# Visual Similarity Methods Applied to Ancient Scripts.

Hasnat Aslam  Ahmadreza Pourghodrat  Manoj Oladri  Ashish Masih

January 28, 2025

## 1 Introduction:

Visual similarity analysis plays a crucial role in the study and interpretation of ancient scripts, offering valuable insights into the historical development and cultural context of diverse writing systems. In this research effort, we explore the application of visual similarity methods to three distinct ancient scripts: Old Turkic, Old Hungarian, and Carian. By employing computational techniques and algorithms, we aim to uncover patterns, relationships, and unique characteristics within these scripts that can deepen our understanding of their linguistic and cultural significance.

In this project we analyzed three different ancient scripts, to determine if they have any visual similarities in them. The analyzed scripts are Old Hungarian, Old Turkic, and Carian. Old Hungarian script was our reference script for comparison with the rest of the scripts.

This research aims to help addressing whether we can classify our target scripts in order to find if they belong to particular script families. On another paper [2], we've seen these methods applied to some other languages with great results. We tried to extend that work as well.

## 2 Background:

Classifying different ancient scripts is beneficial for several reasons:

- **Historical understanding:** By classifying scripts, historians can trace the development and evolution of writing systems over time. This helps in understanding the cultural and linguistic contexts of ancient civilizations.
- **Preservation and decipherment:** Classifying scripts aids in the preservation of ancient texts and facilitates efforts to decipher unreadable or partially understood scripts. It enables scholars to identify similarities and differences between scripts, leading to breakthroughs in translation and interpretation.
- **Cultural identification:** Identifying and classifying scripts can provide insights into the identities and interactions of ancient societies. It helps in distinguishing scripts used by different cultures or regions, shedding light on historical contacts and influences.

- **Educational and Research Purposes:** Classifying scripts contributes to educational resources and research materials. It supports scholars, students, and enthusiasts in studying and documenting the diversity of human writing systems throughout history.

- **Digital Archiving and Analysis:** In the age of digital technology, classifying ancient scripts is crucial for digital archiving and analysis. It facilitates the development of tools for automatic script recognition, text digitization, and computational analysis of ancient texts.

These days using tools computers provide to help in research works has increased dramatically. We also used computers to analyze visual similarities for a few important reason:

- **Objective Insights:** It helps in reducing subjective biases that may arise from human interpretation. By relying on computational algorithms, visual similarity analysis can provide more objective insights and identify patterns based on quantifiable criteria.

- **Efficient Comparison:** It enables researchers to efficiently compare visual patterns and features across a large dataset of images or documents. This can be particularly useful in fields like art history, archaeology, and linguistics for identifying similarities and differences between artifacts or texts.

- **Pattern Recognition:** By leveraging algorithms and computational techniques, visual similarity analysis can recognize intricate patterns and structures that may not be easily discernible to the human eye. This aids in discovering hidden relationships or classifications within visual data.

## 2.1 Research on the current topic:

There are a few research papers that we reviewed for our project:

1. **Multiple scripts analysis** [2]: This paper analyzed eight different scripts Brahmi, Cretan Hieroglyphs, Greek, Indus Valley, Linear B, Phoenician, Proto Elamite and Sumerian pictographs. They used an SVM (support vector machine) machine learning model for analyzing all the mentioned scripts. In their study, they compared the shapes of each alphabet in one language to another and created a correlation matrix between each letter of any two compared scripts. And then selected the characters that had high correlation scores >= 75% in each comparison. Finally, they tabulated the match between each language permutation. This research found that some of the languages had very high correlations. E.g. Greek - Phoenician had the highest score with 22 out of 22 matching characters from Phoenician script.

2. **SVM and CNN classifier for handwriting recognition** [3]: This paper analyzed the handwritten characters using the integrated SVM and CNN classifiers and found that applying the dropout hyperparameter helped in increasing the efficiency as compared to other

approaches mentioned in the literature. They achieved an error rate as low as 14.7% in the best-case scenario.

### 2.2 ML approaches:

Various machine-learning approaches have been used by other scholars and some of them pertinent to our study are mentioned here:-

### 2.3 Feature extraction:

The image of the ancient character is divided into sub-sections and various characteristics are analyzed e.g. stroke direction, stroke length, curvature, presence of specific components, etc. Once that information is available in the machine learning model, it compares the count of and score of each cont with other characters to find a match. These features are extracted manually or by using contour analysis.

### 2.4 Classical ML models:

SVM (support vector machine), random forest, KNN (k-nearest Neighbours), etc. these models can learn about the extracted characters by extracting features.

### 2.5 Convolution Neural Networks (CNN):

These models generate a pyramid-like structure of the image (ancient language character) at various zoom levels and use it for finding matches with other characters.

### 2.6 Approaches before ML became popular:

In the days before the advent of machine learning and neural network tools linguistics would analyze the structure, form, and context of ancient languages and organize those characters into classes based on shared features such as stroke patterns, component radicals, etc..

Lexicographers would compile the dictionaries and character databases from their experience and the knowledge gained over the years on the topic.

