Journal of Imaging Science and Technology® 69(2): 1–10, 2025.

A Building-Block Approach to Character-Level Writer Verification on the Great Isaiah Scrolls

T. Lumban Tobing and P. Bours
Department of Information Security and Communication Technology, Norwegian University of Science and Technology, Gjøvik, Norway
E-mail: tabita.tobing@ntnu.no

Abstract. *This study presents a novel character-level writer verification framework for ancient manuscripts, employing a building-block approach that integrates decision strategies across multiple token levels, including characters, words, and sentences. The proposed system utilized edge-directional and hinge features along with machine learning techniques to verify the hands that wrote the Great Isaiah Scroll. A custom dataset containing over 12,000 samples of handwritten characters from the associated scribes was used for training and testing. The framework incorporated character-specific parameter tuning, resulting in 22 separate models and demonstrated that each character has distinct features that enhance system performance. Evaluation was conducted through soft voting, comparing probability scores across different token levels, and contrasting the results with majority voting. This approach provides a detailed method for multi-scribe verification, bridging computational and paleographic methods for historical manuscript studies.*
Keywords: *writer verification, handwriting analysis, historical document*

[DOI: 10.2352/J.ImagingSci.Technol.2025.69.2.020401]

1. INTRODUCTION

The period of ancient history extends approximately from the fifth millennium BCE to the fifth century CE. During this period, people transcribed mostly on thick paper made from plants and animal skins; those that survived named are referred to as ancient manuscripts. Considering the writing medium used and the thousands of years they have survived, physical quality degradation is inevitable and is a major obstacle for accurate document analysis. Digital image production of ancient manuscripts has been made possible through the vast and sophisticated technology of document imaging, aiming to preserve them at a specific point in time. Furthermore, advancements in digital imaging have enabled the restoration of degraded manuscripts, addressing challenges such as bleed-through removal [1, 2], alignment correction [3], and multispectral text extraction [4], making historical documents more accessible and amenable for computational analysis.

Paleography, a study of ancient writing systems is one of the auxiliary sciences of history. In general, paleography investigates the type of script and ductus (distinctive features of the strokes of particular hands) inscribed on the manuscript to determine whether additional manuscripts studied were written by the same person. Such investigations are needed to identify additional manuscripts written by the same person, as these manuscripts may provide additional content for deeper understanding of the contextual knowledge gleaned from the precedent manuscript [5]. Thus, paleography plays a central role in analyzing ancient manuscripts and verifying their authorship, and offers a foundational approach to the understanding of historical documents.

Simultaneously, the field of computer science has adapted techniques from image processing and pattern recognition to address similar problems through the tasks of writer identification and verification [6]. As summarized by Bensefia et al. [7], writer identification task deals with the retrieval of handwritten samples from a database depending on the graphical analysis of the handwritten samples under study, while writer verification task aims to determine whether two samples of documents were written by the same writer. Based on the description, writer identification and verification tasks are equivalent to the objectives that paleography aimed to achieve. This notion then forms the basis of research in computer-aided writer identification and verification based on digital images of various handwritten scripts in modern and historical documents.

This study integrates the knowledge from paleography with advancements in computer-aided writer recognition to verify the authorship of the Great Isaiah Scrolls, an ancient manuscript believed to have been written by two different scribes. The primary objective of this study is the proposal of a writer verification system that can verify the dual-scribe hypothesis using a novel building-block approach. This approach is designed to improve verification accuracy by iteratively analyzing character-specific data, which is crucial for the detailed and precise identification of the scribes. Additionally, this study critically evaluates the data and methods used in the writer verification system, examining the integration of paleographic expertise with modern computational techniques.

The rest of the article is organized as follows: Section 2 reviews research on paleography-based writer recognition and computer-aided writer recognition. Section 3 focuses on the description of the proposed framework and the methods used in this study. Section 4 describes the experiment setting and Section 5 provides the results and its interpretation. Section 6 summarizes and concludes the study.

Received Dec. 8, 2024; accepted for publication Mar. 10, 2025; published online Apr. 24, 2025. Associate Editor: Steven Simske.

1062-3701/2025/69(2)/020401/10/$25.00

2. RELATED WORK

To perform writer verification on historical manuscripts is to bridge a profound gap between computer science and historical studies (disciplines focused on the historical significance of documents and relics). This bridging process requires alignment of expertise from both fields, where each discipline must temper its assumptions and expectations to find common ground. Not all advanced techniques in computer science can be applied directly to historical manuscripts, nor can traditional historical methods fully address the computational challenges. Therefore, this section explores the background and contributions of both disciplines, highlighting gaps in existing research and the drivers of the proposed framework.

2.1 *1QIsa-a Scrolls and Paleographic Approaches to Identifying Scribes*

1QIsa-a, the Great Isaiah Scroll, discovered in 1947 in Cave 1 at Qumran, is one of the most important biblical manuscripts found in the Dead Sea Scrolls collection. It is notable for its length (734 cm) and preservation, containing the full ancient square script Hebrew text of the Book of Isaiah from ca. 125 BCE, making it one of the oldest surviving biblical texts (additional information and images can be accessed at http://dss.collections.imj.org.il/isaiah) [8]. The study of ancient manuscripts, such as the 1QIsa-a, combines historical research with various methodologies to uncover both the textual content and the scribal practices behind these texts. While the primary focus often lies in extracting historical facts from the content itself, the application of contextualization, sourcing, and corroboration is essential in the scholarly approach to understanding these manuscripts. As Van Drie et al. describe, corroboration involves comparing documents to address historical questions or confirm claims, with the identification of comparable documents—such as those written by the same scribe— providing deeper insights into their creation and context [9]. For ancient manuscripts, this requires identifying comparable documents, often determined by whether they were written by the same scribe. Documents attributed to the same scribe can provide additional comparative material for deeper contextual insights [5].

