

RESEARCH

Open Access



R-GNN: recurrent graph neural networks for font classification of oracle bone inscriptions

Jiang Yuan¹, Shanxiong Chen^{1,2*}, Bofeng Mo³, Yuqi Ma¹, Wenjun Zheng¹ and Chongsheng Zhang⁴

Abstract

Font classification of oracle bone inscriptions serves as a crucial basis for determining the historical period to which they belong and holds significant importance in reconstructing significant historical events. However, conventional methods for font classification in oracle bone inscriptions heavily rely on expert knowledge, resulting in low efficiency and time-consuming procedures. In this paper, we proposed a novel recurrent graph neural network (R-GNN) for the automatic recognition of oracle bone inscription fonts. The proposed method used convolutional neural networks (CNNs) to perform local feature extraction and downsampling on oracle bone inscriptions. Furthermore, it employed graph neural networks (GNNs) to model the complex topography and global contextual information of oracle bone inscriptions. Finally, we used recurrent neural networks (RNNs) to effectively combine the extracted local features and global contextual information, thereby enhancing the discriminative power of the R-GNN. Extensive experiments on our benchmark dataset demonstrate that the proposed method achieves a Top-1 accuracy of 88.2%, significantly outperforming the competing approaches. The method presented in this paper further advances the integration of oracle bone inscriptions research and artificial intelligence. The code is publicly available at: <https://github.com/yj3214/oracle-font-classification>.

Keywords Oracle bone inscriptions, Deep learning, Font classification, Recurrent graph neural network

Introduction

Text is an important carrier that has enabled the transmission of human history and civilization for thousands of years, oracle bone inscriptions refer to the writing system engraved on turtle shells and animal bones during the Shang and early Zhou dynasties in China [1]. Oracle bone inscriptions are a precious intangible cultural heritage of the Chinese nation, this form of writing was

primarily used to record divination results, but it was also employed to document important events such as trade, land registration, and rituals later [2]. Furthermore, oracle bone inscriptions laid the foundation for the early development of language and writing in China, and its strokes and structural methods had a profound influence on the evolution of Chinese characters in later periods [3]. The research on oracle bone inscriptions is beneficial for people to understand the history of the Chinese Shang Dynasty and the evolution of Chinese characters, and it has garnered widespread attention worldwide [4]. Currently, research on oracle bone inscriptions has become an independent discipline and has been extensively applied in various research fields, including history, culture, and linguistics. The font classification of oracle bone inscriptions plays an extremely important role in promoting the progress of oracle bone inscriptions research, which is a crucial topic in the field of oracle bone research. Recently, most of the advancements

*Correspondence:

Shanxiong Chen
csxpmml@163.com

¹ College of Computer and Information Science, Southwest University, Chongqing 400715, China

² Key Laboratory of Ethnic Language Intelligent Analysis and Security Governance of MOE, Minzu University of China, Beijing 100081, China

³ The Center for Oracle Bone Studies, Capital Normal University, Beijing 100048, China

⁴ School of Computer and Information Engineering, Henan University, KaiFeng 475001, China

in oracle bone inscriptions research have been made based on font classification. The font classification of oracle bone inscriptions refers to the categorization and organization of oracle bone characters according to their font and form characteristics. Researchers can quickly determine the age of each oracle bone and the chronological relationships between multiple oracle bones based on the style of their inscriptions. Experts in the field of oracle bone inscriptions can also use the inscriptions on the oracle bones to reconstruct ancient Chinese history. The accuracy of determining the age of historical materials directly affects the completeness of historical reconstruction. Therefore, the font classification of oracle bone inscriptions plays a crucial role in reconstructing the history of the Shang Dynasty in ancient China.

Despite decades of efforts by several generations of researchers, the font classification of oracle bone inscriptions has made certain achievements, but the following challenges still exist: (1) Oracle bone inscriptions are a kind of logographic writing system with strong pictorial characteristics and complex strokes [5], which requires researchers in oracle bone inscriptions research to have a proficient understanding of oracle bone paleography, calligraphy, archaeology, and other specialized knowledge. (2) Describing font styles involves a high level of subjectivity. Merely referring to relevant literature without extensive practice using specific examples makes it equally difficult to master this skill accurately. (3) Currently, the font classification of oracle bone inscriptions heavily relies on expert knowledge, which lacks objective analysis and quantitative classification criteria, which can lead to inconsistencies in the classification results.