## 3 Data Collection:

For analysis we considered three scripts and downloaded data in the form of PNG images, descriptions about each script are given below:-

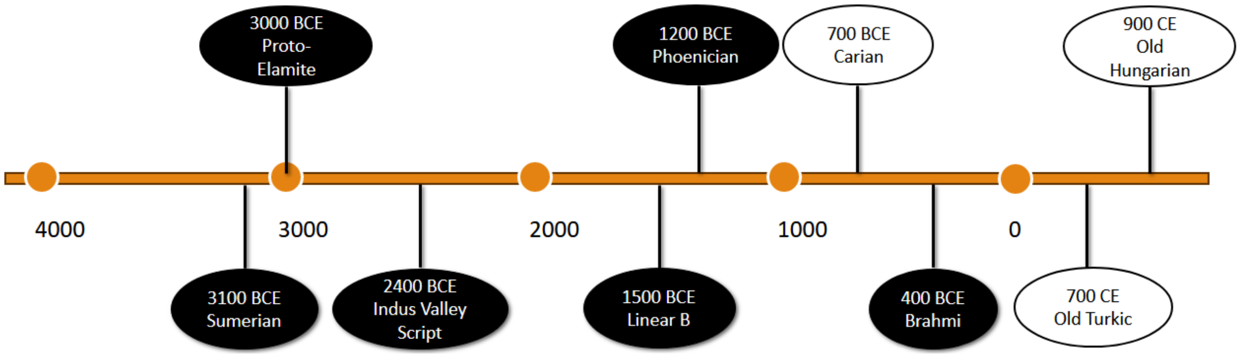



Figure 1: Historical Overview

### 3.1 Old Turkic Script:

The Old Turkic script consists of about 200 inscriptions and several manuscripts. These inscriptions date from the 7th century to the 10th century. This script was used by the Göktürks, who were a nomadic Turkic people. This was derived from the Aramaic script and was influenced by the Sogdian script [6]. Its alphabets are composed of consonants and vowels. Additionally, it consists of 38 to 40 characters including letters and diacritics.

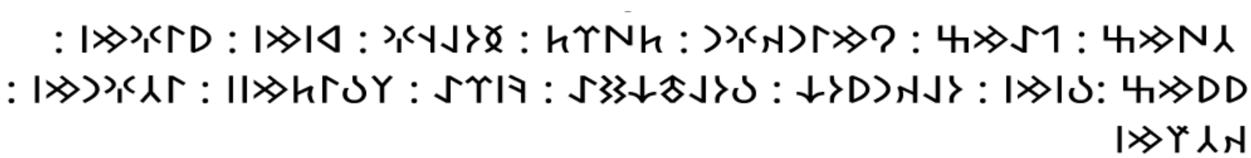


Figure 2: Old Turkic

### 3.2 Old Hungarian Script:

The Old Hungarian script, also known as the Hungarian Runic script or Székely-Hungarian Rovás, is an alphabetic writing system used historically for recording the Hungarian language. The origin of this ancient script has not been revealed yet, the usage might have appeared in the 6th – 7th centuries [4]. This script consists of 42 runic letters and was usually written by Rovás on wooden sticks or rocks.

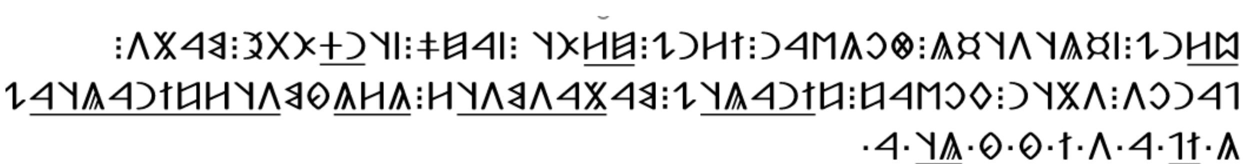


Figure 3: Old Hungarian

### 3.3 Carian Script:

The Carian script was an alphabetic writing system used to write the Carian language, an Anatolian language spoken in ancient Caria (modern-day southwestern Turkey) [5]. While the Carian language is not fully deciphered, some scholars have proposed connections between Carian and the Luwian languages based on onomastic evidence and loanwords in Greek inscriptions [1]. Carian script is estimated to have 28 to 35 characters.

## 4 Design and Research Questions

### 4.1 Design:

Deep learning models require large number of training and testing data to learn different model parameters, to achieve model generalization in this study different data augmentation methods are applied to enhance the degree of freedom for model training and increase number of images.

### 4.2 Research question

#### 4.2.1 RQ1:

Design deep neural network model to predict character using symbol of languages.

#### 4.2.2 RQ2:

Train model on one language and test on other language to analyze the relationship between different languages.

### 4.3 Data Augmentation

#### 4.3.1 Image resizing:

The symbol characters stored in the form of images, different characters have different number of pixels, width and height. Therefore, to transform in to common grid all images are resized into 64*64 images.

#### 4.3.2 Rotation

Different number of rotations are applied

- 0 to 45
- 0 to 90
- 0 to 180

- 0 to 270
- 0 to 360

For each type of rotations random integer values are generated between every range.

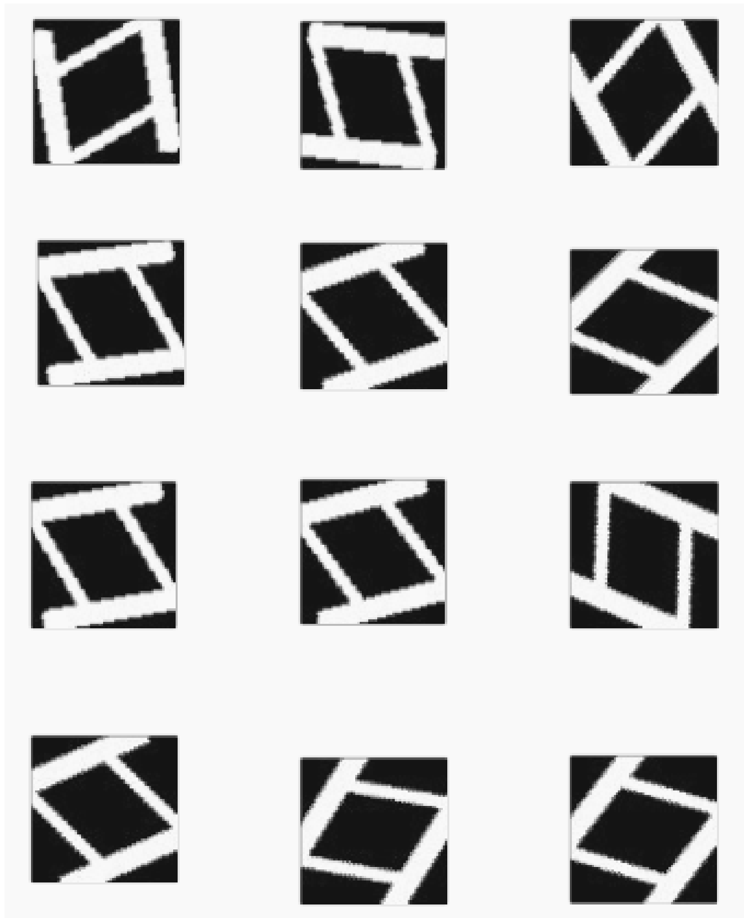


Figure 4: Sample character

### 4.4 Data Sampling

The augmented data is divided into three classes using random sampling method for each language.

