Application of the traveling salesman problem to optimize skeletonization and stroke reconstruction

Alifah[1], Dian Andriana[2], Muhammad Zulhaj Aliansyah[3], Lukman Hakim[1], Kholid Murtadlo[1]
[1]Department of Informatics Engineering, Faculty of Engineering, Universitas Yudharta Pasuruan, Pasuruan, Indonesia
[2]Research Center for Artificial Intelligence and Cybersecurity, 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:

Digitalization
Greedy algorithm
Pegon script
Skeletonization
Traveling salesman problem

ABSTRACT

The preservation of Turots Nusantara manuscripts written in Pegon script faces significant challenges due to physical deterioration and the complexity of handwritten styles. This study proposes a novel digitization approach based on image processing to extract and reconstruct handwriting strokes by combining skeletonization and the travelling salesman problem (TSP) algorithm. The novelty of this research lies in the application of a modified Greedy TSP algorithm capable of recognizing branching and cyclic structures typical of Arabic–Pegon characters, enabling accurate reconstruction of handwritten stroke sequences. The process involves preprocessing (grayscale, thresholding, and morphological operations), skeleton extraction using a thinning method, and weighted graph construction based on Euclidean distance between skeleton points. The proposed system achieved an average precision of 0.552, recall of 0.815, F1-score of 0.657, and accuracy of 0.82. These results demonstrate the method's effectiveness in detecting and reconstructing character shapes from Pegon manuscripts. Practically, this approach offers potential applications in the automatic digitization, preservation, and analysis of Pegon script, contributing to the conservation of Indonesia's Islamic intellectual and cultural heritage.

Corresponding Author:

Lukman Hakim
Department of Informatics Engineering, Faculty of Engineering, Universitas Yudharta Pasuruan
Jl. Yudharta No.7, Kembangkuning, Sengonagung, Purwosari, Pasuruan, East Java 67162, Indonesia
Email: lukman@yudharta.ac.id

1. INTRODUCTION

The development of Islam in Indonesia is closely linked to written traditions as a medium for recording, preserving, and transmitting knowledge. Many works of the "Ulama Nusantara", collectively known as Turots, were written using Arabic Pegon script an adapted Arabic writing system used for local languages such as Javanese, Sundanese, and Madurese [1]. As a form of Islamic intellectual heritage, Pegon manuscripts contain valuable historical, religious, and cultural knowledge.

However, analyzing handwritten Pegon manuscripts remains a challenging task. Arabic-based handwriting is inherently complex due to its cursive nature, contextual letter shapes, overlapping strokes, loops, and diacritical marks [2]. These challenges are further exacerbated by the physical deterioration of Nusantara manuscripts caused by aging, humidity, and inadequate preservation, which often results in faded ink, broken strokes, and damaged paper [3], [4].

Digitization has therefore become an essential effort for preserving Turots Nusantara manuscripts. Beyond long-term conservation, digitization improves accessibility, supports scholarly analysis, and enables the transmission of cultural heritage to future generations [5], [6]. A critical stage in this process is the reconstruction and analysis of handwriting strokes, which aims to recover the original writing structure and stroke sequence from static digital images [7].

Several studies have addressed handwriting skeletonization and stroke reconstruction. Research by Kryzhanovskaya *et al*. [8] proposed reconstructing pen strokes from skeleton images using heuristic graph traversal, achieving paths similar to natural handwriting but struggling with branched or complex characters. Meanwhile, Li *et al*. [9] introduced instance-aware skeleton embedding for curved text detection in scene images, improving detection accuracy but not focusing on stroke reconstruction. Another study by Diaz *et al*. [10] modeled skeletons as graphs and extracted writing paths based on continuity and curvature criteria, producing smooth stroke sequences. However, this approach fails to handle branched or multi-stroke characters commonly found in Arabic-derived scripts such as Pegon or Jawi.

To address these limitations, this study proposes a stroke reconstruction method based on the travelling salesman problem (TSP). TSP is well suited for modeling continuous paths and has the potential to produce stroke representations that closely resemble human writing behavior. Nevertheless, standard TSP formulations restrict each node to a single visit, making them unsuitable for characters with loops or branching structures. Similarly, Greedy TSP solvers, which prioritize local shortest distances, often generate implausible stroke orders for complex handwritten characters [11].

Therefore, this research introduces a modified Greedy TSP algorithm that allows controlled revisiting of skeleton nodes for characters with looping and branching structures, while maintaining single-visit constraints for simpler characters. This modification enables more accurate reconstruction of handwritten Pegon strokes and better represents the original writing sequence, supporting the digitization and preservation of Nusantara manuscripts.

2. METHOD

The research process consists of several stages, including pre-processing, skeletonization, travelling salesman problem, letter segmentation based on bounding box, and freeman chain code (FCC). The flowchart of these stages is shown in Figure 1.

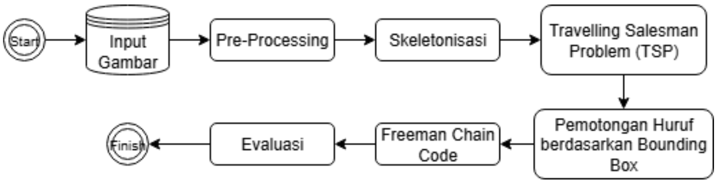


Figure 1. Flow of research

2.1. Dataset

The process of collecting the dataset for this study originated from an ancient manuscript written using the Pegon script, namely the "Kitab Syair Perahu". The initial stage began with scanning the manuscript. The scanned results were then processed by manually cutting each page, which consisted of 13 lines per page. After that, each line was cut again into individual words to facilitate the segmentation process. An example of the image fragments to be analyzed in this study can be seen in Figure 2.

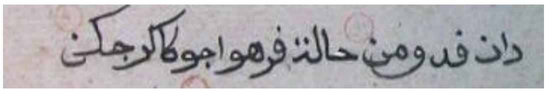


Figure 2. Image line cutouts "Book Syair Perahu"

2.2. Preprocessing

Pre-processing is performed to prepare the data so that it becomes more structured and cleaner, facilitating character shape segmentation and analysis in the next stage. The first step is grayscale conversion. A grayscale image represents each pixel with a single intensity value, where the red, green, and blue components are equal, producing shades between black and white. According to Zeger *et al*. [12], grayscale is a digital image representation that stores only luminance intensity information, typically in 8-bit format with values ranging from 0 to 255. As reported in [13], this process aims to extract important structures or objects from images, particularly under uneven lighting conditions, and is carried out using (1).

$$I = 0{,}2989\ \times R + 0{,}5870\ \times G + 0{,}1141\ \times B \tag{1}$$

The variable I denotes the grayscale intensity value, while R, G, and B represent the red, green, and blue components of the RGB color model.

