



Deep Learning in Archiving Indus Script and Motif Information

RESEARCH ARTICLE

ubiquity press

VAISHNAVI DIXIT

NUSHRAT HUSSAIN

SHUBHAM BASAK

DEVA ATTURU

DEBASIS MITRA

UJJWAL BHATTACHARYA

**Author affiliations can be found in the back matter of this article*

ABSTRACT

This work presents a novel computational system for the automated digitization of image-based data from seals of the ancient Indus Valley Civilization (IVC). The objective of this system's design is to automatically extract and archive key information from seals or images, including the script and motifs. The system operates as a pipeline comprising three deep learning models integrated with a custom-designed database. Two models form the Ancient Script Recognition network (ASR-net), which digitizes sequences of graphemes from Indus seals, similar to Optical Character Recognition for modern languages. The third model, the Motif Identification network (MI-net), identifies recurring motifs—distinctive symbols or iconographic elements with specific functional significance in the IVC. The database stores the extracted information, linking it to the respective seal images in a structured format. This end-to-end pipeline has been fully implemented, from image input to database archival. The overarching aim of this work is to support the application of automated statistical methods in the ongoing efforts to decipher the Indus script.

CORRESPONDING AUTHOR: Debasis Mitra

Florida Institute of Technology,
Melbourne, Florida,
United States
dmitra@fit.edu

KEYWORDS:

Indus Valley script; machine learning; deep learning; automated script recognition; automated motif identification; information archival

TO CITE THIS ARTICLE:

Dixit, V, Hussain, N, Basak, S, Atturu, D, Mitra, D and Bhattacharya, U. 2025. Deep Learning in Archiving Indus Script and Motif Information. *Journal of Computer Applications in Archaeology*, 8(1): 156–169. DOI: <https://doi.org/10.5334/jcaa.175>

I INTRODUCTION

The Indus Valley Civilization (IVC), also known as the Harappan Civilization, was one of the earliest urban societies, flourishing from around 2600 BCE to 1900 BCE in present-day Pakistan and northwest India. Renowned for its advanced urban planning, drainage systems, and standardized weights, the civilization remains enigmatic due to its indecipherable script. Unlike other ancient civilizations, such as Egyptian and Mesopotamian, the IVC lacks bilingual inscriptions like the Rosetta Stone, complicating the decipherment efforts (Kenoyer, 1998; Jun, 2022).

Over the past century, scholars have attempted to decode the Indus script through statistical analyses of grapheme sequences and contextual clues (Parpola, 1994; Mukhopadhyay, 2019). However, these methods are time-consuming and limited in scale. Recent advances in machine learning offer new opportunities for automating the analysis of ancient scripts, but require digital datasets.

Our project addresses this need by developing a machine-learning system for automated script transcription from Indus seals and motif identification. By training machine learning algorithms on annotated datasets, the system allows efficient analysis of large seal collections, aspiring to contribute to the study of the IVC's socio-cultural and economic aspects. This tool supports ongoing efforts by making the transcription and archival processes more efficient. It is extendable to other ancient scripts as well but our current project addresses only Indus seals.

The article is structured as follows: Section II reviews related works. Section III introduces the proposed computational architecture applicable to similar projects, followed by descriptions of our methodology in Section IV and system integration in Section V. Section VI discusses the challenges faced in the project, and Section VII outlines the scope for future work.

II RELATED WORK

In recent years, many researchers have applied statistical techniques to study the information content in Indus seals (Mukhopadhyay, 2023, 2019; Rao and Mohanty, 2015; Fuls, 2013; Yadav et al., 2010). Although Rao et al.'s work is not directly related to deep learning, it provides valuable insights for machine learning approaches (Rao and Mohanty, 2015; Yadav et al., 2010). By comparing the syntactic structure of the Indus script with other writing systems, they attempted to validate whether the script represented a language. This comparative analysis highlights structural characteristics that deep learning models can focus on when analyzing grapheme sequences in the Indus script.

Palaniappan and Adhikari (2017) proposed a deep learning pipeline for automating grapheme identification. Their pipeline segmented images into regions, classified them as textual or non-textual, refined the textual regions, isolated individual symbols, and classified one of the graphemes in a reference corpus as a proof of concept. With 92% accuracy in identifying the grapheme, this work demonstrated the potential of deep learning in expediting corpus creation and advancing research on the digitization of the Indus script. Our current work is motivated by this paper and extends this line of research significantly.

Oakes (2019) and Mukhopadhyay (Mukhopadhyay, 2019) applied rigorous statistical methods to the analysis of the Indus script in line with past works, e.g., by Parpola (1994). They used manually digitized data from Indus inscriptions to examine symbol frequencies, co-occurrences, and positional patterns. By identifying statistically significant patterns, they uncovered underlying rules governing grapheme organization within the script. These insights are valuable for developing deep learning models that can learn such statistical relationships and potentially contribute to deciphering the script.

Daggumati and Revesz (2021) tackled a crucial step in script decipherment by identifying allographs. Their method used Markov Chain language models to analyze the sequential patterns of symbols, aiming to detect symbols that frequently co-occur, possibly representing recurring sounds or grammatical elements. While not directly using deep learning, their approach could serve as a pre-processing step for deep learning models targeting the decipherment of the Indus script. These statistical approaches over large corpora emphasize the necessity of automated digitization tools for extracting and archiving information from Indus seals, which inspired our work.

Additionally, recent machine learning applications in archaeological image analysis provide relevant insights. Barucci et al. (2023) developed deep learning tools for segmenting Egyptian hieroglyphs in an archaeologically sensitive manner, contrasting with our use of YOLOv3 (detailed later). Sobhy et al. (2023) developed an AI-based translator that converts Egyptian hieroglyphs into English. Despite these advancements, to the best of our knowledge, no one has created a complete pipeline for the automated detection and archival of information from Indus seals as we have present here.

III SYSTEM COMPONENTS

First, we outline the conceptual landscapes and tools employed in our project, followed by their adaptation. The integrated workflow utilizing these tools is presented last.

A. CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (CNNs) form the foundation of our tool components. They have transformed computer vision by enabling classification, reconstruction, and even generation of visual data (Venkatesan and Li, 2017). CNNs capture spatial hierarchies of features in images through convolutional layers with filters that slide over input images, allowing for the progressive learning of complex representations. Our motif identification task is essentially performed by a specially designed CNN. Also, the following subsections detail three specific types of CNN utilized in our work.