- Training 60%
- Testing 20%
- Validation 20%

### 4.5 Model Training and Testing

Multiple models are designed with different configurations and different hyper-parameters are used to train the models. For different models different accuracy matrices and optimization methods are used.

- Loss (categoricalhinge, categorical crossentropy)
- Matrix (Accuracy, categorical crossentropy)
- Optimizer (adam, ftrl, adadelta)
- No of Epochs (10-200)
- Batch size (8-128)
- Hidden layers (3-8)
- Dense Layers (1-4)

# 5 Experimental Results

Different models have different training and testing accuracy for different language and different model configurations.

## 5.1 Model Training

```
Epoch 1/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 4s 23ms/step - accuracy: 0.3811 - loss: 1.9794 - val_accuracy: 0.6875 - val_loss: 0.8845
Epoch 2/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 0s 386us/step - accuracy: 0.8750 - loss: 0.2375 - val_accuracy: 0.7500 - val_loss: 0.9217
Epoch 3/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 2s 21ms/step - accuracy: 0.8903 - loss: 0.3608 - val_accuracy: 0.8413 - val_loss: 0.4440
Epoch 4/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 0s 300us/step - accuracy: 1.0000 - loss: 0.0758 - val_accuracy: 0.5000 - val_loss: 0.8608
Epoch 5/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 2s 21ms/step - accuracy: 0.9406 - loss: 0.1465 - val_accuracy: 0.8510 - val_loss: 0.4172
Epoch 6/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 0s 297us/step - accuracy: 1.0000 - loss: 0.0217 - val_accuracy: 1.0000 - val_loss: 0.0142
Epoch 7/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 2s 21ms/step - accuracy: 0.9345 - loss: 0.1719 - val_accuracy: 0.9471 - val_loss: 0.1288
Epoch 8/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 0s 302us/step - accuracy: 1.0000 - loss: 7.3765e-04 - val_accuracy: 1.0000 - val_loss: 0.0018
Epoch 9/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 2s 21ms/step - accuracy: 0.9776 - loss: 0.0711 - val_accuracy: 0.9327 - val_loss: 0.3248
Epoch 10/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 0s 284us/step - accuracy: 1.0000 - loss: 0.0049 - val_accuracy: 1.0000 - val_loss: 0.1042
Epoch 11/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 2s 21ms/step - accuracy: 0.9954 - loss: 0.0401 - val_accuracy: 0.9519 - val_loss: 0.1422
Epoch 12/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 0s 325us/step - accuracy: 1.0000 - loss: 0.0013 - val_accuracy: 1.0000 - val_loss: 0.0065
Epoch 13/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 2s 21ms/step - accuracy: 0.9919 - loss: 0.0373 - val_accuracy: 0.9423 - val_loss: 0.0992
Epoch 14/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 0s 311us/step - accuracy: 0.8750 - loss: 0.3557 - val_accuracy: 1.0000 - val_loss: 0.0382
Epoch 15/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 2s 21ms/step - accuracy: 0.9947 - loss: 0.0185 - val_accuracy: 1.0000 - val_loss: 0.0087
Epoch 16/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 0s 320us/step - accuracy: 1.0000 - loss: 3.5299e-04 - val_accuracy: 1.0000 - val_loss: 3.7218
e-04
Epoch 17/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 2s 21ms/step - accuracy: 0.9933 - loss: 0.0302 - val_accuracy: 0.8990 - val_loss: 0.3596
Epoch 18/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 0s 324us/step - accuracy: 1.0000 - loss: 0.0397 - val_accuracy: 1.0000 - val_loss: 0.0024
Epoch 19/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 2s 21ms/step - accuracy: 0.9780 - loss: 0.0606 - val_accuracy: 0.9519 - val_loss: 0.1643
Epoch 20/20
110/110 ━━━━━━━━━━━━━━━━━━━━ 0s 303us/step - accuracy: 0.8750 - loss: 0.1334 - val_accuracy: 1.0000 - val_loss: 0.0311
```

Figure 5: Model training

## 5.2 Samples

### 5.2.1 Old Hungarian

```
Found 1553 images belonging to 18 classes.
Found 126 images belonging to 18 classes.
Found 126 images belonging to 18 classes.
```

Figure 6: Sample Distribution

### 5.2.2 Old Turkic

```
Found 323 images belonging to 17 classes.
Found 34 images belonging to 17 classes.
Found 34 images belonging to 18 classes.
```

Figure 7: Sample Distribution

### 5.2.3 Carian

```
Found 3792 images belonging to 44 classes.
Found 308 images belonging to 44 classes.
Found 308 images belonging to 44 classes.
```

Figure 8: Sample Distribution

## 5.3 Accuracy and Confusion Matrix (Old Hungarian)

Deep neural network has around 98% testing and approximately 100% training accuracy.