Paleography, the study of ancient handwriting, is a vital tool in this process. This methodology, as defined by Wakelin, involves analyzing features such as handwriting size, letterforms, and corrections to reveal individual scribal practices and styles [10]. Tov's work emphasized that examining these characteristics in Hebrew manuscripts can help identify distinct scribes, allowing scholars to trace the evolution and transmission of texts [11]. The consensus surrounding the authorship of the 1QIsa-a Scroll is not uniform. Traditionally, paleographic methods suggested that the entire manuscript was copied by a single scribe, with subtle variations in handwriting attributed to personal idiosyncrasies or occasional inconsistencies. However, some scholars also proposed that the manuscript was actually the work of two distinct scribes—one responsible for Columns I–XXVII and another for Columns XXVIII–LIV. Recent advancements, particularly through AI techniques that detect micro-level handwriting variations, have revealed the involvement of at least two scribes in its creation [12]. This breakthrough demonstrates the power of combining traditional paleographic methods with modern technology to enhance our understanding of ancient texts, as well as the scribes who worked on them.

2.2 *Computational Writer Verification*

The study of writer verification has grown significantly in recent decades, driven by advances in machine learning and its applications to handwriting analysis. At least three key components differentiate computational approaches: the token level of textual input (e.g., character, word, sentence), the feature extraction methods, and the classification techniques (see Table I).

Across diverse studies, research on Latin and non-Latin scripts often employs similar feature extraction techniques and classifiers, demonstrating the adaptability of machine learning frameworks across scripts. However, the choice of input level significantly influences the system's focus and accuracy. For example, character-level analysis can capture fine-grained handwriting traits, while word- or sentence-level inputs provide broader context. Despite these advances, integrating decision scores across token levels in a systematic manner remains a major challenge, as highlighted in Table I.

For historical manuscripts like the 1QIsa-a Scrolls, it is essential to select features and classifiers that align with paleographic practices. In this study, we employ edge-directional and hinge features, as these methods are intuitive to scholars studying historical manuscripts. These features have been successfully applied in handwriting analysis from their introduction to recent studies in capturing stroke directionality and curvature patterns (Refs. [31] and [32]). Additionally, we use an SVM classifier, which has demonstrated robust performance in tasks involving historical and modern handwriting (Refs. [30] and [33]).

2.3 *Gaps in Existing Research*

Despite significant advancements, computational writer verification faces critical limitations when applied to historical manuscripts:

- **Multi-Scribe Authorship:** Most systems assume a single writer for a document, overlooking the possibility of collaborative manuscripts where multiple scribes contribute.
- **Unit-Level Integration:** While character-level analysis provides precision, there is a lack of robust frameworks for aggregating information hierarchically across characters, words, and sentences.
- **Probabilistic Decision Frameworks:** Few studies adopt probabilistic approaches to integrate evidence across granularities, which is essential for handling complex documents and transitions between scribes.

Table I. Research on writer recognition for Latin and non-Latin scripts.

Dataset name	Script	Input level					Feature extraction method	Classifier
		P	S	L	W	C		
IAM, CVL, Firemaker	Dutch, English, German	Refs. [13–17]	Refs. [18]	Refs. [19] and [21]	Refs. [20] and [22]	—	LBP, LTP, LPQ, FAST, SIFT, HKD, CNN feature maps, SRS-LBP	SVM, Nearest Neighbor, CNN, RNN
ICFHR2012, IFN/ENIT, QUWI HIT-MW, IHP, HWDB1.1, JEITA-HP, KHATT, AHTID/MW, Custom dataset	Arabic, Chinese, Kannada, Devanagari, Japanese	Refs. [13, 19], [26–29]	Refs. [23]	—	Refs. [19] and [23]	Refs. [18, 24, 25] [30]	CNN feature maps, SURF SIFT, CA LDCF, GITF, CLGP, HOG, GLRL, Zoning	SVM Nearest Neighbor, Distance Calculation, CNN

These gaps are particularly relevant for the 1QIsa-a Scrolls, where evidence suggests multi-scribe authorship. The inability to address such challenges limits the scope and accuracy of existing verification methods. Furthermore, due to the varying consensus on the authorship of the 1QIsa-a Scrolls, no standardized protocol exists for verifying proposed authorship claims through a writer verification system. Current systems lack the capability to assess and validate the theories surrounding the manuscript's scribal origins.

2.4 *Motivation for a Building-Block Approach*

This study introduces a building-block approach to address the gaps identified. At the character level, this framework-based approach assigns probability scores to individual characters based on their likelihood of having been written by the same scribe. These scores are then aggregated hierarchically at word and sentence levels, allowing for a comprehensive verification process. This modular approach is well-suited for manuscripts like the 1QIsa-a Scrolls, where the complexity of multi-scribe authorship requires a flexible and scalable framework. By integrating paleographic expertise with computational methods, the proposed system offers an intuitive yet powerful solution for historical manuscript studies. Furthermore, this framework provides scholars with a deeper understanding of the methods employed, bridging the gap between disciplines and enhancing collaboration.

3. PROPOSED METHOD

This section describes the proposed character-level writer verification data flow and additional methods used as validation baseline in the experiments.

3.1 *Overall Architecture*

The writer recognition framework mentioned in Section 2.2 mainly adopts an absolute decision strategy in identifying, verifying, or authenticating a specific token. For example, in a page-level writer recognition system, the recognition rate is usually computed using global features, and the decision process stops at this level. This method typically assesses whether the entire page can be attributed to a particular writer, relying on features that represent the overall style of the text. However, in the context of historical manuscripts, a more granular analysis is often required. In such cases, it is important to consider the recognition rate at smaller units, such as characters, words, or sentences. This is because, in historical manuscript studies, the goal is not only to determine whether an entire page was written by the same hand but also to investigate whether different hands may have written paragraphs within the same text, sentences within a paragraph, or even individual words within a sentence.

Unlike modern datasets, where the identity of the writer is typically labeled (often in controlled lab environments), historical manuscripts lack such clear labeling and may involve multiple hands contributing to the same document. In these cases, a detailed analysis at finer levels is necessary to assess the possibility of multiple authors contributing to a single page, paragraph, or even a specific word. This approach provides a more nuanced understanding of the document's authorship, which is especially important when studying manuscripts where the attribution of authorship is uncertain or disputed. Thus, in this study, we integrated the building-block approach in the decision strategy stage with a machine learning- based writer verification system, as illustrated in Figure 1.

