Improved Deep Learning model for Ancient Cuneiform Symbols Classification

Raed Majeed[1,*], **Hiyam Hatem**[1], **Wael Abd-Alaziz**[1]
[1]Collage of Computer Science and Information Techonology,University of Sumer, Iraq
Emails: raed.m.muttasher@gmail.com; hiamhatim2005@gmail.com; w.abdalaziz@uos.edu.iq

Abstract

Cuneiform script, among the earliest writing systems, poses a distinct challenge for classification because of its complex symbols and varied linguistic contexts. This study investigates the use of Convolutional Neural Network (CNN) architectures for the classification of cuneiform symbols. The preprocessing includes resizing the cuneiform images to a uniform dimension and categorizing them into training, validation, and testing sets. A modified CNN model has been introduced. The CNN model demonstrates a lower parameter count in comparison to other deep learning models, which frequently necessitate significant storage capacity. The results from the CLI dataset indicate that the proposed CNN model reached an impressive accuracy of 99.55%, This study enhances computational approaches for the analysis of ancient scripts and underscores the significance of utilizing deep learning techniques within the fields of historical linguistics and digital humanities.

Keywords: Cuneiform classification; Convolutional Neural Networks (CNNs); Symbol recognition; Ancient script analysis; Deep learning linguistics

1. Introduction

The most renowned kind of ancient writing was cuneiform [1,2]. It was created by the Sumerians in Mesopotamia in south of Iraq between 3500 and 3000 BC. It is considered one of the Sumerians' greatest contributions to general culture [3]. The Latin word (cuneus) means wedge-shape or nails are the base for the writing method name. The writing procedure is done by sketching wedge-like forms on soft clay surface using a reed pen [4]. The Sumerian Documenting with cuneiform system of writing the literature, laws, events, i.e., is crucial for human history [5,6].

Thousands cuneiform tablets were found in Iraq in the early 1800s [7]. Many museums display these collections of tablets, which need skill specialist and patience to decipher. New technology as AI plays an essential factor to assist the cuneiform text classification. Information technology was needed to recognize cuneiform symbols. Cuneiform texts encompass an extensive variety of subjects, such as literary works, theology, physics, and historiography. By adopting mechanisms to categories cuneiform characters, it becomes straightforward to pinpoint their areas of concentration. Consequently, the primary goal of this study is to develop a model for classifying cuneiform symbols and to enhance the translation process of cuneiform texts by examining the effectiveness of deep learning algorithms in this classification task. This paper presents multiple contributions:

1. Develop deep learning models for the classification of cuneiform symbols in the (CLI) dataset.
2. The unigram method will be employed to extract features from cuneiform texts by breaking the text down into its smallest feasible items (symbols), thereby facilitating an optimization in the quantity of data as well as improving evaluating speed.
3. To address the issue of class imbalance in the dataset through the application of the oversampling technique.
4. A modified CNN model has been developed. The CNN model demonstrates a lower parameter count in comparison to pre-trained deep learning models, which frequently necessitate significant storage capacity.

The rest of this paper arranged as: the related works presented in Section 2, Materials and Methods are presented in Section3, Results and Discussions in Section 4, and Conclusions together with some future research directions aregiven in Section 5.

DOI: https://doi.org/10.54216/FPA.190209
Received: December 09, 2024 Revised: February 04, 2025 Accepted: March 01, 2025

2. Related Work

This section presents several studies that predominantly utilize deep learning techniques for the classification of cuneiform symbols. A particular emphasis is placed on balanced datasets provided by Jauhiainen et al. [8], which were made available to participants in the CLI-shared task at VarDial 2019 [9]. This approach aims to tackle the challenge of classifying symbols across various dialects and languages.

Al-Khateeb (2023) investigates the application of machine learning algorithms, such as deep neural networks (DNNs), in the classification of cuneiform symbols. The investigation emphasises the application of unigram features on a balanced dataset comprising seven cuneiform languages, displaying an impressive accuracy of 93% with DNNs [10].

Williams et al. (2023) introduce a systematic approach for the localisation and classification of Elamite cuneiform signs. The system employs a blend of RetinaNet for object detection and ResNet for classification, resulting in notable accuracy in the classification of cuneiform signs. The system achieved a top-5 classification accuracy of 0.89, offering significant transliteration support for academics [11].

