial axis and can be deleted without breaking object connectivity. This definition forms the foundation of the skeletonization process, ensuring that the topological structure of characters is preserved. In the Zhang-Suen algorithm, a pixel p is deleted if it satisfies the following conditions. These conditions ensure that the skeleton remains one pixel wide while preserving connectivity.

Delete pixel p if:

- $2 \leq \mathrm{N}(p) \leq 6$
  $\mathrm{N}(p) =$ number of 8 neighbors of pixel p
- $\mathrm{Z}\,(p) = 1$
  $\mathrm{Z}\,(p) =$ number of transitions from 0 to 1 in the clockwise order of the 8 neighbors
- $p_2.p_4.p_6 = 0$ step 1
- $p_2.p_4.p_6 = 0$ step 2
  $p_2.p_4.p_6 =$ position of the pixel's neighbors in the clockwise direction

To strengthen the justification for the chosen method, a comparative evaluation was also conducted using the Lee algorithm. The performance of Zhang-Suen and Lee skeletonization was quantitatively assessed using three metrics: execution time, number of spurious branches, and pixel reduction ratio (PRR).

This comparison provides a comprehensive basis for selecting the most suitable algorithm for Jawi character segmentation in this research.

2.6. Keypoint detection

After skeletonization of the main strokes, structural key points are extracted to support detailed shape analysis and writing direction detection. The detected key points include start points, end points, intersection points, turn points, centroid points, and loop. Their definitions are as in:

2.6.1. Start point

The skeleton point with one active neighbor in the 3×3 window and located at the top-right position (maximum column value and minimum row value) is defined as the start point. This point is selected because it represents the natural writing direction of Arabic or Jawi script, which begins from the right. Mathematically, the definition can be formulated as in:

$$degree(p) = 1 \tag{6}$$

$$Start\ point = arg\max_{p}(x(p) - y(p)|degree(p) = 1) \tag{7}$$

p is a skeleton pixel, $degree(p)$ represents the number of connected neighbors of pixel p and $x(p) - y(p)$ are the x and y coordinate values of pixel p in the image coordinate system. A pixel p is defined as a starting point if it is an endpoint, i.e., connected to only one neighbor in the skeleton graph ($degree(p) = 1$). Among all such pixels, the one with the largest $(x - y)$ value is chosen as the top-right position for traversal.

2.6.2. End point

A skeleton pixel with one active neighbor, located at the far left (minimum column value) and bottom (maximum row value), is defined as the end point. This point indicates the geometric termination of the character shape. Mathematically, the definition can be formulated as in:

$$degree(p) = 1 \tag{8}$$

$$End\ point = arg\min_{p}(x(p) - y(p)|degree(p) = 1) \tag{9}$$

As with the start point, the end point is also a pixel p that is connected to only one neighbor ($degree(p) = 1$). The difference from the start point is that the end point is located at the bottom-left of the skeleton, determined by a small x (leftmost) and large y (lowest).

2.6.3. Intersection point

A skeleton pixel with three or more active neighbors is defined as an intersection point. This point represents branching in the character structure, which is typical in complex or curved shapes. Mathematically, the definition can be expressed as (10).

$$degree(p) \geq 3 \tag{10}$$

An intersection point has high connectivity, as it is connected to three or more neighboring pixels in the skeleton graph.

2.6.4. Turn point

A skeleton pixel with one horizontal neighbor and one vertical neighbor is defined as a turn point. This point corresponds to a sharp change in direction, typically close to a 90° angle, and is often found in curved or angular characters. Mathematically, a turn point can be defined as (11).

$$\theta = cos^{-1}\left(\frac{(p_i - p_{i-1}) \cdot (p_{i+1} - p_i)}{\|p_i - p_{i-1}\| \, \|p_{i+1} - p_i\|}\right) \tag{11}$$

If p_{i-1}, p_i, p_{i+1} are three consecutive pixels along the skeleton path, then pixel p_i is considered a turn point if:

$$\theta > \theta_{threshold} \tag{12}$$

where $\theta_{threshold}$ is a predefined threshold angle (commonly between 30° and 60°) used to detect significant curvature changes.

2.6.5. Centroid point

The centroid point represents the average position of all skeleton pixels within a character's bounding box. It serves as a global descriptor of the character's position and is also used to establish the relationship between the main stroke and its diacritics. Mathematically, if there are n skeleton points, the centroid coordinates can be computed as:

$$C_x = \frac{1}{n}\sum_{k=1}^{n} x_k \,, C_y = \frac{1}{n}\sum_{k=1}^{n} y_k \quad (13)$$

C_x and C_y are the centroid coordinates representing the average x and y positions of all skeleton pixels, x_k and y_k are the x and y coordinate values of the k-th skeleton pixel in the image coordinate system. Thus, the centroid is defined as $C = (C_x, C_y)$. This point can be used to normalize the position of each character and as a feature in determining relative positions or for character classification.

2.6.6. Loop

A skeleton loop is defined as a closed cycle detected in the character skeleton graph. To ensure accuracy, the system applies a filtering process that distinguishes true loop from noise or small turns. Each cycle is evaluated using the Convex Hull method, and only those with a sufficiently large area are classified as valid loop. Mathematically, loop identification can be expressed as:

$$Loop_w = \{h \in cycle\_basis(\mathcal{G}) \,|\, area(ConvexHull(h)) > \theta\} \quad (14)$$

$\mathcal{G}$ is the skeleton graph representing the connectivity of pixels, h is a A cycle (closed path) extracted from the cycle basis of graph $\mathcal{G}$, and $cycle\_basis(\mathcal{G})$ denotes the set of all fundamental cycles in the skeleton graph. $ConvexHull(h)$ represents the convex hull formed by the pixels belonging to cycle h, and $area(ConvexHull(h))$ is the area enclosed by the convex hull of cycle h. Θ is the minimum area threshold used to filter out small or noisy cycles, and Loop_w is the set of valid loop whose areas exceed the threshold θ.