After grayscale conversion, binarization is performed using the Otsu thresholding method, an intensity-based segmentation technique for grayscale images [14]. This method classifies each pixel as foreground or background depending on whether its intensity is above or below the threshold value [15]. As demonstrated in [16] Otsu thresholding effectively separates images into regions with clear intensity differences, improving segmentation accuracy with minimal computational cost. The conversion from grayscale to a binary image is achieved by determining an optimal threshold value using (2).

$$I(x,y) = \begin{cases} 1 & jika\ I(x,y) > T \\ 0 & jika\ I(x,y) \leq T \end{cases} \tag{2}$$

The function $I(x,y)$ represents the pixel intensity at coordinates (x,y), while T denotes the threshold value.

Subsequently, morphological opening and closing operations are applied. Opening, which consists of erosion followed by dilation using the same structuring element, is used to remove small foreground noise while preserving the main object shape. Conversely, closing is a dilation followed by erosion operation that fills small holes and connects fragmented objects, resulting in smoother object contours [17]–[19]. The formulas for morphological opening and closing are presented in (3) and (4).

$$A \circ B = (A \ominus B) \oplus B \tag{3}$$

$$A \odot B = (A \oplus B) \ominus B \tag{4}$$

A represents the input image, B denotes the structural elements used in morphological processing, $\ominus$ signifies the erosion operation, and $\oplus$ indicates the dilation operation.

2.3. Skeletonization

After preprocessing, the character image is skeletonized to reduce each letter to its medial axis while preserving its essential morphological structure. Skeletonization extracts a one-pixel-wide representation that maintains the object's topology and geometric form, providing a structural basis for stroke extraction from static images [20], [21]. In this study, skeletonization is performed using the Zhang–Suen algorithm, a parallel thinning method that iteratively removes pixels based on local connectivity rules within a 3×3 neighborhood, resulting in a continuous and one-pixel-wide skeleton [22]. This representation reduces data redundancy and supports efficient subsequent processing, as formally expressed in (5).

$$Skeleton_{Zhangsuen}(A) = \lim_{k\to\infty} T_2\,(T_1(A_k)) \tag*{5)}$$

A_k represents the binary image produced in the k-th iteration, while T_1 and T_2 are the two sub-iterations of the thinning process in which pixels are removed based on logical rules. The notation $\lim\limits_{k\to\infty}$ indicates that the process continues until convergence, meaning no further pixel removal occurs and the result no longer changes.

2.4. TSP

The next step applies the TSP, an NP-hard combinatorial optimization method that seeks the shortest path visiting each point once [23], [24]. However, this formulation is not suitable for Arabic letter segmentation because many characters contain loops and branching structures that produce non-linear skeleton paths. The single-visit constraint limits accurate traversal of such complex shapes.

Carrabs *et al.* [25], introduced the carousel Greedy algorithm, which allows local node revisits and achieves near-optimal solutions under high route complexity with low computational cost. Building on this idea, this study adopts a Greedy-with-revisit TSP approach, enabling selective revisits for letters with loops or branches, while maintaining single-visit traversal for simpler letters.

Comparative experiments with the nearest neighbor method show that the Greedy algorithm yields more stable trajectories and better represents natural stroke flow, especially for complex letters such as ح ,ع,

and م. Unlike nearest neighbor, which often causes abrupt jumps or premature loop closure, the Greedy method considers global connectivity, resulting in more coherent paths. This modified TSP approach effectively reconstructs strokes in ancient Arabic manuscripts by preserving stroke continuity, branching behavior, and structural connectivity, producing geometrically accurate and handwriting-consistent results. Mathematically, TSP is defined by (6).

$$\min \textstyle\sum_{i=0}^{n}\ d\ (v_i, v_i + 1), \qquad with\ v_1 = v_n + \tag{6}$$

The term $d\ (v_i, v_i + 1)$ denotes the Euclidean distance between two consecutive points, where $v_i, v_i + 1$ are visited points and n is the total number of points.

The Greedy revisit algorithm works by selecting the location closest to the current position without limiting locations that have not been visited. This strategy is relevant for cases of branching, merging, or complex cyclic structures. The steps of this algoritm are as follows:

– Determine the starting point p_start of p (usually starting from the end point of the skeleton).

– Initialization

The path is initialized as $[p\_start]$, all points $p \in p$ are marked as unvisited ($visited[p] = 0$), the starting point p_start is marked as visited ($visited[p\_start] = 1$), and the variable current is set to p_start.

– If there is still a point $p \in p$ such that $visited[p] < limit(p)$:

a. Find the closest p_next point from current using Euclidean distance. The Euclidean distance formula can be seen in (7).

$$d_{(i,j)} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{7}$$

b. Add p_next to the path

Once the closest point is found, it is added to the path list. This is an important step in forming the sequence of paths that represent the scratch path.

c. $visited[p\_next] += 1$

A marker indicating that the p_next point has already been visited. The visited value is recorded so that the algorithm can track visit values to determine whether a point needs to be visited again (for example, in the case of branching points) or if a single visit is sufficient.

d. current ← p_next

The point that was just visited (p_next) is updated as the current point ($current$). This is important so that the next point is calculated from the latest position in the next iteration.

– Return path

Description:

The visit limit $limit(p)$ defines the maximum number of allowable visits for each skeleton point based on its local structural characteristics: for endpoints (degree = 1), $limit(p) = 1$ since only a single visit is required; for linear points (degree = 2), $limit(p) = 1$ as no revisits are necessary in straight segments; and for branching points (degree ≥ 3), $limit(p) > 1$ to allow revisiting these points so that all connected branches can be explored, with the value typically set equal to or slightly greater than the point's degree. Furthermore, $limit(p) = 1$ can be adjusted according to local skeleton complexity or density, where regions with high connectivity or curved structures such as loops may require higher visit limits to ensure complete path reconstruction, while simpler or straighter regions can use lower limits to avoid unnecessary revisits and improve computational efficiency.