Hagelskjaer (2022) A Case Study on Cuneiform Tablets explores the application of deep learning techniques for classifying extensive point cloud data derived from cuneiform tablets. The study utilises a deep neural network that demonstrates strong performance, effectively applying the model to the metadata of cuneiform artifacts [12].

Kapon et al. (2024) employ advanced machine learning techniques, such as Variation Auto-Encoders (VAEs), to analyze and date cuneiform tablets that span three millennia. The authors demonstrate that the application of machine learning significantly enhances the precision of dating and classifying these historical artifacts [13].

Saeed et al. (2024) introduced a technique for identifying Hebrew letters and classifying cuneiform tablets. The methodology employs YOLOv8 to recognize and categories images of cuneiform tablets according to their Hebrew letter content, thereby improving the precision of both detection and classification [14].

Dencker et al. (2023) address the challenge of weak supervision in the detection of cuneiform signs. The authors showcase enhanced sign detection accuracy through the application of weakly aligned transliterations and deep learning techniques, resulting in a superior precision for identifying cuneiform symbols within clay tablet images [15].

Bernier-Colborne et al. (2021) examined the automatic classification of cuneiform tablets through the application of convolutional neural networks (CNNs) for symbol recognition. The study illustrates the application of CNNs on a balanced dataset of cuneiform symbols, resulting in significant enhancements in symbol identification, thereby contributing to the preservation and comprehension of historical scripts [16].

Doost mohammadi et al. (2023) investigated the capabilities of hybrid deep learning models that integrate convolutional and recurrent neural networks (CNN-RNN) for the recognition of cuneiform symbols. Their efforts advanced the effectiveness of conventional symbol classification systems by deepening the contextual comprehension of symbols within sequence data, resulting in a significant boost in classification accuracy [17].

3. The Model Design and Implementation

The proposed model is composed of two stages. The initial phase involves data Pre-processing focuses on efficiently preparing the input data to enhance the performance of deep learning algorithms. The second stage entails constructing a convolutional neural network (CNN) model. Then, the models undergo training with the data that has been both trained and evaluated. Figure (1) illustrates the overall framework of the proposed system.

DOI: https://doi.org/10.54216/FPA.190209
Received: December 09, 2024 Revised: February 04, 2025 Accepted: March 01, 2025

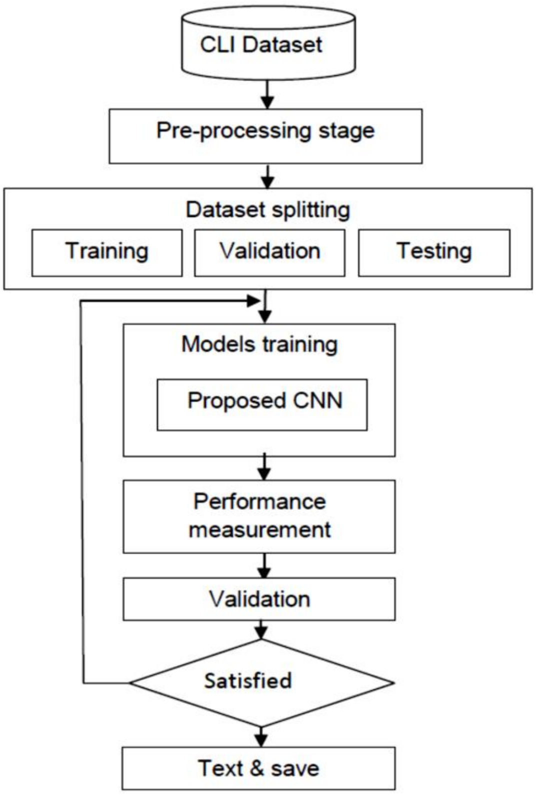


Figure1. Diagram of the proposed method.

A. Description of the CLI Dataset

The CLI [18] dataset comprises 134,000 portions of cuneiform texts associated with seven dialects of languages written in cuneiform, as shown in table 1. Provide a concise overview of the classifications for the various categories of cuneiform classes images. The figure illustrates that the classes SUX and NEA encompass the majority of the data relative to other classes, highlighting the imbalance presenting among classes in the dataset.