B. YOLOv3

YOLOv3 (You Only Look Once version 3) is an advanced object detection model renowned for its efficiency and accuracy, employed to detect bounding boxes around each grapheme. Introduced by Joseph Redmon and Ali Farhadi in 2018 (Redmon and Farhadi, 2018a), YOLOv3 enhances object detection by dividing input images into a grid and predicting bounding boxes and class probabilities. It utilizes a feature pyramid network for multi-scale feature extraction, improving detection accuracy at various scales. In our project, YOLOv3 effectively creates bounding boxes around graphemes. The trained YOLOv3 model is used to detect bounding boxes around each grapheme in the image. It outputs two extreme coordinates (x, y) for the rectangle around each grapheme present in the seal. The coordinates of these bounding boxes are stored in a separate file

associated with the original image. This intermediate file is used in the next step.

Built on the Darknet convolutional neural network backbone (Redmon and Farhadi, 2018b), its key features include multi-scale prediction and anchor boxes, which improve object localization and detection accuracy. Known for its impressive performance in real-time object detection, YOLOv3 is widely used in fields like surveillance, autonomous driving, and robotics. In our project, we use YOLOv3 to create bounding boxes around each of the graphemes.

C. MOBILENET

MobileNet (Howard et al., 2017) is a state-of-the-art convolutional neural network (CNN) architecture designed to address the computational constraints of mobile and embedded devices while achieving high accuracy in image classification tasks. In our research, MobileNet was employed for the identification of individual graphemes. Developed by researchers at Google, MobileNet introduces the concept of depth-wise separable convolutions, which significantly reduce the computational complexity associated with traditional CNN architectures. This innovative approach decomposes standard convolution operations into two distinct layers: a depth-wise convolution and a point-wise convolution. By employing these layers sequentially, MobileNet achieves a substantial reduction in the number of parameters and computational demands, making it particularly advantageous for deployment on resource-constrained platforms (Figure 1).

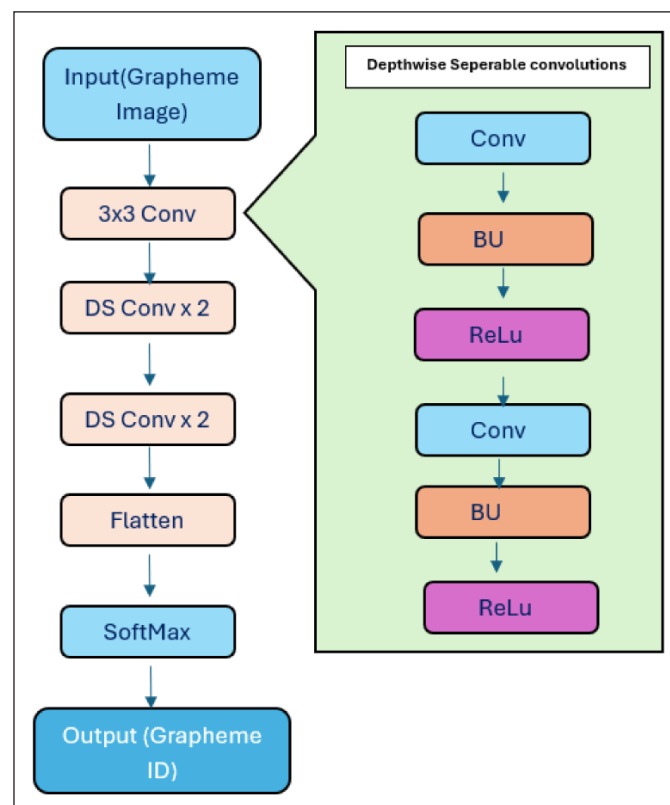


Figure 1 MobileNet architecture designed for detecting individual graphemes.

MobileNet's design, characterized by depth-wise separable convolutions, provides efficient inference capabilities and a low memory footprint without compromising performance. This architecture facilitates the democratization of deep learning applications on mobile devices, encompassing tasks such as image classification, object detection, and semantic segmentation. In the context of our study, MobileNet was utilized for grapheme identification within seal images, leveraging transfer learning techniques from established models such as ResNet and DenseNet to enhance character recognition accuracy within the designated bounding boxes. Our selection of MobileNet for grapheme identification was driven not only by its computational efficiency but also by the objective of implementing our system on mobile devices to support field archaeologists in their research endeavors.

D. ANCIENT SCRIPT RECOGNITION NETWORK (ASR-NET)

The objective of the Ancient Script Recognition Network (ASR-net) is to extract the Indus Valley Civilization (IVC) script present in seal images. It incorporates two key tools: first, it utilizes YOLOv3 to create bounding boxes around each grapheme on a seal image, serving as the initial step in organizing and preparing visual data for further analysis. ASR-net then decodes the sequence of graphemes represented by these bounding boxes using MobileNet, which identifies the corresponding grapheme for each bounding box. Subsequently, ASR-net analyzes the spatial arrangement of graphemes within the image, sorting them from top to bottom and by each row from right to left, leveraging the coordinates returned by YOLOv3 for accurate spatial arrangement. This sequential

processing mirrors the known reading pattern in Indus script, ensuring precise character recognition even in cases involving multiple lines of text on a seal.

E. MOTIF IDENTIFICATION NETWORK (MI-NET)

This component is crucial for recognizing recurring iconic patterns in seals that may hold significant archaeological relevance, contributing to a deeper understanding of the seals' cultural context. MI-net operates independently within the pipeline, utilizing a CNN to process the entire seal image and identify any present motifs. The MI-net model analyzes the whole original seal image to identify the motif present on the seal, in parallel to the above step of script transcription. Since a motif takes large part of the image, MI-net does not create any bounding box around the motif. Instead, it uses a conventional custom designed CNN trained over full seal images.

F. END-TO-END INTEGRATION

The workflow for analyzing ancient seals from IVC consists of three phases, as illustrated in Figure 2. In the **Training** phase, the system trains three models—YOLOv3, ASR-net, and MI-net—on pre-labeled seals. The YOLOv3 model detects bounding boxes around individual symbols (graphemes) within seal images from the FIT_ISI_Corpora, followed by MobileNet model identifying each individual grapheme in the bounding box. ASR-net comprises of these two models so that a sequence of graphemes in the script present in the seal image is the final output. In parallel to ASR-net, MI-net identifies motifs present in the seals. Each trained model undergoes validation independently with respective test data that includes known ground truth labels. In the **Inferencing** phase, these trained models are employed