The existing font classification of oracle bone inscriptions heavily relies on expert knowledge, where experts in oracle bone research comprehensively use features such as the size of characters, spacing between characters, and the thickness of strokes to classify the oracle bone inscription fonts, which requires researchers to possess extremely strong discernment ability and accumulate long-term experience, which consumes a significant amount of time and energy. The font classification method relying solely on expert knowledge is a method that lacks objective analysis and quantitative indicators. Even experts in font classification find it difficult to use quantitative indicators to describe the features of various fonts. This makes it challenging to design a rule-based algorithm based solely on expert knowledge to automatically classify the fonts of oracle bone inscription.

In recent years, significant progress has been made in the research of ancient inscriptions due to the effective learning of inherent patterns and feature representations from samples using deep learning. We believe these statistical learning methods hold great potential in the font

classification of oracle bone inscriptions. However, to our knowledge, very few researchers have applied deep learning to the font classification of oracle bone inscriptions. To address this gap, we propose a recurrent graph neural network (R-GNN) for the font classification of oracle bone inscriptions. By comprehensively extracting both local detailed features and global contextual information from the oracle bone inscriptions, our R-GNN effectively models the features of the oracle bone inscriptions. We achieved an impressive Top-1 accuracy of 88.2% on our benchmark dataset.

Dataset description and challenges

Some factors contributing to the success of deep learning are the availability of real-world samples and effective feature representation [6, 7]. So we have created a new dataset provided by researchers from the oracle bone inscription research center at Capital Normal University in China, named the oracle font classification dataset (OFCD), specifically for training and testing the R-GNN. The OFCD comprises a total of 1473 handprinted oracle bone images, covering 16 different fonts. Figure 1 shows different types of oracle bone images. The handprinted oracle bone images are copied by oracle experts from the colored oracle bone images.

The colored oracle bone image contains various sources of noise, and there are significant differences in the distribution and quantity of characters among different samples. These challenges pose significant difficulties in computing effective feature representations for the samples and learning their intrinsic patterns. Therefore, this paper uses the handprinted oracle bone images to verify the feasibility and effectiveness of deep learning to recognize oracle bone inscription fonts. Due to the severe fragmentation of the oracle bones, the amount of writing on each oracle bone varies greatly and there are large areas of blank space. To balance the number of oracle characters on each sample and remove white space, we preprocessed the oracle bone. Since the original dataset contains a substantial amount of noise and there is significant disparity in the distribution of character counts within each sample, preprocessing of the original dataset is essential. Initially, we manually extract a series of text segments from the samples, after which we concatenate text segments of the same font along their edges. Subsequently, these concatenated segments are uniformly scaled to a size of 128×128 . After the aforementioned processing, we obtain a dataset with reduced noise and a more evenly distributed character quantity among samples. Moreover, even for the same font, variations exist in handwriting due to different environments and writing instances. Our synthesized samples incorporate text information from diverse original samples of the same