Where $Loop_w$ denotes the set of valid loop, c represents a cycle from $cycle\_basis(\mathcal{G})$, and θ is the minimum area threshold (set to 1.5 in this study). Cycles with an area greater than θ are retained as valid loop, while smaller ones are discarded as noise. This approach ensures that only loop reflecting true circular character structures are preserved in the analysis.

2.7. Character segmentation

Character segmentation was performed after identifying key points in the skeleton structure, using three approaches: character segmentation is conducted after skeleton-based keypoint extraction using a three-stage strategy. First, connected component labeling (CCL) is applied to directly segment isolated characters, with centroids used to guide cropping and associate diacritics with the main stroke. Second, for connected components, skeleton intersection points are employed as structural cut cues, with cuts applied one pixel before the intersection to preserve stroke continuity. Third, loop (closed-path) analysis is integrated to resolve ambiguities in multi-character connections: cropping is adjusted relative to the intersection position depending on loop location, double cuts are applied for three-character connections, and components without loops and with only two endpoints are retained as single characters. This combined strategy enables accurate and context-aware segmentation of complex handwritten Jawi characters.

While these segmentation methods (e.g., connected component, skeletonization, and keypoint detection) are not entirely new, the novelty of this work lies in their adaptation to the Jawi script. Previous studies have provided important insights into Arabic script segmentation, but they remain largely limited to standard Arabic and have not been specifically applied to Jawi. The Jawi script introduces distinct structural complexities: its characters exhibit more intricate structures, diacritic placement differs, handwriting thickness varies considerably, and ligatures are often irregular. These characteristics make the segmentation of connected characters particularly challenging, thereby requiring a more robust and adaptive approach to accurately identify character boundaries under complex conditions.

Although this research focuses on character segmentation, the proposed method can be integrated into a complete OCR workflow. Segmented characters serve as input for later stages such as feature extraction, classification, and post-processing, enabling the recognition of Pegon script in historical manuscripts. This integration highlights the practical application of segmentation methods in broader OCR systems.

2.8. Evaluation

This evaluation stage was conducted to measure how well the segmentation method separated Jawi characters correctly. The evaluation was carried out by comparing the automatic segmentation results generated by the system with the manually defined ground truth data. This process used several standard evaluation metrics in the field of pattern recognition, namely accuracy, precision, recall, and F1-score [47].

3. RESULTS AND DISCUSSION

3.1. Pre-processing

Figures 3(a) to (c) illustrates the pre-processing stages, including grayscale conversion, thresholding, and noise removal using CCA. These steps effectively enhance foreground-background separation and provide a clean basis for segmentation. However, some diacritics are removed during CCA due to their small size and spatial separation resembling noise, which affects characters that rely on dot patterns. This limitation indicates that fixed threshold settings are not fully aligned with Jawi paleographic characteristics, suggesting the need for adaptive diacritic-preservation strategies to improve segmentation consistency.

(a) (b) (c)

Figure 3. Step of pre-processing: (a) grayscale image result, (b) thresholding image result, and (c) cleaned thresholding image result

3.2. Bounding box

Bounding box analysis successfully distinguishes main strokes and diacritics, as visualized in Figure 4, preserving their spatial relationships, which are critical for Jawi recognition. Nevertheless, diacritics located far from the main stroke are sometimes misclassified as noise or incorrectly associated due to overlapping strokes. This limitation reduces segmentation accuracy for dot-sensitive characters. Adaptive distance thresholds normalized to character size and the incorporation of Jawi-specific linguistic rules are recommended to improve robustness across handwriting styles and manuscript quality variations.

Figure 4. Visualization of diacritics and main strokes

3.3. Skeletonization

The skeletonization results are presented in Figure 5, comparing the Zhang-Suen and Lee (1994) algorithms. Zhang-Suen Figure 5(a) produces thin skeletons and preserves main stroke structures efficiently, but introduces excessive branches that increase visual roughness. In contrast, the Lee algorithm Figure 5(b) generates smoother skeletons with fewer branches, although some fine structural details are lost due to oversimplification.

(a) (b)

Figure 5. Step of skeletonization: (a) Zhang-Suen skeletonization algorithm and (b) Lee 94 skeletonization algorithm

Quantitative results in Table 2 show that Zhang-Suen achieves faster processing time (0.28 ms) than Lee (0.70 ms), but produces more excessive branches (35 vs. 27). Both methods yield identical PRR values of 31%. This indicates a clear trade-off between structural detail preservation and skeleton smoothness. Considering its efficiency and comparable PRR, Zhang-Suen was selected as the preferred method, with the acknowledgment that branch-pruning post-processing is required to mitigate noise and prevent segmentation errors. Future work should investigate adaptive or hybrid skeletonization strategies to balance efficiency and structural fidelity for Jawi character recognition.

