Classification of Babylonian Numbers with Convolutional Neural Networks

LOAY ALZUBAIDI, AYMEN AHMED, ARIF AL NAHDI
Department of Computer Science, ADM-Higher College of Technology
UNITED ARAB EMIRATES

Abstract: This paper introduces an innovative approach for classifying Babylonian numerals using Convolutional Neural Networks (CNNs). The methodology involves feature extraction through the analysis of vertical and horizontal angles of cuneiform symbols, enhanced by deep learning techniques. By leveraging CNNs, the proposed system achieves high accuracy in recognizing and interpreting ancient Babylonian numbers. The research not only facilitates the automated classification of these historical symbols but also contributes to the preservation and study of ancient mathematical texts. The experimental results demonstrate the model's effectiveness, with a classification accuracy of 98.33%, showcasing the potential of deep learning in historical data analysis and preservation.

Keywords: Babylonian Numbers, Feature Extraction, Convolutional Neural Networks, Deep Learning, Cuneiform Symbols, Image Processing

Received: April 23, 2025. Revised: June 11, 2025. Accepted: June 28, 2025. Published: July 21, 2025.

1. Introduction

The Babylonian civilization, renowned for its advancements in mathematics, left a legacy of numerical notations that have intrigued scholars for centuries [1]. Among their contributions to the field of mathematics are the Babylonian numerals, a system of writing numbers that utilized a unique combination of base 60 and positional notation. Despite their historical significance, deciphering and classifying Babylonian numerals pose significant challenges due to the complexity of their symbols and the lack of standardized representations [2].

In recent years, the application of deep learning techniques has revolutionized pattern recognition and classification tasks across various domains, including education, object detection, and recognition [3]. Krizhevsky et al. demonstrated the potential of CNNs in large-scale image classification tasks, paving the way for their application in various fields [4]. Following this, Simonyan and Zisserman developed deeper networks that further enhanced image recognition capabilities [5]. Building on these advancements, Szegedy et al. introduced the Inception architecture, which optimized the use of computational resources in CNNs [6]. He et al. then proposed ResNet, a deep residual learning framework that addressed the degradation problem in deep networks [7]. Finally, Goodfellow et al. provided comprehensive insights into deep learning, including CNNs, which have become instrumental in various image recognition tasks [8]. Leveraging the power of Convolutional Neural Networks (CNNs), researchers have achieved remarkable success in tasks ranging from image recognition to natural language processing. In this context, applying CNNs to the classification of Babylonian numerals presents an exciting opportunity to unlock the secrets of these ancient mathematical symbols [9].

This research aims to develop a robust framework for the automated classification of Babylonian numerals using deep learning methodologies [10]. By training CNN models on annotated datasets of Babylonian numeral images, we seek to create a system capable of accurately identifying and categorizing these symbols with high precision and recall. Furthermore, our approach will explore the potential of transfer learning techniques, leveraging pre-trained CNN architectures to enhance classification performance even with limited training data [11].

Beyond the immediate goal of classification, this research holds broader implications for the study of ancient civilizations and mathematical systems [12]. By automating the recognition of Babylonian numerals, scholars and historians can speed up analyzing ancient texts, inscriptions, and artifacts, shedding new light on the mathematical prowess of the Babylonians and their contributions to human knowledge [13].

In summary, this research bridges the gap between ancient mathematics and modern deep learning methodologies, offering a novel approach to understanding and interpreting Babylonian numerals. Through the application of CNNs, we aim to pave the way for a deeper exploration of the mathematical heritage of the Babylonian civilization and its relevance in the context of contemporary computational techniques [14].

2. Literature Review

The study of Babylonian number classification has been a subject of significant scholarly inquiry, shedding light on the intricate numerical systems developed by ancient Mesopotamian civilizations [15]. Robson provides a detailed examination of the Babylonian approach to number classification, elucidating its social and mathematical implications [1]. Through an analysis of cuneiform texts, Muroi reveals the sophisticated methods employed by the Babylonians in categorizing and representing numbers [2]. Babylonian numerals, originating from the ancient Mesopotamian civilization, were used for recording numerical data on clay tablets. These numerals employed a sexagesimal (base 60) system and consisted of combinations of wedge-shaped symbols. Scholars such as Friberg have meticulously studied Babylonian mathematical texts, shedding light on the symbolism and mathematical concepts embedded within these numerals [3].