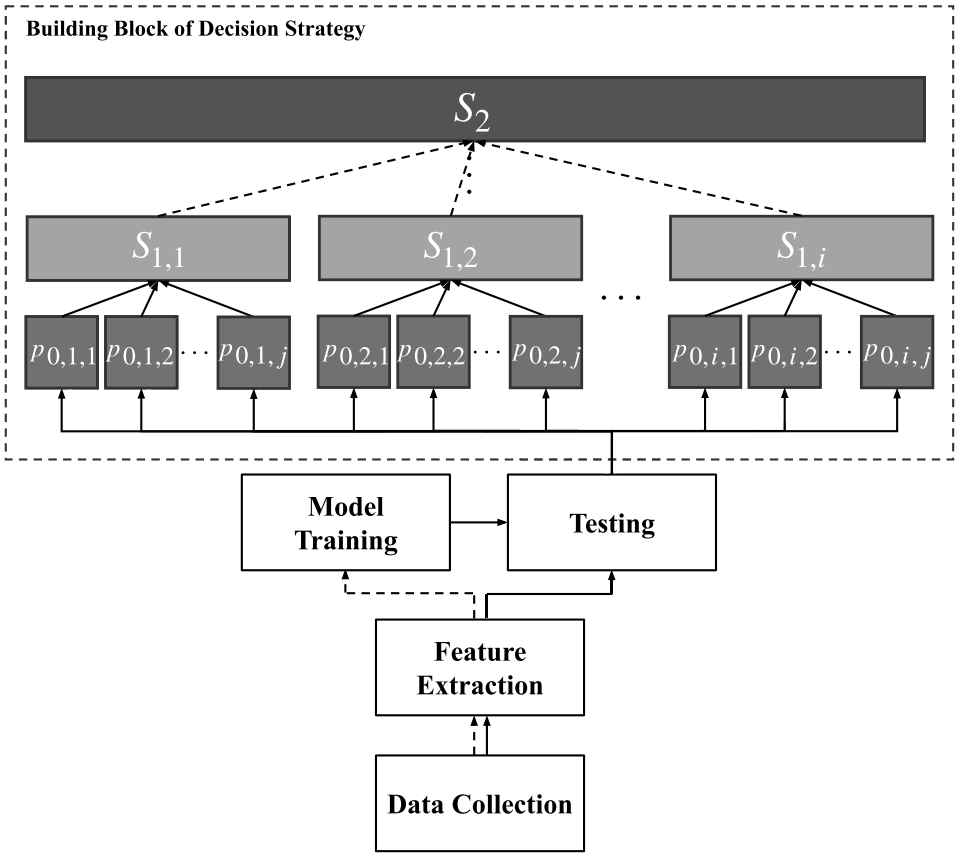

Figure 1. Character-Level Writer Verification Framework: from machine learning processes (data collection, feature extraction, model training, and testing) to the construction of a hierarchical decision strategy for writer verification.

The building-block approach uses a layered decision-making strategy built on the outcomes of a machine learning model. At the lowest level, raw probability scores result from testing and evaluation within the machine learning framework. These scores serve as foundational inputs, which are further used to compute the intermediate-level and highest-level scores, allowing for a hierarchical assessment of writer verification. This decision strategy consists of three hierarchical layers. At the highest level (Layer 2), S_2 represents the sentence- level probability score, indicating the likelihood that the entire sentence is written by a specific scribe. The intermediate level (Layer 1), where $S_{1,i}$ represents the word-level probability score for the ith word in the sentence, indicating the likelihood that the word is associated with a specific writer. At the lowest level (Layer 0), $p_{0,i,j}$ represents the raw probability scores of each individual character in word i, with j indexing the character within the word. This layered structure permits a detailed analysis, ranging from characters to words and sentences, facilitating writer verification at multiple levels.

3.2 *Validation Baseline*

To implement the proposed framework, we built an adaptation system to investigate the hands that wrote an ancient manuscript written in square script Hebrew, known as the Great Isaiah Scrolls. Our baseline used edge-directional and hinge feature extraction methods proposed by Bulacu et al. [30]. Edge-directional and hinge methods extract angular information differently—Edge-directional features compute the angle between handwriting strokes and a horizontal reference line, while hinge features measure angles formed between pairs of adjacent strokes. In both methods, these angles are categorized into structured bins based on their frequency of occurrence, with each bin assigned an empirical probability. These binned values represent the feature vectors, which are then used as input for training a machine learning model. Training-wise, we used an RBF kernel Support Vector Machine (SVM) to train and predict data [33]. SVM is known as a robust classifier that is suitable for datasets with high-dimensional and non-linearly separable features, which aligns with the characteristics of the data used in this study. To assess the contribution of the building-block approach, we leveraged SVM's probabilistic outputs by opting for class probability scores as the output of the testing stage. These scores were then processed for soft voting and majority voting systems.

3.2.1 *Decision Strategy*

The machine learning-based verification system generates class probability scores for each token at the lowest level. At the character level (Layer 0), the model produces $p_{0,i,j}$ representing the probability that this character j in word i belongs to the reference scribe. This probability is constrained within: $p_{0,i,j} \in [0, 1]$. Since this is a binary

classification task (two scribes), the probability of the second scribe is implicitly given by: $1 - p_{0,i,j}$.

To compute higher-level scores, soft voting and majority voting were applied separately at the word level (Layer 1) and the sentence level (Layer 2). At the word level (Layer 1), the aggregated score is defined as: $S_{1,k} = \{S_{1,k}^{\text{soft}}, S_{1,k}^{\text{majority}}\}$, where k is the word index, $S_{1,k}^{\text{soft}}$ is the soft voting score for word k, and $S_{1,k}^{\text{majority}}$ is the majority voting score for word k. At the sentence level (Layer 2), the final aggregated score is expressed as: $S_2 = \{S_2^{\text{soft}}, S_2^{\text{majority}}\}$, where S_2^{soft} is the sentence-level soft voting score and S_2^{majority} is the sentence-level majority voting score. These scores are further explained in the following section.