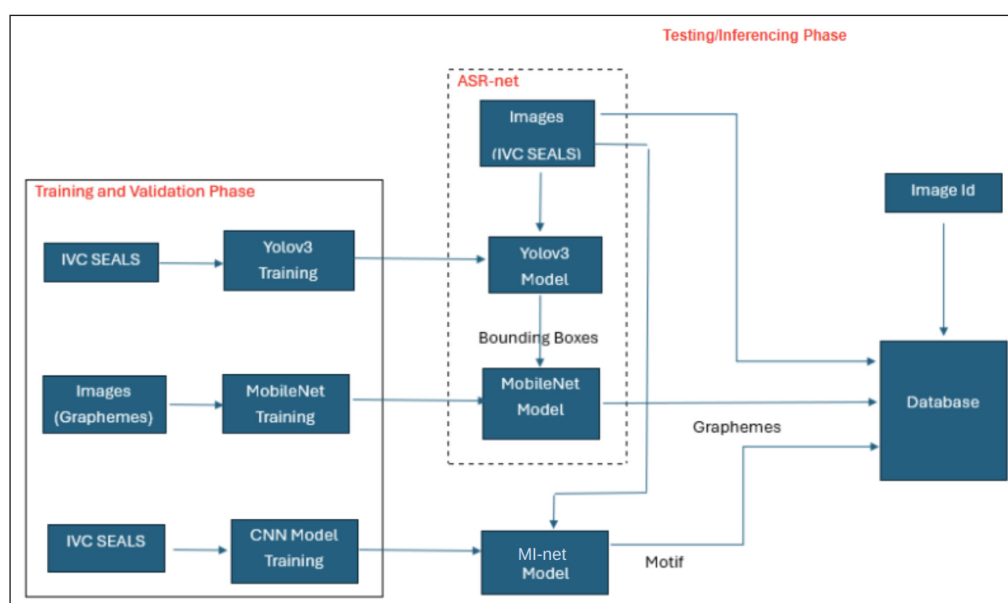


Figure 2 End-to-end workflow illustrating the project's training and inference phases. The training phase consists of two stages: ASR-net and MI-net. In contrast, the inference phase utilizes the trained models for transcribing seal image data and automatically archiving it in a database.

in an integrated and automated workflow to decode relevant information from any new unlabeled seal. When a new seal image is input, the system first generates boxes around the graphemes, then utilizes the trained models to decode the individual symbols (ASR-net) and any motifs present in the image (MI-net). Finally, in the **Archiving** phase, the workflow uses a relational database to store the extracted information for future reference and analysis.

IV METHODOLOGY

We initially collected 1004 images from *CISI volumes* (Joshi and Parpola, 1987) and Mahadevan's compendium (Mahadevan, 1977). After eliminating duplicates, 963 images remained, which we refer to as the FIT_ISI_Corpora, representing the respective institutions. We manually annotated these images with information on scripts and motifs for training and testing purposes.

A. BOUNDING BOX CREATION

A1. Data Collection

The process began with the collection of cropped images containing graphemes and motifs. In total, 232 such images were manually annotated for training and 13 for validation purposes in bounding box generation. We deliberately did not use the whole dataset due to the labor-intensive process required to create bounding boxes around each grapheme on hundreds of seals, and owing to the known robustness of YOLOv3 model.

A2. Annotation

Each image was meticulously annotated to mark the location of individual graphemes. This annotation process also involved labeling each grapheme with the respective identifier. The annotations were stored in XML files.

A3. Model Training

YOLOv3 was trained using the 232 annotated images to predict bounding boxes around each grapheme. A separate set of 13 images, not part of the training set, was used for testing and selected to represent a variety of scenarios. Following the MobileNet layer, the model included flattening and dense layers to complete the recognition process.

B. GRAPHEME IDENTIFICATION

B1. Dataset

We utilized a dataset comprising 40 classes or labels of graphemes, which we identified as the most frequently occurring in available Indus scripts: **M8, M12, M15, M17, M19, M28, M48, M51, M53, M59, M102, M104, M141, M162, M173, M174, M176, M204, M205, M211, M216, M245, M249, M267, M287, M294, M296, M302, M307, M326, M327, M328, M330, M336, M342, M387, M389, M391, and Other**. The prefix 'M' denotes Mahadevan, while the subsequent integer corresponds to the grapheme's assigned identifier in Mahadevan's compendium (Mahadevan, 1977). The "Other" category encompasses all graphemes not included in the labeled set for this pilot project. For training the grapheme identification model, we obtained 12,264 augmented images cropped from Indus seal images (exceeding 300 images per class). Augmentation involves applying artificial geometric transformations and adding random noise to expand the training dataset's size. This process also ensures dataset balance, equalizing the number of instances per class, as real-world data is often unbalanced, which can lead to bias in training.

B2. MobileNet Training and Validation

We explored several relevant deep learning architectures before selecting MobileNet. The MobileNet architecture, depicted in [Figure 3](#), was chosen for grapheme

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 64, 64, 3)	6
mobilenet_1.00_224 (Functional)	(None, None, None, 1024)	3228864
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 64)	262208
dense_1 (Dense)	(None, 40)	2600
Total params: 3493678 (13.33 MB)		
Trainable params: 3471790 (13.24 MB)		
Non-trainable params: 21888 (85.50 KB)		

Figure 3 Table showing the parameters of the MobileNet detailing the model architecture used for grapheme identification.

identification within each bounding box generated in the previous stage. Table II outlines the performance metrics for various model architectures assessed for the grapheme identification task. The first two entries correspond to conventional convolutional neural networks (CNNs) with a limited number of layers, representing shallow learning compared to deep learning. The next four entries include different variants of Residual Networks, followed by four configurations of Dense Networks or fully connected artificial neural networks (ANNs). Two additional entries represent variations of Inception networks, and the final entry showcases the performance of the MobileNet architecture previously discussed.

Among all the models evaluated, MobileNet demonstrated the highest validation accuracy of 94.5% and the best validation F1 score of 95%. Additionally, the confusion matrix presented in Figure 4 demonstrates the high accuracy attained during testing of the model after training.

After training MobileNet, we performed the final grapheme identification step. The model was trained

on 40 classes using approximately 12,264 images (with augmentations). To validate the robustness of the model, we performed a 5-fold cross-validation over the augmented training dataset five times. We conducted experiments with up to 30 epochs, where the validation accuracy stabilized. Figure 6 presents the training and validation accuracy/loss curves across these five folds. The validation accuracy and F1 score results across the five cross-validation folds were as follows (Table I).

The mean validation accuracy across folds was **85.28%** with a standard deviation of **±1.51%**, while the mean validation F1 score, which balances precision and recall, was **85.22%** with a standard deviation of **±1.53%**. These results demonstrate that the model's performance remains stable across different data partitions, reinforcing its reliability for grapheme identification, even though class balance was not maintained in an automated random selection of validation data during cross-validation.

The model was tested on a class-balanced 200 un-augmented test set that was reserved for the purpose. The result from this testing is shown in Figure 4 as a confusion matrix. This matrix shows clearly that most classes were identified by the model accurately.