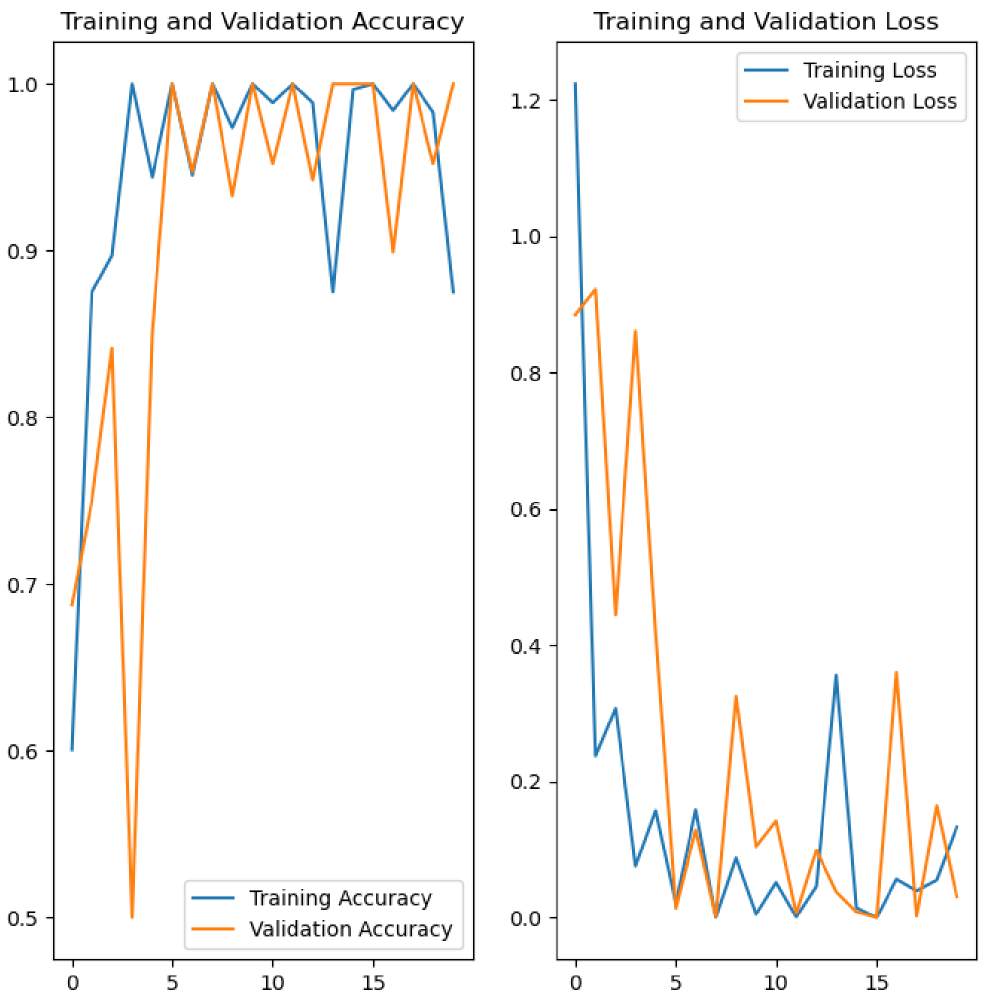



Figure 9: Model Accuracy Plot

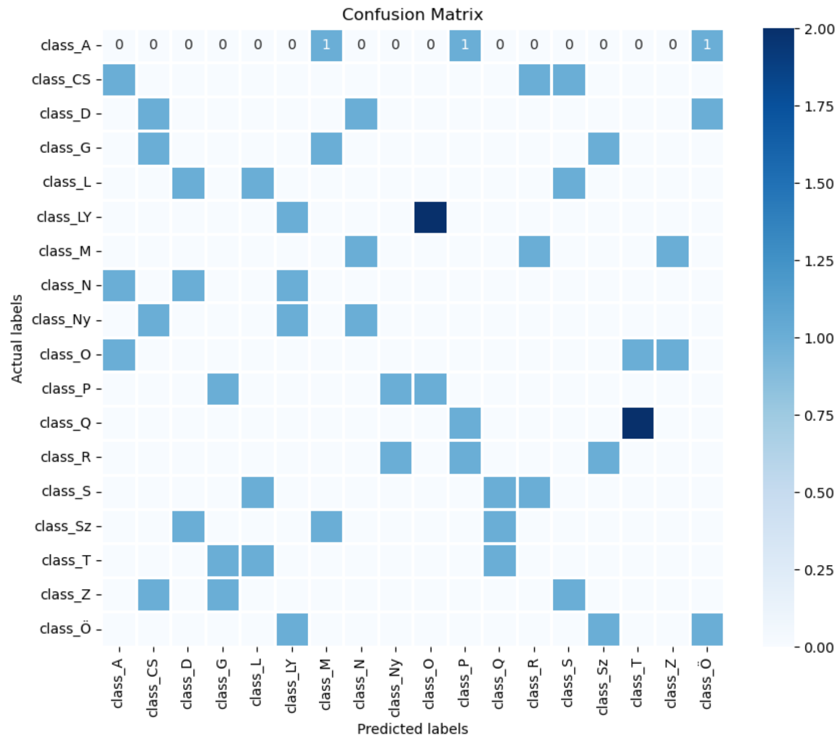



Figure 10: Confusion Matrix for model 1

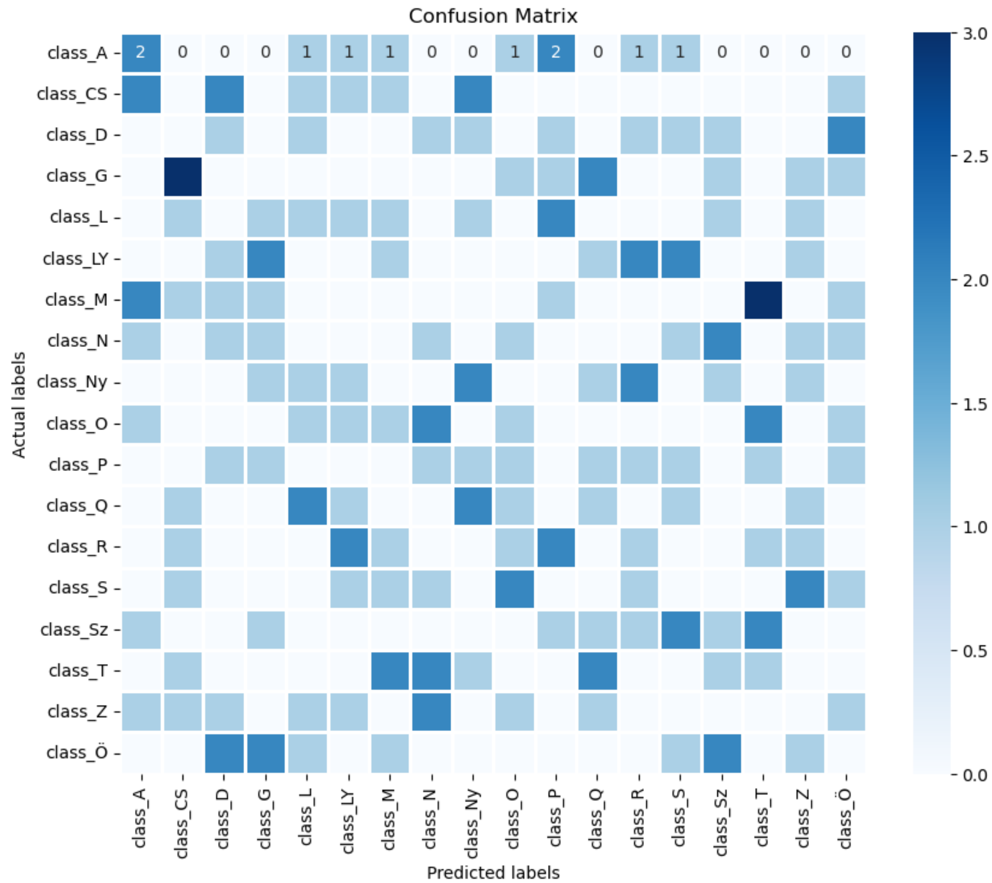



Figure 11: Confusion Matrix for model 2

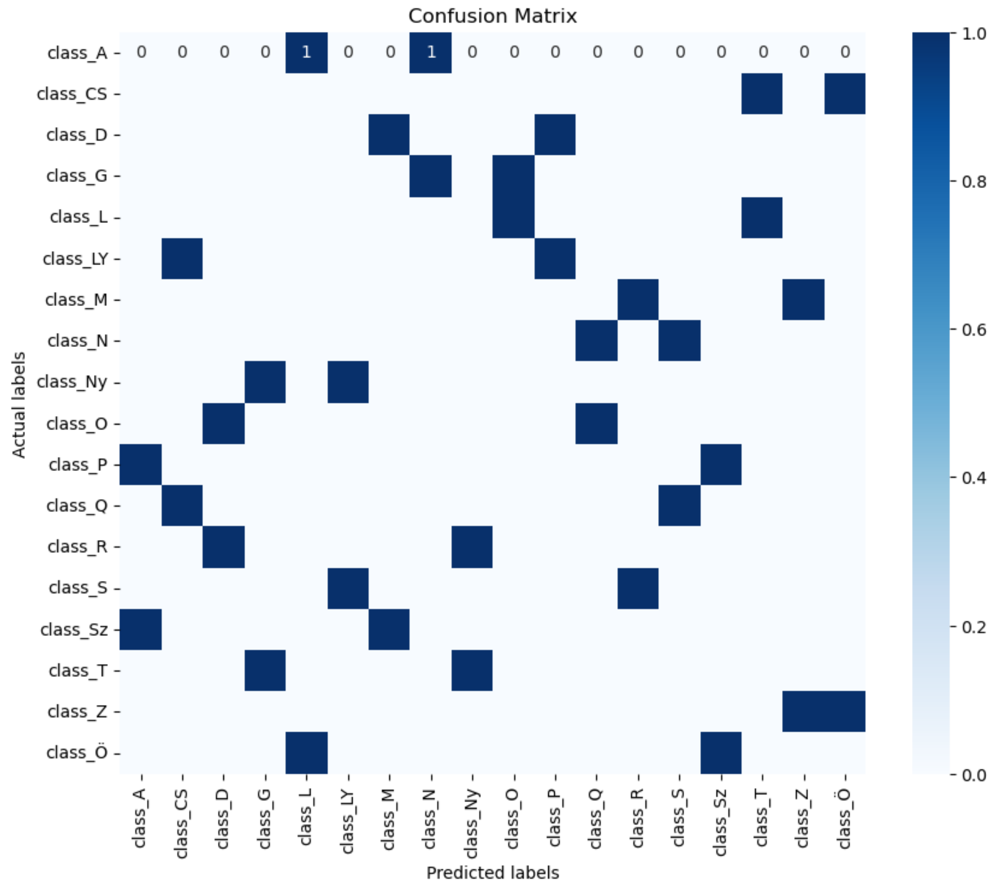



Figure 12: Confusion Matrix for model 3

### 5.3.1 MNIST Model

The state of the art MNIST model that is used for handwritten number classification also trained on the old Hungarian symbols, the accuracy of MNIST model is comparatively very low as compare to other models. The confusion matrix shows that model has biases with respect to specific characters.

