algorithms

MDPI

Article

A Hybrid Deep Learning Framework for Multi-Symbol Recognition and Positional Decoding of Handwritten Babylonian Numerals

Loay Alzubaidi, Kheir Eddine Bouazza * and Islam Al-Qudah

Computer Information Science Division, Higher Colleges of Technology, Abu Dhabi P.O. Box 41012, United Arab Emirates; lalzubaidi@hct.ac.ae (L.A.); qudahuws@hotmail.com (I.A.-Q.)
* Correspondence: kbouazza@hct.ac.ae

Abstract

The Babylonian numeral system, developed more than four thousand years ago, is one of the earliest known positional number systems, employing a sexagesimal (base-60) structure and a limited set of wedge-shaped symbols. Despite their visual simplicity, Babylonian numerals exhibit substantial structural and positional complexity, particularly when multiple symbols are combined to represent larger numerical values. This complexity presents significant challenges for modern computational recognition, especially in handwritten and degraded archaeological contexts. Most existing research has focused on the recognition of isolated Babylonian numeral symbols, which does not adequately reflect real inscriptions where numerals typically appear as composite sequences. To address this limitation, this paper proposes a hybrid deep learning framework capable of identifying, interpreting, and computing the decimal values of multi-symbol handwritten Babylonian numerals. Building on prior work in single-symbol recognition, we construct a synthetic yet realistic dataset of composite numeral images by combining handwritten glyphs into sequences of two to four symbols while incorporating natural variations in spacing, alignment, and handwriting style. The proposed framework integrates a Convolutional Neural Network (CNN) for visual feature extraction with optional structural feature fusion, followed by a Support Vector Machine (SVM) classifier for reliable multi-class discrimination. A rule-based positional decoder is then applied to convert recognized symbol sequences into their corresponding decimal values using Babylonian base-60 logic. By combining visual recognition with positional numerical reasoning, the proposed system enables end-to-end interpretation of handwritten Babylonian numeral sequences. To the best of our knowledge, this work represents one of the first approaches to jointly classify, decode, and compute numerical values from multi-symbol handwritten Babylonian numerals, contributing to digital epigraphy, archaeological text analysis, and cultural heritage preservation.

Keywords: Babylonian numerals; cuneiform numeral recognition; handwritten digit recognition; deep learning; CNN–SVM hybrid model; multi-symbol numeral classification; positional number system; structural feature extraction; epigraphy; cultural heritage computing

Check for updates

Academic Editors: Andrea Barucci, Rocco Furferi, Francesco Buonamici and Igor Aizenberg

Received: 7 February 2026
Revised: 3 April 2026
Accepted: 13 April 2026
Published: 20 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The ancient Babylonian numeral system represents one of the earliest and most influential positional number systems in human history. Originating in Mesopotamia, this system

is based on a sexagesimal (base-60) structure and employs only two fundamental wedge-shaped symbols to construct numerical values through repetition and positional placement. Unlike modern decimal numerals, Babylonian numbers lack explicit digit separators or a fixed symbol for zero, resulting in representations that are highly context-dependent and stylistically variable. Differences in scribal practices, writing tools, and historical periods further contribute to visual diversity across inscriptions.

These characteristics pose substantial challenges for computational recognition, particularly when dealing with handwritten or degraded cuneiform numerals preserved on clay tablets. While the historical and mathematical foundations of Babylonian numerals have been extensively documented by scholars such as Robson [1], Friberg [2], and Muroi [3], their automatic interpretation remains a largely unresolved problem in the fields of pattern recognition and digital humanities.

Recent advances in machine learning and computer vision have stimulated increasing interest in the automated analysis of ancient scripts, including cuneiform writing. However, most existing computational approaches to Babylonian numeral recognition focus primarily on isolated symbols or limited numeric ranges. Such approaches do not adequately reflect realistic historical inscriptions, where numerals typically appear as composite sequences composed of multiple symbols arranged according to positional rules. In addition, many previous studies rely on synthetic datasets that do not fully capture the structural and positional complexity of authentic Babylonian numeral inscriptions.

In our previous work, we addressed the recognition of single handwritten Babylonian numerals using convolutional neural networks enhanced with structural features and margin-based classification [4]. Building upon this foundation, the present study extends the problem from isolated symbol recognition to the interpretation of composite multi-symbol numeral sequences. This task involves identifying individual glyphs within a sequence and decoding their positional relationships to compute the corresponding numerical value under Babylonian base-60 logic. To address these challenges, we propose a hybrid recognition framework that combines deep learning-based visual classification with rule-based positional decoding tailored specifically to the Babylonian numeral system. By integrating data-driven feature learning with explicit numerical reasoning, the proposed approach bridges the gap between glyph-level recognition and full numerical interpretation.

The key contributions of this paper are summarized as follows:

- Multi-symbol Dataset Construction: A curated dataset of handwritten Babylonian numeral sequences composed of two to four symbols is introduced, reflecting realistic variability in spacing, alignment, and handwriting style.
- Hybrid Recognition Framework: A hybrid CNN–SVM architecture is proposed, combining deep visual feature extraction with optional structural feature fusion to robustly classify composite Babylonian numeral images.
- Positional Numerical Decoding: A rule-based positional decoder grounded in Babylonian base-60 logic is integrated, enabling automatic conversion of recognized symbol sequences into their corresponding decimal values.
- End-to-End Interpretation: The proposed system jointly performs symbol recognition, positional reasoning, and numerical computation, providing an end-to-end solution for interpreting handwritten Babylonian numerals.

This work aims to advance the automatic digitization and interpretation of ancient numerical records by introducing a hybrid deep learning framework capable of recognizing multi-symbol handwritten Babylonian numerals and converting them into their corresponding decimal values. The proposed system provides practical tools for archaeological analysis, digital epigraphy, and cultural heritage preservation.

2. Literature Review

The recognition of ancient numerical scripts, particularly handwritten Babylonian numerals, has attracted increasing attention in recent years within the domains of digital epigraphy, pattern recognition, and historical document analysis. This section reviews relevant work spanning archaeological foundations, early computational approaches to cuneiform recognition, and recent advances in deep learning-based handwritten numeral recognition.

2.1. Historical and Archaeological Background

The mathematical structure and symbolic conventions of Babylonian numerals have been extensively studied in archaeological and historical scholarship. Foundational works by Robson [1], Friberg [2], and Muroi [3] document the development of the sexagesimal positional system and its use in administrative, astronomical, and mathematical texts. These studies highlight both the elegance and ambiguity of Babylonian numerical notation, particularly the absence of explicit place-value separators and standardized glyph forms.

2.2. Early Computational Efforts on Cuneiform and Numerical Symbols

Early computational approaches to cuneiform recognition primarily focused on general sign classification rather than numerical interpretation. Dencker et al. [5] proposed weakly supervised Convolutional Neural Networks aligned with transliteration data, enabling large-scale sign recognition without exhaustive manual annotation. Kriege et al. [6] adopted graph-based representations that explicitly modeled spatial relationships among strokes and wedges, capturing structural dependencies inherent in cuneiform writing. These approaches demonstrated the feasibility of automated cuneiform analysis but were not specialized for numerical semantics.

Subsequent studies sought to improve detection accuracy using object detection frameworks. Williams et al. [7] proposed DeepScribe, which employs a RetinaNet object detector combined with a ResNet classifier to localize and classify Elamite cuneiform signs. Similarly, Stötzner et al. [8] applied convolutional neural networks trained on annotated 3D renderings and illumination-augmented photographs to enhance robustness in cuneiform sign detection.

Research specifically targeting Babylonian numerals has been comparatively limited. Ahmed in [9] developed a backpropagation neural network to recognize handwritten Babylonian digits in the range 1–10, reporting a training accuracy of 99.34% and a testing accuracy of approximately 75%. Ahlawat et al. [10] proposed a hybrid CNN–SVM framework that achieved improved performance on benchmark numeral datasets. However, their approach focused on treating digits as isolated classes and did not incorporate explicit structural or positional descriptors, which are particularly important for interpreting composite Babylonian numerals.

Overall, while these early and intermediate efforts laid important groundwork for cuneiform symbol recognition, they largely overlooked the challenges posed by multi-symbol numeral sequences and the need for positional decoding within a sexagesimal number system. This gap motivates the hybrid, sequence-aware approach proposed in the present work.

2.3. Handwritten Digit Recognition and Deep Learning Advances

Deep learning has significantly advanced handwritten digit recognition, with Convolutional Neural Networks (CNNs) emerging as the dominant paradigm. Benchmark datasets such as MNIST [11] established CNNs as powerful models for handwritten numeral clas-

sification, while extended datasets such as EMNIST demonstrated their scalability across multiple scripts and character sets.