2.5. Letter segmentation based on bounding box

At this stage, Arabic letter segmentation is performed using skeletonization and the modified TSP path. Each letter is automatically separated from a line of Arabic text by utilizing skeleton-based sub-paths generated in the previous TSP modification process.

According to Aanchal *et al.* [26], a bounding box is an axis-aligned rectangle defined by x–y coordinates, width, and height that encloses handwriting regions in scanned document images. Accordingly, this study applies bounding box–based letter cropping as the initial segmentation step. Unlike previous approaches, cropping is performed for each sub-path produced by the modified TSP algorithm, allowing segmentation to consider both spatial location and stroke order that reflects natural handwriting structure. This approach effectively separates individual letters from the background and irrelevant elements such as noise.

The bounding box for each sub-path is computed using the minimum and maximum x and y coordinate values, as defined in (8).

$$x_{min} = \min(x) - \delta, \; x_{max} = \max(x) + \delta \\ y_{min} = \min(y) - \delta, \; y_{max} = \max(y) + \delta \quad (8)$$

The symbol δ is an additional margin used to cut the area so that it is not too narrow. The image area in the bounding box will be cut directly from the combined_skeleton, resulting in a cut image of the letters that corresponds to each sub-path.

2.6. Freeman chain code

The next step encodes the movement direction between skeleton points using the FCC method for each sub-path generated by the TSP algorithm. FCC represents object contours in binary or skeletonized images using eight discrete directions (0–7), providing a concise description of character structure [27]. This method is widely used in image compression, pattern recognition, and shape analysis because it effectively captures directional changes [28].

Fethi *et al.* [29] applied FCC to extract contour features of handwritten Arabic characters and proposed a modified FCC that simplifies the code sequence without losing essential structural information, leading to improved recognition accuracy. In this study, FCC is used to transform skeleton paths into numerical movement sequences that can be spatially analyzed [29], [30], such as identifying writing direction, loops, and characteristic turns in Arabic letters. The eight-directional system of FCC is illustrated below.

```
3  2  1
\ | /
4 ----- 0
/ | \
  5  6  7
```

The numbers 0 to 7 above represent 8 directions of movement relative to a point. In the FCC representation, each number indicates the direction of pixel movement, where 0 indicates right, 1 indicates upper right, 2 indicates up, 3 indicates upper left, 4 indicates left, 5 indicates lower left, 6 indicates down, and 7 indicates lower right [31].

2.7. Evaluation metrics

Evaluation is conducted quantitatively using standard multi-class classification metrics, including accuracy, precision, recall, and F1-score [32], which are computed based on true positive (TP), false positive (FP), true negative (TN), and false negative (FN) values [33]. Accuracy measures overall classification correctness, precision evaluates the reliability of positive predictions, Recall measures the ability to detect all actual positive samples, and the F1-score provides a balanced measure by combining precision and recall, particularly under class imbalance conditions [34], [35]. In addition, intersection over union (IoU) is used to assess the overlap between predicted and ground truth (GT) regions in segmentation tasks.

Beyond quantitative metrics, a qualitative node-based analysis is performed on the skeletonization results to evaluate spatial structure and inter-node connectivity, demonstrating the system's effectiveness in separating letter shapes and diacritics. To verify that the classification results are not due to random chance, a Binomial Test is applied. This non-parametric statistical test determines whether the observed classification accuracy is significantly higher than a random baseline (typically 50%), thereby confirming the validity of the system's performance. The binomial test formulation is presented in (14).

$$P\,(X = k) = \binom{n}{k}\, p^{k}\, (1 - p)^{n-k} \quad (14)$$

n denotes the total number of trials, such as the number of samples or predictions. k represents the number of successes, for example the number of correct predictions. p is the probability of success under the null hypothesis (commonly set to 0.5 for a random baseline). And $\binom{n}{k}$ is the binomial coefficient, computed as $\frac{n!}{k!(n-k)!}$.

For a one-tailed test, the p-value is calculated using (15).

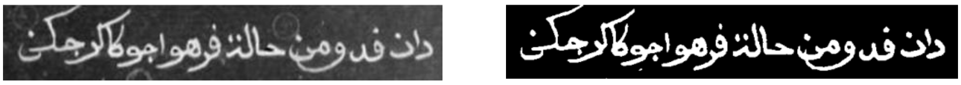
$$p\ value = \sum_{i=k}^{n} \binom{n}{k}\, p^{i}\, (1 - p)^{n-i} \quad (15)$$

This test is used to determine whether the actual proportion of successes is significantly greater (or smaller, depending on the direction of the test) than the probability assumed under the null hypothesis.

3. RESULTS AND DISCUSSION

3.1. Pre-processing results

The initial image processing step is grayscale conversion, which removes color information and retains only pixel intensity to facilitate subsequent processing. As shown in Figure 3(a), the grayscale image preserves the basic shapes of the Arabic letters in the Syair Perahu manuscript despite variations in ink and background color, allowing clearer separation between text and background based on intensity differences. Following this step, thresholding is applied to convert the grayscale image into a binary (black-and-white) image, enabling effective separation of letter objects from the background. The Otsu thresholding results display the letters in white (value 255) against a black background (value 0), making the Arabic script clearly distinguishable and highlighting the curved morphological features of Pegon characters, as shown in Figure 3(b).

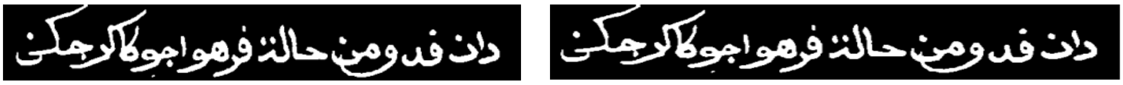


(a) (b)

Figure 3. Grayscale and thresholding results: (a) grayscale image and (b) binary image after thresholding

Before skeletonization, an advanced pre-processing stage is applied using morphological opening and closing to improve letter structure quality. These operations clean and complete the binary image, thereby enhancing skeleton extraction. Opening removes small noise introduced during digitization, while closing fills small gaps to preserve stroke continuity.