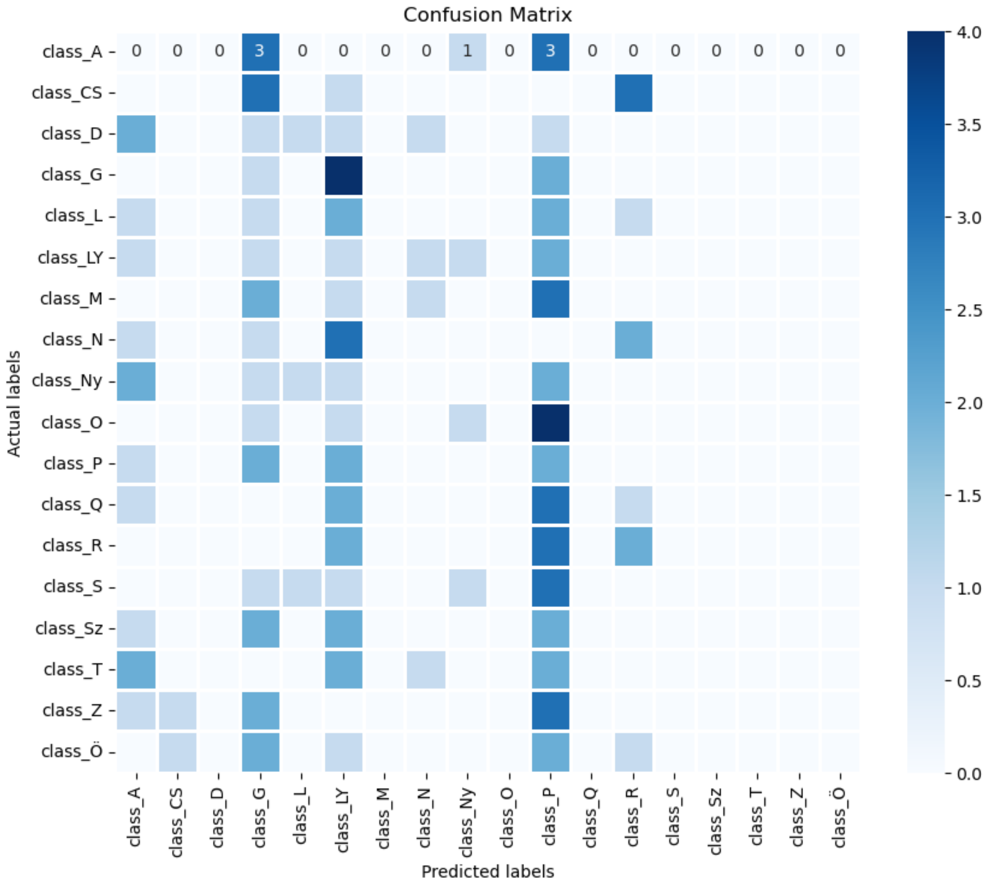
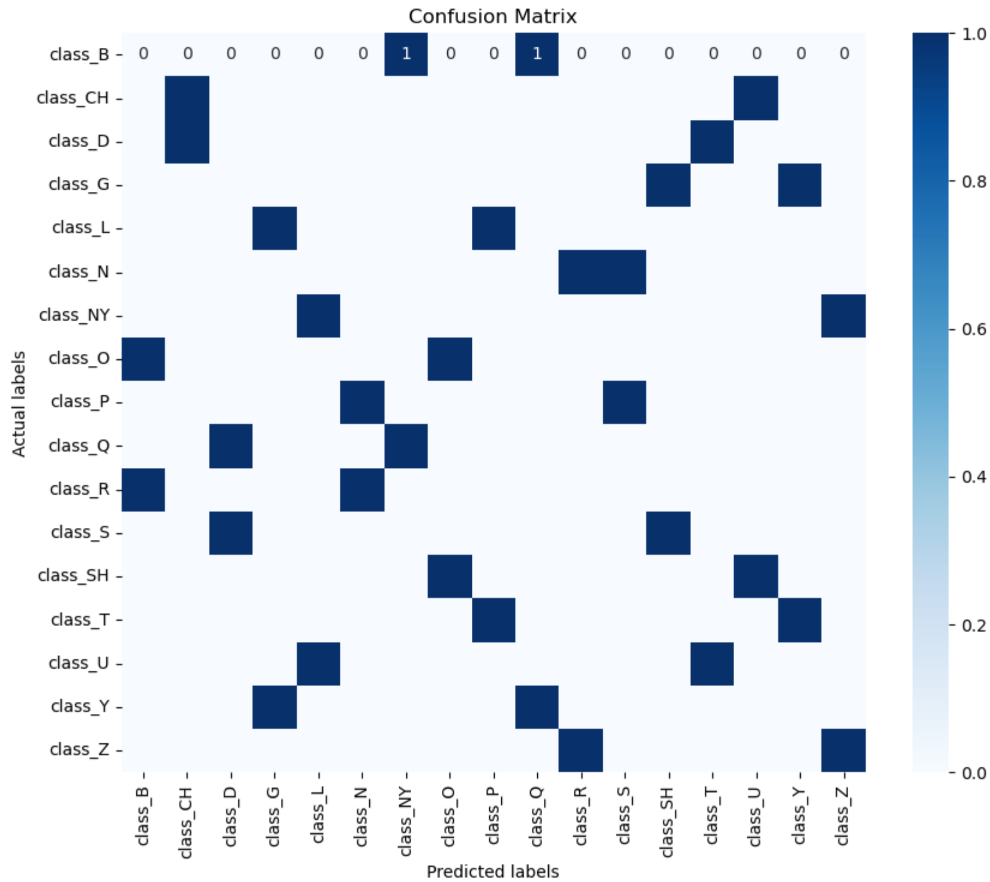



Figure 13: Confusion Matrix for MNIST model

## 5.4 Accuracy and Confusion Matrix (Old Turkic)

The models also showed very high accuracy for old Turkic language characters.