Building on these foundations, several studies have explored CNN-based models for non-Latin and historically complex numeral systems. Goel et al. [12], for example, demonstrated that fine-tuned CNN architectures can achieve high generalization performance when applied to handwritten Gujarati numerals, highlighting the adaptability of deep learning models to script-specific variations.

Beyond standard classification pipelines, recent research has emphasized the importance of spatial reasoning and contextual awareness. Liu et al. [13] introduced the Path Aggregation Network (PANet), which enhances feature aggregation across spatial hierarchies and improves information propagation in complex visual recognition tasks. Similarly, Ren et al. [14] proposed region proposal networks for robust object detection in challenging visual environments. Such approaches are particularly relevant for ancient manuscripts, where surface degradation, incomplete strokes, and background artifacts frequently occur.

Transfer learning and hybrid learning strategies have also been investigated to address data scarcity and visual ambiguity. Alzubaidi and colleagues [15–17] demonstrated the effectiveness of integrating handcrafted structural features with CNN-based representations, showing that the fusion of low-level geometric descriptors and high-level semantic features can significantly improve performance in low-resource and degraded image domains, including medical imaging and historical document analysis.

Collectively, these studies demonstrate the effectiveness of deep learning techniques for handwritten numeral recognition across various scripts. However, most existing approaches focus primarily on modern numeral systems and treat characters as isolated symbols. In contrast, Babylonian numerals exhibit unique structural and positional characteristics within a sexagesimal numeral system, where numerical meaning is determined not only by symbol appearance but also by the arrangement of symbols within a sequence.

Consequently, purely visual recognition models are often insufficient for interpreting Babylonian numerals in realistic scenarios. These limitations highlight the need for recognition frameworks that integrate visual classification with structural and positional reasoning. This observation motivates the hybrid CNN–SVM framework proposed in this study, which combines deep visual feature extraction with structural feature integration and rule-based positional decoding to support the interpretation of composite Babylonian numeral sequences.

2.4. Structural Feature Integration in Symbol Recognition

Recognizing the limitations of purely visual learning models, a growing body of research has explored the integration of handcrafted structural descriptors with deep learning architectures. Such approaches aim to explicitly encode geometric and topological properties that may be difficult for purely data-driven models to learn reliably, particularly in low-resource or highly variable settings. Kass and Vats [18], for instance, proposed an attention-based encoder–decoder model for handwritten text recognition, demonstrating improved robustness to structural variability in character sequences. Similarly, recent work has explored the integration of structural and trajectory-based features for handwriting recognition, enhancing the ability of models to capture stroke dynamics beyond raw pixel patterns. Garrido-Muñoz et al. [19], for example, provide a comprehensive survey of handwritten text recognition approaches that discuss structural and contextual enhancements in modern deep learning models.

In archaeological and historical script analysis, structural features have proven especially effective due to the stylized nature of glyphs and the frequent presence of degradation. Features such as line orientation, stroke counts, corner angles, and junction points have

been successfully employed to classify numerals and symbols in Indo-Arabic scripts [16], Brahmi inscriptions [20], and Mayan glyphs [21]. These studies consistently report that hybrid models—combining learned visual representations with explicit structural cues—exhibit improved generalization and interpretability when confronted with ambiguous, eroded, or stylistically diverse symbols.

The relevance of structural feature integration is particularly pronounced for Babylonian numerals, whose visual simplicity masks a rich geometric and positional structure. This body of work motivates the incorporation of domain-specific structural descriptors into the proposed recognition framework, complementing CNN-based feature learning and enhancing robustness in the presence of handwriting variability and inscriptional noise.

2.5. Summary of Comparative Studies

As summarized in Table 1, prior research on cuneiform and Babylonian numerals spans a range of domains and methodological paradigms, from weakly supervised CNNs for general cuneiform scripts to hybrid CNN–SVM models for isolated numeral recognition. While recent studies have increasingly adopted hybrid deep learning approaches, most existing methods remain limited to single-symbol classification or narrowly scoped numeric ranges. Moreover, object detection-based frameworks have primarily targeted general script recognition rather than numerical interpretation. Notably, none of the surveyed approaches explicitly address the recognition and positional decoding of multi-symbol Babylonian numeral sequences within a unified framework.

Table 1. Comparative Summary of Existing Methods for Cuneiform and Babylonian Numeral Recognition.

Study	Domain Focus	Method
Dencker et al. [5]	Cuneiform scripts	Weakly Supervised CNN
Kriege et al. [6]	General cuneiform	Graph + CNN
Stötzner et al. [8]	Cuneiform scripts	CNN-based sign detection
Ahmed [9]	Babylonian numerals (0–10)	BPNN
Ahlawat et al. [10]	Babylonian numerals	CNN + SVM
Alzubaidi et al. [15–17]	Deep learning hybrids	Hybrid CNNs
This work	Babylonian numerals (1–50, sequences)	CNN + Structural Features

2.6. Research Gap and Motivation

Despite notable progress in cuneiform and numeral recognition, existing studies exhibit several important limitations when viewed collectively. Most prior work has focused on isolated numeral symbols, often restricted to narrow ranges such as 0–10, or on general cuneiform sign recognition without explicit numerical interpretation. In addition, many approaches rely predominantly on synthetic datasets and treat Babylonian numerals as purely visual patterns, thereby underutilizing the structural and positional semantics that are intrinsic to the sexagesimal numeral system.

As a result, current methods struggle to generalize to realistic archaeological scenarios, where numerals commonly appear as composite sequences, display substantial handwriting variability, and are affected by surface degradation and inscriptional noise. The lack of integrated positional reasoning further limits the ability of these approaches to perform complete numerical interpretation rather than isolated symbol classification.

Motivated by these gaps, the present work addresses the recognition and interpretation of multi-symbol handwritten Babylonian numeral sequences. We introduce a dataset composed of numerals formed by combining two to four handwritten symbols, designed to reflect realistic inscription characteristics. Moreover, we integrate geometric structural

descriptors such as line orientation, contour structure, and angular relationships into a CNN-based recognition pipeline. By coupling visual classification with explicit positional decoding based on Babylonian base-60 logic, the proposed framework addresses ambiguity, degradation, and positional interpretation within a unified system.

3. Methodology

This section presents a detailed description of the proposed framework for recognizing and decoding handwritten Babylonian numerals. Although Babylonian numeral symbols are visually simple, their interpretation is governed by strict positional semantics rooted in the sexagesimal (base-60) numeral system. As a result, accurate recognition requires not only robust visual classification of individual symbols but also explicit modeling of their positional relationships within a numeral sequence.

To address these requirements, we propose a hybrid deep learning architecture capable of both classifying composite numeral sequences and converting them into their corresponding decimal values using Babylonian base-60 logic. The proposed methodology integrates dataset construction, image preprocessing, visual and structural feature extraction, model architecture design, and rule-based symbolic decoding into a unified end-to-end pipeline. This integrated design enables the system to bridge low-level visual perception with high-level numerical reasoning, forming the foundation for the subsequent experimental evaluation.

3.1. Problem Formulation

Unlike conventional handwritten digit recognition tasks, Babylonian numerals are composed of repeated wedge-shaped symbols whose numerical meaning is determined not only by their visual form but also by their positional arrangement within a sequence. Values are encoded using a sexagesimal (base-60) system, where combinations of unit and ten symbols are grouped positionally to represent numbers from 1 to 59, and higher values are formed through left-to-right positional placement. For example, the number 59 is expressed using two distinct symbols representing five tens and nine units, whereas the number 1 is represented by a single unit symbol. This dependence on positional semantics introduces a level of structural and interpretive complexity that is absent in modern decimal numeral systems.

In this work, we formulate the recognition of handwritten Babylonian numerals as a joint visual classification and positional decoding problem. Each input image contains a sequence of two to four handwritten symbols that together represent a single Babylonian number. Given such an image, the system is required to perform the following:

- Accurately identify each constituent glyph within the composite sequence;
- Determine the correct left-to-right reading order of the symbols;
- Compute the corresponding decimal value by applying Babylonian base-60 place-value logic.

This formulation naturally leads to a multi-stage recognition process that combines low-level visual perception with high-level numerical reasoning. An effective solution must therefore balance the representational power of deep learning for visual feature extraction with explicit symbolic reasoning grounded in the mathematical structure of the Babylonian numeral system. This problem formulation directly motivates the hybrid recognition framework proposed in the following sections.