Table 1: The imbalance Classes in CLI Dataset

No.	Language	Class	Samples No.	Percentage
1	Sumerian	SUX	53,673	38.44%
2	Neo-Assyrian	NEA	32,966	23.68%
3	Late Babylonian	LTB	15,947	11.44%
4	Standard Babylonian	STB	17,817	12.77%
5	Neo-Babylonian	NEB	9,707	6.97%
6	Middle Babylonian Peripheral	MPB	5,508	3.96%
7	Old Babylonian	OLB	3,803	2.73%

B. The Pre-processing stage

The Pre-processing stage is essential for optimizing the performance of CNN models. Preparing and transforming raw data is necessary to efficient model training. The execution of models is impacted by the conditions of the images.

DOI: https://doi.org/10.54216/FPA.190209
Received: December 09, 2024 Revised: February 04, 2025 Accepted: March 01, 2025

B.1. Image Resize and enhancement

The dataset images underwent resizing to 224 x 224 to align with the specifications of the proposed models. The objective of resizing the images is to minimize the data size. The precision of feature extraction may be adversely affected when images differ in size; a smaller size results in reduced time for training, validating, and testing the CNN models [19]. The subsequent step in image preprocessing involves the application of a specific degree of sharpness. Sharpening improves the edges of an image and refines the details by elevating the contrast across all pixels. Enhancing image sharpness resulted in improved feature extraction, a crucial aspect in the process for CNN applications, as the quality of the input data can greatly influence the model's accuracy [20].Additionally, the dataset was divided into training, validation, and test sets to facilitate model evaluation.

B.2. Unigram enumeration

N-gram mining is a technique employed in the analysis of natural languages to locate and extract continuous strings of n elements from a specified text. This step serves in the categorization of cuneiform symbols [21]. A unigram was applied. After extracting the n-grams, they may be evaluated to yield valuable insights into the text. The final count for every n-gram may be used to determine the rate of its recurrence within the text.

B.3. CLI dataset classes Balancing

The CLI dataset exhibits an imbalance in class distribution; this phase executes the up sampling of minority classes within the CLI dataset. New samples are generated by randomly selecting samples from the minority classes with replacements. The original dataset has been supplemented with new samples to achieve a balanced distribution, resulting in 53,673 samples per class, as illustrated in Table 2.

Table 2: Classes after the Balancing

No.	Class	Number of Samples	Samples no. after the Balance	Percentage
1	LTB	15,947	53,673	29.7%
2	MPB	5,508	53,673	10.3%
3	NEA	32,966	53,673	61.5%
4	NEB	9,707	53,673	18.1%
5	OLB	3,803	53,673	7.1%
6	STB	17,817	53,673	33.2%
7	SUX	53,673	53,673	100%

C. CLI Dataset Splitting

The dataset is divided into three primary sections:

1. The training set constitutes the largest portion of the dataset, accounting for 80%, and serves the purpose of training the proposed models while adjusting the weights through the observation and learning of the correct output.
2. The validation set represents a portion of the dataset, specifically (10%). This component is utilized to assess the model through the adjustment of hyper parameters. This data exerts an indirect influence on the models, as it is observed by them but not employed for learning objectives.
3. The testing set (10%) serves as a distinct part of the dataset, employed to ensure an impartial and precise assessment of the models following the conclusion of the entire training process.

4. The proposed CNN model architectural

Our CNN architecture consists of nine convolutional layers, each accompanied by activation function and batch normalization with eight max pooling layers. We incorporate a Global Average Pooling layer and two dense layers. The first convolutional layer including 256 filters, each with a size of 3x3 and stride is 3×3 with the same padding. The ReLU is selected as an activation function, followed by batch normalization, serving to stabilize the learning

DOI: https://doi.org/10.54216/FPA.190209
Received: December 09, 2024 Revised: February 04, 2025 Accepted: March 01, 2025

process by normalizing the activations. A max pooling layer is applied using a (3x3) window and stride its (3×3) to reduce the spatial dimensions of the feature maps, to reduce the computational requirements in the network. The convolutional layers and filters have been increased to 256, 64, and 512 with the same filter. After each convolution, added batch normalization and max pooling with (3×3) filter and stride (3×3). The network was then further expanded to include four additional convolutional layers with filters (128,512,256 and 512) respectively, each followed by the batch normalization and (3x3) max pooling layer. The last convolutional layer with 128 filters of dimensions 3x3, followed by batch normalization without the inclusion of an extra pooling phase.