In this study, a 2×2 square structuring element (np.ones((2,2), np.uint8)) is used, as it effectively removes noise while preserving fine stroke details and preventing adjacent letters from merging, which is suitable for Pegon manuscripts with thin strokes and narrow spacing. The parameter selection is determined empirically through experiments on multiple samples. For manuscripts with higher noise levels, larger structuring elements (e.g., 3×3 or 5×5) may be used, provided that stroke details are preserved. The results of morphological opening and morphological closing are shown in Figures 4 (a) and (b).



(a) (b)

Figure 4. Morphological opening and morphological closing results: (a) image after morphological opening and (b) image after morphologi closing

3.2. Skeletonization results

The skeleton extraction process in this study uses the morphological thinning method with the Zhang-Suen algorithm, which is implemented through the skeletonize() function from the skimage.morphology library. This method works iteratively by selectively removing edge pixels without altering the main topological structure of the letters, resulting in a one-pixel skeleton that represents the medial axis of each letter stroke. The skeletonization results can be seen in Figure 5.

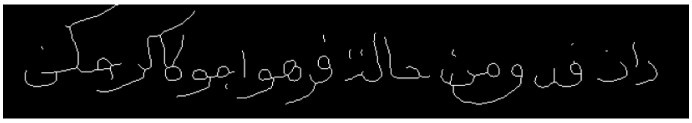


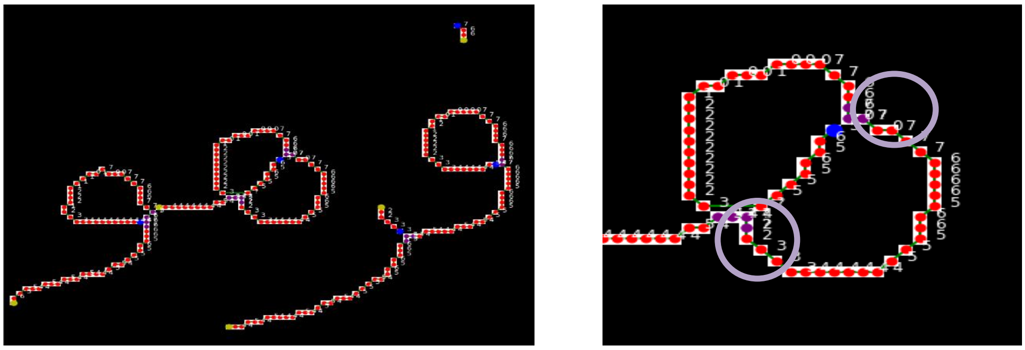
Figure 5. Skeletonization results

Figure 5 shows that the skeleton (frame lines) successfully follows the main contours of Arabic letters, including curves and connections between letters. The strokes of the letters have been successfully reduced to the medial axis, but still retain the shape and direction of the original strokes. This shows that the morphological thinning method using the Zhang-Suen algorithm is effective in extracting important features from complex letter shapes.

3.3. TSP results

In this study, the TSP algorithm is applied to skeletonized Pegon word images to generate sequential stroke paths that closely reconstruct the original handwriting. The Greedy revisit algorithm enables the TSP path to follow complex letter structures, including loops and branches, while preserving connectivity between stroke segments. Each resulting path is visualized as a directed graph to analyze its correspondence with the original letter shapes.

For evaluation, one word containing multiple looping letters was selected from each segmented line to facilitate clear observation and analysis of the generated paths. The detailed TSP path results are shown in Figure 6(a), while highlighted loop handling and revisiting behavior are illustrated in Figure 6(b).



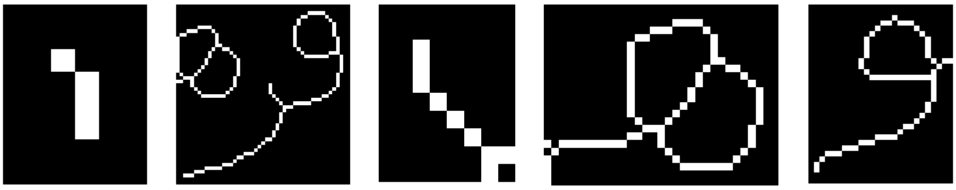
(a) (b)

Figure 6. TSP results: (a) overall generated TSP path and (b) detail view of loop traversal and revisiting behavior

It can be seen that the TSP path generated has the ability to follow the entire structure of the letter without breaking the path, even when the structure is circular or branched. Points visited more than once are marked in purple, indicating that revisiting occurs in those areas due to the complexity of the letter structure. This method is able to create paths that more closely resemble the hand-drawn strokes that form letters. Unlike conventional TSP, which only allows one visit per point.

3.4. Letter segmentation results

After obtaining the TSP paths and sub-paths, letter segmentation is performed using bounding boxes for each sub-path. The process starts by extracting skeleton points and generating a right-to-left TSP path based on Euclidean distance. The TSP path is then divided into sub-paths using the split_tsp_path_by_distance() function, where a new sub-path is created if the distance between consecutive points exceeds a maximum threshold (e.g., 3 pixels). For each sub-path, a bounding box is computed using the minimum and maximum x and y coordinates, enabling individual letters to be cropped into separate image segments. These segmented letter images are saved in a designated directory, with a “label.txt” file prepared for subsequent labeling. An example of the bounding box–based letter segmentation results is shown in Figures 7(a) to (e).



(a) (b) (c) (d) (e)

Figure 7. Letter segmentation results: (a) first segmented letter, (b) second segmented letter, (c) third segmented letter, (d) fourth segmented letter, and (e) fifth segmented letter

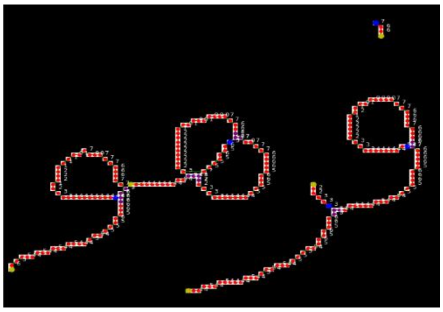
Figure 7 shows that this method is effective in following the complex shapes of Arabic script through skeleton paths and separating letters based on the bounding box of each TSP sub-path. The test results are displayed in the form of letter segment visualizations and extracted image files. These two files serve as the basis for the letter recognition stage in the next process.