The dataset consists of composite handwritten Babylonian numeral sequences formed by combining previously collected handwritten numeral glyphs. Each composite sample contains between two and four Babylonian numeral symbols arranged sequentially. The dataset is organized into three classes according to the number of symbols present in the

sequence: Class 2 (two-symbol numerals), Class 3 (three-symbol numerals), and Class 4 (four-symbol numerals). Each class contains 1012 samples, resulting in a total dataset size of 3036 composite numeral images. This balanced dataset structure ensures that the classification model is trained and evaluated under consistent conditions across all composite numeral categories.

3.2. Dataset Construction

To simulate realistic Babylonian numeral inscriptions, we constructed a structured dataset of handwritten Babylonian symbols and programmatically combined them into composite multi-symbol numeral sequences.

We begin with a base set of 1000 handwritten samples for each fundamental Babylonian numeral, including unit symbols (1–9) and tens symbols (10, 20, 30, 40, and 50). Representative samples of these handwritten numerals are illustrated in Figure 1, highlighting the natural variability in stroke shape, orientation, and writing style.

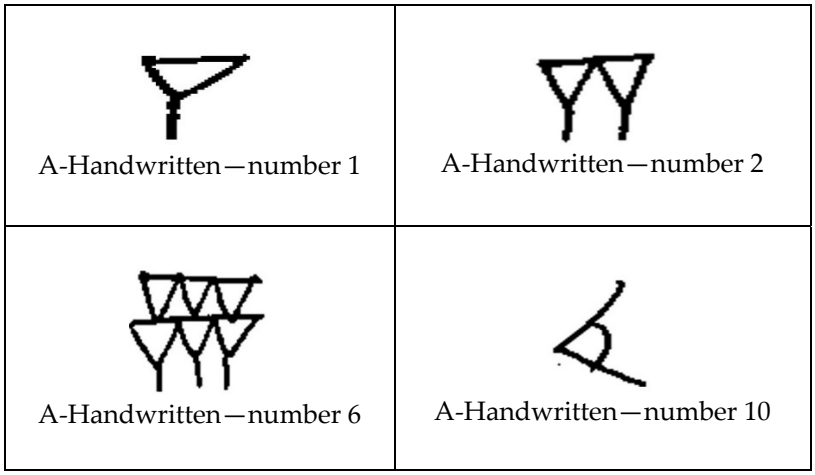


Figure 1. Sample of the handwritten Babylonian numbers.

Using this base set, we constructed composite numeral images containing two to four symbols, corresponding to realistic Babylonian number representations. Composite images were generated programmatically by combining isolated symbols according to controlled alignment, spacing, and placement rules that preserve valid Babylonian numeral structure. This process enables the creation of multi-symbol sequences while maintaining consistency with historical writing conventions.

To further enhance realism and improve model generalization, several data augmentation techniques were applied during dataset generation [22], including the following:

- Rotation
- Scaling
- Shearing
- Translation
- Background noise injection

Each composite image was annotated with both its symbolic representation and its corresponding decimal value, enabling supervised learning for both classification and numerical decoding tasks. An example of the image combination process for a composite Babylonian numeral is shown in Figure 2.

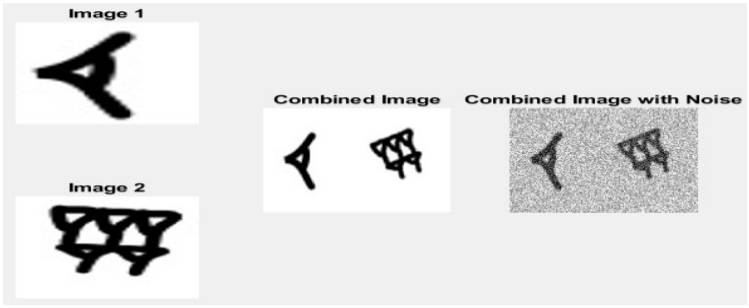


Figure 2. Image Combination for Babylonian number 15.

The final dataset was organized into classes based on the number of symbols per sequence and their associated numerical values, ensuring balanced representation across categories. A sample of the generated composite numeral images is presented in Figure 3.

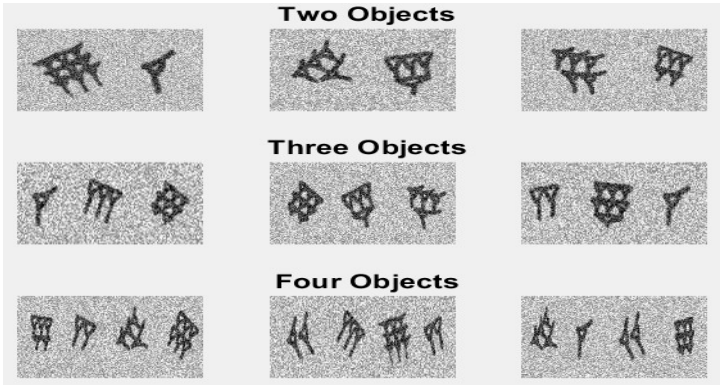


Figure 3. Sample of the generated dataset.

To ensure diversity and reduce potential bias in symbol placement, the composite images were generated using randomized alignment and spacing parameters within predefined structural constraints that preserve valid Babylonian numeral compositions.

3.3. Image Preprocessing

All input images were subjected to a standardized preprocessing pipeline designed to enhance glyph visibility, suppress noise, and normalize input dimensions prior to feature extraction and classification. This preprocessing stage aims to reduce variability caused by handwriting style, stroke thickness, and background artifacts while preserving the structural characteristics that are critical for accurate recognition.

The preprocessing pipeline consists of the following steps:

- Grayscale conversion: Input images are converted to grayscale to reduce computational overhead and to emphasize intensity-based structural information rather than color variations.
- Gaussian filtering: A Gaussian smoothing filter is applied to suppress high-frequency noise introduced during image acquisition and data augmentation.
- Binary thresholding: Images are binarized to isolate glyph structures from the background, facilitating subsequent visual and structural analysis. After binarization, the main glyph region is implicitly localized through centering and normalization, which helps suppress background noise introduced during augmentation and improves the robustness of subsequent feature extraction.

- Resizing and centering: Each image is resized and centered to a fixed resolution (e.g., 64 × 64 pixels), ensuring consistent input dimensions across the dataset and improving training stability.
- Optional edge detection: When structural feature extraction is enabled, the Canny edge detector is applied to enhance contour visibility and highlight stroke boundaries. These preprocessing operations standardize the input representation and improve the visibility of structural features, thereby facilitating more reliable feature extraction and classification by the subsequent CNN–SVM pipeline.

Figures 4 and 5 illustrate the outcome of the preprocessing and feature enhancement steps for representative handwritten Babylonian numerals, showing examples for both single-symbol and multi-symbol inputs.

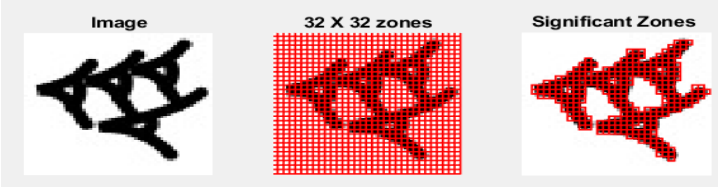


Figure 4. Feature extraction for single number.

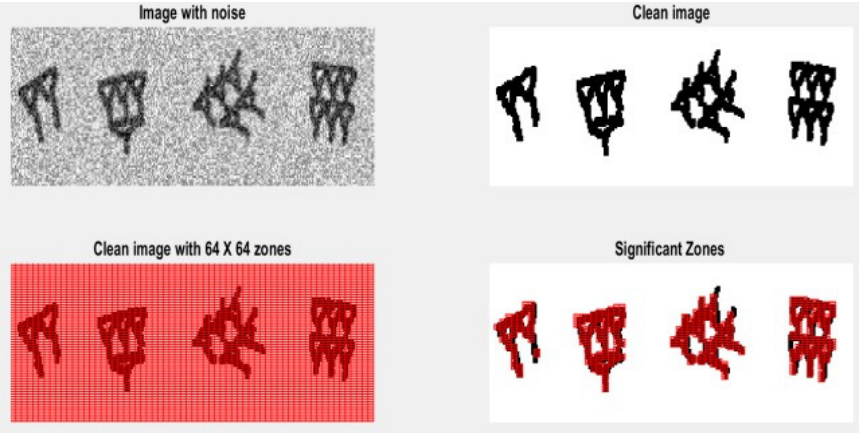


Figure 5. Structural feature extraction for multi-symbol Babylonian numerals.