Confusion Matrix
Actual labels
Predicted labels
class_B
class_CH
class_D
class_G
class_L
class_N
class_NY
class_O
class_P
class_Q
class_R
class_S
class_SH
class_T
class_U
class_Y
class_Z
0
1
1.0
0.8
0.6
0.4
0.2
0.0

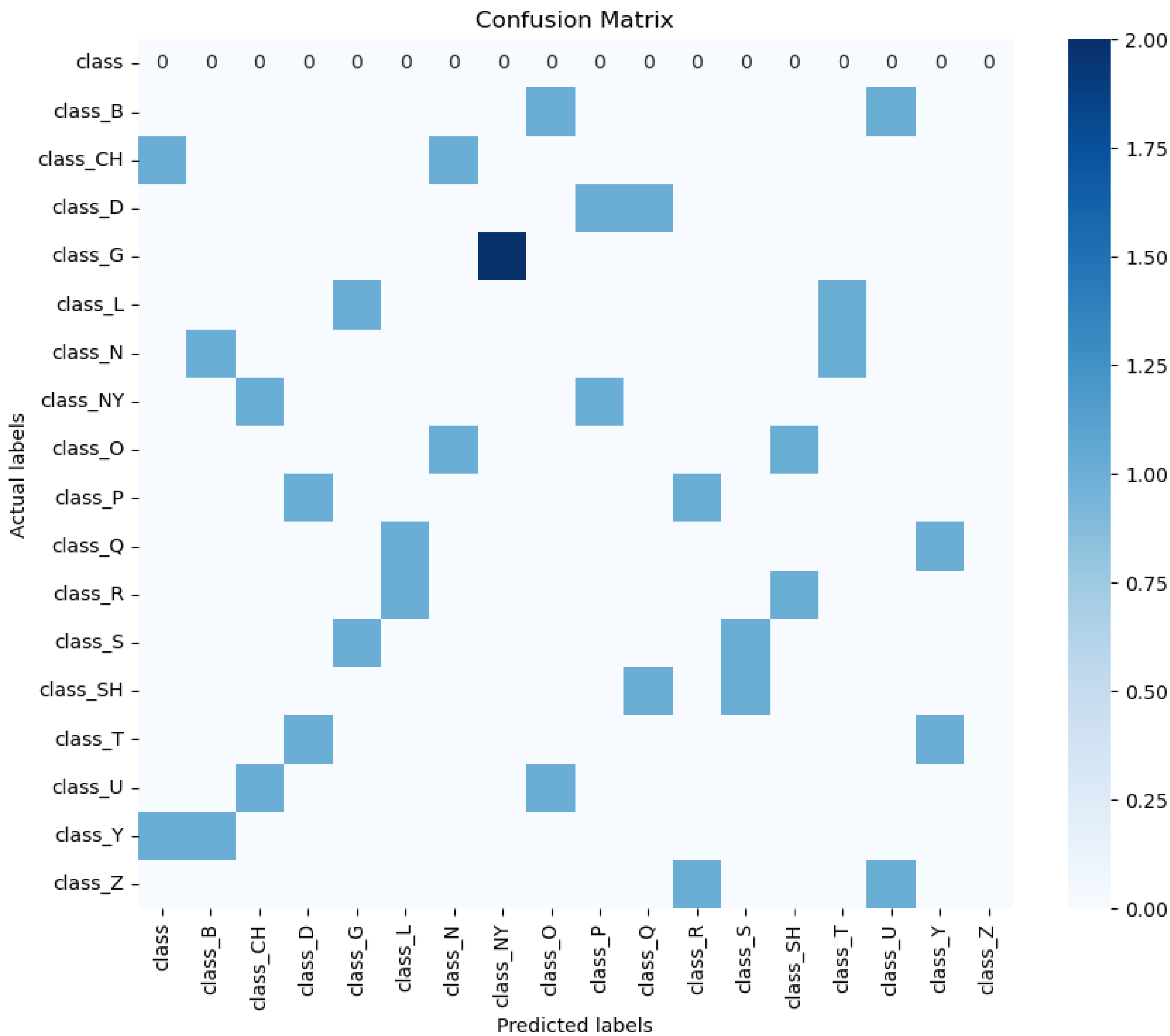



Figure 14: Confusion Matrix model accuracy

## 5.5 Accuracy and Confusion Matrix (Old Turkic)

The models also showed similar accuracy for Carian language characters

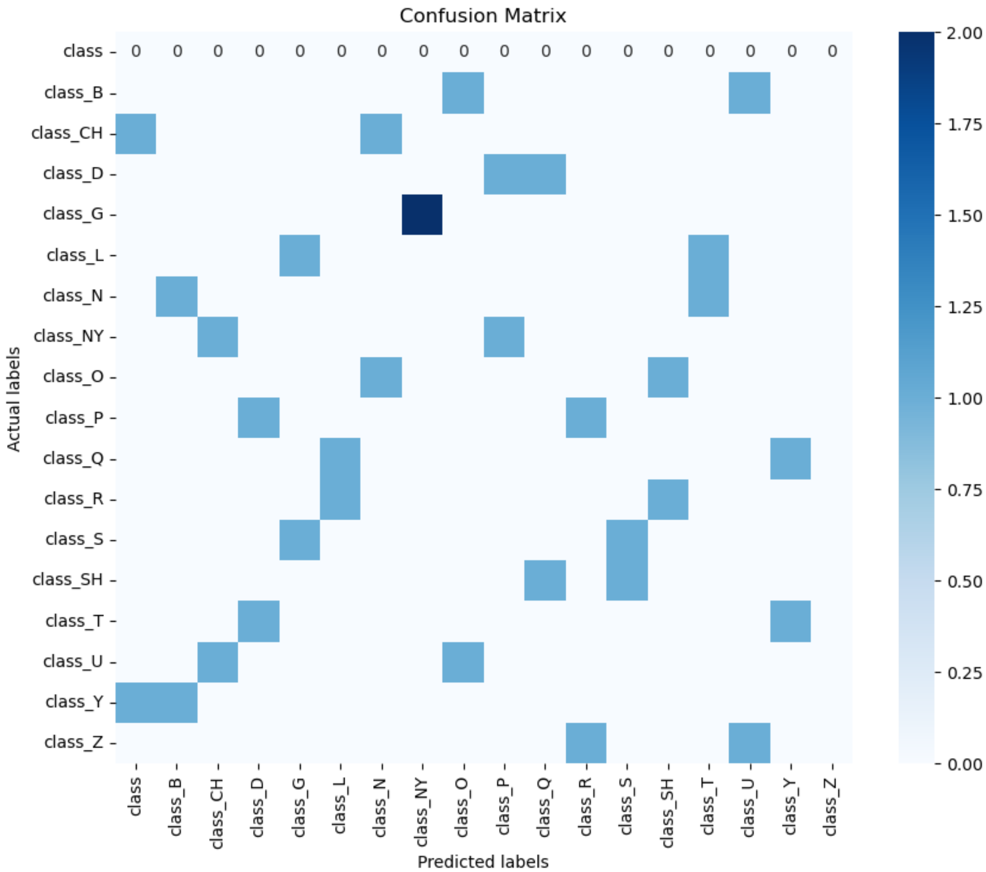



Figure 15: Confusion Matrix model accuracy

### 5.6 Evaluate Old Hungarian model on Old Turkic

The deep neural network model trained on old Hungarian data and applied on old Turkic to identify the character of different symbols. To analyze the relationship between these two languages. The accuracy is around 11% which does not provide any specific information related to the relationship between two languages. The MNIST dataset has ten characters and trained on 60000 images which shows that one charcter has more than 1000 images. In this study we used maximum 360 images for any character. Training on large augmented dataset may be improve the accuracy that can help to understand the relationship between Old Hungarian and Old Turkic.

```
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.1213 - loss: 7.9250
Evaluation Result:
Loss: 8.04805850982666
Accuracy: 0.11764705926179886
```

```
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - categorical_crossentropy: 440.3127 - loss: 1.9222
Evaluation Result:
Loss: 1.9392201900482178
Accuracy: 471.04156494140625
```

```
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.1369 - loss: 12.2048
Evaluation Result:
Loss: 12.206172943115234
Accuracy: 0.11764705926179886
```

# 6 Conclusions and future work:

In this research, several avenues for future investigation emerge from our current findings, providing opportunities to enhance the robustness and scope of the study:

- Enhancement of Data Augmentation: To fortify the performance and diversity of our neural network models, future efforts will focus on refining data augmentation techniques tailored specifically to historical script recognition. This involves exploring advanced augmentation methods such as geometric transformations, color manipulations, and synthetic data generation to further diversify the training set and improve model resilience to variations in script styles and conditions.

- Addressing Model Overfitting and Generalization: A critical aspect highlighted by our study is the challenge of model overfitting and the need for improved generalization. Future work will delve into the development of regularization strategies, exploring techniques like dropout, L1/L2 regularization, and early stopping. Additionally, investigating domain adaptation methods to enable better performance across script styles and eras will be pursued to enhance the model's applicability.

- Expansion of Symbolic Image Database: Incorporating a broader spectrum of historical scripts, particularly symbols from Old Hungarian, Old Turkic, Carian, and others, will enrich the dataset's representational diversity. This expansion will involve extensive data collection efforts focused on acquiring more varied and intricate symbols to bolster the model's recognition capabilities across a wider range of historical scripts.