Table 2. Comparison of skeletonization algorithms

Algorithm	Processing time (ms)	Excessive branches	PRR (%)	Notes
Zhang-Suen	0.28	35	31	Faster, but generates more noise/branches
Lee (1994)	0.70	27	31	Cleaner skeleton, but slower and loses details

3.4. Key point detection

Key point detection results shown in Figure 6 demonstrate reliable identification of start points, end points, centroids, and intersections, effectively supporting character segmentation. However, turn points are not successfully detected, as curved strokes are often misinterpreted as intersections. In practice, this limitation has minimal impact on segmentation reliability because segmentation decisions primarily rely on intersection and centroid points rather than turn points. No explicit correction strategy was applied at this stage; instead, segmentation robustness was maintained by prioritizing stable keypoints and excluding ambiguous turn-point detections. This limitation indicates that the current rule-based approach is insufficient to capture curvature variations. Future improvements should incorporate curvature-sensitive analysis or data-driven methods to enhance the detection of directional changes in Jawi character strokes.

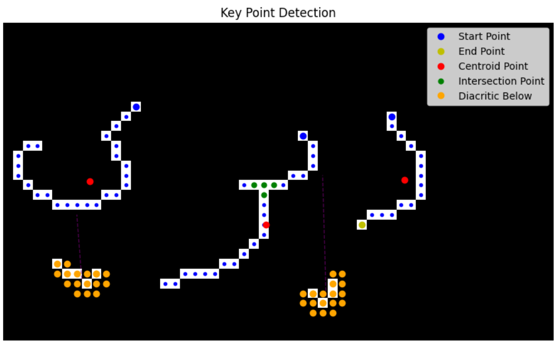

Figure 6. Key point detection results for the word "ديري"

3.5. Character segmentation

In the character segmentation section, the characters in the figure are arranged according to the natural Jawi reading direction, from right to left. However, the subfigure labels follow the standard journal convention of left-to-right ordering.

3.5.1. Based on centroid of CCL

This section presents the segmentation results obtained using CCL. The method is designed to separate individual Jawi characters by identifying components without structural connections and extracting them as single units. To improve segmentation consistency, the centroid of each connected component, representing its spatial center of mass, was used as a reference. This allows the system to separate the main strokes from diacritical elements such as dots or harakat more precisely. Figure 7(a) shows *ya'*, Figure 7(b) shows *ya'* and *ra'*, and Figure 7(c) shows *dal*. These examples demonstrate that CCL performs effectively on isolated components with no structural overlap.

As shown in Figure 7, single characters were segmented correctly when their structures were not connected, demonstrating the effectiveness of CCL in handling isolated strokes. However, segmentation failures were observed in tightly connected characters, such as *ya'* and *ra'* (Figure 7(b)), where adjacent centroids were too close, causing the system to interpret multiple characters as a single unit.

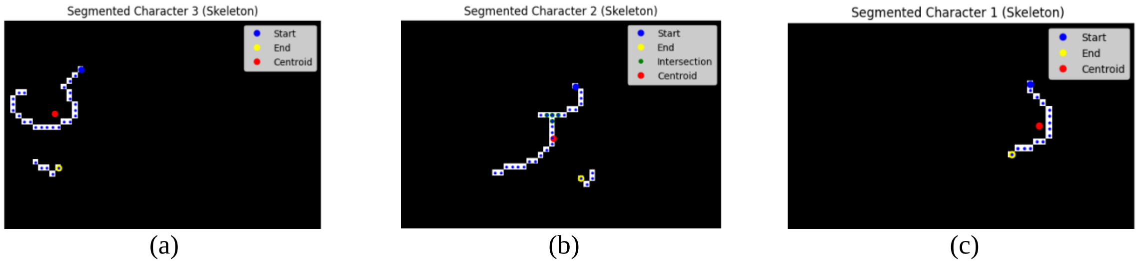

Figure 7. Step of segmentation: (a) results of *ya'* character segmentation, (b) results of *ya'* and *ra'* character segmentation, and (c) results of dal character segmentation

This highlights a fundamental limitation of centroid-based segmentation: its lack of discriminative power in dealing with the irregular spacing and complex ligatures of Jawi script. To address this, more context-aware strategies are required. Potential improvements include normalizing centroid distances relative to character size or integrating additional heuristics, such as skeleton-based intersection points, to guide more accurate segmentation.

3.5.2. Based on centroid from CCL and intersection points

Despite the observed improvements, segmentation errors persist in complex character structures. Figures 8(a) to (d) show the results of character segmentation using centroid information from CCL combined with skeleton intersection points for different Jawi characters. In Figures 8(a) and (b), characters with serrated or highly curved strokes (e.g., *ya* and *ra*) exhibit over-segmentation, as multiple centroid and intersection points are generated along irregular skeleton paths. Conversely, Figures 8(c) and (d) illustrate cases where smoother ligatures (e.g., *ya* and *dal*) lead to under-segmentation due to weak or missing intersection points in the skeleton representation. These results indicate that the effectiveness of the proposed approach is highly sensitive to skeleton quality and residual noise, which directly influence the accuracy of centroid detection and intersection point extraction.