(a) Soft Voting. The soft voting score for a word, denoted as $S_{1,i}^{\text{soft}}$, is computed as the average of the probability scores of its characters. Assuming a word i consists of n characters, each with a probability score $p_{0,i,j}$ for $j = 1, 2, \ldots, n$, the soft voting score is expressed as:

$$S_{1,i}^{\text{soft}} = \frac{1}{n} \sum_{j=1}^{n} p_{0,i,j}, \tag{1}$$

where $p_{0,i,j}$ is the probability score of the jth character in word i, and n is the total number of characters in the word.

For example, consider word 1 ($i = 1$) consisting of four characters with probability scores: $p_{0,1,1} = 0.5$, $p_{0,1,2} = 0.55$, $p_{0,1,3} = 0.5$, and $p_{0,1,4} = 0.6$. The soft voting score for this word is:

$$S_{1,1}^{\text{soft}} = \tfrac{1}{4}(0.50 + 0.55 + 0.50 + 0.60) = 0.5375. \tag{2}$$

At the sentence level (Layer 2), the soft voting score is computed by averaging the soft voting scores of the words in the sentence:

$$S_2^{\text{soft}} = \frac{1}{m} \sum_{i=1}^{m} S_{1,i}^{\text{soft}}, \tag{3}$$

where m is the number of words in the sentence.

(b) Majority Voting. In majority voting, the decision score for a word, denoted as $S_{1,i}^{\text{majority}}$, is computed by counting the number of characters with a probability score greater than 0.5. The indicator function I ($p_{0,i,j} > 0.5$) is used, returning 1 if $p_{0,i,j} > 0.5$ or 0 otherwise.

The majority voting score is expressed as:

$$S_{1,i}^{\text{majority}} = \frac{1}{n} \sum_{j=1}^{n} \text{I}(p_{0,i,j} > 0.5), \tag{4}$$

where n is the number of characters in the word.

For example, consider word 1 ($i = 1$) consisting of four characters with probability scores: $p_{0,1,1} = 0.5$, $p_{0,1,2} = 0.55$, $p_{0,1,3} = 0.50$, and $p_{0,1,4} = 0.60$. Applying the indicator function, the majority voting score for this word is:

$$\begin{aligned} S_{1,1}^{\text{majority}} &= \tfrac{1}{4}\,(\text{I}(0.50 > 0.5) + \text{I}(0.55 > 0.5) \\ &\quad + \text{I}(0.50 > 0.5) + \text{I}(0.60 > 0.5)) \\ &= \tfrac{1}{4}(0 + 1 + 0 + 1) \\ &= 0.50. \end{aligned} \tag{5}$$

At the sentence level (Layer 2), the majority voting score is computed in a similar way by averaging the decision scores obtained from each word in the sentence:

$$S_2^{\text{majority}} = \frac{1}{m} \sum_{i=1}^{m} S_{1,i}^{\text{majority}}, \tag{6}$$

where m is the number of words in the sentence.

4. EXPERIMENT SETTING

This section explains the rationale behind the selection of data, methods, and approaches of this study and its description.

4.1 *Data Collection*

The Great Isaiah scroll consists of 54 pages, commonly called columns, of which it was assumed (by humanities [20]) that Scribe A wrote Col. I–XXVII and Scribe B Col. XXVIII–LIV. Research done by Popovic et al. [29] proved that assumption by implementing unsupervised learning for all columns. Based on these findings, the adaptation system aims to build a dataset and verify the data based on the claimed columns belonging to each scribe. We present an illustrated version of the letter samples, traced from the original images (see Figure 2). To consider all tokens that are available in large quantities, we chose tokens to be investigated from the lowest level, i.e., characters, words, and one sentence. We constructed the character-specific model based on the non-final form of the 22 letters of ancient square script Hebrew. We excluded the five final form letters since their frequency of occurrence was so limited that the machine learning-based system was unlikely to benefit from the small dataset.

For data collection, we expanded our custom dataset using the same approach described in our earlier study [34]. A total of 283 samples representing each letter (except for the letter *tet*, with only 145 samples, and the letter *samekh* with 232 samples, due to limited availability) were taken from each scribe's corresponding columns. The training-validation set referring to Scribe A comprised single characters extracted from Col. I–XXVI, while the set referring to Scribe B included single characters extracted from Col. XXVIII–LIII. The test set consisted of one sentence from Col. XXVII (referring to Scribe A's handwriting) and one from Col. LIV (referring to Scribe B's handwriting).

This separation of data into Scribe A and Scribe B's columns follows the presumed authorship as well as AI-based writer identification methods discussed in Section 2. The goal of this work was to verify these assumptions using the proposed system, leveraging the computational approach to confirm the authorship of the individual columns attributed to each scribe.

4.2 *Validation Baseline*

To validate the proposed framework, we utilized edge-directional and hinge feature extraction methods

Figure 2. Illustration of the 22 isolated letters with their corresponding names.

and an SVM-based classification technique. A major advantage of implementing handcrafted features, specifically edge-directional and hinge features, is that unique information based on the slant angle and curvatures of handwriting projection can be extracted. The collaborative nature of SVM and handcrafted features also brings an advantage for robust implementation, especially with our relatively small dataset, which could perform poorly during model training with a deep learning network.

4.2.1 *Edge-directional and Hinge Feature Extraction*

In this study, four pixel-length variations were applied: 2 px, 3 px, 4 px, and 5 px, resulting in different numbers of feature elements, i.e., 9, 13, 17, and 21 elements, respectively. In addition, unlike the Sobel edge detection method used in the original article, we implement the skeleton transformation method. This transformation aims to reduce the pixel width of binary objects to a 1-pixel-wide representation. Our earlier study [35] demonstrated that using edge-directional features for writer verification of the Great Isaiah Scrolls yields better and more consistent accuracy scores when combined with skeleton transformation. Based on this study, each individual character image must be skeletonized before feature extraction. Once the skeleton transformation was applied to each character, these instances were then fed into the feature extraction stage.