Every convolutional layer employs 'same' padding to preserve the spatial dimensions of the feature maps. Global average pooling is utilized to decrease each feature map to a single value, thereby effectively summarizing the essential information from the whole network prior to the classification phase. The classification stage of the network consists of two dense layer the first dense layer with 256 units. This layer incorporates L2 regularization on the kernel and L1 regularization on the activity and bias. Thekernel_regularizer applies an L2 regularization strength of 0.015, while the activity_regularizer and bias_regularizer apply an L1 regularization strength of 0.004, 0.01 respectively. Then, the ReLU activation is utilized to introduce non-linearity to the decision-making process. The second dense layer is the output layer consists of 39 units that utilize a softmax activation function. Softmax computes the probability score for each class and determines the final decision labels. Figure (2) provides the architectural framework of the CNN model.

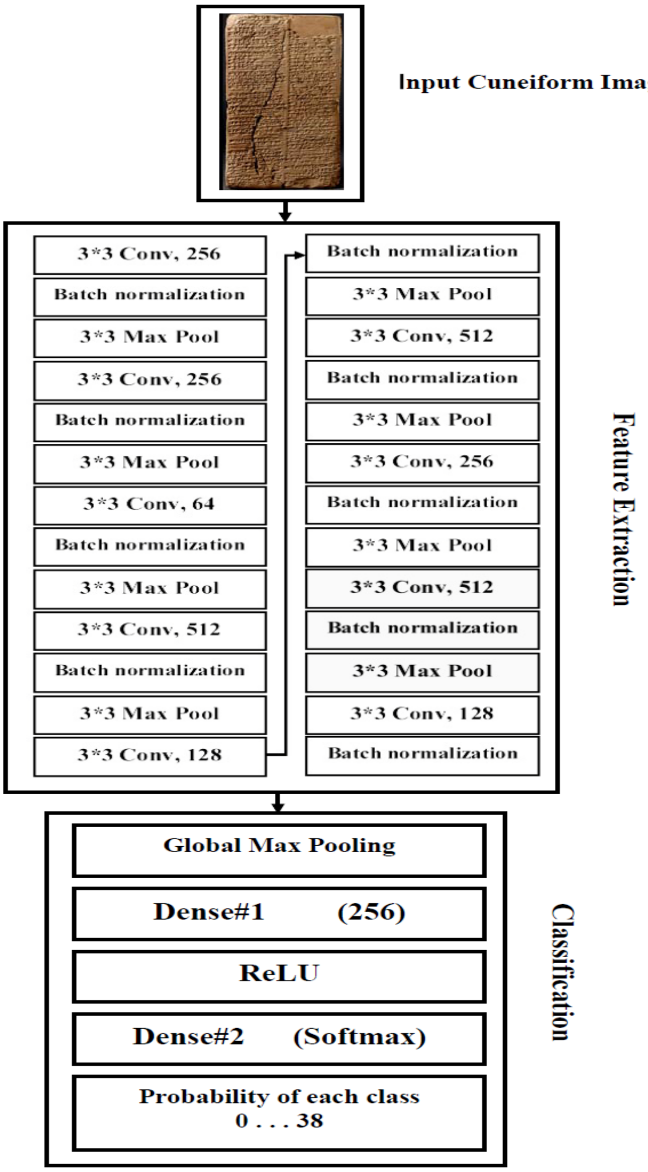


Figure 2. Architectural Framework of proposed CNN Model.

DOI: https://doi.org/10.54216/FPA.190209
Received: December 09, 2024 Revised: February 04, 2025 Accepted: March 01, 2025

A. Proposed model Training and evaluation

Training the proposed system involves several key steps. The process begins with feeding the training set into the models. The training set is divided into distinct sets known as batches. The training set is inserted into the network layers. The convolutional layer extracts feature from the images by applying filters, which detect patterns like edges and textures. These features are then reduced in complexity using pooling layers. The fully connected layers take the flattened output of the convolutional and pooling layers to perform the classification.