3.4. Structural Feature Extraction

To enhance recognition performance beyond raw pixel-based analysis, structural feature extraction is incorporated as a complementary component to the deep learning pipeline. Babylonian numerals, although visually simple, are composed of distinct wedge-shaped strokes arranged in vertical and horizontal orientations, forming characteristic angular patterns. Capturing these geometric and spatial properties explicitly provides valuable discriminative information that is difficult for purely data-driven models to learn reliably, particularly under conditions of handwriting variability and image degradation.

These structural descriptors complement the CNN-based feature representations by explicitly encoding geometric characteristics of Babylonian numeral shapes, thereby improving discrimination between visually similar symbols.

3.4.1. Preprocessing and Edge Detection

The structural feature extraction process begins with standardized image preprocessing, as described in Section 3.3. Each input image is converted to grayscale to reduce computational complexity and emphasize intensity-based structural information. The Canny edge detector is then applied to highlight glyph contours and stroke boundaries,

producing a binary edge map that serves as the basis for subsequent geometric analysis. This step enhances the visibility of line segments and junctions that are critical for characterizing Babylonian numeral structure.

3.4.2. Line and Corner Detection

Following edge detection, the Hough Transform is employed to identify prominent straight-line segments within the image. Detected lines are categorized into vertical and horizontal orientations, corresponding to the fundamental stroke components of Babylonian numeral symbols. These line segments represent the primary building blocks from which numeral shapes are constructed.

In addition, the Harris corner detector is applied to identify intersection points and angular features within the glyphs. Corners and junctions capture the complexity of stroke interactions and curvature, which often differentiate visually similar numerals. These angular features are particularly useful in distinguishing numerals that share similar stroke counts but differ in geometric arrangement.

3.4.3. Feature Vector Construction

From the detected structural elements, a compact feature vector is constructed for each image. The core components of the feature vector include the following:

- Number of vertical line segments
- Number of horizontal line segments
- Total number of detected corners or angular points
- (Optional) Line lengths and inter-line angles for finer discrimination

This representation provides a concise yet expressive summary of the numeral's geometric composition. Table 2 presents sample feature vectors extracted from representative Babylonian numerals, illustrating distinct structural patterns across units and tens symbols.

Table 2. Structural feature vectors for representative Babylonian numerals.

Babylonian-N	# V-Lines	# H-Lines	# Angles
1	1	4	16
2	1	2	18
3	2	4	21
4	2	0	31
5	1	0	35
6	2	0	43
7	2	0	37
8	3	0	35
9	3	2	57
10	0	0	7
20	0	2	14
30	0	1	20
40	0	0	28
50	0	0	34

3.4.4. Structural Observations

Several observations can be drawn from the extracted structural features. Numerals greater than nine (i.e., 10, 20, 30, 40, and 50) exhibit a gradual increase in angular complexity, with approximately seven additional angular features per multiple of ten. In contrast, numerals between 1 and 9 are dominated by vertical line segments, which are largely absent in tens symbols. These consistent structural distinctions enable the model to disambiguate visually similar glyphs, particularly in noisy or degraded conditions.

3.4.5. Visual Representation

To facilitate interpretation, Figure 6 illustrates the structural feature extraction process for selected Babylonian numerals. Detected line segments and corner points are overlaid on the original glyph images, providing visual insight into the geometric cues utilized by the model during training.

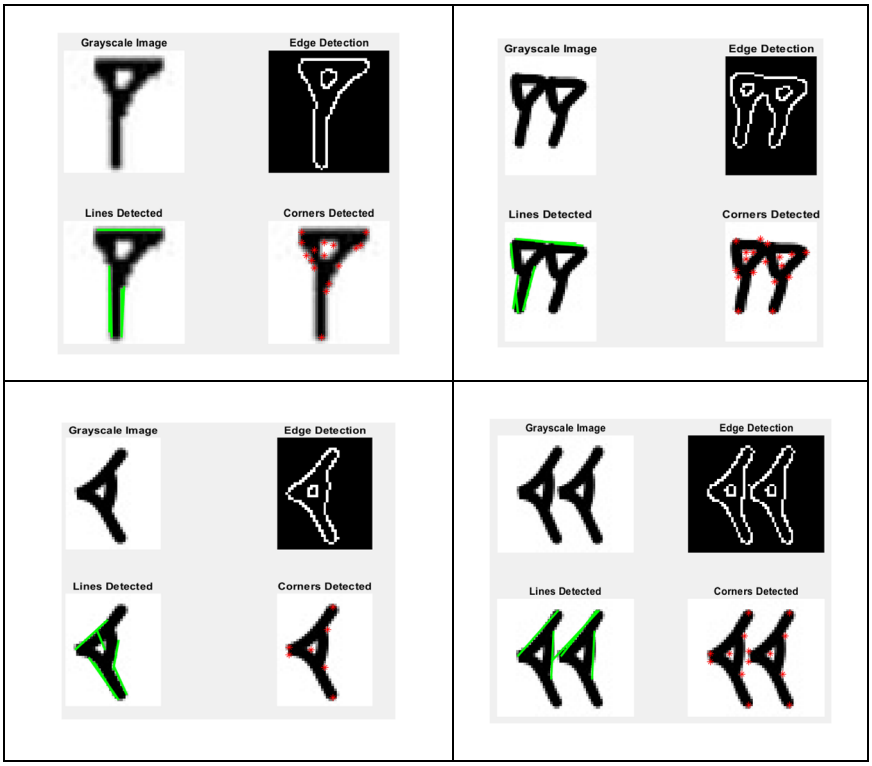


Figure 6. Feature extraction for 4 Babylonian numerals.

3.5. Model Architecture and Integration

To accurately recognize and classify composite handwritten Babylonian numerals, we design a hybrid recognition architecture that combines the hierarchical feature learning capabilities of Convolutional Neural Networks (CNNs) with the discriminative strengths of Support Vector Machines (SVMs). This design choice is motivated by the need to robustly handle visually similar numeral patterns, limited training data, and structural ambiguity inherent in ancient handwritten scripts.

In the proposed architecture, the CNN component serves as a powerful feature extractor, learning high-level visual representations from preprocessed numeral images. These learned representations are optionally augmented with handcrafted structural features described in Section 3.4, enabling the model to exploit both data-driven and domain-specific geometric cues. The resulting feature vectors are then passed to an SVM classifier, which performs the final decision-making using margin-based separation, providing improved robustness in multi-class and overlapping feature spaces.

This hybrid CNN–SVM integration enables the system to effectively process multi-symbol numeral sequences while maintaining high classification accuracy under ambiguous and noisy conditions. The overall architecture is designed to support modular integration of visual and structural features, facilitating reliable recognition and forming the basis for subsequent positional decoding and numerical interpretation.

3.5.1. CNN Feature Extraction Module

The core component of the proposed architecture is a custom Convolutional Neural Network (CNN) designed to extract high-level visual representations from preprocessed Babylonian numeral images. This CNN serves as a feature extractor, learning hierarchical spatial patterns that characterize handwritten cuneiform strokes and their arrangements within composite numeral sequences. The use of deep convolutional architectures for hierarchical feature extraction follows the paradigm established by early CNN models for large-scale image recognition [23].

The CNN architecture is composed of multiple convolutional layers that apply learnable filters to the input images, enabling the detection of local spatial features such as edges, stroke orientations, and texture variations that are characteristic of Babylonian numerals, following deep hierarchical convolutional design principles established in modern architectures [24]. Each convolutional layer is followed by a Rectified Linear Unit (ReLU) activation function, introducing non-linearity and allowing the network to model complex relationships between visual features and numeral classes.

To improve computational efficiency and enhance invariance to small spatial translations, max-pooling layers are interleaved between convolutional blocks. These pooling operations progressively reduce the spatial resolution of the feature maps while retaining the most salient activations. Batch normalization layers are applied after convolutional operations to stabilize training by maintaining consistent activation distributions, which accelerates convergence and improves generalization performance.

At the final stage of the CNN, the learned feature maps are flattened and passed through one or more fully connected layers to produce a compact intermediate feature vector. This vector encapsulates the essential spatial and structural characteristics of the composite numeral image and serves as a high-level representation for subsequent classification and feature fusion stages, consistent with established principles of deep representation learning [25].