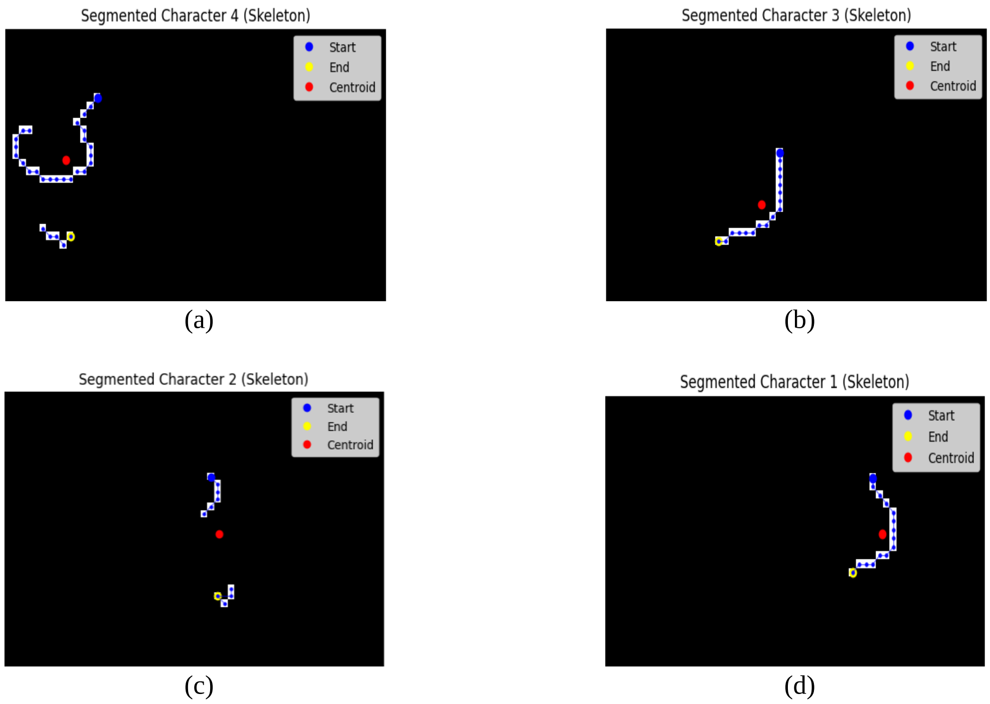

Figure 8. Step of segmentation: (a) results of *ya'* character segmentation, (b) results of *ra'* character segmentation, (c) results of *ya'* character segmentation and (d) results of *dal* character segmentation

3.5.3. Based on centroid from CCL, intersection points, and loop

To further address connected characters, loop features were integrated with centroid and intersection information. As shown in Figures 9 and 10, this approach improves segmentation accuracy for characters with loop structures (e.g., *ba'*, *ha'*), outperforming the previous methods in handling complex shapes. Nevertheless, limitations remain for characters with multiple loops or closely located intersections, as well as for sequences of three connected characters (e.g., mim-nun-waw), where incorrect cut placement may cause structural loss. These findings indicate that fixed cutting rules are still insufficient for highly complex Jawi handwriting. Future improvements should incorporate adaptive intersection filtering, branch pruning, and Jawi-specific linguistic constraints to enhance segmentation robustness.

The segmentation results for characters with loop structures are shown in Figures 10(a) and (b). Figure 10(a) shows *ha'*, and Figure 10(b) shows *ba'*. These examples demonstrate that the proposed method performs effectively for both independent and connected characters with loop structures. Compared to the two previous approaches, this method provides improved segmentation accuracy, particularly in distinguishing characters with complex shapes.

Despite its advancements, the current segmentation method faces structural challenges, particularly with characters featuring double loops or intersections located very close to a loop, such as ha' or mim. In these instances, cutting rules often damage the loop area, leading to structural degradation. Similarly, in complex sequences like *mim-nun-waw*, the system occasionally fails to position cut points accurately, causing the middle character (*nun*) to be lost or incorrectly merged. To overcome these limitations and enhance robustness against the high variability of Jawi script, future refinements should focus on adaptive intersection filtering, skeleton branch pruning, and the integration of linguistic rules to better preserve character integrity.

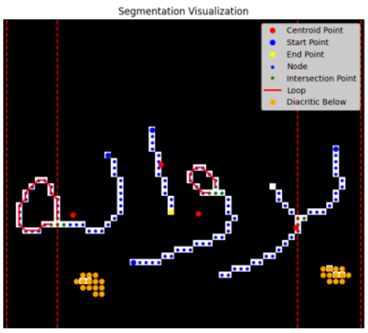

Figure 9. The result of visualization of character segmentation that has loop part *ba'* and *ha'*

(a) (b)

Figure 10. Step of segmentation: (a) results of *ha'* character segmentation and (b) results of *ba'* character segmentation

3.6. Evaluation

The evaluation stage focused on comparing the system's character segmentation results against a manually verified ground truth (GT) from the "Kitab Syair Perahu" manuscript. Using a sample of 10 manuscript lines, the analysis was conducted on a per-line (rasm) basis to ensure accuracy. From a total of 269 ground-truth characters, the system generated 236 segments, of which 187 were true positives (TP). The remaining data recorded 17 false positives (FP) and 31 false negatives (FN). While the full evaluation utilized all 10 lines, the segmentation process is visually demonstrated through four selected examples in Figures 11 to 14.