For hinge feature extraction, we employed a similar skeletonization process as a prerequisite. Hinge features capture the angular relationships between pairs of edge directions, providing a detailed representation of how strokes curve and connect within a character. For every pixel on the skeletonized character, pairs of edge directions within a predefined radius were identified. The computed angles were grouped into predefined bins, creating a histogram that represented the distribution of angular relationships for the given character. In this study, four pixel-length variations were applied: 2 px, 3 px, 4 px, and 5 px, which resulted in a different number of feature elements, i.e., 104, 252, 464, and 740 feature elements, respectively.

4.2.2 *SVM Model Training*

For classification optimization, we used 81 combinations of parameter C and γ with $C = 2^{-3}$, 2, 2^3, 2^5, 2^7, 2^9, 2^{11}, 2^{13}, 2^{15}, and $\gamma = 2^{-15}, 2^{-13}, 2^{-11}, 2^{-9}, 2^{-7}, 2^{-5}, 2^{-3}$, 2, 2^3. We used the exponentially growing sequences of the parameters as recommended in a practical guide [36] to SVM classification. In the model training stage, we distributed data for training and validation with an 80:20 ratio. The feature extraction from each character with four different pixel-length options was accomplished. Assuming that we choose features that belong to the letter *alef* and were extracted using the 2 px pixel-length setting, we then need to split the training and validation data to ensure an equal representation of samples from each scribe. This is illustrated by the following explanation (note: this does not apply to the letters *tet* and *samekh*). Since the letter alef has a set of 283 samples belonging to Scribe A and another set from Scribe B, the 112 validation samples (20% of total samples) should consist of 56 samples from Scribe A and 56 samples from Scribe B. For training, each scribe should be represented by 227 training samples. To ensure an equally distributed result, we employed a stratified cross-validation method. Furthermore, we implemented 5-fold cross- validation to create five different sets of training and validation data to avert the proneness of underfitting or overfitting. Next, we recorded the average of the training and validation accuracy scores obtained from 5-fold cross-validation. Finally, we obtained 22 combinations of parameters and a pixel-length type specific to each letter with the best training accuracy scores with low training-validation scores difference, to avoid

Table II. Best average scores of training and validation of the 22 trained character-specific models.

Letter	Edge-directional		Hinge	
	$\mu_{Training}$ (%)	$\mu_{Validation}$ (%)	$\mu_{Training}$ (%)	$\mu_{Validation}$ (%)
alef	70.4	58.2	81.8	63.0
ayin	68.0	58.0	81.1	60.3
bet	67.0	56.9	79.3	63.7
dalet	72.4	61.1	81.7	64.7
gimel	70.7	58.9	83.0	61.0
he	69.8	58.0	81.1	58.7
het	65.2	60.7	76.7	62.4
kaf	64.1	53.0	80.4	58.2
lamed	65.3	52.8	74.8	57.2
mem	70.7	59.4	79.7	63.2
nun	66.0	57.5	80.6	59.3
pe	63.6	56.4	79.7	60.2
qof	72.1	60.5	83.3	63.5
resh	69.9	61.1	80.1	65.3
samekh	63.9	57.2	71.5	59.6
shin	64.8	57.3	81.7	60.2
tav	66.2	62.4	75.0	66.9
tet	70.1	63.5	82.2	62.3
tsadi	62.7	56.2	78.3	58.5
vav	67.3	58.4	71.4	60.2
yod	66.4	59.1	79.4	58.8
zayin	61.9	58.9	72.7	61.6

overfitting. These combinations were used to test the new data derived from the testing set of Col. XXVII and Col.LIV.

5. RESULT AND DISCUSSION

5.1 *Character-Specific Model*

Table II presents the best results of parameter tuning obtained from the training and validation section of the proposed writer verification system. To determine which letters are more representative or less representative of their respective features (edge- directional and hinge), we can analyze the training, validation, and gap scores provided for each letter. Representative letters typically have higher scores (indicating better alignment with the feature set), while less representative letters present with lower scores, suggesting that the features struggle to capture the stylistic traits of those letters. Letters like *dalet*, *tet*, and *qof* are the most representative, as evidenced by their consistently high scores in both training and validation datasets. Their distinct stylistic traits make it easier to classify them with edge-directional and hinge features. Letters like *zayin*, *vav*, and *samekh* are less representative due to lower training and validation scores. This indicates that these letters either lack strong distinguishing features or that the current feature extraction methods struggle to model their traits effectively.

Table III. Mean scores for training, validation, and gaps.

Feature	Training mean (%)	Validation mean (%)	Gap mean (%)
Hinge features	78.9	61.3	17.6
Edge-directional features	67.2	58.4	8.8
Mean difference	11.7	2.9	8.8

To assess the effectiveness of the hinge and edge-directional features, we analyzed the mean differences in training scores, validation scores, and the gaps between these scores. Figure 3 intuitively depicts the trends in training and validation scores for each feature extraction method, complementing the results summarized in Table III.

The mean training score for hinge method was 78.9%, while edge-directional achieved a mean score of 67.2%, resulting in a mean difference of 11.7%. This indicates that hinge consistently outperformed edge-directional in capturing patterns within the training data. The higher training scores for hinge suggest that it is more effective at modeling the stylistic differences in the handwriting captured during training, which is crucial for distinguishing between writers. For validation scores, hinge achieved a mean of 61.3%, whereas edge- directional attained a mean score of 58.4%, resulting in a smaller mean difference of 2.9%. While hinge maintains an advantage on unseen data, the reduced gap between the validation scores suggests that edge-directional performs closer to hinge when generalizing to new data. This indicates that edge-directional might generalize more consistently but is overall less accurate. The gap between training and validation scores for hinge was 17.6%, while edge- directional exhibited a smaller gap of 8.8%, with a mean difference of 8.8%. The larger gap for hinge highlights a potential overfitting issue, as its performance on training data was significantly higher than on validation data. In contrast, edge-directional had a smaller gap, suggesting better generalization to new data despite its lower overall accuracy. While hinge demonstrated higher accuracy on both training and validation data, its larger gap between training and validation scores suggests it may be more prone to overfitting. Edge-directional, with its smaller gap, appears to generalize better but sacrifices some level of accuracy. Based on these results, hinge is recommended if achieving the highest possible accuracy is the primary goal, and overfitting can be mitigated through regularization techniques or additional data augmentation. However, if generalization is more critical, edge-directional may be a more robust choice.