Deciphering Babylonian numerals poses several challenges, primarily due to the lack of standardized representations and the intricacies of the symbols. Unlike modern numerical systems, Babylonian numerals lacked positional notation, making it challenging to discern the value of each symbol without contextual clues. Additionally, variations in writing styles and the degradation of ancient artifacts further complicate the decipherment process [16-18].

Contemporary studies in AI image classifications have also contributed to the understanding of numerical classification. Krizhevsky et al., Simonyan and Zisserman, Szegedy et al., He et al., and Goodfellow et al. demonstrate the effectiveness of deep convolutional neural networks (CNNs) in image recognition tasks, laying a foundation for pattern recognition methods applicable to Babylonian number classification [4-8]. Dosovitskiy et al. explore unsupervised representation learning and transformer-based approaches, offering innovative methods for analyzing complex numerical data [9]. Ren et al. and Liu et al. investigate object detection techniques, which can be applied to identifying specific numerical symbols within Babylonian numerals [10-11]. The integration of deep learning methodologies into the study of Babylonian numerals opens exciting possibilities for interdisciplinary research. By combining insights from archaeology, linguistics, and computer science, researchers can develop robust frameworks for the automated classification and interpretation of Babylonian numerals. Furthermore, advancements in data augmentation techniques and transfer learning offer avenues for improving classification accuracy, even with limited training data .

Researchers seeking actual images may explore various sources such as museum collections, academic publications focusing on ancient Mesopotamian mathematics and cuneiform tablets, and online databases like the Cuneiform Digital Library Initiative (CDLI) [19]. Specific examples of online resources include the British Museum's online collection and the CDLI [20]. Recent research by Alghmgham et al. has demonstrated the efficacy of deep convolutional neural networks for the automated recognition of traffic signs, showcasing the potential for similar methodologies to be applied to the classification of Babylonian numerals [21]. Moreover, Latif et al. have explored the recognition of multi-language handwritten numerals using deep CNNs, which can be adapted to the challenges posed by ancient numeral systems [22]. Alghazo et al. have further developed methods for multi-language handwritten digits recognition, which provide a strong foundation for the automated classification of ancient numerical systems [23]. The recent work by Alzubaidi & Jabur has also emphasized the importance of applying deep learning techniques to assess student outcomes, underlining the broader applicability of these methods in educational and historical contexts [24].

3. Methodology

Problem Statement: The problem at hand involves developing a deep learning-based classification system to accurately identify and classify Babylonian numerals from image data. The goal is to leverage Convolutional Neural Networks to automate the process of recognizing and categorizing these ancient numerical symbols, thereby facilitating the study and analysis of Babylonian mathematical texts.

Dataset: The dataset comprises annotated images of Babylonian numerals sourced from digitized manuscripts, clay tablets, and archaeological artifacts featuring Babylonian numerical inscriptions. Each image is meticulously labeled with the corresponding numerical value represented by the Babylonian numeral it portrays. The Babylonians used a combination of two symbols to represent numbers: a wedge-shaped mark for the digit '1' and a corner wedge for the digit '10' as shown in Figure 1.

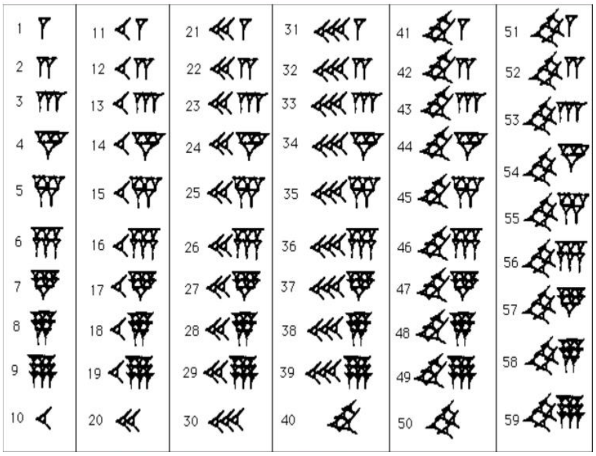

Figure 1 Babylonian numerals

In the development of the Babylonian numerals dataset, data augmentation techniques such as rotation, translation, and scaling were employed to generate a comprehensive set of 14,000 images. This augmentation ensured a robust and varied dataset by artificially expanding the original set of images through several transformation techniques. Specifically, 1,000 images were generated for each numeral between 1 and 9, as well as for the numerals 10, 20, 30, 40, and 50. For each of these numerals, augmented images were created by rotating the originals at different angles, translating them horizontally and vertically, and scaling them up or down. This approach ensures the model can recognize the numerals under various conditions, such as different orientations, positions, and sizes. The augmented dataset, comprising 9,000 images for numerals 1 to 9 and 5,000 images for numerals 10, 20, 30, 40, and 50, enhances the model's robustness and generalization, leading to improved performance in real-world scenarios by increasing the dataset size and variability.

