International Journal of Innovation Studies

CLASSIFICATION OF BABYLONIAN NUMBERS USING CONVOLUTIONAL NEURAL NETWORKS

Loay Alzubaidi, Aymen Ahmed, Arif Al Nahdi

Department of Computer Science, ADM-Higher College of Technology.

lalzubaidi@hct.ac.ae

aahmed1@hct.ac.ae

anahdi@hct.ac.ae

Abstract

This study presents a novel approach for classifying Babylonian numerals through the use of Convolutional Neural Networks (CNNs). The method combines structural feature analysis focusing on vertical and horizontal angles of cuneiform symbols with the capabilities of deep learning. By applying CNN architectures, the proposed system achieves a high level of accuracy in recognizing and interpreting these ancient numerical forms. Beyond offering an efficient classification tool, our research contributes to the preservation and analysis of historical mathematical texts. Experimental results show a classification accuracy of 98.33%, demonstrating the potential of deep learning methods in the study and safeguarding of ancient cultural heritage.

Keywords — *Babylonian Numerals, Feature Extraction, Convolutional Neural Networks, Deep Learning, Cuneiform Symbols, Image Processing*I.

INTRODUCTION

The Babylonian civilization, renowned for its mathematical achievements, developed a distinctive numerical system that has intrigued historians and mathematicians for centuries [1]. This system, combining a base-60 structure with positional notation, is preserved through cuneiform inscriptions. Despite its historical importance, the complexity of these symbols and the absence of standardized representations present ongoing challenges for accurate interpretation and classification [2].

In recent years, advances in deep learning have transformed the field of pattern recognition, with applications ranging from education and object detection to large scale image classification [3]. The seminal work of Krizhevsky et al. demonstrated the potential of Convolutional Neural Networks (CNNs) for large scale image recognition tasks [4]. Subsequent developments, such as the deeper architecture proposed by Simonyan and Zisserman [5], the computationally efficient Inception model introduced by Szegedy et al. [6], and the ResNet framework by He et al. [7], have further advanced CNN performance. Goodfellow et al. have also provided comprehensive insights into CNN methodologies and their broad applicability [8]. such advancements create an opportunity to apply CNN based approaches to the classification of these numerals, enabling faster and more accurate analysis of these ancient mathematical symbols [9].

This study seeks to develop a robust framework for automatically classifying these numerals using annotated image datasets [10]. By training CNN models enhanced where appropriate through transfer learning [11], Our proposed system seeks to achieve high precision and recall in classification. Such automation could significantly benefit archaeological and historical research [12], enabling scholars to analyze inscriptions and artifacts more efficiently [13].

It is worth emphasizing that, this investigation bridges the gap between ancient mathematics and modern computational methods [14], offering not only a practical classification tool but also a means to deepen our understanding of the mathematical legacy of one of the world's most influential civilizations.

II. LITERATURE REVIEW

This investigation 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, largely because of the lack of standardized representations and the intricacies of the symbols. Unlike modern numerical systems, Babylonian numerals lacked positional notation, posing challenges to discern the value of each symbol without contextual clues. Equally important, variations in writing styles and the degradation of ancient artifacts make even more challenging 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, applicable to identifying specific numerical symbols within Babylonian numerals [10-11]. The integration of deep learning methodologies into This investigation of Babylonian numerals opens exciting possibilities for interdisciplinary research. Through the combination of insights from archaeology, linguistics, and computer science, researchers can develop robust frameworks for the automated classification and interpretation of Babylonian numerals. In practice, 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]. A further point to consider is that, Latif et al. have explored the recognition of multilanguage 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 multilanguage 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].

III. 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. our objective is to leverage Convolutional Neural Networks to automate the process of recognizing and categorizing these ancient numerical symbols, thereby facilitating This investigation 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 Fig. 1.

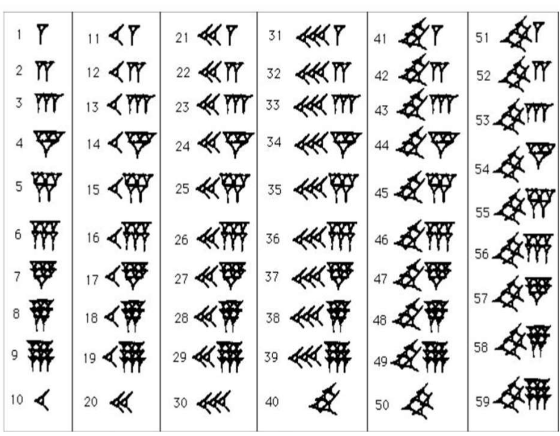

Fig. 1 Babylonian numerals

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. The adopted methodology 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 Fig. 2. These preprocessing steps grayscale conversion, noise reduction, thresholding, normalization, resizing, centering and padding, and optional edge detection—ensured the dataset standardized and enhanced, crucial for effective training and improved accuracy and robustness in numeral recognition.

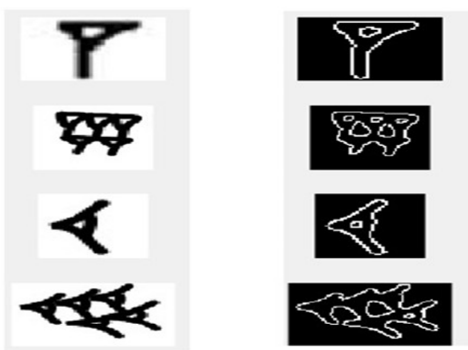

Fig. 2 Image preprocessing

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 Fig. 3), while a tabulated feature vector demonstrates distinct patterns between numerals (as shown in Table 1).