CNN Training Configuration

The CNN model was implemented using a sequential architecture consisting of three convolutional layers followed by max-pooling operations. The convolutional layers employ 3×3 kernels with 32, 64, and 128 filters respectively. Each convolutional layer is followed by batch normalization and a ReLU activation function to improve training stability and introduce non-linearity.

The feature maps produced by the final convolutional block are flattened and passed to a fully connected layer, producing a compact feature representation used by the classifier. During training, the CNN was optimized using stochastic gradient descent with momentum (SGDM) with a learning rate of 0.01. The model was trained for 10 epochs with a batch size of 32. These settings provided stable convergence while preventing overfitting on the moderately sized dataset.

3.5.2. SVM Classification Module

Instead of employing a conventional Softmax classifier at the output of the CNN, the proposed framework utilizes a Support Vector Machine (SVM) for the final classification stage. This design choice is motivated by the strong generalization capabilities of SVMs and their effectiveness in constructing margin-based decision boundaries, particularly in challenging classification scenarios.

SVMs are well suited for multi-class problems involving closely spaced feature distributions, where class boundaries may overlap in the learned feature space. They have also been shown to perform robustly in low-resource or moderately imbalanced datasets,

conditions under which deep neural classifiers are more prone to overfitting. These properties are especially relevant in the context of ancient handwritten numerals, where symbol variability, noise, and degradation introduce additional ambiguity.

In the proposed architecture, the CNN functions exclusively as a feature extractor, producing rich, high-dimensional descriptors that encode the visual characteristics of each composite numeral image, following established hybrid CNN–SVM classification paradigms for handwritten digit recognition [26]. These descriptors are then provided as input to the SVM classifier, which performs the final label prediction based on support vectors and kernel-based decision boundaries. By decoupling feature learning from classification, the system benefits from the representational power of deep learning while leveraging the discriminative precision of SVMs.

This hybrid CNN–SVM configuration combines deep feature abstraction with high-confidence separation, resulting in improved robustness and classification accuracy when handling overlapping classes and noisy handwritten inputs.

More recent studies have further explored hybrid CNN–SVM architectures for handwritten digit and image classification, reporting consistent improvements in robustness and generalization performance [27]. Hybrid CNN–SVM architectures have been shown to provide improved generalization in scenarios involving limited training data or overlapping feature distributions. In such cases, replacing the conventional Softmax classifier with an SVM allows the model to construct more stable margin-based decision boundaries in the learned feature space. This property is particularly beneficial for handwritten Babylonian numerals, where handwriting variability and structural similarity between symbols can lead to ambiguous visual representations.

SVM Configuration

The SVM classifier operates on the feature vectors extracted by the CNN. A radial basis function (RBF) kernel was used to model nonlinear decision boundaries in the learned feature space. Multi-class classification was implemented using a one-vs-one strategy, which constructs pairwise classifiers between classes and selects the final class label based on a voting mechanism. This configuration provides strong generalization performance when separating classes with overlapping visual characteristics.

3.5.3. Structural Feature Fusion (Optional Enhancement)

In selected configurations, the proposed CNN–SVM pipeline is further enhanced through the fusion of manually extracted structural features with deep visual representations. Specifically, the structural descriptors described in Section 3.4—such as counts of vertical and horizontal line segments, angular features, and contour-based measures—are concatenated with the CNN-derived feature vector prior to classification by the SVM.

This feature-level fusion enables the classifier to exploit complementary sources of information. While the CNN captures high-level, data-driven visual patterns, the handcrafted structural features explicitly encode domain-specific geometric properties of Babylonian numerals that may not be reliably inferred from pixel data alone. By integrating these two feature streams, the model benefits from both learned representations and expert-driven structural cues.

The dual-stream representation improves classification robustness, particularly in cases involving visually similar numeral patterns that differ primarily in geometric structure. For example, numerals such as “30” and “40” may appear visually ambiguous under noisy or degraded conditions yet exhibit distinct structural configurations that can be effectively discriminated using fused features. This enhancement is especially valuable in low-quality handwritten inputs, where reliance on visual features alone may be insufficient.

3.5.4. Model Summary

The complete workflow of the proposed CNN–SVM hybrid recognition framework is illustrated in Figure 7. The architecture integrates deep visual feature extraction, optional structural feature fusion, and margin-based classification into a cohesive recognition pipeline.

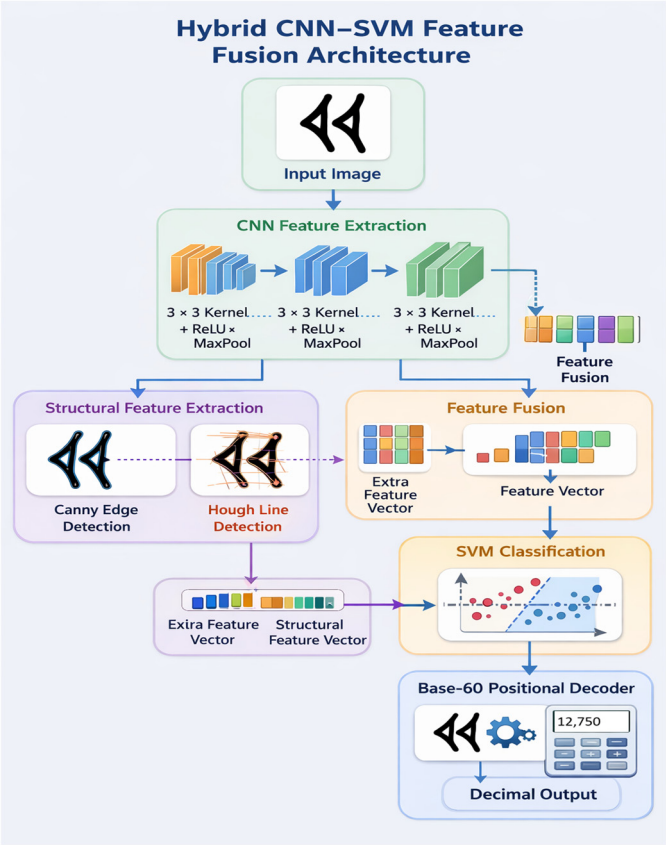


Figure 7. Overview of the CNN–SVM hybrid recognition framework.

As illustrated in Figure 7, the proposed recognition framework follows a hybrid architecture that combines deep visual feature extraction with margin-based classification. The CNN component is responsible for extracting hierarchical visual features from the composite Babylonian numeral images through successive convolutional and pooling layers. These learned feature representations are then flattened and used as input to a Support Vector Machine (SVM) classifier, which performs the final class discrimination. Unlike traditional CNN classifiers that rely on a Softmax layer, the SVM provides stronger margin-based separation between classes in the feature space, improving robustness when dealing with visually similar numeral sequences.

In this design, the CNN component learns hierarchical visual representations from preprocessed numeral images, capturing stroke-level and spatial patterns. When enabled, handcrafted structural descriptors are fused with these learned features to enhance geometric discrimination. The resulting feature vectors are then classified using an SVM, providing robust decision boundaries in the presence of overlapping classes and noisy inputs.

This modular architecture enables the system to accurately recognize both individual Babylonian numeral symbols and their combinations within composite, multi-symbol sequences. By decoupling visual recognition from numerical interpretation, the framework provides a reliable foundation for subsequent positional decoding. The following section

describes how the recognized symbol sequences are translated into their corresponding decimal values using Babylonian base-60 positional logic.

3.6. Positional Decoding of Babylonian Numerals

After the composite numeral sequence is classified, a rule-based positional decoder is applied to compute the corresponding decimal value according to the Babylonian sexagesimal numeral system. The decoder interprets the ordered sequence of recognized symbols and assigns positional weights based on powers of sixty. In this representation, each symbol contributes to the final numerical value depending on its position within the sequence. This step enables the system not only to recognize the visual structure of the numeral sequence but also to translate it into its equivalent decimal representation, thereby providing a complete end-to-end interpretation pipeline for Babylonian numerical inscriptions.

The decimal value of a Babylonian numeral sequence is computed using positional base-60 arithmetic as follows:

$$V = \sum_{i=0}^{n-1} d_i \times 60^{(n-1-i)}$$

where d_i represents the numerical value of the Babylonian glyph at position i, and n is the number of symbols in the sequence.

For example, a sequence of three symbols with values [3, 12, 5] is decoded as follows:

$$3 \times 60^2 + 12 \times 60^1 + 5 \times 60^0 = 10,800 + 720 + 5 = 11,525$$

Formalization of the Sexagesimal Positional System

To explicitly model the structural role of the Babylonian sexagesimal system, we formalize the positional representation of a numeral sequence as follows.