The CNN underwent training using the Adamax optimizer, the learning rate was set at 0.001, the Adam optimizer in the pre-trained models, and the learning rate is 0.001. This process iterates over multiple epochs, where the model continuously adjusts its parameters to improve accuracy. During system training, performance is evaluated using validation data to determine its suitability for the testing phase. This validation data is crucial for preventing over fitting and under fitting. Flexibly modifying the learning rate can enhance the model's performance by optimizing the adjustments made to the weights. This evaluation is carried out using several metrics, including accuracy, recall, precision, and the F1-score.

B. Hyperparameter Tuning:

The hyperparameters of the proposed model are shown in Table 3.

Table 3: Hyperparameters of the Proposed System.

Models	Hyperparameters
Presented CNN model	No. of convolution layer = 9 No. of pooling layer =8 No. of denes layer =2 Activation function ReLU Training epochs 60 Mini batch sizes 32

5. Experimental Results

The studies are conducted in an environment equipped with the following hardware: Central Processing Unit (CPU): Intel(R) Core(TM) i5- 11400H @ 2.70GHz 2.69 GHz, and a RAM capacity of 16.0 GB. The operating system is Windows 11, specifically the 64-bit version. The code was implemented in Python 3.8 programming language within the PyCharm environment. The library and programming environments utilized in the project included TensorFlow, Scikit-learn, Keras, Pandas, OpenCV2, Matplotlib, Pickle, and NumPy. the best result was obtained when using nine convolutional layers with filter sizes (256,256,64,512,128,512,256,512,128), respectively, each followed by batch normalization and max pooling, except for the last convolutional layer, which does not contain max pooling. Two dense layers were used, the first layer followed by L1 and L2 regularization techniques and the second layer for the output. The Adam optimizer was replaced with Adamax with a learning rate of 0.001.

The CNN model demonstrated outstanding accuracy, precision, recall, and minimal loss during training. The training and validation accuracies of the CNN model are 99.13% and 99.60%, respectively, with a training loss of 0.72% and 1.14 validation loss. Training and validation losses decreased consistently over the 60 epochs, with training loss decreasing from 5.86 to 0.72 and validation loss decreasing from 5.95 to 1.14. Additionally, the proposed model's precision and recall are 99.03% and 98.30%, respectively; Figures (4) and (5) show the accuracy during the training and loss function of the CNN model.

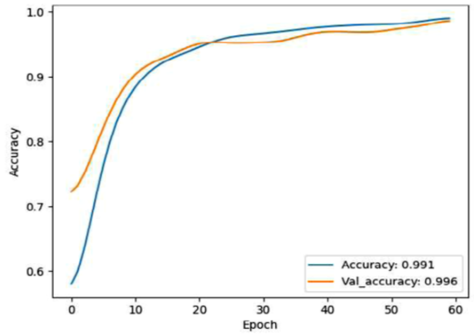


Figure 3. Accuracy of Training and Validation for CNN.

DOI: https://doi.org/10.54216/FPA.190209
Received: December 09, 2024 Revised: February 04, 2025 Accepted: March 01, 2025

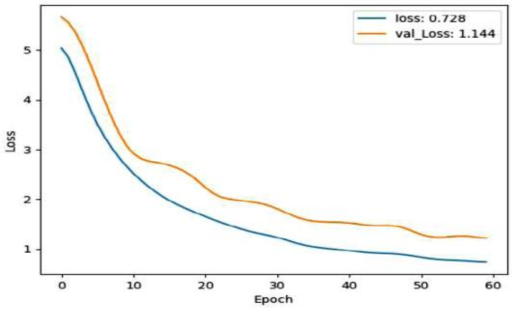


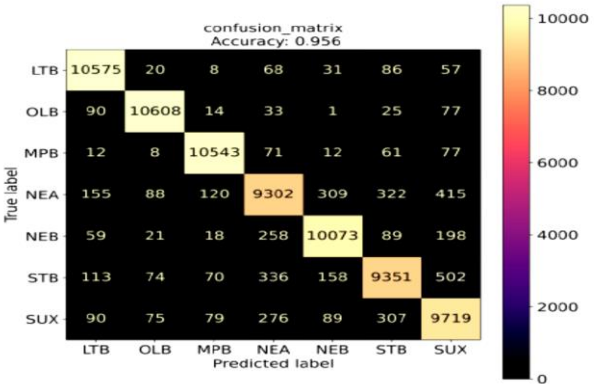
Figure 4. Loss Function for CNN Model.