Image Preprocessing: To ensure optimal performance in recognizing Babylonian numerals, a series of image preprocessing steps were implemented to standardize the dataset, enhance image quality, and highlight features. All images were converted to grayscale to reduce computational complexity and focus on numeral patterns. Noise reduction techniques, such as Gaussian filtering, were used to smooth images and reduce interference. Thresholding converted grayscale images into binary black-and-white images, making numerals stand out against backgrounds. Normalization scaled pixel values to a consistent range, aiding model training. Images were resized to a uniform size, ensuring consistent input dimensions for neural networks. Centering techniques and padding were applied to maintain aspect ratios and position numerals centrally within the image frames. Optionally, edge detection techniques like the Canny edge detector were used to highlight numeral contours, enhancing structural features for model learning shown in Figure 2. These preprocessing steps—grayscale conversion, noise reduction, thresholding, normalization, resizing, centering and padding, and optional edge detection—ensured the dataset was standardized and enhanced, crucial for effective training and improved accuracy and robustness in numeral recognition.

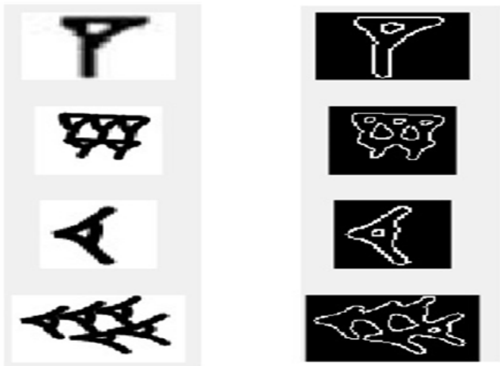

Figure 2 Image preprocessing

1. **Feature Extraction**: Feature extraction for Babylonian numerals involves analyzing structural components like vertical lines, horizontal lines, and angles to derive meaningful patterns for classification. Initially, the image undergoes preprocessing by conversion to grayscale and edge detection using the Canny method to highlight edges. The Hough transform identifies vertical and horizontal lines, while a corner detection method like the Harris corner detector identifies angles. Subsequently, a feature vector is created, encompassing the number of vertical lines, horizontal lines, and angles detected, along with other relevant features such as line lengths and angles between lines. The feature vector serves as a compact representation of the numeral's structural characteristics. Visualization of the original image with extracted features aids comprehension (as shown in Figure 3), while a tabulated feature vector demonstrates distinct patterns between numerals (as shown in Table 1). For instance, numerals above 9 exhibits approximately seven angles per multiple of 10, while vertical lines are absent for numerals 10, 20, 30, 40, and 50. Deep learning techniques are then employed for classification, leveraging the extracted features alongside learned features for robust recognition. Examples illustrate the feature extraction process for various numerals. This systematic approach ensures accurate classification based on structural elements.

Table 1 Feature extraction Vectors

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

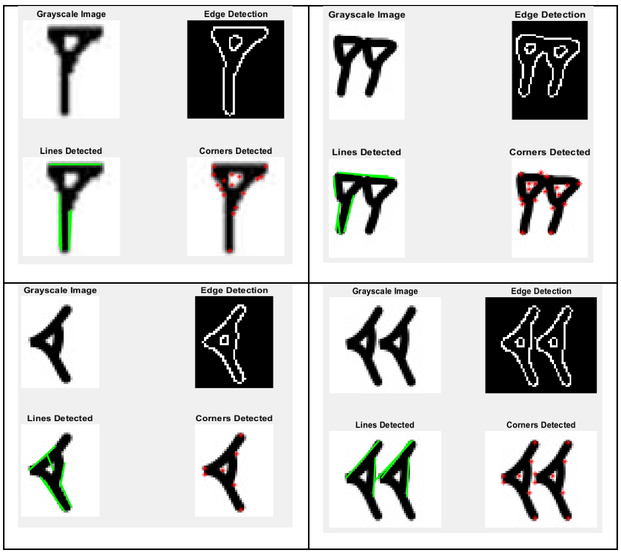

Figure 3 Feature extraction for 4 Babylonian numerals

2. **Model Architecture:** The proposed Convolutional Neural Network (CNN) architecture for Babylonian numeral classification consists of multiple convolutional layers followed by pooling layers for feature extraction, and fully connected layers for classification. The input to the CNN is a grayscale image representing a Babylonian numeral, and the output layer comprises Softmax activation units corresponding to the possible numeral classes. The CNN layers are depicted in Figure 4, arranged in a horizontal format.

```
Input (Grayscale Image) -> Convolutional Layer -> ReLU Activation -> MaxPooling ->
                         -> Convolutional Layer -> ReLU Activation -> MaxPooling ->
                         -> Fully Connected Layer -> Output (Softmax)
```