Let a Babylonian numeral be represented as an ordered sequence of k symbol groups:

$$S = (s_1, s_2, \ldots, s_k)$$

Each symbol group s_i corresponds to a digit value $d_i \in \{1, \ldots, 59\}$, obtained by aggregating unit and tens symbols within the group. The numerical value of the sequence in decimal form is computed using positional base-60 expansion:

$$N = \sum_{i=1}^{k} d_i \cdot 60^{k-i}$$

where:

- d_i is the decoded value of the i-th symbol group;
- k is the total number of positional groups;
- the leftmost group corresponds to the highest power of 60.

This formulation is equivalent to the positional decoding expression defined in Section 3.6, with a change in indexing convention from zero-based to one-based notation.

Decoding Pipeline

The positional decoding process is implemented in three stages:

a. Symbol Classification:
Each detected glyph is classified as a unit or tens symbol using the CNN–SVM model.

b. Digit Formation (1–59):

Symbols within a positional group are aggregated to form a digit value:

$$d_i = 10 \cdot t_i + u_i$$

where t_i and u_i denote the number of tens and unit symbols, respectively.

c. Positional Weighting:
The final decimal value is computed using base-60 positional expansion as defined above.

Treatment of Zero and Empty Positions

Historically, Babylonian numerals lack a consistent explicit zero symbol, particularly in early periods. In this work, we assume the following:

- Numeral sequences are contiguous, with no internal empty positions.
- Each position contains at least one symbol group representing a value in [1, 59].

This assumption aligns with the construction of the synthetic dataset and ensures unambiguous positional decoding. Handling missing positions or placeholder symbols is identified as an important direction for future work.

4. Experimental Setup

This section describes the experimental configuration used to evaluate the proposed hybrid CNN–SVM framework, including dataset composition, training protocol, and performance characteristics.

4.1. Dataset and Model Training

To assess the effectiveness of the proposed hybrid CNN–SVM model, we constructed a balanced dataset consisting of three composite Babylonian numeral classes, corresponding to different multi-symbol sequence configurations. Each class contains 1012 samples, resulting in a total of 3036 composite numeral images.

The dataset was randomly partitioned into training and validation subsets as follows:

- Training set: 708 samples per class
- Validation set: 304 samples per class

Each sample represents a Babylonian numeral composed of two to four handwritten glyphs, reflecting realistic inscription patterns rather than isolated symbols. To enhance robustness and reduce overfitting, data augmentation techniques were applied during training, including rotation, scaling, translation, and background noise injection. These augmentations simulate natural handwriting variability and surface degradation commonly observed in archaeological inscriptions.

To clarify the dataset organization, the classes correspond to the number of Babylonian glyphs present in each composite numeral sequence. Specifically, Class 2 contains numerals composed of two handwritten symbols, Class 3 contains numerals composed of three symbols, and Class 4 contains numerals composed of four symbols. Each class contains 1012 samples, resulting in a total dataset size of 3036 composite numeral images. The corresponding class distribution is summarized in Table 3.

Table 3. The table summarizes the composition of the dataset used in the experiments.

Class.	Number of Symbols in Sequence	Samples per Class	Description
Class 2	Two-symbol Babylonian numerals	1012	Numerals composed of two handwritten glyphs
Class 3	Three-symbol Babylonian numerals	1012	Numerals composed of three handwritten glyphs
Class 4	Four-symbol Babylonian numerals	1012	Numerals composed of four handwritten glyphs

This balanced dataset design ensures equal representation across all sequence configurations and allows the model to learn robust representations for composite Babylonian numerals of varying structural complexity.

4.2. Training Process and Performance

The hybrid CNN–SVM model was trained for 10 epochs using a fixed learning rate of 0.01. During training, the CNN component was optimized to learn discriminative visual representations, while the SVM classifier was trained on the resulting feature vectors for final class prediction.

The training process exhibited rapid and stable convergence. Classification accuracy exceeded 90% by the third epoch and continued to improve, stabilizing between 98% and 99% by the final epoch. Concurrently, the training loss decreased sharply from an initial value of approximately 14 to near zero, indicating effective optimization and strong model convergence. To further evaluate the effectiveness of the proposed hybrid architecture, we compared the CNN–SVM model with a baseline CNN classifier that uses a standard Softmax output layer. The baseline CNN uses the same convolutional feature extraction architecture but replaces the SVM classifier with a Softmax classification layer. This comparison allows us to assess the contribution of the margin-based SVM classifier in separating visually similar Babylonian numeral sequences. The comparative performance of the baseline and proposed models is summarized in Table 4.

Table 4. Performance comparison between baseline and proposed models.

Model	Classifier	Accuracy (%)
Baseline CNN	Softmax	95.62
CNN + Structural Features	Softmax	96.41
Proposed Hybrid Model	**CNN–SVM**	**97.14**

As shown in Table 4, the proposed CNN–SVM hybrid model achieves the highest classification accuracy among the evaluated configurations. The results demonstrate that replacing the conventional Softmax classifier with an SVM improves classification performance by providing stronger margin-based separation in the learned feature space. The integration of structural features further enhances discrimination between visually similar composite numeral patterns. These findings confirm that the hybrid CNN–SVM architecture provides improved robustness for recognizing complex handwritten Babylonian numeral sequences. The evolution of training accuracy and loss over epochs is illustrated in Figure 8.

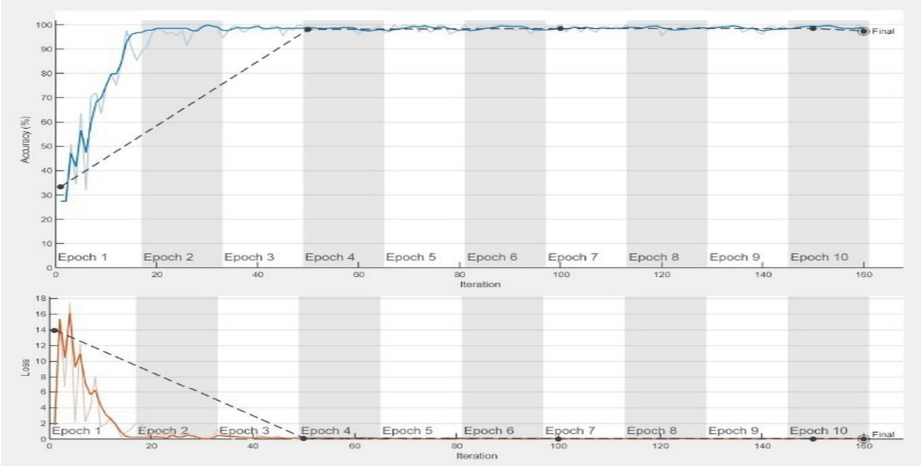


Figure 8. Training and validation accuracy and loss curves over 10 epochs.

4.3. Confusion Matrix and Classification Metrics

The classification performance of the proposed hybrid CNN–SVM framework was evaluated using standard metrics including precision, recall, and F1-score. These metrics provide a comprehensive evaluation of the model's ability to correctly identify composite Babylonian numeral sequences while minimizing false positives and false negatives.

Precision measures the proportion of correctly predicted samples among all predictions for a given class, while recall indicates the ability of the model to correctly identify samples belonging to that class. The F1-score represents the harmonic mean of precision and recall and provides a balanced measure of classification performance.

Based on the predictions obtained from the validation dataset, the classification metrics were computed for each of the three composite numeral classes. The results are summarized in Table 5.

Table 5. Classification performance metrics on the validation set.

Class	Precision (%)	Recall (%)	F1-Score (%)
2	98.67	97.37	98.02
3	97.01	96.05	96.53
4	95.82	98.03	96.91

The results indicate consistently high performance across all classes. In particular, Class 2 achieves the highest F1-score, reflecting strong model performance in recognizing two-symbol composite numerals. Class 4, which contains longer symbol sequences, exhibits slightly lower precision but maintains high recall, suggesting that most samples are correctly detected even when minor confusion occurs with structurally similar patterns. Overall, the balanced precision, recall, and F1-scores confirm the robustness of the proposed CNN–SVM hybrid model for recognizing composite handwritten Babylonian numeral sequences under variations in symbol arrangement and handwriting style.

4.4. Baseline Model Comparison

To evaluate the effectiveness of the proposed hybrid CNN–SVM framework, its performance was compared with two baseline configurations. The first baseline consists of a standard CNN model using a Softmax classifier for end-to-end classification. The second baseline extends this model by incorporating handcrafted structural feature extraction, while retaining the Softmax classifier. The proposed hybrid framework replaces the Softmax layer with a Support Vector Machine (SVM) classifier and integrates structural feature fusion to improve discrimination in the learned feature space.