C. MOTIF IDENTIFICATION

The MI-net model (Figure 5), which stands for Motif Identification Network, is a six-layer CNN specifically developed for identifying iconic motifs in seal images. It employs *categorical cross-entropy* as the loss function to minimize the difference between predicted probabilities and actual one-hot encoded labels for motif identification. The *Adam optimizer* is used to facilitate

FOLD	ACCURACY (%)	F1 SCORE (%)
Fold 1	84.92	84.85
Fold 2	85.76	85.70
Fold 3	86.01	85.94
Fold 4	84.51	84.42
Fold 5	85.19	85.19

Table I Cross-validation performance metrics.

MODEL ARCHITECTURE	TEST ACCURACY (%)	TEST F1 SCORE (%)
2D CNN (2 Layers)	83.94	81.95
2D CNN (3 Layers)	79.77	75.12
ResNet50 (He et al., 2016)	79.66	79.34
ResNet50V2 (He et al., 2016)	80.34	79.04
ResNet101 (He et al., 2016)	76.10	79.44
ResNet101V2 (He et al., 2016)	76.58	78.14
ResNet152 (He et al., 2016)	73.66	78.50
ResNet152V2 (He et al., 2016)	72.68	73.64
DenseNet121 (Huang et al., 2017)	80.49	82.37
DenseNet169 (Huang et al., 2017)	82.44	84.18
DenseNet201 (Huang et al., 2017)	80.95	82.72
EfficientNetB0 (Tan and Le, 2019)	80.98	82.03
InceptionV3 (Szegedy et al., 2016b)	72.65	74.19
InceptionResNetV2 (Szegedy et al., 2016a)	72.16	72.44
MobileNet (Howard et al., 2017)	94.50	95.00

Table II Comparison of Various Model Architectures Evaluated for Grapheme Identification.

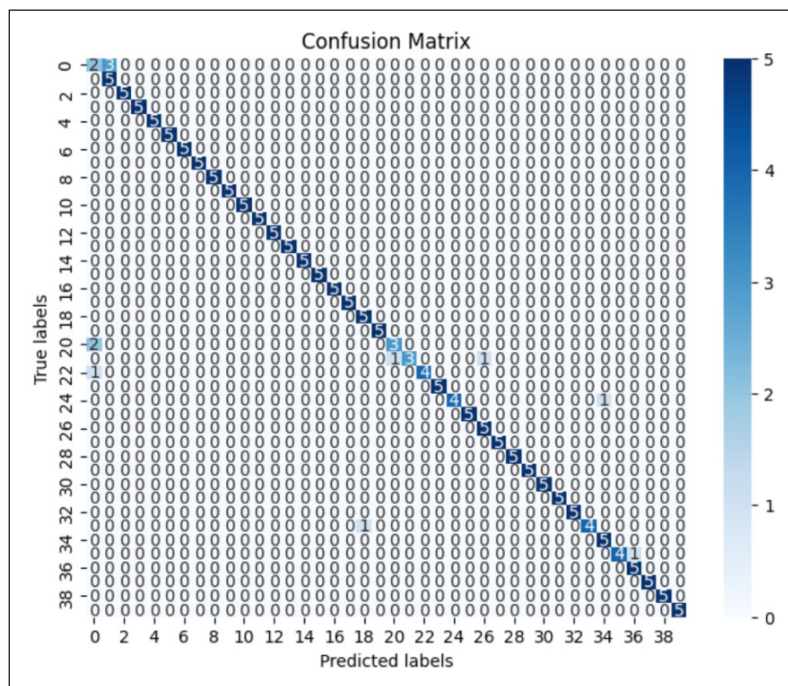


Figure 4 Confusion Matrix illustrating the actual and predicted labels for the MobileNet grapheme detection model during the validation stage.

```
CNNAttention(
  (conv1): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv2): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv3): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv4): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv5): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (fc1): Linear(in_features=25088, out_features=512, bias=True)
  (fc2): Linear(in_features=512, out_features=11, bias=True)
  (attention): Sequential(
    (0): Conv2d(512, 1, kernel_size=(1, 1), stride=(1, 1))
    (1): Sigmoid()
  )
)

Hyperparameters:
loss_function: categorical_crossentropy
optimizer: adam
metrics: ['custom_f1', 'accuracy']
```

Figure 5 MI-net layers using CNN in Pytorch for motif identification.

faster convergence, particularly with smaller datasets, due to its adaptive learning rate and momentum. Model performance is assessed using accuracy and a custom F1 score metric. The *ReLU* activation function is applied in the convolutional layers to enhance convergence speed, while the final softmax layer generates the probability distribution for the motif classes. The optimizer used is Adam, which has default hyperparameters: learning rate ($lr = 0.001$), β_1 (0.9), β_2 (0.999), and epsilon ($1e-7$) for adaptive learning and faster convergence.

C1. MI-net Training and Validation

Each image typically features a single motif, and the model is trained to calculate the probability of each motif in the input image using *one-hot encoding*, returning the motif with the highest probability. Figure 8 illustrates the confusion matrix of the model, showing its performance during validation.

The dataset consists of eleven motifs (labels): buffalo, bull, elephant, horned ram, man holding tigers, Pashupati, sharp horn with a long trunk, short-horned bull with its head lowered toward a trough, swastika, tiger-looking man on a tree, and unicorn. The data is combined and then split into k-folds, with each fold representing a separate validation subset. For fold 1, the training set consists of augmented images, while the validation set contains un-augmented images reserved for this fold. This process is repeated for each fold, allowing the model to be trained on different subsets of the data and validated on distinct portions, helping assess the model's generalization capabilities.

The mean accuracy over all 5 folds is 0.85, providing an overall indication of the model's performance across different validation sets. We conducted experiments with up to 100 epochs, as this is where validation performance stabilized. Figure 7 displays the training