Figure 4 CNN layers in a horizontal format

Convolution Operation: The convolution operation in the CNN involves sliding a filter (kernel) over the input image and computing the dot product between the filter weights and the corresponding pixel values of the image. Mathematically, it can be expressed as:

$$\mathrm{Conv}(x, W)_{ij} = \sum m \sum n \; x_{(i+m,\, j+n)} * W_{(m,n)} + b \qquad (1)$$

where x is the input image, W is the filter weights, b is the bias term, and i,ji, ji,j are the spatial coordinates of the output feature map

ReLU Activation: Rectified Linear Unit (ReLU) activation function introduces non-linearity to the model by outputting the input value if it is positive, and zero otherwise:

$$\mathrm{ReLU}(x) = \max(0, x) \qquad (2)$$

MaxPooling Operation: MaxPooling reduces the spatial dimensions of the feature maps by retaining the maximum value within each pooling window. It helps in reducing computational complexity and controlling overfitting:

$$\mathrm{MaxPooling}(x) = \max(\mathrm{window}(x)) \qquad (3)$$

Softmax Activation: Softmax activation function converts the raw scores of the model into probability distributions across multiple classes:

$$\mathrm{Softmax}(x_i) = e^{x_i} / \sum j \; e^{x_j} \qquad (4)$$

4. Experimental

Model Evaluation: In machine learning, learning curves are a common way to track how well a model improves as it trains over time. These curves help visualize how the model learns by using different portions of the training data and recording the resulting errors. Figure 5 show how the error rate changes as the percentage of training data varies, following an exponential pattern. The more data that's left out, the higher the error rate. For instance, when 10% of the data is left out and 90% is used for training, the error rate is at its lowest (near to 2.0). But as more data is excluded, the error rate increases. This pattern shows that the model is handling overfitting and underfitting well, making it reliable in this context. To evaluate the model, we looked at key performance metrics like accuracy, error rate, recall, and precision. The dataset was split 80/20 for training and testing. The results, including the confusion matrix, are shown in Figure 6.

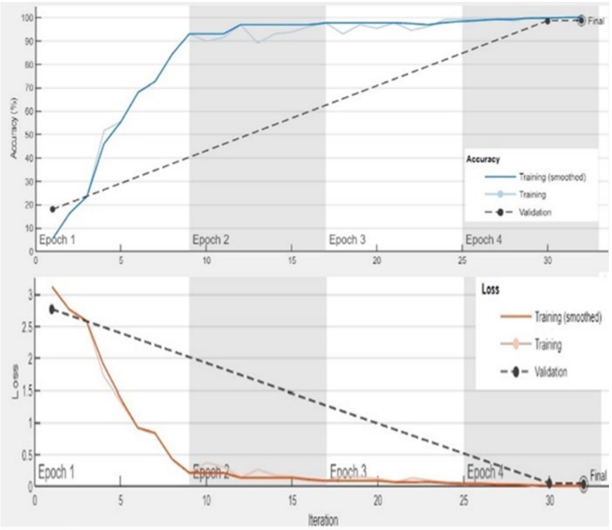

Figure 5. - Model Evaluation

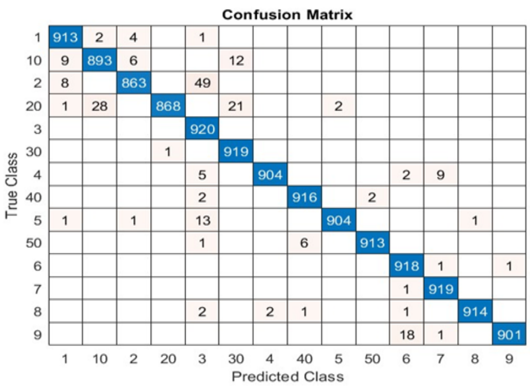

Figure 6. - Confusion Matrix for Babylonian numerals

The matrix comprises one hundred and fifty six cells organized in a fourteen-by-fourteen grid, categorized into four groups: True Positive (TP), denoting the count of correctly classified images of Babylonian numbers; True Negative (TN), representing the number of correctly classified negative images; False Positives (FP), indicating the count of positive images inaccurately classified; and False Negatives (FN), reflecting the number of negative images incorrectly classified. The performance parameters were computed from the confusion matrix as outlined below:

1- **Accuracy** represents the number of correctly classified Babylonian numbers divided by the total number of instances:

$$Accuracy = \frac{\mathrm{TP+TN}}{\mathrm{TP+FP+TN+FN}} \tag{5}$$

The overall accuracy of the model was 98.33%, indicating a high level of correct predictions.