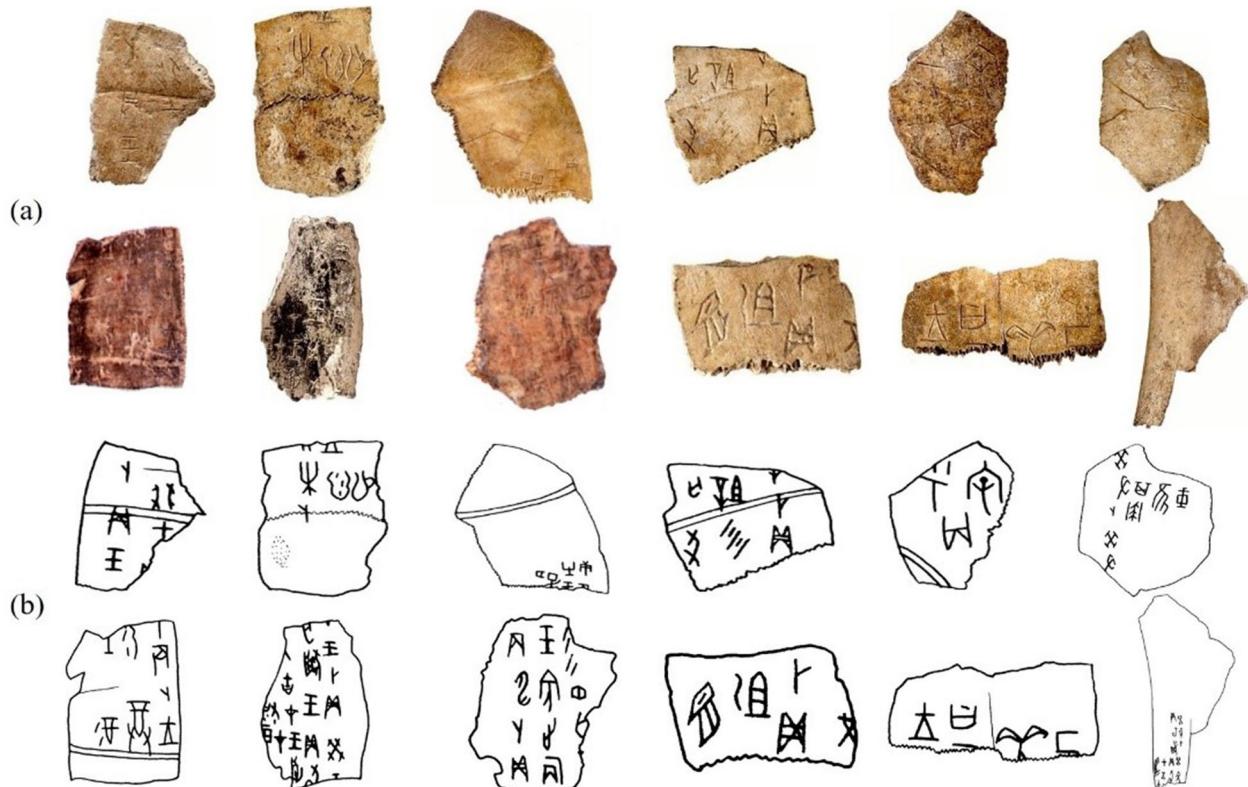


Fig. 1 Different types of oracle bone images. **a** colorful oracle bone images. **b** handprinted oracle bone images

font, thus compelling the network to learn consistent features of the same font. Note that text segments from the training set and the testing set are not concatenated with each other. This ensures that there are no shared text segments between samples in the training and testing sets, ensuring the reliability of the experimental results. Through the aforementioned preprocessing, we ultimately obtained a total of 9,426 samples, with 7578 in the training set and 1848 in the testing set, and some are shown in Fig. 3. Notably, the similarity between different fonts of oracle bone inscriptions is relatively high, which poses a challenge for computer-based approaches in classifying oracle bone inscription fonts.

After collecting the OFCD dataset, we considered two main issues in designing an oracle font classification algorithm: (1) How to describe the complex topological structure of oracle bone inscriptions? (2) How to integrate global contextual information and local features of oracle bone inscriptions? Oracle bone inscriptions are a logographic writing system with highly intricate arrangement rules for the overall inscriptions, effectively modeling the complex topological structure of oracle bone inscriptions becomes a crucial consideration in the design of a network. On the other hand, the overall writing style of the oracle bone inscriptions and the writing

form of the character both contribute to improving the classification performance. Thus, carefully designing the integration of global contextual information and local features of oracle bone inscriptions is equally important.

To address these challenges, we have proposed a recurrent graph neural network (R-GNN) that effectively combines the advantages of graph neural networks and recurrent neural networks. As shown in Fig. 2, we represent the oracle bone images as graphs, and utilize graph neural networks to learn the complex topological structure of the oracle bone inscriptions. Simultaneously, we consider using gated recurrent units [8] (GRUs) to integrate the local segment features extracted by convolutional neural networks (CNN) with the global contextual information extracted by the graph neural networks. The proposed method thus incorporates the global contextual information from GNNs and the local features from CNNs. Finally, we employ this integrated feature representation for font classification. The main contributions of this paper are as follows:

- (1) We have collected a dataset named OFCD specifically for font classification of oracle bone inscriptions. The OFCD dataset comprises 1473 hand-printed oracle bone images, covering 16 different