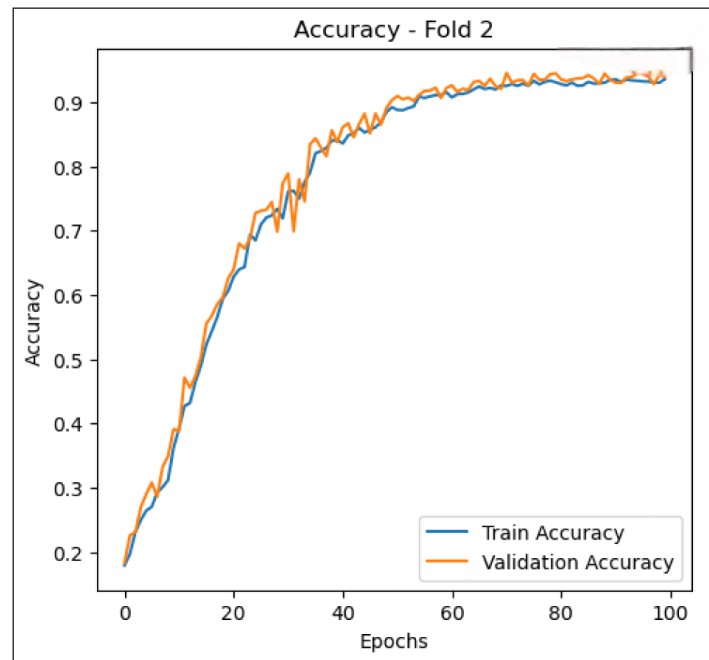


Figure 6 MobileNet learning curves of training and validation accuracies in grapheme identification versus training epoch numbers for one of the folds of the 5-fold cross validation based experiments. The model, trained on 40 classes using approximately 12,264 pre-augmented images, displays performance trends across epochs. The 5-fold cross-validation was repeated 5 times, yielding a mean accuracy of 85.28% ($\pm 1.51\%$) and a mean F1 score of 85.22% ($\pm 1.53\%$).

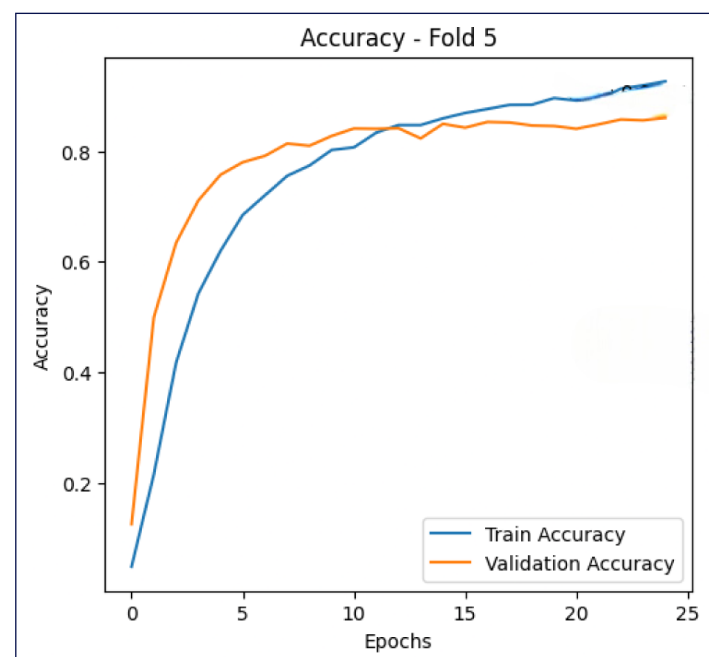


Figure 7 MI-net learning curves of training and validation accuracies in motif identification versus training epoch numbers for one of the folds of the 5-fold cross validation based experiments. The model achieved a cross-validation F1 score of 0.94 for this fold, while the mean F1 score across all five folds was (0.8034 ± 0.2038) .

accuracy achieved throughout the training epochs for fold 2, offering insight into the model's performance during the first folds of the cross-validation process.

D. ARCHIVING EXTRACTED INFORMATION IN A DATABASE

Extracted information from prior phases was archived in a SQL (Structured Query Language) database to enhance

data management and accessibility. The database, implemented in *MySQL*, features a schema designed to support the project results, comprising relational tables for image data, grapheme sequences, and motifs, each with pertinent columns such as ImageID, Image, Grapheme sequence, and Motif. Data insertion is facilitated through a Python pipeline that accepts Indus seal images as input. Source images are stored either as binary large objects

(BLOBs) or as file paths referencing external storage, while grapheme sequences and motifs are maintained as `varchar` data types. SQL queries enable precise data extraction and analysis. This structured repository provides a centralized framework for data access and supports collaborative research efforts.

V MODEL INTEGRATION FOR INFERENCE PIPELINE

The pipeline begins with an input image, as illustrated in Figure 9, which displays sample inputs of IVC seals. The image is automatically resized and reshaped to conform

to the specific format needed by the trained model, ensuring compatibility and optimal processing.

A. BOUNDING BOX CREATION, GRAPHEME ENCODING, AND MOTIF IDENTIFICATION

As mentioned before, the trained YOLOv3 model is used to detect bounding boxes around each grapheme in the image. The MobileNet model, trained to identify graphemes, uses the bounding boxes obtained from the previous step as input. It identifies each grapheme and organizes them into a sequence based on their geometric locations on the seal, as detected by the YOLOv3 model. This sequence is then saved in a separate file linked to the original seal image. (Sample Figure 10). The MI-net

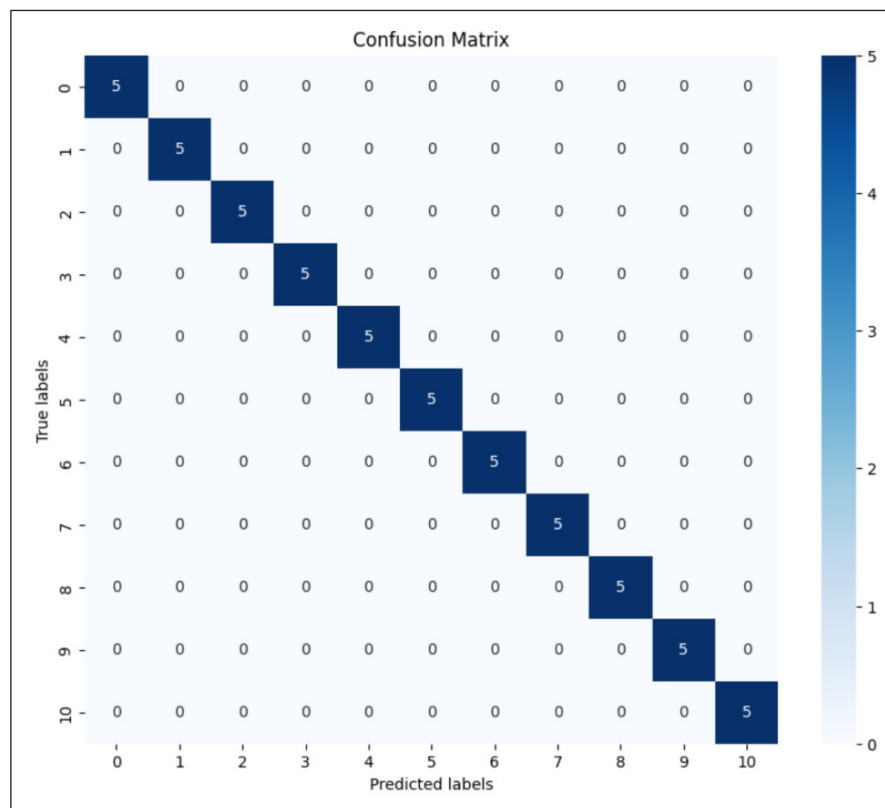


Figure 8 Confusion matrix for motif identification model between the true and predicted labels during validation from MI-net.