3.5. FCC results

In this study, FCC is used to analyze reconstructed letter strokes by examining the shape and direction of pen movement. After skeletonization and TSP-based path tracking, multiple sub-paths representing letter skeletons are obtained. Each sub-path is then converted into a FCC through the following steps.

First, each sub-path is processed sequentially as a series of (x, y) skeleton coordinates. Next, direction codes are assigned for each pair of consecutive points using the direction_code() function when the points are direct 8-connected neighbors. If consecutive points are not adjacent, stepwise interpolation is applied by incrementally moving toward the target point based on the sign of coordinate differences, converting each step into a Freeman direction code. A maximum step limit is imposed to avoid infinite loops in invalid paths.

All generated direction codes are stored as an array representing the FCC for each sub-path. An example of FCC extraction for the word “Perahu” is shown in Figure 8. In Figure 8, the small white numbers on each path indicate the direction of movement based on the FCC. Meanwhile, the blue and yellow dots indicate the starting and ending points of each sub-path.



- Sub-path 1: [7, 6, 6]
- Sub-path 2: [4, 5, 4, 4, 4, 4, 4, 4, 3, 3, 3, 2, 2, 2, 2, 2, 1, 2, 1, 0, 1, 0, 0, 0, 0, 7, 7, 7, 6, 6, 6, 7, 6, 6, 6, 6, 7, 6, 6, 6, 6, 5, 6, 6, 6, 5, 5, 5, 4, 5, 4, 4, 5, 4, 4, 4, 4, 5, 4, 4, 6, 5, 6, 6, 5, 6, 5, 6, 5, 4, 5, 5, 5, 5, 4, 4, 5, 4, 5, 5, 4, 4, 5, 4, 4, 4, 4, 5, 4, 4, 5, 4, 4]
- Sub-path 3: [3, 3, 3, 2, 2]
- Sub-path 4: [4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 3, 3, 2, 1, 2, 2, 2, 1, 1, 1, 0, 1, 7, 0, 0, 7, 7, 7, 6, 6, 6, 7, 7, 5, 6, 6, 6, 6, 6, 5, 6, 5, 5, 5, 4, 5, 5, 4, 4, 4, 5, 4, 4, 5, 4, 4, 5, 4, 4, 5, 5, 6]
- Sub-path 5: [5, 6, 5, 6, 5, 5, 5, 4, 4, 4, 4, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 1, 0, 0, 1, 0, 0, 0, 7, 7, 6, 6, 6, 0, 7, 0, 7, 7, 7, 6, 6, 6, 6, 5, 6, 6, 5, 5, 5, 4, 4, 4, 4, 4, 4, 3, 3, 3, 2, 2, 4, 4, 5, 4, 5, 4, 4, 4, 4, 4, 4, 4, 4]

Figure 8. FCC generated from the word, start from the top right to the left

3.6. Evaluation metrics results

After completing all pre-processing, segmentation, and TSP-based processing stages, an evaluation is conducted to assess the system’s ability to detect and segment letters. The evaluation compares the expected number of characters GT with the number of characters detected by the system (prediction). GT values are obtained manually from reference data, while predictions are derived from the segmentation results produced by the algorithm.

In this study, GT data are taken from the first page of the Syair Perahu manuscript by randomly selecting three lines (rasm) and evaluating them manually. Prior to line-level evaluation, a preliminary word-level experiment is performed using the word “Perahu”, as discussed in the previous section. A detailed breakdown of the evaluation results for “Perahu” is presented in Table 1. Then, for a detailed explanation of each Evaluation value per row, please refer to Table 2.

As a complement to the visual analysis, Table 3 displays a quantitative evaluation of the segmentation results on each line of the manuscript, which includes the number of GT characters, detected characters (DT), correctly segmented characters TP, FP, FN, as well as performance metrics such as accuracy, precision, recall, F1-score and IoU.

Table 1. Evaluation results of the word “Perahu”

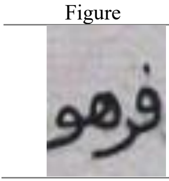
Figure	Precision	Recall	F1-score	Accuracy	IoU
	0.600	0.750	0.667	0.75	0.700

Table 2. Evaluation results per row

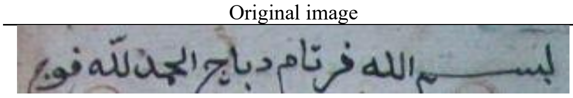
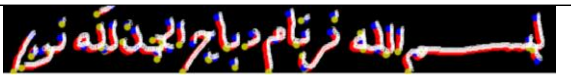
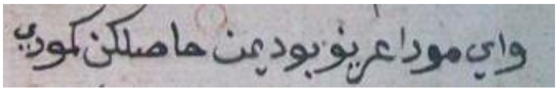
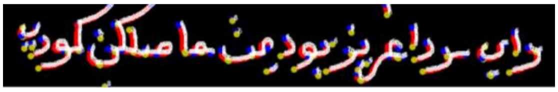
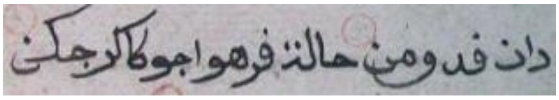
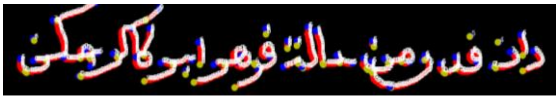
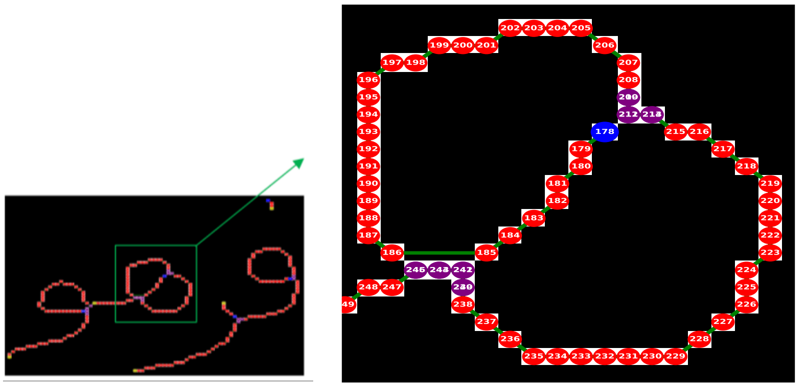
Original image		Processed result		
				