5.2 *Decision Strategy Results: Scribe A versus Scribe B*

Following the interpretation of character-specific training and validation scores, it is imperative to analyze the results of the decision strategy on the building-block approach for determining the probabilities at word and sentence levels. This approach provides a practical perspective on how the trained features translate into the testing phase and helps

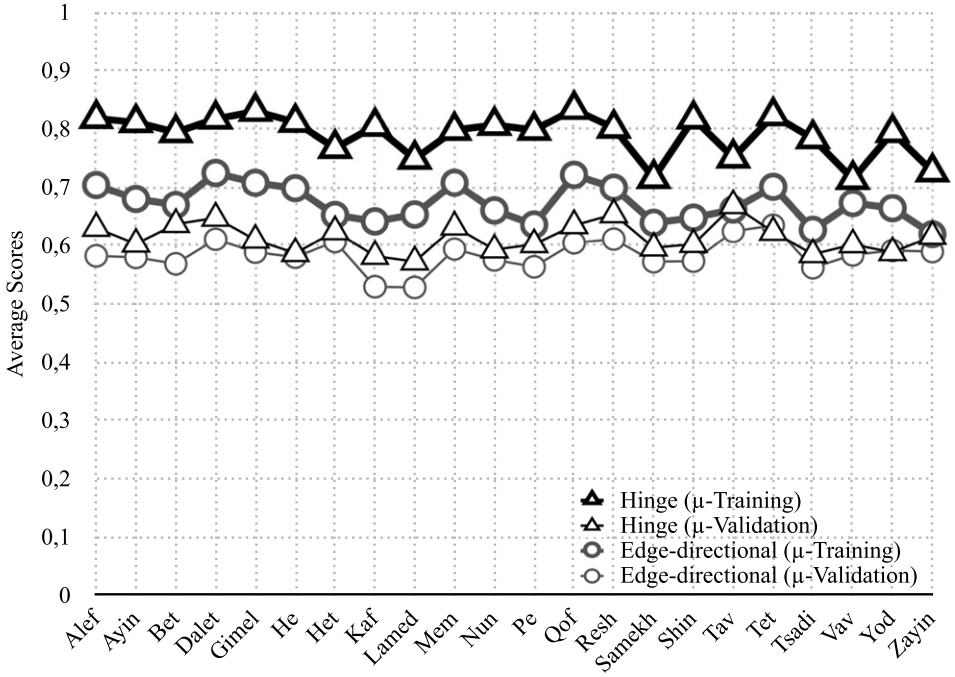

Figure 3. Best average scores of training and validation of the 22 trained character-specific models.

Table IV. The probability scores at word- and sentence-levels, a unit of analysis perceived to be written by Scribe A (Col. XXVII).

Word #	Edge-directional		Hinge	
	$S_{1,i}^{soft}$ (%)	$S_{1,i}^{majority}$ (%)	$S_{1,i}^{soft}$ (%)	$S_{1,i}^{majority}$ (%)
1	51.5	75.0	54.6	50.0
2	33.2	25.0	56.0	50.0
3	51.4	57.1	55.0	42.9
4	49.7	40.0	49.9	60.0
5	66.4	80.0	65.8	60.0
6	52.4	50.0	46.9	50.0
7	60.7	50.0	29.5	0
8	45.8	33.3	51.6	33.3
9	59.3	60.0	49.6	20.0
10	55.9	75.0	64.4	75.0
11	35.8	25.0	55.2	50.0
12	73.6	100	55.8	25.0
13	71.8	100	65.3	50.0
Sentence #	S_2^{soft} (%)	$S_2^{majority}$ (%)	S_2^{soft} (%)	$S_2^{majority}$ (%)
1	54.4	53.8	53.8	23.1

Table V. The probability scores on word- and sentence-level, a unit of analysis perceived to be written by Scribe B (Col. LIV).

	Edge-directional		Hinge	
Word #	$S_{1,i}^{soft}$ (%)	$S_{1,i}^{majority}$ (%)	$S_{1,i}^{soft}$ (%)	$S_{1,i}^{majority}$ (%)
1	49.3	66.7	62.6	66.7
2	51.4	42.9	34.1	14.3
3	51.1	40.0	37.9	30.0
4	53.7	50.0	53.8	50.0
5	39.0	0	33.1	33.3
6	60.7	62.5	56.8	62.5
7	40.1	0	40.7	66.7
8	40.1	25.0	64.2	100
9	55.6	57.1	54,3	42.9
Sentence #	S_2^{soft} (%)	$S_2^{majority}$ (%)	S_2^{soft}(%)	$S_2^{majority}$(%)
1	49.0	33.3	48.6	44.4

identify characteristics associated with each scribe's writing. The results are detailed in Table IV (showing probabilities for Scribe A, Col. XXVII) and Table V (showing probabilities for Scribe B, Col. LIV).

As shown in Table IV, for Scribe A, edge-directional outperformed hinge at sentence- level probabilities under both soft voting and majority voting strategies. Under soft voting, edge-directional achieved a sentence-level probability of 54.4%, which is slightly higher than hinge's 53.8%. Word-level probabilities for edge-directional under this strategy ranged from 33.2% to 73.6%, indicating a slightly wider spread but with consistently fewer extremely low values compared to hinge. Under majority voting, edge-directional also demonstrated stronger performance with a sentence-level probability of 53.8%, significantly higher than hinge's 23.1%. This marked difference indicates that edge-directional, despite variability in word-level probabilities, aggregates better in majority voting.