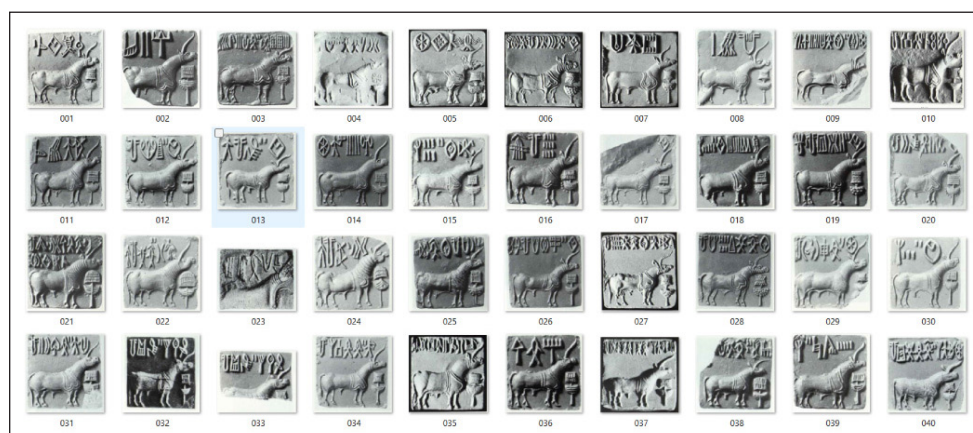


Figure 9 A few samples from the image dataset of IVC seals.

model analyzes the whole original seal image to identify the motif present on the seal, in parallel to the above step of script transcription.

B. DATABASE STORAGE AND SAMPLE QUERIES

As a final step, all the extracted information is stored in the database (Figure 11).

The database contains four attributes or columns, described as follows:

- Image ID: A unique identifier assigned to each image.
- Image: The original image itself.
- Grapheme sequence: The arrangement of grapheme encodings derived from step 3.
- Motif: Information pertaining to the identified motifs from step 4.



Figure 10 Bounding boxes indicating the locations of graphemes on the seal, following the processing of the seal image through the YOLOv3 model.

As we have indicated before, we used SQL for retrieving and manipulating information stored in the database. A few sample queries are described below:

- `SELECT COUNT (DISTINCT GraphemeSequence) AS unqiuesequences FROM details;`
This query finds the total number of unique grapheme sequences in the 'details' table.
- `SELECT ImageID FROM details WHERE SUBSTRING_INDEX (SUBSTRING_INDEX (GraphemeSequence, ',', 3), ',', -1) = 'M307';`
This query returns all the 'ImageID's where the third grapheme present in the sequence is 'M307'.
- `SELECT ImageID FROM details WHERE GraphemeSequence LIKE '%M17%';`
This query retrieves all the 'ImageID's where the grapheme sequence contains 'M17' as a prefix.

VI CHALLENGES

A. ADDRESSING CLASS IMBALANCE IN MOTIF CATALOGING

As mentioned before, some motifs are sparsely present in our FIT_ISI_Corpora. (Atturu, 2024) Please note that any corpora happen to be only a small subset of the seals produced by IVC over nearly a thousand years and over a large geographical region. Even if a motif is observed only once, it needs to be cataloged, and needs to be recognized when observed the next time on a newly found seal or sealing, possibly for deriving crucial information. While this may not be a problem for an experienced archaeologist, it is not feasible for most ML algorithms to learn a motif from only one sample. Extreme imbalance in sample sizes over multiple classes was our first challenge. In the future we intend to address this with a promising field in machine learning, namely,

ImageID	Image	GraphemeSequence	Motif
1	BLOB	M302,M51,M307,M173	unicorn
2	BLOB	M174,M328,M328,M173	unicorn
3	BLOB	M328,M245,M245,M48,M328,M17,M336,M328,M245	unicorn
4	BLOB	M141,M102,M59,M216,M104,M389	elephant
5	BLOB	M28,Other,M17,M162,M391,M391	unicorn
6	BLOB	M307,M294,M8,M294,M8,M287,M342,M17,M40	unicorn
7	BLOB	M174,M17,M336	unicorn
8	BLOB	M205,M330,M28,M328	unicorn
9	BLOB	Other,M17,M328,M389,Other,M336,M328,M245	unicorn
10	BLOB	M48,M249,M307,M389,M162,M328,M245,M249	unicorn
11	BLOB	M28,M19,Other,M173	unicorn
12	BLOB	M267,M328,M204,M102,M307,M267,M102,M336	unicorn
13	BLOB	M205,M328,M48,M342,M19	unicorn
14	BLOB	M216,M328,M328,M328,M19,M391	unicorn

Figure 11 Sample of a database table depicting automatically archived information by the pipeline.

the *zero-shot learning* where only a few (or even no) training data is needed. The technique is primarily based on transfer learning, i.e., learning from other domains, e.g., Sumerian scripts (Hu et al., 2023; Wang et al., 2022).

B. IMAGES OF BROKEN SEALS WITH PARTIALLY VISIBLE MOTIFS

Many seals are damaged from their lengthy burial periods or may have been discarded while still in use. However, broken seals can still hold significant value in archaeological studies. For instance, the context of a seal's usage and the motifs it displays can lead to the same conclusions, regardless of whether the seal is intact. Often, motifs on broken or damaged seals may be only partially visible, yet they can still be easily recognized by a human observer from just a small portion of the design (Figure 12). This raises the question: can our machine learning model achieve a level of accuracy comparable to that of a human in identifying motifs from limited visual information? In this study, we did not explore this issue and chose to exclude such images from our dataset. However, we plan to address this in future work by incorporating degraded seals into our training and validation sets, ensuring they are accurately labeled.

C. STYLISTIC VARIATIONS AND UNCERTAIN CLASS

Artisans from IVC have curved motifs in many different styles and variations, either for artistic reasons or for conveying some different meaning. For identification purpose, archaeologists group all such variability under one motif class or type. For example, the most frequently found Unicorn motif may have two to twelve thread patterns on their necks. Wide variations within a class is a challenge for deep learning algorithms, unless each variation is frequently present in the training set. Another problem is that a motif may look like a different one, even

to the human eyes. For example, a horned-zebra may look like a unicorn. While most such cases could be discerned easily by an expert with only a closer examination, it is not clear how one can train an ML algorithm to make such discrimination over different motifs that look very similar (e.g., Figure 13). This challenge can only be solved by carefully curating and labeling the training data according to a broad consensus within the relevant archaeology community.