These configurations correspond to different components of the proposed framework, allowing assessment of their individual contributions to classification performance. This staged comparison provides insight into the contribution of structural feature integration and margin-based classification within the proposed framework.

All models were trained and evaluated using the same dataset, preprocessing pipeline, and training configuration to ensure a fair comparison. Classification performance was measured using overall accuracy on the validation dataset. The performance comparison between baseline models and the proposed CNN–SVM framework is summarized in Table 6.

The results demonstrate that incorporating structural features improves classification performance by providing additional geometric information. Furthermore, replacing the conventional Softmax classifier with an SVM leads to additional performance gains by enabling stronger margin-based separation in the learned feature space. These findings

confirm that the proposed hybrid CNN–SVM architecture provides improved robustness for recognizing complex handwritten Babylonian numeral sequences.

Table 6. Performance comparison of baseline and proposed models.

Model	Accuracy (%)	Description
CNN + Softmax	94.12	Baseline CNN classifier
CNN + Structural Features (Softmax)	95.86	CNN with additional handcrafted geometric features
CNN–SVM + Structural Features	**97.14**	Proposed hybrid model with SVM classifier and structural feature fusion

4.5. Validation and Accuracy Improvements

Unless otherwise stated, experiments were conducted using the baseline CNN–SVM configuration without structural feature fusion. Structural features were introduced in additional experiments to evaluate their contribution as an optional enhancement to the visual feature representation. During the validation phase, the proposed CNN–SVM hybrid model leverages both learned visual representations and extracted structural features to improve recognition accuracy and glyph counting reliability. The inclusion of structural descriptors—such as zoning information and connected component analysis—enhances robustness when distinguishing between composite numeral sequences with overlapping visual characteristics.

A key advantage of the proposed framework lies in its ability to move beyond symbol-level classification and perform semantic numerical interpretation. Once a sequence of Babylonian glyphs is recognized, the system applies Babylonian base-60 positional logic to compute the corresponding decimal value. In this representation, each glyph contributes to the final value according to its position in the sequence, with the leftmost glyph corresponding to the highest positional weight and the rightmost glyph representing the units position.

Using the positional decoding scheme described in Section 3.6, a Babylonian numeral sequence is converted into its corresponding decimal representation through a base-60 positional expansion. Each recognized symbol is first mapped to its numerical value, and the final decimal value is obtained by applying positional weighting based on powers of 60.

For example, a recognized glyph sequence [10, 30] is decoded as follows:

$$10 \times 60^1 + 30 \times 60^0 = 600 + 30 = 630$$

This highlights that correct symbol classification does not always guarantee correct numerical decoding, particularly when errors occur in higher positional weights due to the exponential nature of the base-60 system.

This decoding approach significantly enhances the practical applicability of the proposed model, enabling accurate interpretation of multi-symbol Babylonian numerals rather than isolated glyph recognition. Such end-to-end numerical decoding is essential for epigraphic analysis, archaeological data interpretation, and large-scale digitization of ancient numerical records.

5. Discussion

5.1. Model Sensitivity to Numerical Base

Although the proposed framework is designed for the Babylonian sexagesimal (base-60) numeral system, its architecture is inherently modular and can be adapted to other positional number systems.

Formally, the positional decoding process can be generalized for any base b as follows:

$$N_b = \sum_{i=1}^{k} d_i \cdot b^{k-i}$$

where d_i represents the decoded value of the i-th positional group and k is the sequence length.

Within this formulation, the CNN feature extractor and SVM classifier are independent of the numerical base, as they operate solely on visual characteristics of symbols. The decoder, however, is explicitly dependent on the base through the positional weighting scheme.

Adapting the model to a different base (e.g., base-10 or base-20) would therefore primarily require the following:

- redefining symbol-to-value mappings;
- modifying positional weights;
- retraining the model only if the symbol set changes.

This demonstrates that the proposed architecture is not intrinsically tied to base-60 but rather represents a generalizable framework for positional numeral recognition and decoding.

This separation between perception (CNN–SVM) and numerical reasoning (decoder) highlights the modular design of the proposed framework.

5.2. Generalization to Other Numerical Systems

The proposed hybrid framework can be extended beyond Babylonian numerals to other positional number systems, including the following:

- Decimal (base-10) systems used in modern notation;
- Vigesimal (base-20) systems such as Mayan numerals;
- Binary (base-2) and hexadecimal (base-16) systems in computing.

Such generalization requires three main adaptations:

1. Symbol Representation:
   The visual recognition module must be trained on the appropriate symbol set.
2. Digit Construction Rules:
   The mapping from visual symbols to numerical values must reflect the structural conventions of the target system.
3. Positional Decoding:
   The decoder must apply the appropriate base b using the generalized positional formulation.

These considerations position the proposed method as a generic pipeline for structured numeral interpretation, rather than a domain-specific solution limited to Babylonian numerals.

5.3. Error Propagation and Positional Sensitivity

A key characteristic of positional numeral systems is that decoding errors are not uniformly distributed across positions. Due to exponential weighting, errors in higher-order positions have a significantly larger impact on the final numerical value.

For a positional error occurring at index i, the induced deviation scales proportionally to the following:

$$\Delta N \propto b^{k-i}$$

In the Babylonian system ($b = 60$), this effect is particularly pronounced, meaning that even small classification errors in leading positions can produce large numerical deviations.

This has several implications:

- High symbol-level accuracy does not necessarily guarantee correct numerical decoding.
- Robust performance requires sensitivity to positional importance.
- Evaluation should consider numerical correctness in addition to classification accuracy.

To better reflect this, future evaluations should incorporate the following:

- Absolute error: $\mid N_{\text{pred}} - N_{\text{true}} \mid$
- Relative error: $\frac{|N_{\text{pred}} - N_{\text{true}}|}{N_{\text{true}}}$

These metrics provide a more comprehensive assessment of system performance in positional decoding tasks.

5.4. Synthetic vs. Real Data and Domain Shift

The dataset used in this study is synthetically constructed from real handwritten Babylonian numeral symbols. While this approach enables controlled experimentation and sufficient data generation, it may not fully capture the variability present in real archaeological tablet inscriptions, such as:

- surface erosion;
- incomplete strokes;
- irregular spacing;
- noise and degradation.

This introduces a potential domain shift between training data and real-world scenarios. Although data augmentation techniques were applied to mitigate this gap, further validation on real cuneiform datasets is necessary.

Future work will focus on the following:

- incorporating real tablet imagery;
- applying domain adaptation techniques;
- evaluating robustness under realistic archaeological conditions.

5.5. Comparison with End-to-End Learning Approaches

Recent advances in deep learning have introduced end-to-end architectures, such as CNN–LSTM and Transformer-based models, capable of jointly modeling visual and sequential dependencies.

While such models offer strong sequence modeling capabilities, the proposed hybrid approach provides several advantages:

- Interpretability:
  The separation between visual recognition and symbolic decoding allows transparent inspection of each stage.
- Modularity:
  Individual components (feature extraction, classification, decoding) can be independently modified or improved.
- Robustness in low-data settings:
  Hybrid approaches are less prone to overfitting when training data is limited.
- Integration of domain knowledge:
  The rule-based decoder explicitly encodes positional logic, which may be difficult for purely data-driven models to learn reliably.

These properties make the proposed framework particularly suitable for historical and low-resource contexts.

5.6. Toward Base-Agnostic and Learnable Decoders

The current decoding mechanism is deterministic and assumes prior knowledge of the numerical base. While effective, this limits flexibility in scenarios involving multiple or unknown numeral systems.

A promising direction for future work is the development of base-agnostic or learnable decoding mechanisms, including the following:

- Parametric decoders, where the base b is treated as an explicit variable;
- Sequence learning models (e.g., LSTM, Transformers) that learn positional relationships directly from data;
- Hybrid neuro-symbolic approaches that combine learned representations with rule-based constraints.

Such extensions would enable the system to operate in more complex and uncertain environments, including multi-base or partially observed numeral systems.

Future work will also consider probabilistic decoding strategies to handle ambiguous or partially degraded symbols by generating multiple plausible interpretations.

5.7. Broader Implications and Interdisciplinary Applications

Beyond digital epigraphy, the proposed framework represents a general paradigm combining the following:

- visual perception (CNN);
- structured classification (SVM);
- symbolic reasoning (rule-based decoding).