Precision	Recall	F1-score	Accuracy	IoU
0.595	0.815	0.687	0.82	0.524
				
Precision	Recall	F1-score	Accuracy	IoU
0.550	0.786	0.647	0.79	0.543
				
Precision	Recall	F1-score	Accuracy	IoU
0.512	0.846	0.638	0.85	0.520

Table 3. Detailed evaluation results of segmentation system

Image	GT	DT	TP	FP	FN	Precision	Recall	F1-Score	Accuracy	IoU
P01-1	27	37	22	15	5	0.595	0.815	0.687	0.82	0.524
P01-9	28	40	22	18	6	0.550	0.786	0.647	0.79	0.543
P01-10	26	43	22	21	4	0.512	0.846	0.638	0.85	0.520
Average	81	120	66	54	15	0.552	0.815	0.657	0.82	0.529

As a result, the evaluation metrics show that the system is quite good at recognizing the number of letters based on the number detected. However, because the evaluation only considers the suitability of the number without verifying the correctness of the identity or position of the letters, the results do not fully reflect the overall accuracy of the system. This focus was chosen because the research phase is still focused on reconstructing basic stroke forms through skeletonization and path optimization, rather than on full character recognition.

To evaluate the temporal accuracy of the stroke reconstruction sequence, this study adds nodes at each point of the TSP trajectory. Each node represents the algorithm's traversal sequence so that the reconstruction process can consider both the spatial dimension and the temporal sequence of the strokes. The results of the node addition strategy in the TSP trajectory are shown in Figure 9(a), which presents the overall word-level trajectory, and Figure 9(b), which provides a detailed view of the reconstructed stroke sequence.



(a) (b)

Figure 9. Implementation of adding nodes on the TSP path: (a) global trajectory after node insertion and (b) magnified view illustrating the refined and continuous stroke reconstruction

The addition of sequential node numbering along the TSP path is intended to record the traversal order (temporal order) of stroke reconstruction. Each node is assigned a global sequence number across all sub-paths, allowing every skeleton point to be represented not only spatially but also temporally based on its visit order. This enables evaluation from two perspectives:

- Spatial accuracy, which assesses whether the TSP path fully covers the character shape without missing stroke components

– Temporal accuracy, which evaluates whether the reconstructed stroke order matches natural handwriting patterns, such as correct traversal direction or loop ordering. Thus, node numbering functions as a time-step indicator in stroke reconstruction.

To further validate the evaluation results, a Binomial Test is applied to confirm that the observed accuracy is not due to chance and exceeds the random baseline (50%). The hypotheses are defined as: hipotesis nol (H_0); accuracy ≤ 0.50 and hipotesis alternatif (H_1); accuracy > 0.50. For the word "Perahu", the obtained p-value is 0.06250. Although slightly above the 0.05 significance threshold, this result indicates that the model's accuracy is marginally but consistently higher than random guessing, supporting the claim that the proposed method demonstrates meaningful generalization capability.

3.7. Stroke reconstruction evaluation

To complement the character count-based evaluation, this study added a quantitative evaluation of the accuracy of the stroke paths generated by the TSP algorithm. The evaluation was conducted using two main metrics: average path deviation (APD), which measures the spatial accuracy of the stroke paths, and temporal sequence accuracy (TSA), which measures the correspondence of the stroke sequence to the original writing pattern. The results of this quantitative evaluation are presented in Table 4.

Table 4. Quantitative accuracy assessment of reconstructed stroke paths

Image	APD (px)	TSA (%)
P01-1	1.92	86,3%
P01-9	2.15	82,5%
P01-10	2.03	88.1%
Average	2.03	85.6%

The evaluation results in Table 4 show that the TSP reconstruction path has low spatial deviation, with an average APD value of 2.03 pixels, indicating that the stroke shape is close to the ground-truth pattern. In addition, the TSA value of 85.6% indicates that most of the traversal sequence of the TSP path follows the actual letter writing pattern. Thus, the stroke reconstruction is not only visually correct, but also quantitatively measurable in terms of both shape and writing sequence.

4. CONCLUSION

This study successfully demonstrates a skeleton-based image processing framework for Pegon script that integrates thinning-based skeletonization, a modified TSP, and Greedy revisit optimization to reconstruct handwritten stroke paths. The proposed approach effectively handles complex Arabic letter structures involving loops and branches, enabling accurate stroke traversal and letter segmentation using Euclidean-distance–based sub-path separation and bounding box cropping. Experimental results show that the system achieves an average precision of 0.552, recall of 0.815, F1-score of 0.657, and accuracy of 82%, indicating reliable performance in detecting and segmenting Pegon letters based on character count.

To further validate the effectiveness of stroke reconstruction, a quantitative evaluation using APD and TSA was conducted. The results show a low average APD of 2.03 pixels and a high TSA of 85.6%, confirming that the reconstructed paths closely follow both the spatial shape and temporal writing sequence of the original handwriting. These findings demonstrate that the proposed TSP-based stroke reconstruction is not only visually coherent but also quantitatively accurate. Although the current evaluation focuses on segmentation and stroke reconstruction rather than full character recognition, the results provide a strong foundation for future work that incorporates character classification and spatial context analysis to support complete Pegon manuscript transcription.

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
Alifah	✓	✓	✓	✓	✓	✓		✓	✓		✓			
Dian Andriana	✓	✓			✓		✓			✓		✓	✓	✓
Muhammad Zulhaj Aliansyah	✓	✓		✓	✓		✓			✓		✓	✓	✓
Lukman Hakim	✓			✓			✓			✓		✓	✓	✓
Kholid Murtadlo	✓			✓			✓			✓		✓	✓	✓

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

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data supporting the findings of this study are publicly available at: https://data.brin.go.id/dataset.xhtml?persistentId=hdl%3A20.500.12690%2FRIN%2F1ESGFP.

REFERENCES

[1] R. Hikmawati and N. Muthohirin, "The role of pesantren in preserving the tradition of pegon Arabic writing: A study in the pesantren ilmu Al-Quran Singosari," *El-Jaudah : Jurnal Pendidikan Bahasa dan Sastra Arab*, vol. 5, no. 1, pp. 54–70, Jun. 2024, doi: 10.56874/ej.v5i1.1820.