To a certain degree, all the challenges outlined here apply equally to the tasks of grapheme or text identification. Additionally, these issues will arise if the research is expanded beyond IVC seals to include the automated digitization of other ancient scripts and symbols.

VII CONCLUSION & FUTURE SCOPE

The integration of the ASR-net and MI-net within our pipeline signifies a significant advancement in the analysis of ancient scripts, particularly the enigmatic Indus script. The ASR-net demonstrates a high success rate in identifying individual symbols, fulfilling the critical need for automated digitization techniques for ancient Indus seals, similar to Optical Character Recognition systems used for modern languages. Additionally, this research introduces the Motif Identification problem, highlighting recurring motifs on Indus seals that are believed to serve specific functions within particular periods of the civilization. Despite the challenges of applying deep learning to motif identification, the creation of an open-source dataset of annotated seals represents a crucial step in bridging field archaeology with theoretical research on IVC. Through the integration of advanced technologies and interdisciplinary collaboration, this research aims to contribute to the ongoing efforts

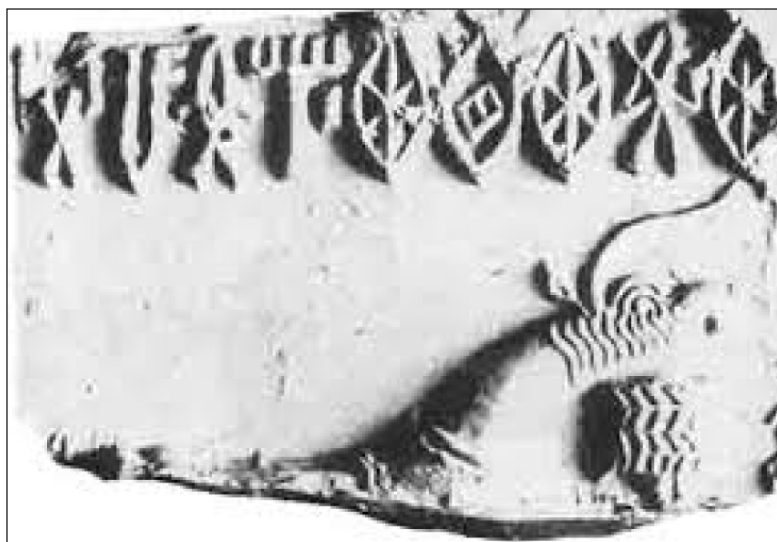


Figure 12 Broken seal sample.



Figure 13 Stylistic variations in the dataset.

to decipher the ancient mysteries embedded in IVC artifacts, as noted in (Wells, 2006).

In our immediate future work, we plan to: (1) expand the FIT_ISI_Corpora with additional seal images, such as those available at (Wells and Fuls, 2024); (2) label the data in accordance with community consensus; and (3) enhance our machine learning capabilities, for instance, by utilizing YOLOv5 (Jocher et al., 2022) instead of the current YOLOv3 implementation. These initiatives will broaden our collaboration (Fuls, 2015) and address challenges outlined in the previous sections. Looking ahead, our database could be enriched with relevant archaeological information, such as seal thickness, quality, and metadata regarding their discovery locations. Alternatively, we could collaborate with existing online databases (Wells and Fuls, 2024).

Moreover, expanding our scope beyond the IVC to encompass other ancient civilizations, such as those from Mesopotamia, Egypt, and Mesoamerica, offers opportunities to explore diverse writing systems and cultural practices across different geographical regions and time periods. Furthermore, pioneering techniques for the automatic description of images could enhance the efficiency and accuracy of analyzing visual artifacts, leveraging advancements in machine learning and computer vision algorithms. These interdisciplinary efforts present exciting possibilities for illuminating the richness and complexity of ancient civilizations and contributing to a broader scholarly understanding of human cultural heritage in the future.

The authors have no conflict of interest to declare. Dixit, Hussain, Basak, and Atturu did the experiments under supervision of Mitra and Bhattacharyya. Atturu performed initial set of experiments, developed the database and the end-to-end workflow under Mitra's guidance.

DATA ACCESSIBILITY STATEMENT

github.com/DM-BiCLab/Deep-Learning-in-Archiving-Indus-Script-and-Motif-Information.git.

ACKNOWLEDGEMENTS

We would like to extend our gratitude to Professor Jonathan Mark Kenoyer and Greg Jamison for many discussions. Andreas Fuls and unknown reviewers have provided many valuable feedback on the draft that led to its much improvement. Furthermore, we acknowledge harappa.com for resources that enriched our project.

FUNDING INFORMATION

We appreciate a support from the National Endowment for the Humanities under grant PR-290075-23.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Vaishnavi Dixit orcid.org/0009-0005-7487-0927

Florida Institute of Technology, Melbourne, Florida, United States

Nushrat Hussain orcid.org/0000-0002-5138-5495

Indian Statistical Institute, Kolkata, India

Shubham Basak orcid.org/0000-0003-4243-7983

Indian Statistical Institute, Kolkata, India

Deva Atturu

Florida Institute of Technology, Melbourne, Florida, United States

Debasis Mitra orcid.org/0000-0002-4351-1252

Florida Institute of Technology, Melbourne, Florida, United States

Ujjwal Bhattacharya orcid.org/0000-0002-8546-6453

Indian Statistical Institute, Kolkata, India

REFERENCES

- Atturu, DMR.** 2024. *Deep Learning in Indus Valley Script Digitization*. Master's thesis. Florida Institute of Technology. Available at: <https://repository.fit.edu/etd/1416>.