It can be inferred from Table V that Scribe B presented a different trend. Under soft voting, edge-directional again slightly outperformed hinge, with a sentence-level

probability of 49.0% compared to 48.6%. However, the margin of difference was narrower for Scribe B than for Scribe A. Word-level probabilities for edge- directional ranged from 39.0% to 60.7%, showing less spread compared to Scribe A's data and a more consistent performance. For majority voting, hinge surprisingly outperformed edge- directional, with a sentence-level probability of 44.4% compared to 33.3%. This indicates that hinge may be more robust when handling shorter sentence structures, as Scribe B's dataset consists of only 9 words per sentence compared to Scribe A's 13 words.

Analyzing the results further, several key observations emerge when comparing the impact of sentence length and voting strategies on edge-directional and hinge features:

Sentence Length Impact: Scribe B's shorter sentences (9 words) amplify the impact of low word-level probabilities on the aggregated sentence-level result. This effect is particularly evident in majority voting, where the edge-directional feature's performance drops significantly for Scribe B (33.3%) compared to Scribe A (53.8%).

Edge-directional's Stability: Edge-directional consistently outperformed hinge under soft voting for both scribes, indicating that edge-directional is more stable when aggregating probabilities using this strategy. However, its performance under majority voting varies more, particularly for shorter sentences (Scribe B).

Hinge's Robustness in Majority Voting: While hinge performance was lower than edge-directional, it showed better sentence-level probabilities for Scribe B under majority voting. This suggests that hinge may handle extreme variability at the word level better than edge- directional in certain conditions.

The results emphasize the importance of selecting an appropriate voting strategy and feature representation. Soft voting appears more stable across scribes and features, making it a preferred approach when building sentence-level probabilities. Edge-directional emerges as the stronger candidate overall due to its higher sentence-level probabilities and more consistent word-level predictions under soft voting. However, the lower performance of both features for Scribe B highlights challenges in handling shorter sentences. Shorter units of analysis amplify inconsistencies in word-level predictions, especially under majority voting, where extreme values (e.g., 0%) can disproportionately affect the aggregated result.

5.3 *Evaluation of Probability Scores*

The relatively low probability scores observed in the analysis can be attributed to several underlying factors related to the data, feature extraction, and the nature of the scribes' handwriting. Two key possibilities contributing to these results are discussed below.

- **Limitations in Feature Representation and Overlapping Handwriting Styles** A significant factor influencing the low probability scores is the limited ability of the features— specifically, edge-directional and hinge—to capture the distinct differences between Scribe A and Scribe B. Both scribes may share overlapping stylistic traits, which make it difficult for the system to differentiate their handwriting accurately. Edge-directional and hinge features, which are used to represent characteristics such as stroke angles and curvatures, may not be sensitive enough to the subtle differences in each scribe's unique writing style. When the handwriting styles between two scribes exhibit significant overlap, such as similar slant angles and letter shapes, these features may fail to highlight the necessary distinctions, resulting in lower confidence in the classification and relatively low probability scores.
- **Human-Generated Assumptions and Predefined Labels** The system's training process is based on human-generated assumptions and predefined labels from AI-driven research mentioned in Section 2.1, particularly involving unsupervised writer identification of ancient texts such as the 1QIsa-a scrolls. In this study, data augmentation techniques were used to artificially expand the dataset and increase the robustness of the model. However, the assumptions regarding the scribes' styles and the division of the columns (such as the plane separation between Columns 1-54) might not fully capture the actual complexity of the scribes' handwriting. These assumptions—while valuable for training purposes—may not fully account for the nuanced and dynamic nature of handwriting. The predefined labels that were assigned to the scribes based on these assumptions may not be entirely accurate, leading to mismatches in the system's classification and, ultimately, to low or inconsistent probability scores.

6. CONCLUSION

While notable advancements have been made in computational writer verification, significant challenges persist, particularly in the analysis of multi-scribe manuscripts such as the 1QIsa-a Scroll. The building-block approach presented in this study, which aggregates character-level probability scores to derive word- and sentence-level outcomes, offers a versatile and scalable solution to these challenges. By integrating computational techniques with paleographic analysis, this method provides a promising tool for historical manuscript studies and encourages interdisciplinary collaboration.

Future research should focus on refining feature extraction techniques, particularly enhancing edge-directional and hinge features to better capture the distinguishing characteristics of handwriting. Further evaluation using diverse writer identification datasets may contribute to refining and optimizing these features, ensuring robust performance across different script styles and manuscript conditions.

REFERENCES

[1] R. H. Johnston, R. Easton, K. Knox, R. Eschbach, J. Tusinski, and M. C. Zapan, "Digital image restoration technology as applied to ancient degraded textual material using color imaging systems," Color Imaging Conf. **3**, 191–194 (1995).

[2] E. Dubois and P. Dano, "Joint compression and restoration of documents with bleed-through," Archiving **2**, 170–174 (2005).

[3] J. Wang and C. L. Tan, "Non-rigid registration and restoration of double-sided historical manuscripts," *2011 Int'l. Conf. on Document Analysis and Recognition* (IEEE, Beijing, China, 2011), pp. 1374–1378.

[4] R. Hedjam and M. Cheriet, "Novel data representation for text extraction from multispectral historical document images," *2011 Int'l. Conf. on Document Analysis and Recognition* (IEEE, Beijing, China, 2011), pp. 172–176.

[5] D. Longacre, "*A Contextualized Approach to the Hebrew Dead Sea Scrolls Containing Exodus*," Ph.D. thesis (University of Birmingham, Birmingham, UK, 2015).

[6] V. Klement, "Forensic writer recognition," in *Digital Image Processing*, edited by J. C. Simon and R. M. Haralick (Springer Netherlands, Dordrecht, 1981), pp. 519–524.

[7] A. Bensefia, T. Paquet, and L. Heutte, "A writer identification and verification system," Pattern Recognit. Lett. **26**, 2080–2092 (2005).

[8] Israel Museum, "The great isaiah scroll," (2024), accessed: March 29, 2025.

[9] J. van Drie and C. van Boxtel, "Historical reasoning: towards a framework for analyzing students' reasoning about the past," Educ. Psychol. Rev. **20**, 87–110 (2008).

[10] D. Wakelin, "Paleography," *The Encyclopedia of Medieval Literature in Britain* (John Wiley Sons, Ltd, Chichester, UK, 2017), pp. 1–6.