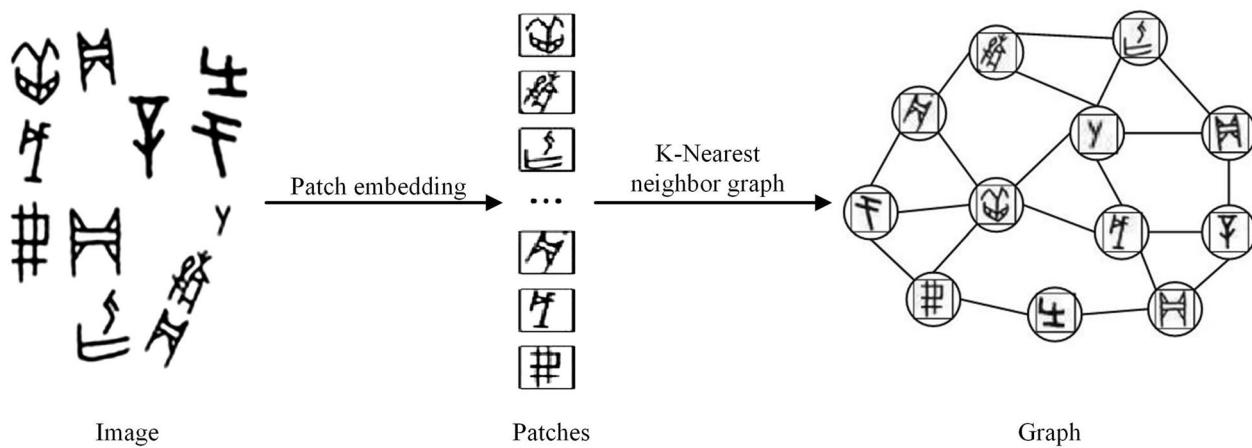


Fig. 2 A schematic diagram representing the complex topological structure of oracle bone inscriptions using a graphical structure



Fig. 3 Some synthesized samples from the OFCD dataset

- fonts of oracle bone inscriptions. The dataset serves as a benchmark platform for comparative evaluations.
- (2) We propose a network called R-GNN for font classification of oracle bone inscriptions. R-GNN effectively combines the strengths of graph neural networks and recurrent neural networks. It can comprehensively integrate the global contextual information and local features of oracle bone inscriptions, enabling effective identification of different fonts.

- (3) Extensive experimental results on our benchmark dataset demonstrate the effectiveness of R-GNN. Furthermore, we validate the robustness of R-GNN in representing the font features of oracle bone inscriptions through visualization.

Related work

This section mainly reviews the related work from the following two aspects: font classification of oracle bone inscriptions and character classification of oracle bone inscriptions.

Font classification of oracle bone inscriptions

The publication of “Classification and Chronology of Oracle Bone Inscriptions from the Yin Ruins” [9] in 1991 is an important reference in the field of font classification of oracle bone inscription. It subdivides oracle bone inscriptions into 20 categories and summarizes the features of various fonts. Building upon this foundation, subsequent researchers conducted more detailed studies within the same font of inscriptions. The results of these related studies are included in books such as

“Compilation and Research of oracle bone inscriptions with the font named Wuming” [10] and “Compilation and Research of Yin Ruins Village South Series Oracle Bone Inscriptions.” [11] The authors of these books have made significant contributions to font classification of oracle bone inscription. Professor Mo, Bofeng from Capital Normal University has provided a systematic description of the research progress in the field of font classification of oracle bone inscription in [12], emphasizing the significance of researching the font of oracle bone inscription. However, in the literature, there are few instances where computer technology has been combined with font classification of oracle bone inscription, it is noteworthy that our proposed R-GNN represents the first attempt to employ deep learning to tackle this challenge.