- Barucci, A, Amendola, M, Argenti, F, Canfailla, C, Cucci, C, Guidi, T, Python, L and Franci, M.** 2023. Discovering Ancient Egyptian Hieroglyphs with Deep Learning. Italian National Council (CNR). Available at: <https://www.ifac.cnr.it/wp-content/BOOKS/BOOK/HORUS/testoHORUS.pdf>.
- Daggumati, S and Revesz, PZ.** 2021. 'A method of identifying allographs in undeciphered scripts and its application to the indus valley script'. *Humanities and Social Sciences Communications*, 8: 1–14. DOI: <https://doi.org/10.1057/s41599-021-00713-0>
- Fuls, A.** 2013. 'Positional analysis of indus signs'. *Epigrafika*, 7: 253–275.
- Fuls, A.** 2015. 'Appendix I: Automated Segmentation of Indus Texts'. Archaeopress Publishing Ltd. pp. 100–118. DOI: <https://doi.org/10.2307/j.ctvr43jmf.15>
- He, K, Zhang, X, Ren, S and Sun, J.** 2016. 'Deep residual learning for image recognition'. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. DOI: <https://doi.org/10.1109/CVPR.2016.90>
- Howard, AG, Zhu, M, Chen, B, Kalenichenko, D, Wang, W, Weyand, T, Andreetto, M and Adam, H.** 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR abs/1704.04861. Available at: <http://arxiv.org/abs/1704.04861>, arXiv:1704.04861.
- Hu, W, Zhan, HJ, Liu, C, Yin, B and Lu, Y.** 2023. Ots: A one-shot learning approach for text spotting in historical manuscripts. arXiv preprint arXiv:2304.00746. DOI: <https://doi.org/10.2139/ssrn.4419850>
- Huang, G, Liu, Z, Van Der Maaten, L and Weinberger, KQ.** 2017. 'Densely connected convolutional networks'. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269. DOI: <https://doi.org/10.1109/CVPR.2017.243>
- Jocher, G, Chaurasia, A, Stoken, A, Borovec, J, NanoCode012, Kwon, Y, Michael, K, Xie, T, Fang, J, Imyhyx, Lorna, Yifu, Z, Wong, C, Abhiram V, Montes, D, Wang, Z, Fati, C, Nadar, J, Laughing, UnglvKitDe, Sonck, V, Tkianai, YxNONG, Skalski, P, Hogan, A, Nair, D, Strobel, M and Jain, M.** 2022. ultralytics/yolov5: v7.0 – yolov5 sota realtime instance segmentation. Available at: <https://zenodo.org/record/3908559>.
- Joshi, JP and Parpola, A.** 1987. *Corpus of Indus Seals and Inscriptions*, Volumes I and II. Memoirs of the Archaeological Survey of India; no. 86, 96, 116; nide 359, 383, 386; tom. 239–240, 359, 383, 386. Suomalainen Tiedeakatemia, Helsinki.
- Jun.** 2022. 'Historical review of mohenjo-daro and harappan civilization in pakistan'. *Pacific International Journal*, 5: 31–42. DOI: <https://doi.org/10.55014/pjj.v5i2.185>
- Kenoyer, J.** 1998. *Ancient cities of indus valley civilization*. Karachi, Islamabad: Oxford University Press.
- Mahadevan, I.** 1977. The indus script: Text, concordance and tables. Issue 77 of Memoirs of the Archaeological Survey of India.
- Mukhopadhyay, A.** 2023. 'Semantic scope of indus inscriptions comprising taxation, trade and craft licensing, commodity control and access control: archaeological and script-internal evidence'. *Humanities and Social Sciences Communications*, 10: 1–12. DOI: <https://doi.org/10.1057/s41599-023-02320-7>
- Mukhopadhyay, BA.** 2019. 'Interrogating indus inscriptions to unravel their mechanisms of meaning conveyance'. *Palgrave Communications*, 5: 73. DOI: <https://doi.org/10.1057/s41599-019-0274-1>
- Oakes, MP.** 2019. 'Statistical analysis of the tables in mahadevan's concordance of the indus valley script'. *Journal of Quantitative Linguistics*, 26: 401–422. DOI: <https://doi.org/10.1080/09296174.2017.1406294>
- Palaniappan, S and Adhikari, R.** 2017. Deep learning indus script. PLOS Submission: arXiv preprint arXiv:1702.00523. DOI: <https://doi.org/10.48550/arXiv.1702.00523>
- Parpola, A.** 1994. *Deciphering the Indus Script*. Cambridge University Press.
- Rao, VN and Mohanty, MK.** 2015. 'Comparative visual analysis of symbolic and illegible indus valley script with other languages'. *IOSR Journal of Humanities and Social Science (IOSR-JHSS)*, 20: 66–72.
- Redmon, J and Farhadi, A.** 2018a. YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.
- Redmon, J and Farhadi, A.** 2018b. Yolov3: An incremental improvement. Available at: <https://arxiv.org/abs/1804.02767>, arXiv:1804.02767.
- Sobhy, A, Helmy, M, Khalil, M, Elmasry, S, Boules, Y and Negied, N.** 2023. 'An ai based automatic translator for ancient hieroglyphic language—from scanned images to english text'. *IEEE Access*, 11: 38796–38804. DOI: <https://doi.org/10.1109/ACCESS.2023.3267981>
- Szegedy, C, Ioffe, S and Vanhoucke, V.** 2016a. 'Inception-v4, inception-resnet and the impact of residual connections on learning'. In: *CoRR*. Available at: <http://arxiv.org/abs/1602.07261>, arXiv:1602.07261.
- Szegedy, C, Vanhoucke, V, Vinyals, O and Wojna, Z.** 2016b. 'Rethinking the inception architecture for computer vision'. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826. DOI: <https://doi.org/10.1109/CVPR.2016.308>
- Tan, M and Le, QV.** 2019. 'Efficientnet: Rethinking model scaling for convolutional neural networks'. In: *2019 International Conference on Machine Learning (ICML)*, pp. 6105–6114. Available at: <http://proceedings.mlr.press/v97/tan19a.html>.
- Venkatesan, R and Li, B.** 2017. *Convolutional Neural Networks in Visual Computing: A Concise Guide*. CRC Press. DOI: <https://doi.org/10.4324/9781315154282>
- Wang, W, Duan, L, En, Q, Zhang, B and Liang, F.** 2022. 'Tpsn: Transformer-based multi-prototype search network for few-shot semantic segmentation'. *Computers and Electrical Engineering*, 103: 108326. DOI: <https://doi.org/10.1016/j.compeleceng.2022.108326>

Wells, B. 2006. Epigraphic approaches to indus writing.
Dissertation, Harvard University.

Wells, B and **Fuls, A.** 2024. Online indus writing database.
www.indus.epigraphica.de.

Yadav, N, Joglekar, H, Rao, RPN, Vahia, MN, Adhikari, R and
Mahadevan, I. 2010. 'Statistical analysis of the indus
script using n-grams'. *PLoS ONE*, 5: e9506. DOI: <https://doi.org/10.1371/journal.pone.0009506>

TO CITE THIS ARTICLE:

Dixit, V, Hussain, N, Basak, S, Atturu, D, Mitra, D and Bhattacharya, U. 2025. Deep Learning in Archiving Indus Script and Motif Information. *Journal of Computer Applications in Archaeology*, 8(1): 156–169. DOI: <https://doi.org/10.5334/jcaa.175>

Submitted: 29 July 2024 **Accepted:** 08 April 2025 **Published:** 19 May 2025

COPYRIGHT:

© 2025 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

Journal of Computer Applications in Archaeology is a peer-reviewed open access journal published by Ubiquity Press.