This paradigm is applicable to a wide range of domains where structured patterns coexist with noise and uncertainty, including the following:

- Physical sciences: analysis of structured patterns in noisy experimental data;
- Environmental and geospatial sciences: modeling spatial-temporal patterns with uncertainty;
- Economic and financial systems: integration of learned patterns with rule-based constraints;
- Cognitive science: modeling perception and symbolic reasoning processes.

These broader applications highlight the potential of hybrid systems that integrate deep learning with structured reasoning.

6. Conclusions

This paper presents a hybrid CNN–SVM framework for the recognition and interpretation of multi-symbol handwritten Babylonian numeral sequences. By integrating a standardized preprocessing pipeline, explicit structural feature extraction, and margin-based classification, the proposed approach effectively addresses both visual recognition and positional decoding within the Babylonian sexagesimal numeral system. Experimental results demonstrate that the framework achieves high classification accuracy (97.14%) and maintains robust performance across multi-symbol numeral configurations.

The primary contributions of this work can be summarized as follows:

- A unified recognition pipeline for handwritten Babylonian numerals composed of multiple glyphs, reflecting realistic inscription patterns rather than isolated symbols.
- The integration of handcrafted structural descriptors—such as vertical and horizontal line counts and angular features—to complement CNN-based visual representations and improve generalization.
- A hybrid CNN–SVM classification strategy that enhances robustness when handling overlapping classes and noisy handwritten inputs.

- An explicit base-60 positional decoding mechanism that translates recognized symbol sequences into their corresponding decimal values, enabling semantic numerical interpretation.

Overall, the results confirm the feasibility of applying hybrid machine learning techniques to the digitization and interpretation of complex historical numeral systems. The proposed framework contributes to ongoing efforts in digital epigraphy and computational archaeology, and it provides a foundation for the automated analysis of ancient numerical records beyond isolated symbol recognition.

7. Future Work

Several directions for future research can further extend the scope and applicability of the proposed framework. One promising avenue is the integration of Vision Transformer-based architectures, which may improve contextual awareness across symbol sequences and enhance robustness when handling longer or more irregular numeral compositions.

Another important extension involves expanding the dataset to include real and heavily degraded tablet inscriptions, as well as a broader range of handwriting styles and historical variations. Such data would enable a more comprehensive evaluation of generalization performance under realistic archaeological conditions.

Finally, the proposed recognition and decoding framework may be extended beyond numerical symbols to support full cuneiform sign recognition, enabling automatic interpretation of more complex Mesopotamian texts. This extension would contribute to large-scale digitization efforts in digital epigraphy and cultural heritage preservation.

Author Contributions: Conceptualization, L.A.; methodology, L.A.; software, L.A.; validation, L.A., I.A.-Q. and K.E.B.; formal analysis, L.A.; investigation, L.A.; resources, L.A.; data curation, L.A.; writing—original draft preparation, L.A.; writing—review and editing, L.A., I.A.-Q. and K.E.B.; visualization, L.A.; supervision, I.A.-Q. and K.E.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study, along with the implementation of the proposed framework (including training scripts, model configuration, positional decoder, and dataset generation procedures), will be made publicly available through a GitHub repository upon acceptance.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Robson, E. *Mathematics in Ancient Iraq: A Social History*; Princeton University Press: Princeton, NJ, USA, 2009.
2. Friberg, J. A remarkable collection of Babylonian mathematical texts. *Not. Am. Math. Soc.* **2008**, *55*, 1076–1086.
3. Muroi, K. Babylonian number theory and trigonometric functions. In *Seki, Founder of Modern Mathematics in Japan*; Springer: Tokyo, Japan, 2013; pp. 31–47.
4. Alzubaidi, L.; Al-Qudah, I.; Bouazza, K.E. Deep Learning Approaches to Babylonian Numeral Recognition: A CNN–SVM Hybrid Model. In Proceedings of the IEEE International Conference on Computing (ICOCO), Kuala Lumpur, Malaysia, 12–14 December 2024; pp. 140–145.
5. Dencker, T.; Klinkisch, P.; Maul, S.M.; Ommer, B. Deep learning of cuneiform sign detection with weak supervision using transliteration alignment. *PLoS ONE* **2020**, *15*, e0243039. [CrossRef] [PubMed]
6. Kriege, N.M.; Fey, M.; Fisseler, D.; Mutzel, P.; Weichert, F. Recognizing cuneiform signs using graph-based methods. In Proceedings of the International Workshop on Cost-Sensitive Learning, San Diego, CA, USA, 5 May 2018; Volume 88, pp. 31–44.
7. Williams, E.C.; Su, G.; Schloen, S.R.; Prosser, M.C.; Paulus, S.; Krishnan, S. DeepScribe: Localization and Classification of Elamite Cuneiform Signs via Deep Learning. *ACM J. Comput. Cult. Herit.* **2025**, *18*, 1–32. [CrossRef]
8. Stötzner, E.; Homburg, T.; Mara, H. CNN Based Cuneiform Sign Detection Learned from Annotated 3D Renderings and Mapped Photographs with Illumination Augmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Paris, France, 2–6 October 2023; pp. 1680–1688.

9. Ahmed, H.B. Intelligence Recognition of Babylon Numerals System Using Back Propagation Neural Network. *Eur. Int. J. Sci. Technol.* **2015**, *4*, 207–212.
10. Ahlawat, S.; Choudhary, A. Hybrid CNN–SVM Classifier for Handwritten Digit Recognition. *Procedia Comput. Sci.* **2020**, *167*, 2554–2560. [CrossRef]
11. Cohen, G.; Afshar, S.; Tapson, J.; Van Schaik, A. EMNIST: Extending MNIST to handwritten letters. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 2921–2926.
12. Goel, P.; Ganatra, A. Handwritten Gujarati numerals classification based on deep convolution neural networks using transfer learning scenarios. *IEEE Access* **2023**, *11*, 20202–20215. [CrossRef]
13. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
14. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
15. Alghmgham, D.A.; Latif, G.; Alghazo, J.; Alzubaidi, L. Autonomous traffic sign (ATSR) detection and recognition using deep CNN. *Procedia Comput. Sci.* **2019**, *163*, 266–274. [CrossRef]
16. Latif, G.; Alghazo, J.; Elhassan, A.; Alzubaidi, L. Multi-Language Handwritten Digits Recognition Based on Novel Structural Features. *J. Imaging Sci. Technol.* **2019**, *63*, 020406. [CrossRef]
17. Latif, G.; Alghazo, J.M.; Alzubaidi, L.; Naseer, M.M.; Alghazo, Y. Deep convolutional neural network for recognition of unified multi-language handwritten numerals. In Proceedings of the IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), London, UK, 12–14 March 2018; pp. 90–95.
18. Kass, D.; Vats, E. AttentionHTR: Handwritten text recognition based on attention encoder–decoder networks. In Proceedings of the 15th IAPR International Workshop on Document Analysis Systems (DAS 2022), La Rochelle, France, 22–25 May 2022; pp. 507–522. [CrossRef]
19. Garrido-Munoz, C.; Rios-Vila, A.; Calvo-Zaragoza, J. Handwritten Text Recognition: A Survey. *arXiv* **2025**, arXiv:2502.08417. [CrossRef] [PubMed]
20. Pal, U.; Jayadevan, R.; Sharma, N. Handwriting recognition in indian regional scripts: A survey of offline techniques. *ACM Trans. Asian Lang. Inf. Process. (TALIP)* **2012**, *11*, 1–35. [CrossRef]
21. Roman-Rangel, E.; Pallan, C.; Odobez, J.M.; Gatica-Perez, D. Analyzing ancient maya glyph collections with contextual shape descriptors. *Int. J. Comput. Vis.* **2011**, *94*, 101–117. [CrossRef]
22. Kumar, T.; Brennan, R.; Mileo, A.; Bendechache, M. Image data augmentation approaches. *IEEE Access* **2024**, *12*, 187536–187571. [CrossRef]
23. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
24. Garrido-Muñoz, C.; Ríos-Vila, A.; Calvo-Zaragoza, J. Handwritten text recognition: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2026**, *48*, 4367–4385. [CrossRef] [PubMed]
25. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
26. Niu, X.X.; Suen, C.Y. A novel hybrid CNN–SVM classifier for recognizing handwritten digits. *Pattern Recognit.* **2012**, *45*, 1318–1325. [CrossRef]
27. Pal, U.; Wakabayashi, T.; Kimura, F. Handwritten numeral recognition of six popular Indian scripts. *Pattern Recognit.* **2010**, *43*, 2685–2697.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.