[2] 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, Jan. 2024, doi: 10.1016/j.jksuci.2023.101872.

[3] M. A. Dhali, T. Reynolds, A. Z. Alizadeh, S. H. Nijdam, and L. Schomaker, "Pattern recognition techniques in image-based material classification of ancient manuscripts," in *Pattern Recognition Applications and Methods (ICPRAM 2023)*, 2024, pp. 124–150, doi: 10.1007/978-3-031-54726-3_8.

[4] S. Gaikwad, R. Kachhoria, and G. Yadav, "AI-based OCR for digitizing ancient Indian texts: Preserving linguistic heritage and overcoming script challenges," *International Journal of Linguistics Applied Psychology and Technology (IJLAPT)*, vol. 2, no. 03(Mar), pp. 47–50, Apr. 2025, doi: 10.69889/ijlapt.v2i03(Mar).102.

[5] F. Brinis, M. O. Soualah, and F. Bouarab, "Comparative study of encoding formats for digitized ancient Arabic manuscripts," in *2023 24th International Arab Conference on Information Technology (ACIT)*, Dec. 2023, pp. 1–11, doi: 10.1109/ACIT58888.2023.10453686.

[6] E. Abdoh, "Utilizing modern technology for the preservation of sncient manuscripts and rare Books: The digitization project at King Abdulaziz complex for endowment libraries as a model," *Restaurator. International Journal for the Preservation of Library and Archival Material*, vol. 46, no. 1, pp. 35–58, Mar. 2025, doi: 10.1515/res-2024-0016.

[7] R. Senatore, A. Santoro, A. Parziale, and A. Marcelli, "A biologically inspired approach for recovering the trajectory of offline handwriting," *Memetic Computing*, vol. 15, no. 3, pp. 355–375, Sep. 2023, doi: 10.1007/s12293-023-00397-w.

[8] S. Kryzhanovskaya, S. Arseev, and L. Mestetskiy, "Pen trace reconstruction with s0keleton representation of a handwritten text image," *Proceedings of the 30th International Conference on Computer Graphics and Machine Vision (GraphiCon 2020). Part 2*, pp. 1–10, Dec. 2020, doi: 10.51130/graphicon-2020-2-3-27.

[9] H. Li, X. Hu, and H. Lu, "T-skeleton: Accurate scene text detection via instance-aware skeleton embedding," *IET Image Processing*, vol. 18, no. 6, pp. 1491–1503, May 2024, doi: 10.1049/ipr2.13043.

[10] M. Diaz, G. Crispo, A. Parziale, A. Marcelli, and M. A. Ferrer, "Writing order recovery in complex and long static handwriting.," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. 4, pp. 171–184, Jun. 2022, doi: 10.9781/ijimai.2021.04.003.

[11] Y. Li and J. W. Siegel, "Entropy-based convergence rates of greedy algorithms," *Mathematical Models and Methods in Applied Sciences*, vol. 34, no. 05, pp. 779–802, May 2024, doi: 10.1142/S0218202524500143.

[12] I. Zeger, S. Grgic, J. Vukovic, and G. Sisul, "Grayscale image colorization methods: Overview and evaluation," *IEEE Access*, vol. 9, pp. 113326–113346, 2021, doi: 10.1109/ACCESS.2021.3104515.

[13] W. Liu, L. Wang, and M. Cui, "Quantum image segmentation based on grayscale morphology," *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–12, 2022, doi: 10.1109/TQE.2022.3223368.

[14] M. T. Nyo, F. Mebarek-Oudina, S. S. Hlaing, and N. A. Khan, "Otsu's thresholding technique for MRI image brain tumor segmentation," *Multimedia Tools and Applications*, vol. 81, no. 30, pp. 43837–43849, Dec. 2022, doi: 10.1007/s11042-022-13215-1.

[15] J. Li and X. Gui, "Fully automatic grayscale image segmentation: Dynamic thresholding for background adaptation, improved image center point selection, and noise-resilient start/end Point determination," *Applied Sciences*, vol. 14, no. 20, Oct. 2024, doi: 10.3390/app14209303.

[16] A. T. H. Al-Rahlawee and J. Rahebi, "Multilevel thresholding of images with improved Otsu thresholding by black widow optimization algorithm," *Multimedia Tools and Applications*, vol. 80, no. 18, pp. 28217–28243, Jul. 2021, doi: 10.1007/s11042-

021-10860-w.

[17] H. A. Saeed, S. Hamad, and A. T. Hussain, “Analysis the digital images by using morphology operators,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 3, pp. 1654–1662, Dec. 2021, doi: 10.11591/ijeecs.v24.i3.pp1654-1662.

[18] S. Bhutada, N. Yashwanth, P. Dheeraj, and K. Shekar, “Opening and closing in morphological image processing,” *World Journal of Advanced Research and Reviews*, vol. 14, no. 3, pp. 687–695, Jun. 2022, doi: 10.30574/wjarr.2022.14.3.0576.

[19] H. Fujisaki and M. Nakashizuka, “Deep morphological filter networks for gaussian denoising,” in *2020 IEEE International Conference on Image Processing (ICIP)*, Oct. 2020, pp. 918–922, doi: 10.1109/ICIP40778.2020.9190657.

[20] A. Bærentzen and E. Rotenberg, “Skeletonization via local separators,” *ACM Transactions on Graphics*, vol. 40, no. 5, pp. 1–18, Oct. 2021, doi: 10.1145/3459233.

[21] 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, Mar. 2022, doi: 10.1016/j.patcog.2021.108416.

[22] 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.

[23] C. Wu, X. Fu, J. Pei, and Z. Dong, “A novel sparrow search algorithm for the traveling salesman problem,” *IEEE Access*, vol. 9, pp. 153456–153471, 2021, doi: 10.1109/ACCESS.2021.3128433.

[24] Z. Zhang and Y. Han, “Discrete sparrow search algorithm for symmetric traveling salesman problem,” *Applied Soft Computing*, vol. 118, Mar. 2022, doi: 10.1016/j.asoc.2022.108469.