[11] E. Tov, *Scribal Practices and Approaches Reflected in the Texts Found in the Judean Desert* (Brill, Leiden, The Netherlands, 2018).

[12] M. Popović, M. A. Dhali, and L. Schomaker, "Artificial intelligence based writer identification generates new evidence for the unknown scribes of the dead sea scrolls exemplified by the great isaiah scroll (1QIsa^a)," PLoS One **16**, 1–28 (2021).

[13] P. Kumar and A. Sharma, "Segmentation-free writer identification based on convolutional neural network," Comput. Electr. Eng. **85**, 106707 (2020).

[14] A. Srivastava, S. Chanda, U. Pal, C. Wallraven, Q. Liu, and H. Nagahara, "Exploiting multi-scale fusion, spatial attention and patch interaction techniques for text-independent writer identification," *Pattern Recognit.* (Springer International Publishing, Cham, 2022), pp. 203–217.

[15] A. Nicolaou, A. D. Bagdanov, M. Liwicki, and D. Karatzas, "Sparse radial sampling LBP for writer identification," *2015 13th Int'l. Conf. on Document Analysis and Recognition (ICDAR)* (IEEE, Tunis, Tunisia, 2015), pp. 716–720.

[16] A. Bennour, C. Djeddi, A. Gattal, I. Siddiqi, and T. Mekhaznia, "Handwriting based writer recognition using implicit shape codebook," Forens. Sci. Intl. **301**, 91–100 (2019).

[17] S. He and L. Schomaker, "Gr-rnn: global-context residual recurrent neural networks for writer identification," Pattern Recognit. **117**, 107975 (2021).

[18] L. Xing and Y. Qiao, "Deepwriter: a multi-stream deep CNN for text-independent writer identification," *2016 15th Int'l. Conf. on Frontiers in Handwriting Recognition (ICFHR)* (IEEE, Shenzhen, China, 2016), pp. 584–589.

[19] A. Chahi, Y. El merabet, Y. Ruichek, and R. Touahni, "Cross multi-scale locally encoded gradient patterns for off-line text-independent writer identification," Eng. Appl. Artif. Intell. **89**, 103459 (2020).

[20] H. Sheng and L. Schomaker, "Fragnet: writer identification using deep fragment networks," IEEE Trans. Inform. Forens. Security **15**, 3013–3022 (2020).

[21] S. Chen, Y. Wang, C.-T. Lin, W. Ding, and Z. Cao, "Semi-supervised feature learning for improving writer identification," Inform. Sci. **482**, 156–170 (2019).

[22] V. Kumar and S. Sundaram, "Siamese-based offline word level writer identification in a reduced subspace," Eng. Appl. Artif. Intell. **130**, 107720 (2024).

[23] Y. Hannad, I. Siddiqi, C. Djeddi, and M. E.-Y. El-Kettani, "Improving arabic writer identification using score-level fusion of textural descriptors," IET Biometr. **8**, 221–229 (2019).

[24] R. Nasuno and S. Arai, "Writer identification for offline Japanese handwritten character using convolutional neural network," *The 5th IIAE Int'l. Conf. on Intelligent Systems and Image Processing 2017 (ICISIP2017)* (Institute of Industrial Applications Engineers (IIAE), Hawaii, USA, 2017), pp. 94–97.

[25] H. T. Nguyen, C. T. Nguyen, T. Ino, B. Indurkhya, and M. Nakagawa, "Text-independent writer identification using convolutional neural network," Pattern Recognit. Lett. **121**, 104–112 (2019).

[26] A. Durou, S. Al-Maadeed, I. Aref, A. Bouridane, and M. Elbendak, "A comparative study of machine learning approaches for handwriter identification," *2019 IEEE 12th Int'l. Conf. on Global Security, Safety and Sustainability (ICGS3)* (IEEE, London, UK, 2019), pp. 206–212.

[27] P. Kumar and A. Sharma, "DCWI: distribution descriptive curve and cellular automata based writer identification," Expert Syst. Appl. **128**, 187–200 (2019).

[28] A. Rehman, S. Naz, M. I. Razzak, and I. A. Hameed, "Automatic visual features for writer identification: a deep learning approach," IEEE Access **7**, 17149–17157 (2019).

[29] A. Durou, I. Aref, S. Erateb, T. El-mihoub, T. Ghalut, and A. Emhemmed, "Offline writer identification using deep convolution neural network," *2022 IEEE 2nd Int'l. Maghreb Meeting of the Conf. on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)* (IEEE, Sabratha, Libya, 2022), pp. 43–47.

[30] S. Dargan, M. Kumar, A. Garg, and K. Thakur, "Writer identification system for pre-segmented offline handwritten devanagari characters using k-nn and svm," Soft Comput. **24**, 10111–10122 (2020).

[31] M. Bulacu, L. Schomaker, N. Petkov, and M. A. Westenberg, "Writer style from oriented edge fragments," *Computer Analysis of Images and Patterns* (Springer, Berlin, Heidelberg, 2003), pp. 460–469.

[32] P. Diamantatos, E. Kavallieratou, and S. Gritzalis, "Directional hinge features for writer identification: the importance of the skeleton and the effects of character size and pixel intensity," SN Comput. Sci. **3**, 1–18 (2022).

[33] R. G. Brereton and G. R. Lloyd, "Support vector machines for classification and regression," Analyst **135**, 230–267 (2010).

[34] T. Tobing, S. Yildirim Yayilgan, S. George, and T. Elgvin, "Isolated handwritten character recognition of ancient hebrew manuscripts," Archiv. Conf. **19**, 35–39 (2022).

[35] T. Tobing, P. Škrabánek, S. Yildirim Yayilgan, S. George, and T. Elgvin, "Character-based writer verification of ancient hebrew square-script manuscripts: on edge-direction feature," Archiv. Conf. **20**, 155–158 (2023).

[36] C.-w. Hsu, C.-c. Chang, and C.-J. Lin, "A practical guide to support vector classification," (2003). https://www.csie.ntu.edu.tw/ cjlin/papers/guide/guide.pdf. Accessed: March 29, 2025.