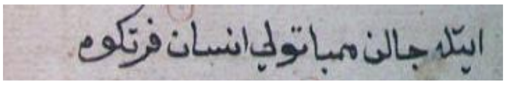

Figure 11. Page p01-lineimg 12

Figure 12. Page p06-lineimg0

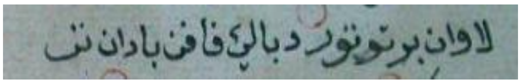

Figure 13. Page p06-lineimg5

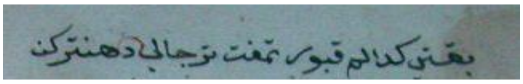

Figure 14. Page p08-lineimg0

To provide a more comprehensive view of the evaluation, Table 3 presents the detailed segmentation results for each manuscript line, including the number of GT characters, detected (DT) characters, correctly segmented characters TP, FP, and FN, along with their corresponding accuracy, precision, recall, and F1-score.

Table 3. Detailed evaluation results of segmentation system

Page-line	GT	DT	TP	FP	FN	Accuracy	Precision	Recall	F1-score
P01-12	28	24	20	1	3	0.833	0.952	0.870	0.909
P06-0	30	24	18	1	5	0.750	0.947	0.783	0.857
P06-5	28	22	18	1	3	0.818	0.947	0.857	0.900
P08-0	30	23	18	1	4	0.826	0.950	0.864	0.905
P09-15	29	29	24	3	2	0.828	0.889	0.923	0.906
P12-4	29	23	18	0	5	0.783	1.000	0.783	0.878
P13-0	25	21	15	0	5	0.762	1.000	0.762	0.865
P13-12	21	22	17	3	2	0.773	0.850	0.895	0.872
P14-4	25	28	22	5	1	0.786	0.815	0.957	0.880
P14-7	24	20	17	2	1	0.850	0.895	0.944	0.919
Average	269	236	187	17	31	0.801	0.925	0.864	0.889

The evaluation results in Table 3 demonstrate robust performance, with an accuracy of 0.801 and a high precision of 0.895. A recall of 86.38% and an F1-score of 88.91% further confirm the system's reliability in balancing detection and accuracy. However, performance inconsistencies persist, with per-line accuracy ranging from 0.75 to 0.85; these are primarily caused by under-segmentation in looped characters and over-segmentation in dense connections. Given the small, homogeneous dataset and the absence of cross-validation, these findings should be considered preliminary. Future work should focus on adaptive strategies such as skeleton pruning and curvature analysis-integrated with hybrid machine learning approaches and expanded datasets to enhance generalizability across diverse manuscripts.

The results highlight both the system's strengths and its limitations, particularly its inconsistent performance with dense ligatures and looped structures. Furthermore, a direct comparison with other segmentation systems was not feasible due to the restricted sample size and the absence of publicly available Jawi script benchmark datasets. Consequently, these findings serve as an initial baseline, with future efforts aimed at incorporating broader datasets to enable formal benchmarking against alternative segmentation approaches.

4. CONCLUSION

This study developed a Jawi character segmentation system for the "Syair Perahu" manuscript, integrating CCA, skeletonization, and keypoint detection. The system achieved an accuracy of 0.801, a precision of 0.895, a recall of 86.38% and an F1-score of 88.91%, proving its effectiveness in handling complex handwritten forms. However, limitations remain: the small, single-manuscript dataset restricts generalizability, while segmentation errors persist in dense ligatures and looped structures, highlighting the need for further refinements to capture the full variability of Jawi handwriting.

Future research should expand the dataset to encompass diverse handwriting styles and manuscripts to enhance system robustness. Leveraging deep learning architectures, such as lightweight CNNs or transformers, could significantly improve accuracy for complex or degraded texts. Furthermore, a hybrid framework incorporating keypoint detection would bolster adaptability. Ultimately, integrating this method into a complete OCR pipeline supported by mobile or hardware-based digitization tools is a vital step toward scalable transcription and the digital preservation of Jawi manuscripts.

FUNDING INFORMATION

The authors are grateful for the funding support from Universitas Yudharta Pasuruan and Research Center for Artificial Intelligence and Cybersecurity, National Research and Innovation Agency (BRIN) Bandung which has contributed to the implementation of this research.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Fadhilatul Fitriyah	✓	✓	✓	✓	✓	✓		✓	✓		✓			
Dian Andriana	✓	✓			✓		✓			✓		✓	✓	✓
Muhammad Zulhaj Aliansyah	✓	✓		✓	✓		✓			✓		✓	✓	✓
Lukman Hakim	✓			✓			✓			✓		✓	✓	✓
Muhammad Faishol Amrulloh	✓			✓			✓			✓		✓	✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

ETHICAL APPROVAL

Not applicable. This study did not involve human or animal subjects.

DATA AVAILABILITY

The data that support the findings of this study consist of scanned images of "Syair Perahu" manuscripts and their labeled datasets. This data is openly available on the BRIN RIN Dataverse https://data.brin.go.id/dataset.xhtml?persistentId=hdl:20.500.12690/RIN/1ESGFP.

REFERENCES

[1] C. Gu, "Scripting English in Jawi: English disguised in Arabic-based 'Tulisan Jawi' in Brunei's linguistic landscape," *Asian Englishes*, vol. 2, no. 3, pp. 591-624, May 2025, doi: 10.1080/13488678.2025.2497010.

[2] C. Gu and P. Coluzzi, "Presence of 'ARABIC' in Kuala Lumpur's multilingual linguistic landscape: heritage, religion, identity, business and mobility," *International Journal of Multilingualism*, vol. 22, no. 3, pp. 1473-1503, May 2024, doi: 10.1080/14790718.2024.2356215.