[25] F. Carrabs, C. Cerrone, R. Cerulli, and B. Golden, “An adaptive heuristic approach to compute upper and lower bounds for the close-enough traveling salesman problem,” *INFORMS Journal on Computing*, pp. 1030–1048, Sep. 2020, doi: 10.1287/ijoc.2020.0962.

[26] A. Aanchal, N. Nidhi, P. Preeti, and G. Gurpratap, “Automatic cropping of handwritten scanned documents with object detection algorithm,” *Procedia Computer Science*, vol. 218, pp. 1733–1741, 2023, doi: 10.1016/j.procs.2023.01.151.

[27] R. Dhief, R. Youssef, and A. Benazza, “An ensemble learning approach using decision fusion for the recognition of Arabic handwritten characters,” in *Proceedings of the 11th International Conference on Pattern Recognition Applications and Methods*, 2022, pp. 51–59, doi: 10.5220/0010839500003122.

[28] B. Žalik, D. Mongus, K. R. Žalik, D. Podgorelec, and N. Lukač, “Lossless chain code compression with an improved binary adaptive sequential coding of zero-runs,” *Journal of Visual Communication and Image Representation*, vol. 75, Feb. 2021, doi: 10.1016/j.jvcir.2021.103050.

[29] M. R. Fethi, O. Farhaoui, I. Zeroual, and A. El Allaoui, “A progressive approach to arabic character recognition using a modified freeman chain code algorithm,” *Data and Metadata*, vol. 2, May 2023, doi: 10.56294/dm2023178.

[30] Y. He, S. Wang, and X. Gao, “Edge extraction of ancient books based on freeman chain code,” in *2021 IEEE 3rd Eurasia Conference on IOT, Communication and Engineering (ECICE)*, Oct. 2021, pp. 257–260, doi: 10.1109/ECICE52819.2021.9645594.

[31] B. Žalik, D. Mongus, N. Lukač, and K. R. Žalik, “Can burrows-wheeler transform be replaced in chain code compression?,” *Information Sciences*, vol. 525, pp. 109–118, Jul. 2020, doi: 10.1016/j.ins.2020.03.073.

[32] H. M. Balaha, H. A. Ali, M. Saraya, and M. Badawy, “A new Arabic handwritten character recognition deep learning system (AHCR-DLS),” *Neural Computing and Applications*, vol. 33, no. 11, pp. 6325–6367, Jun. 2021, doi: 10.1007/s00521-020-05397-2.

[33] D. Chicco and G. Jurman, “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation,” *BMC Genomics*, vol. 21, no. 1, Dec. 2020, doi: 10.1186/s12864-019-6413-7.

[34] H. Chen, N. Wang, X. Du, K. Mei, Y. Zhou, and G. Cai, “Classification prediction of breast cancer based on machine learning,” *Computational Intelligence and Neuroscience*, vol. 2023, no. 1, Jan. 2023, doi: 10.1155/2023/6530719.

[35] S. Riyanto, I. S. Sitanggang, T. Djatna, and T. D. Atikah, “omparative analysis using various performance metrics in imbalanced data for multi-class text classification,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 6, 2023, doi: 10.14569/IJACSA.2023.01406116.

BIOGRAPHIES OF AUTHORS

Alifah undergraduate student at Universitas Yudharta Pasuruan, class of 2021. She participated in the MSIB 6 – Essential Programming Course: Full Stack Web Development (2024), where he engaged in hands-on learning through individual and group projects, including the development of a school registration system using the Python flask framework as his final project. In the same year, she joined an internship at Research Center for Artificial Intelligence and Cybersecurity, National Research and Innovation Agency (BRIN), Bandung, contributing to the AI-Based Turots Nusantara Digitalization project, particularly focusing on the digitization of Pegon script manuscripts, such as Syair Perahu. Since 2023, she has also been teaching information and communication technology (ICT) at MTs and MA Maarif Al-Utsmani. She can be contacted at email: alifah2563@gmail.com.

Dian Andriana has been a senior researcher at the Research Center for Artificial Intelligence and Cybersecurity, National Research and Innovation Agency (BRIN), Bandung, Indonesia, since 1998. She graduated from the School of Electrical Engineering and Informatics, Institut Teknologi Bandung (ITB), Indonesia. Her expertise includes software engineering, artificial intelligence, image processing, and database systems. She has also received research funding from the Ministry of Research, Technology, and Higher Education of Indonesia through the Science and Technology Grant Scheme (Hibah Saintek). She can be contacted at email: dian009@brin.go.id.

Muhammad Zulhaj Aliansyah received the S.T. degree in Electrical Engineering from Universitas Brawijaya, Indonesia, the M.Eng. degree in System Cybernetics from Hiroshima University, Japan, and the Dr.Eng. degree in System Cybernetics from Hiroshima University, Japan, as part of the TAOYAKA Program. He is currently a lecturer with the Department of Data Science, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Surabaya, Indonesia, and a researcher at Research Center for Artificial Intelligence and Cybersecurity, National Research and Innovation Agency (BRIN), Bandung, Indonesia. His current research interests include wireless sensing systems, image processing, and structural health monitoring. One of his notable works is "A Tandem Marker-Based Motion Capture Method for Dynamic Small Displacement Distribution Analysis", published in the Journal of Robotics and Mechatronics (2019). He can be contacted at email: muhammad_zulhaj.sada@upnjatim.ac.id.

Lukman Hakim received the S.Kom. degree in Universitas Yudharta Pasuruan, Indonesia, in 2010, the M.Kom. degree in Computer Science from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, in 2014, and the Ph.D. degree in Information Engineering from Hiroshima University, Japan, in 2021. From 2018 to 2022, he was a research assistant at Hiroshima University, Japan. His current research interests include deep learning, computer vision, image processing, and pattern recognition with applications in biomedical imaging and agriculture. He can be contacted at email: lukman@yudharta.ac.id.

Kholid Murtadlo earned the Doctoral (Dr.) degree in Economics and Management, Brawijaya University. His academic passion lies in the preservation and study of Nusantara's Islamic manuscripts (turats). Bridging the humanities and computational sciences, his role in this research was essential for the philological grounding of the technical workflow. He oversaw the critical stages of data validation and script interpretation, ensuring that the digital stroke reconstruction remained faithful to the graphical and linguistic nuances of the original Pegon manuscripts. He can be contacted at email: ahmad.kholid@fisip.yudharta.ac.id.