- Inclusion of Additional Scripts: Expanding the scope of the research to encompass more ancient scripts like Linear B presents an exciting opportunity to broaden the applicability of our approach. Future work will involve integrating these new scripts into the training and evaluation processes, enabling a comprehensive assessment of the model's adaptability and performance across a more extensive script repertoire.

- Model Complexity Optimization: Further exploration of neural network architectures by increasing the depth (layers) and breadth (nodes) of the network will be undertaken. This entails conducting experiments to determine optimal network configurations, balancing model complexity with computational efficiency, and assessing the impact on recognition accuracy and speed. Fine-tuning the neural architecture will be pivotal in achieving state-of-the-art performance in historical script recognition tasks.

- Comparative Analysis of Scripts: Leveraging the insights gained from our model's performance across various historical scripts, a promising avenue for future research involves conducting a comparative analysis to elucidate relationships and similarities among these scripts. By examining patterns of recognition accuracy and error types across different script families, we can infer linguistic or cultural connections and potentially uncover evolutionary or historical relationships between script systems. This comparative approach not only enriches our understanding of individual scripts but also contributes to broader linguistic and cultural studies, shedding light on the interconnectedness of ancient writing systems.

By pursuing these future directions, we aim to advance the capabilities of historical script recognition systems, fostering greater accuracy, adaptability, and applicability across diverse script domains and historical eras. Each avenue presents unique challenges and opportunities that promise to enhance the efficacy and depth of our research in this compelling domain.

## References


[1] *"Who is Gyges?" once again: assessing the Carian connections of the first Mermnad king of Lydia*, volume Studies on the history and archaeology of Lydia from the Early Lydian period to Late Antiquity of *Collection de l'Institut des Sciences et Techniques de l'Antiquité*. Presses universitaires de Franche-Comté, 2023.

[2] Shruti Daggumati and Peter Z. Revesz. Data mining ancient scripts to investigate their relationships and origins: Proceedings of the 23rd international database applications and engineering symposium, Jun 2019.

[3] Mohamed Elleuch, Rania Maalej, and Monji Kherallah. A new design based-svm of the cnn classifier architecture with dropout for offline arabic handwritten recognition. *Procedia Computer Science*, 80:1712–1723, 2016. International Conference on Computational Science 2016, ICCS 2016, 6-8 June 2016, San Diego, California, USA.

[4] Valentina Leanyfalu. Rovás, The Székely – Hungarian Alphabet — dailynewshungary.com. `https://dailynewshungary.com/rovas-the-szekely-hungarian-alphabet/`. [Accessed 14-05-2024].

[5] Wikipedia. Carian language - Wikipedia — en.wikipedia.org. `https://en.wikipedia.org/wiki/Carian_language`. [Accessed 14-05-2024].

[6] Wikipedia. Old Turkic script - Wikipedia — en.wikipedia.org. `https://en.wikipedia.org/wiki/Old_Turkic_script`.