[3] R. Abughannam, "The Counter-Colonial: The Agency of Architectural Rehabilitation as a form of Resistance in Hebron, Palestine," Ph.D. thesis, Carleton University, Ottawa, ON, Canada, 2024, doi: 10.22215/etd/2024-15989

[4] C. Mahfud, R. Astari, A. Kasdi, M. A. Mu'ammar, M. Muyasaroh, and F. Wajdi, "Islamic cultural and Arabic linguistic influence on the languages of Nusantara; From lexical borrowing to localized Islamic lifestyles," *Wacana, Journal of the Humanities of Indonesia*, vol. 22, no. 1, May 2021, doi: 10.17510/wacana.v22i1.914.

[5] Y. Ruldeviyani, H. Suhartanto, B. A. Sotardodo, M. H. Fahreza, A. Septiano, and M. F. Rachmadi, "Character recognition system for pegon typed manuscript," *Heliyon*, vol. 10, no. 16, p. e35959, Aug. 2024, doi: 10.1016/j.heliyon.2024.e35959.

[6] P. Coluzzi, "Jawi, an endangered orthography in the Malaysian linguistic landscape," *International Journal of Multilingualism*,

vol. 19, no. 4, pp. 630-646, Oct. 2022, doi: 10.1080/14790718.2020.1784178.

[7] A. S. Bania and B. Akob, "Preserving the Jawi Script in Aceh: Assessing Literacy, Cultural Heritage, and Modern Paradigm Challenges," *Studies in English Language and Education (SiELE)*, vol. 12, no. 1, pp. 457-470, 2025, doi: 10.24815/siele.v12i1.36629.

[8] F. Fakhriati, M. Mu'jizah, M. Holil, and T. Permadi, "Don't Leave Indonesian Manuscripts in Danger: An Analysis of Digitalization and Preservation," *Preservation, Digital Technology & Culture*, vol. 51, no. 1, 2022, pp. 3-15, doi: 10.1515/pdtc-2021-0017.

[9] K. Romeo-Pakker, A. Ameur, C. Olivier, and Y. Lecourtier, "Structural analysis of Arabic handwriting: segmentation and recognition," *Machine Vision and Applications*, vol. 8, no. 4, pp. 232-240, Jul. 1995, doi: 10.1007/BF01219591.

[10] P. Dutta and N. B. Muppalaneni, "A top-down character segmentation approach for Assamese and Telugu handwritten documents," *Journal of Ambient Intelligence and Humanized Computing*, vol. 15, no. 9, pp. 3275-3287, Sep. 2024, doi: 10.1007/s12652-024-04805-y.

[11] A. M. A. Al Masri, M. S. Hitam, W. N. J. H. W. Yussof, and A. Al-Shatnawi, "Novel Algorithm for Baseline Detection of Offline Arabic Handwritten Text Recognition," *Journal of Advanced Research in Applied Sciences and Engineering Technology (ARASET)*, vol. 37, no. 1, pp. 56-68, Jan. 2024, doi: 10.37934/araset.37.1.5668.

[12] H. A. Al Hamad and M. Shehab, "Improving the Segmentation of Arabic Handwriting Using Ligature Detection Technique," *Computers, Materials & Continua*, vol. 79, no. 2, pp. 2015-2034, 2024, doi: 10.32604/cmc.2024.048527.

[13] A. A. Shiekh, M. S. Azmi, M. A. Aziz, M. N. Al-Mhiqani, and S. S. Bafjaish, "Framework of diacritic segmentation for Arabic handwritten document," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 24, no. 2, pp. 1001-1008, Nov. 2021, doi: 10.11591/ijeecs.v24.i2.pp1001-1008.

[14] S. Alghyaline, "Arabic Optical Character Recognition: A Review," *Computer Modeling in Engineering & Sciences*, vol. 135, no. 3, pp. 1825-1861, 2023, doi: 10.32604/cmes.2022.024555.

[15] M. Elkhayati, Y. Elkettani, and M. Mourchid, "Segmentation of Handwritten Arabic Graphemes Using a Directed Convolutional Neural Network and Mathematical Morphology Operations," *Pattern Recognition*, vol. 122, p. 108288, Feb. 2022, doi: 10.1016/j.patcog.2021.108288.

[16] 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, p. 4584, Apr. 2023, doi: 10.3390/app13074584.

[17] A. Saidi, A. Moulay Lakhdar, and M. Beladgham, "Recognition of Offline Handwritten Arabic Words Using a Few Structural Features," *Computers, Materials & Continua*, vol. 66, no. 3, pp. 2875-2889, 2021, doi: 10.32604/cmc.2021.013744.

[18] M. El Khayati, I. Kich, and Y. Taouil, "CNN-based Methods for Offline Arabic Handwriting Recognition: A Review," *Neural Processing Letters*, vol. 56, no. 2, p. 115, Mar. 2024, doi: 10.1007/s11063-024-11544-w.

[19] M. P. Ayyoob and P. Muhamed Ilyas, "Stroke-Based Data Augmentation for Enhancing Optical Character Recognition of Ancient Handwritten Scripts," *IEEE Access*, vol. 12, pp. 186794-186802, 2024, doi: 10.1109/ACCESS.2024.3505238.

[20] R. Najam and S. Faizullah, "A scarce dataset for ancient Arabic handwritten text recognition," *Data in Brief*, vol. 56, p. 110813, Oct. 2024, doi: 10.1016/j.dib.2024.110813.

[21] L. Berriche and A. Al-Mutairy, "Seam carving-based Arabic handwritten sub-word segmentation," *Cogent Engineering*, vol. 7, no. 1, Jan. 2020, doi: 10.1080/23311916.2020.1769315.