A. Comparison with Related Work

The outcomes of the suggested methodology are juxtaposed with the pertinent literature indicated in Table 5. Our proposed technique achieved the maximum performance, with 99.55% accuracy, 99.03% precision, 98.30% recall, and 98.70% F1 score. The suggested strategy surpasses earlier research due to the critical significance of class balance in preventing classifiers from favoring majority classes. The unigram approach is superior at extracting characteristics from cuneiform writings as it decomposes the text into the smallest feasible units (symbols). These works illustrate the advancing complexity of deep learning techniques for the classification of cuneiform symbols, emphasizing the utilization of diverse algorithms including DNNs, CNNs, CNN-RNN hybrids, and VAEs to tackle specific issues in symbol identification and classification.

Table 5: Comparison of Proposed System with Previous Works.

Author(s)	year	Method	Accuracy	Precision	Recall	F1-score
Al-Khateeb	(2023)	(DNNs) with unigram features	93%	92%	90%	91%
Williams et al.	(2023)	RetinaNet object detection	89%	N/V	N/V	N/V
Hagelskjaer	(2022)	Deep Learning Classification	N/V	N/V	N/V	88%
Kapon et al.	(2024)	Variational Auto-Encoders (VAEs)	N/V	70%	56%	61%
Saeed et al.	(2024)	YOLOv8 object detection	96%	93.2%	89.8%	N/V
Dencker et al.	(2023)	Deep Learning	N/V	N/V	N/V	N/V
Bernier-Colborne et al.	(2021)	(CNNs)	83%	82%	81%	81%
Doostmohammadi and Nassajian	(2023)	CNN-RNN hybrid model	N/V	N/V	N/V	N/V
Proposed model	(2025)	CNN	99.5%	99.03%	98.30%	98.70%



	LTB	OLB	MPB	NEA	NEB	STB	SUX
LTB	10575	20	8	68	31	86	57
OLB	90	10608	14	33	1	25	77
MPB	12	8	10543	71	12	61	77
NEA	155	88	120	9302	309	322	415
NEB	59	21	18	258	10073	89	198
STB	113	74	70	336	158	9351	502
SUX	90	75	79	276	89	307	9719

Figure 5. Confusion matrix for the proposed Model with balancing dataset.

DOI: https://doi.org/10.54216/FPA.190209
Received: December 09, 2024 Revised: February 04, 2025 Accepted: March 01, 2025

6. Conclusions and future work

Cuneiform, is a writing system that originated in Sumer (southern Iraq) about 3200 BC. Cuneiform is among the earliest known writing systems in history. Cuneiform characters were utilized to engrave texts on clay tablets. The interpretation of cuneiform symbols is a challenging endeavor necessitating specialized knowledge. This paper develops an intelligent model capable of distinguishing cuneiform symbols from various civilisations. Experiments were performed on the CLI dataset to categories it into seven classifications. The methods exhibited optimal performance following the balancing approach, with the suggested DNN model achieving an accuracy of 99.55%. The use of unigram features markedly enhanced classifier performance and expedited their efficacy relative to prior research.

Our findings highlight the efficacy of deep learning methodologies, especially convolutional neural networks, in automating the examination of old scripts such as cuneiform. Archaeologists can optimize the decoding and interpretation of cuneiform inscriptions, therefore expediting discoveries and enhancing our comprehension of ancient civilizations. Future work will focus on Examine the efficacy of fine-tuning pre-trained CNN models for cuneiform classification tasks to utilize transfer learning and enhance model performance. Examine multi-modal methodologies that integrate picture classification with text analysis techniques to improve comprehension of cuneiform writings and extract semantic data.

Funding: “This research received no external funding”

Conflicts of Interest: “The authors declare no conflict of interest.”

References

[1] S. Mahmood et al., "Classifying cuneiform symbols using machine learning algorithms with unigram features on a balanced dataset," *J. Intell. Syst.*, vol. 32, no. 1, p. 20230087, 2023.

[2] L. Boadt, R. J. Clifford, and D. J. Harrington, *Reading the Old Testament: An Introduction*. New York, NY, USA: Paulist Press, 2012.