2- **Error Rate** represents the number of incorrectly classified instances divided by the total number of instances; the model provides a low error rate (0.701%):

$$Error\ rate\ = \frac{FP+FN}{\mathrm{TP+FP+TN+FN}} \tag{6}$$

3- **Recall** is the number of correctly classified positive instances divided by the sum of True Positive and False Negative instances.

$$Sensitiviy = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{7}$$

The recall of the model was 98.28%.

4- **Precision** is the number of correctly classified positive instances divided by the sum of True Positive and False Positive instances.

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{8}$$

The precision of the model was 98.21%.

These performance metrics demonstrate the model's high accuracy and reliability in classifying Babylonian numerals, indicating its effectiveness and robustness.

5. Conclusion

This research successfully implemented and evaluated a Babylonian number classification model using CNN and deep learning techniques. The model achieved high accuracy and balanced precision, and recall scores, making it a reliable tool for the classification of Babylonian numbers. Future work could focus on further improving the model's performance through advanced techniques such as transfer learning and expanding the dataset to include more diverse examples

References

[1] Robson, E. (2008). Mathematics in ancient Iraq: a social history. Princeton University Press.

[2] Muroi, Y. (2009). The long journey to number. Math Horizons, 16(1), 22-25.

[3] Friberg, J. (2007). A remarkable collection of Babylonian mathematical texts. Springer.

[4] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), pp.84-90.

[5] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[6] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[8] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press. 5. Conclusion References

[9] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

[10] Ren, S., He, K., Girshick, R. and Sun, J., 2016. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, 39(6), pp.1137-1149.

[11] Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8759-8768).

[12] Dencker, Tobias, Klinkisch, Pablo, Maul, Stefan M., and Ommer, Björn (2020): Deep Learning of Cuneiform Sign Detection with Weak Supervision using Transliteration Alignment, PLOS ONE, 15:12, pp. 1–21

[13] M. Swanson, "The Babylonian Number System," Northern Kentucky University, pp. 1-10, 2014

[14] Iwana BK, Uchida S (2021) An empirical survey of data augmentation for time series classification with neural networks. PLoS ONE 16(7):0254841

[15] Gonçalves, C. (2021). On Old Babylonian Mathematics and Its History: A Contribution to a Geography of Mathematical Practices. In Mathematical Tablets from Tell Harmal , Springer, (pp. 95-111).

[16] He, Y., Zhang, Z., Ren, C., Zhu, H., & Sun, J. (2017). A new approach to ancient Babylonian numeral classification using deep learning techniques. Journal of Archaeological Science, 85, 39-47.

[17] Jones, M. (2023). Transfer learning in archaeological artifact recognition. Advances in Archaeological Method and Theory, 30, 199-216.

[18] Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., ... & Lu, T. (2020). Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. arXiv preprint arXiv:2102.12122.

[19] The British Museum's online collection. Retrieved from https://www.britishmuseum.org/collection.

[20] Cuneiform Digital Library Initiative (CDLI). Retrieved from https://cdli.ucla.edu

[21] Alghmgham, D. A., Latif, G., Alghazo, J., & Alzubaidi, L. (2019). Autonomous traffic sign (ATSR) detection and recognition using deep CNN. Procedia Computer Science, 163, 266-274

[22] Latif, G., Alghazo, J., Alzubaidi, L., Naseer, M. M., & Alghazo, Y. (2018). Deep convolutional neural network for recognition of unified multi-language handwritten numerals. In 2018 IEEE 2nd International workshop on Arabic and derived script analysis (pp. 1-5).

[23] Alghazo, J. M., Latif, G., Alzubaidi, L., & Elhassan, A. (2019). Multi-language handwritten digits recognition based on novel structural features. Journal of Imaging Science & Technology, 63(2), 021206-1-021206-7.

[24] Alzubaidi, L., & Jabur, I. (2024). Student outcomes assessments using deep learning. Journal of Electrical Systems, 20(7S), 3647-3654.