[22] O. A. Boraik, M. Ravikumar, and M. A. N. Saif, "Characters Segmentation from Arabic Handwritten Document Images: Hybrid Approach," *International Journal of Advanced Computer Science and Applications(IJACSA)*, vol. 13, no. 4, 2022, doi: 10.14569/IJACSA.2022.0130447.

[23] L. Berriche, A. Alqahtani, and S. RekikR, "Hybrid Arabic handwritten character segmentation using CNN and graph theory algorithm," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 1, p. 101872, Jan. 2024, doi: 10.1016/j.jksuci.2023.101872.

[24] N. M. Diah, R. Z. Ramli, N. A. M. Zin, and A. Abdullah, "Real-time feedback engine for online jawi character recognition," *International Journal of Advanced Technology and Engineering Exploration*, vol. 9, no. 89, pp. 477-489, Apr. 2022, doi: 10.19101/IJATEE.2021.874758.

[25] S. Razali, F. Arnia, R. Muharrar, K. Muchtar, and A. Bintang, "Improved Classification of Handwritten Jawi Script Based on Main Part of Script Body," *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, vol. 7, no. 1, pp. 94-104, Feb. 2023, doi: 10.29207/resti.v7i1.4600.

[26] H. M. Al-Barhamtoshy, K. M. Jambi, M. A. Rashwan, and S. M. Abdou, "An Arabic Manuscript Regions Detection, Recognition and Its Applications for OCRing," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 1, pp. 1-28, Jan. 2023, doi: 10.1145/3532609.

[27] B. Bataineh *et al.*, "A Comprehensive Review on Document Image Binarization," *Journal of Imaging*, vol. 11, no. 5, p. 133, Apr. 2025, doi: 10.3390/jimaging11050133.

[28] Z. Aliansyah and D. Andriana, "Graph-based Optical Character Recognition for Syair Perahu," 2025, *RIN Dataverse*. [Online]. Available: https://data.brin.go.id/dataset.xhtml?persistentId=hdl:20.500.12690/RIN/1ESGFP

[29] A. Waly, B. Tarek, A. Feteha, R. Yehia, G. Amr, and A. Fares, "Arabic Handwritten Document OCR Solution with Binarization and Adaptive Scale Fusion Detection," in *2024 6th Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, Giza, Egypt, 2024, pp. 316-319, doi: 10.1109/NILES63360.2024.10753216.

[30] M. Liao, Z. Zou, Z. Wan, C. Yao, and X. Bai, "Real-Time Scene Text Detection With Differentiable Binarization and Adaptive Scale Fusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 919-931, Jan. 2023, doi: 10.1109/TPAMI.2022.3155612.

[31] G. Ning, "Two-dimensional Otsu multi-threshold image segmentation based on hybrid whale optimization algorithm," *Multimedia Tools and Applications*, vol. 82, no. 10, pp. 15007-15026, Apr. 2023, doi: 10.1007/s11042-022-14041-1.

[32] Y. Du, H. Yuan, K. Jia, and F. Li, "Research on Threshold Segmentation Method of Two-Dimensional Otsu Image Based on Improved Sparrow Search Algorithm," *IEEE Access*, vol. 11, pp. 70459-70469, 2023, doi: 10.1109/ACCESS.2023.3293191.

[33] S. Singh, N. Mittal, H. Singh, and D. Oliva, "Improving the segmentation of digital images by using a modified Otsu's between-class variance," *Multimedia Tools and Applications*, vol. 82, no. 26, pp. 40701-40743, Nov. 2023, doi: 10.1007/s11042-023-15129-y.

[34] M. D. Hamanrora, Y. N. Kunang, I. Z. Yadi, and M. Mahmud, "Image segmentation of Komering script using bounding box," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 35, no. 3, pp. 1565-1578, Sep. 2024, doi: 10.11591/ijeecs.v35.i3.pp1565-1578.

[35] X. Zhang, H. Li, F. Meng, Z. Song, and L. Xu, "Segmenting Beyond the Bounding Box for Instance Segmentation," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 2, pp. 704-714, Feb. 2022, doi:

10.1109/TCSVT.2021.3063377.
[36] A. Qaroush, A. Awad, A. Hanani, K. Mohammad, B. Jaber, and A. Hasheesh, "Learning-free, divide and conquer text-line extraction algorithm for printed Arabic text with diacritics," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 7699-7709, Oct. 2022, doi: 10.1016/j.jksuci.2022.04.021.
[37] H. Butt, M. R. Raza, M. J. Ramzan, M. J. Ali, and M. Haris, "Attention-Based CNN-RNN Arabic Text Recognition from Natural Scene Images," *Forecasting*, vol. 3, no. 3, pp. 520-540, Jul. 2021, doi: 10.3390/forecast3030033.
[38] M. A. Al Ghamdi, "A Novel Approach to Printed Arabic Optical Character Recognition," *Arabian Journal for Science and Engineering*, vol. 47, no. 2, pp. 2219-2237, Feb. 2022, doi: 10.1007/s13369-021-06163-9.
[39] J. Ma, X. Ren, V. Y. Tsviatkou, and V. K. Kanapelka, "A novel fully parallel skeletonization algorithm," *Pattern Analysis and Applications*, vol. 25, no. 1, pp. 169-188, Feb. 2022, doi: 10.1007/s10044-021-01039-y.
[40] C. Yang, B. Indurkhya, J. See, and M. Grzegorzek, "Towards Automatic Skeleton Extraction With Skeleton Grafting," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 12, pp. 4520-4532, Dec. 2021, doi: 10.1109/TVCG.2020.3003994.
[41] T.-Q. Wang, X. Jiang, and C.-L. Liu, "Query Pixel Guided Stroke Extraction with Model-Based Matching for Offline Handwritten Chinese Characters," *Pattern Recognition*, vol. 123, p. 108416, Mar. 2022, doi: 10.1016/j.patcog.2021.108416.
[42] D. Peng *et al*., "Recognition of Handwritten Chinese Text by Segmentation: A Segment-Annotation-Free Approach," in *IEEE Transactions on Multimedia*, vol. 25, pp. 2368-2381, 2023, doi: 10.1109/TMM.2022.3146771.
[43] J. Baloun, M. Prantl, L. Lenc, J. Martínek, and P. Král, "On self-supervision in historical handwritten document segmentation," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 28, pp. 329–344, Jul. 2025, doi: 10.1007/s10032-025-00538-6.
[44] M. Hamdan and M. Cheriet, "ResneSt-Transformer: Joint attention segmentation-free for end-to-end handwriting paragraph recognition model," *Array*, vol. 19, p. 100300, Sep. 2023, doi: 10.1016/j.array.2023.100300.
[45] Z. Li, C. Yin, and X. Zhang, "Crack Segmentation Extraction and Parameter Calculation of Asphalt Pavement Based on Image Processing," *Sensors*, vol. 23, no. 22, p. 9161, Nov. 2023, doi: 10.3390/s23229161.
[46] J. Ma, J. Wang, J. Li, and D. Zhang, "Real-time skeletonization for sketch-based modeling," *Computers & Graphics*, vol. 102, pp. 56-66, Feb. 2022, doi: 10.1016/j.cag.2021.11.005.
[47] G. M. Foody, "Challenges in the real world use of classification accuracy metrics : From recall and precision to the Matthews correlation coefficient," *PLoS One*, vol. 18, no. 10, pp. 1-27, 2023, doi: 10.1371/journal.pone.0291908.

BIOGRAPHIES OF AUTHORS

Fadhilatul Fitriyah is a student of Informatics Engineering Study Program at Universitas Yudharta Pasuruan, Indonesia. His academic interests include computer vision, software development, and artificial intelligence. He has participated in an internship program at the Research Center for Artificial Intelligence and Cybersecurity, National Research and Innovation Agency (BRIN) Bandung, with a research focus on the application of computer vision for Pegon script recognition in "Syair Perahu" manuscripts. In 2023, he successfully obtained funding from the Student Entrepreneurship Development Program (P2MW) for the development of Resik, a digital waste bank application. He also passed the Internship and Certified Independent Study (MSIB) program in 2023 and 2024. He is active in information technology research and development activities and is committed to contributing to scientific publications. She can be contacted at email: fadilatulfitria9@gmail.com.

Dian Andriana is a senior researcher at the Research Center for Artificial Intelligence and Cybersecurity, National Research and Innovation Agency (BRIN), Indonesia, since 1998. He graduated from the School of Electrical Engineering and Informatics, Bandung Institute of Technology, Indonesia. His areas of expertise include software engineering, artificial intelligence, image processing, and database systems. She has received research funding from the Ministry of Research, Technology and Higher Education through the Saintek Grant scheme. She actively publishes scientific papers in reputable international journals and conferences, and is registered in Scopus and Google Scholar databases. She can be contacted at email: dian009@brin.go.id.

Muhammad Zulhaj Aliansyah received his B.E. degree from Brawijaya University, Indonesia, and his M.Eng. and Dr.Eng. degrees from Hiroshima University, Japan, in System Cybernetics. He has served as a research associate and postdoctoral researcher in various laboratories at Hiroshima University. Currently, he is a researcher at the Research Center for Artificial Intelligence and Cybersecurity, National Research and Innovation Agency (BRIN), and a Data Science lecturer at the Universitas Pembangunan Nasional "Veteran" of Jawa Timur. His research interests include robotics vision, structural health monitoring, and digital manufacturing. He can be contacted at email: muha350@brin.go.id, muhammad_zulhaj.sada@upnjatim.ac.id.

Lukman Hakim received his Bachelor's degree in Computer Science from Universitas Yudharta Pasuruan, Indonesia, his Master's degree from Sepuluh Nopember Institute of Technology (ITS), and his Ph.D. in Information Engineering from Hiroshima University, Japan. His research focuses on computer vision, image segmentation, deep learning, and intelligent systems. He has published various scientific articles in reputable journals and conferences, and actively establishes research collaborations between Indonesia and Japan. Currently, he serves as Vice Rector at Universitas Yudharta Pasuruan. He can be contacted at email: lukman@yudharta.ac.id.

Muhammad Faishol Amrulloh received his Bachelor's degree in Computer Science from Universitas Yudharta Pasuruan, Indonesia. His Master's degree is from Dian Nuswantoro University, Semarang. His research focuses on the internet of things, machine learning, and artificial intelligence. He has published various scientific articles in leading journals and conferences, and actively carry out the Tri Dharma of Higher Education. He currently serves as the Head of the Informatics Engineering Study Program, Faculty of Engineering, Universitas Yudharta Pasuruan. He can be contacted at email: faishol@yudharta.ac.id.