[3] H. Mara, S. Krömker, S. Jakob, and B. Breuckmann, "GigaMesh and Gilgamesh: 3D multiscale integral invariant cuneiform character extraction," in *Proc. 11th Int. Conf. Virtual Reality, Archaeol. Cultural Heritage*, 2010, pp. 131–138.

[4] N. A. Rasheed and W. L. Nados, "Recognition of cuneiform symbols using neural network," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 17, 2018.

[5] D. Charpin, *Writing, Law, and Kingship in Old Babylonian Mesopotamia*. Chicago, IL, USA: Univ. of Chicago Press, 2010.

[6] E. Uchida and R. Watanabe, "Blackening of the surfaces of Mesopotamian clay tablets due to manganese precipitation," *Archaeol. Discovery*, vol. 2, no. 4, pp. 107–115, 2014.

[7] C. Woods, "Visible language," *Spring*, vol. 45, no. 1–2, p. 155, 2011, doi: 10.1037/020683.

[8] T. Jauhiainen, H. Jauhiainen, T. Alstola, and K. Lindén, "Language and dialect identification of cuneiform texts," *arXiv preprint arXiv:1903.01891*, 2019.

[9] M. Zampieri et al., "A report on the third VarDial evaluation campaign," in *Workshop on NLP for Similar Languages, Varieties and Dialects*, The Association for Computational Linguistics, 2019, pp. 1–16.

[10] B. Al-Khateeb, "Classifying cuneiform symbols using machine learning algorithms with unigram features on a balanced dataset," *J. Intell. Syst.*, vol. 32, no. 1, p. 20230087, 2023, doi: 10.1515/jisys-2023-0087.

[11] E. C. Williams et al., "DeepScribe: Localization and classification of Elamite cuneiform signs via deep learning," *arXiv preprint arXiv:2306.01268*, 2023.

[12] F. Hagelskjær, "Deep learning classification of large-scale point clouds: A case study on cuneiform tablets," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2022, pp. 826–830.

[13] D. Kapon, M. Fire, and S. Gordin, "Shaping history: Advanced machine learning techniques for the analysis and dating of cuneiform tablets over three millennia," *arXiv preprint arXiv:2406.04039*, 2024.

DOI: https://doi.org/10.54216/FPA.190209
Received: December 09, 2024 Revised: February 04, 2025 Accepted: March 01, 2025

[14] E. A. Saeed, A. D. Jasim, and M. A. A. Malik, "Hebrew letters detection and cuneiform tablets classification by using the YOLOv8 computer vision model," *arXiv preprint arXiv:2407.06133*, 2024.

[15] T. Dencker, P. Klinkisch, S. M. Maul, and B. Ommer, "Deep learning of cuneiform sign detection with weak supervision using transliteration alignment," *PLoS One*, vol. 15, no. 12, p. e0243039, 2020.

[16] Bernier-Colborne et al., "Automatic classification of cuneiform tablets using convolutional neural networks," in *Conf. Proc.*, 2021.

[17] M. Doost Mohammadi and M. Nassajian, "Hybrid CNN-RNN deep learning models for cuneiform symbol recognition," *J. Historical Comput.*, 2023.

[18] S. Sharma et al., "A review of image preprocessing techniques for deep learning applications," in *Proc. Int. Conf. Mach. Learn. Data Eng. (ICMLDE)*, 2020.

[19] A. Zhang et al., "Image enhancement and feature extraction for convolutional neural networks in biomedical image analysis," *J. Vis. Commun. Image Represent.*, vol. 62, p. 101084, 2019, doi: 10.1016/j.jvcir.2019.101084.

[20] M. Ali, S. Shiaeles, G. Bendiab, and B. Ghita, "MALGRA: Machine learning and N-gram malware feature extraction and detection system," *Electronics*, vol. 9, no. 11, p. 1777, 2020.

[21] M. Zampieri et al., "A report on the third VarDial evaluation campaign," in *Workshop on NLP for Similar Languages, Varieties and Dialects*, The Association for Computational Linguistics, 2019, pp. 1–16.

DOI: https://doi.org/10.54216/FPA.190209
Received: December 09, 2024 Revised: February 04, 2025 Accepted: March 01, 2025