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

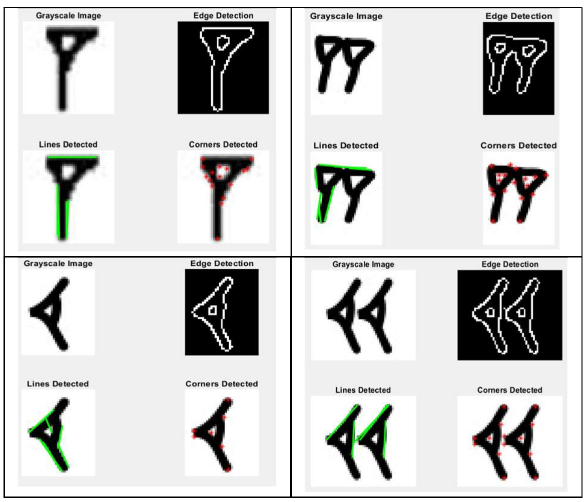

Fig. 3 Feature extraction for 4 Babylonian numerals

For example, 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.

Model Architecture: Our proposed Convolutional Neural Network 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 Fig. 4, arranged in a horizontal format.

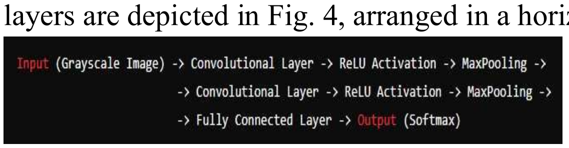

Fig. 4 CNN layers in a horizontal format

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:

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

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

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

$ReLU(x) = \max(0, x)$ (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:

$MaxPooling(x) = \max(window(x))$ (3)

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

$Softmax(x_i) = e^{xi} / \sum j\ e^{xj}$ (4)

Implementation The CNN model can be constructed using the Deep Learning Toolbox. This involves defining the layers of the network, specifying training options, and training the model using the annotated dataset. MATLAB provides pre-trained CNN architectures (e.g., AlexNet, ResNet) that can be finetuned for Babylonian numeral classification, along with tools for data augmentation, training visualization, and performance evaluation.

By following the adopted methodology, we aim to develop a robust deep learning-based classification system able to accurately identifying and categorizing Babylonian numerals, thereby contributing to This investigation and preservation of ancient mathematical heritage.

Model Evaluation: In the realm of machine learning, learning curves are frequently employed for algorithms that acquire knowledge progressively over time. These curves illustrate how effectively the model is learning by employing varying proportions of the training dataset to fit the classifier and document errors. Figures 5 & 6 depict the Error rate versus the Percentage of training data randomly extracted, revealing an exponential trend. As the percentage of excluded data rises, so does the error rate. for example, when 10% of the data is randomly omitted and 90% utilized, the error rate is at a minimum (2.07). As the percentage of excluded data increases, the error rate also escalates. these results indicate that the model successfully addresses overfitting and underfitting challenges during the training process, rendering it reliable in this context.

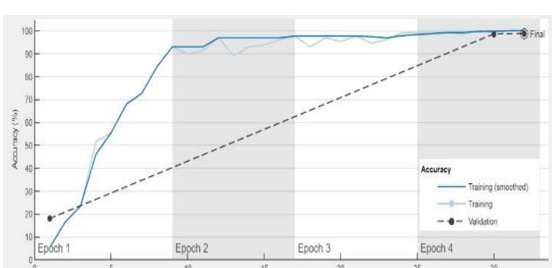

Fig. 5 Model Accuracy Curve

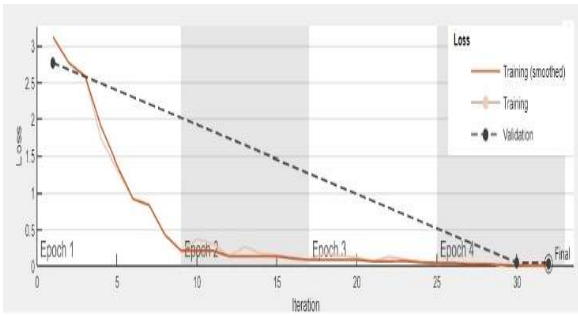

Fig. 6 Model Loss Curve

IV. EXPERIMENTAL

To assess the model, various performance parameters including accuracy, error rate, recall, and precision were considered. The dataset was divided into an 80/20 ratio for training and testing purposes. The experimental outcomes, including the confusion matrix, are presented in Fig. 7.

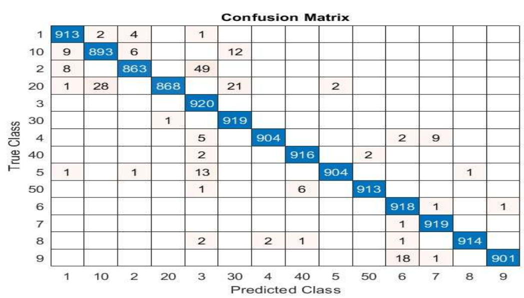

Fig.7 Confusion Matrix for Babylonian numerals

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:

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

$$Accuracy = \frac{\mathrm{TP+TN}}{\mathrm{TP+FP+TN+F}} \qquad (5)$$

The overall accuracy of the model was 98.33%, suggesting, perhaps, a notably high rate of correct predictions.

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%):

$$Error\ rate\ = \frac{F\mathrm{P}+F\mathrm{N}}{\mathrm{TP+FP+TN+FN}} \qquad (6)$$

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

$$Sensitiviy = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \qquad (7)$$

The recall of the model was 98.28%.

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

$$Precision = \frac{\text{TP}}{\text{TP+FP}} \quad (8)$$

The precision of the model was 98.21%.

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

V. 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

VI. 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.

[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.