Font classification of general text

Some researchers have already explored the use of deep learning for font classification on general text. Wang [13] proposed a method that combines convolutional neural networks with SCAE-based domain adaptation for font classification. However, this approach requires a large amount of unlabeled real-world data and millions of synthetic data, resulting in significant training costs and making it unsuitable for font classification of oracle bone inscription, which typically involves a smaller number of samples. On the other hand, Zhang [14] introduced a convolutional neural network with Squeeze-Excitation modules and Haar transform layers for Chinese calligraphy style classification. While this method achieved good performance in classifying four styles of Chinese characters, its effectiveness was limited when applied to font classification of oracle bone inscription. He Sheng and Schomaker [15] proposed a Convolutional Neural Network (FragNet) for English font classification, which involves multi-scale feature extraction and feature fusion. This method achieved good results but overlooked the spatial context information of the text. Subsequently, they further introduced the Global Context Residual Recurrent Neural Network [16] (GR-RNN) for English font classification. This approach leverages the complementary information between global context and local features to further enhance the accuracy of font classification. Srivastava [17] utilized deep learning and proposed three different methods for font classification. These methods include spatial attention mechanisms, multi-scale feature fusion, and patch-based Convolutional Neural Networks. Mohammadian [18] introduced the first publicly available datasets for Persian font recognition and proposes a Visual Font Recognition (VFR) system using Convolutional Neural Networks. Wang [19] introduced a deep learning-based writer adaptation method for handwritten text recognition, utilizing a Style

Extractor Network (SEN) trained by identification loss to explicitly extract personalized writer information. Abderazak [20] presented WriterINet, a CNN-based approach for writer identification by decomposing handwritten documents into segmented images, employing a powerful deep feature architecture, and achieving competitive or superior performance on various benchmark datasets. The above-mentioned methods have demonstrated excellent performance in mainstream font recognition tasks; however, due to the limited volume of oracle bone inscription data, the complex layout of characters, and the high similarity between fonts, achieving satisfactory results solely using CNN becomes challenging in the context of oracle bone inscriptions. R-GNN can be compared with these methods, and the experimental results will be presented in subsequent chapters.

Character classification of oracle bone inscriptions

Compared to font classification of oracle bone inscriptions, some researchers have utilized computer technology to address the issue of character classification of oracle bone inscriptions. Researchers treat oracle bone inscriptions as sketches, Yu [21] proposed a multi-scale, multi-architecture CNN framework along with two data augmentation strategies. By fusing the features from multiple sub-networks, they achieved high classification performance. Liu [22] designed a fully convolutional neural network for classifying oracle bone inscription characters. They demonstrated the superiority of their method on a dataset containing 44,868 oracle bone inscription characters, achieving an accuracy of 94.38%. Huang [23] presented OBC306, the largest oracle bone inscription dataset with over 300,000 character-level samples, addressing the scarcity of labeled data for automatic recognition. Li [24] introduced a deep learning framework based on metric learning for character retrieval of oracle bone inscriptions. This method demonstrated excellent performance in character retrieval tasks. These methods contribute to the advancement of character classification of oracle bone inscriptions using computer-based methods.

The aforementioned methods were primarily trained and tested on datasets with balanced sample distributions. However, some researchers are actively working to address the issue of imbalanced sample distributions in the character classification of oracle bone inscription. Zhang [25] proposed a deep metric learning method that maps character features to Euclidean space to calculate the similarity between characters. They further employed the nearest neighbor approach for classification, which partially alleviated the problem of imbalanced sample distribution. Li [26] proposed a mix-up strategy to overcome imbalances in limited oracle bone character

datasets, this study achieves a new state of the art in automatic recognition by incorporating softmax and triplet losses. Li [27] further introduced a generative adversarial network (GAN) framework to enhance the classification of challenging oracle characters. This method achieved optimal results on multiple datasets. Overall, character classification of oracle bone inscription is still in its preliminary exploration stage, and researchers are actively working to solve the above challenges.

Methods

In this section, we will provide a detailed introduction to the proposed font classification method of oracle bone inscriptions, R-GNN. Both the overall style of oracle bone inscriptions and the glyph structure of individual characters contribute to improving the model's classification performance. Therefore, the R-GNN focuses on integrating the local features extracted by convolutional neural networks and the global contextual information extracted by graph neural networks.

Overall network architecture design

The R-GNN mainly consists of one convolutional feature extraction block (Convolution block), three groups of graph convolutional feature extraction blocks (GC blocks), and three residual gated recurrent units (Residual GRUs), as illustrated in Fig. 4. The input image size is $1 \times 128 \times 128$. After passing through each block, the resulting feature map size is represented as $c \times w \times h$, where c , w , and h denote the depth (number of channels), width, and height of the feature map, respectively. The convolutional block is used to extract local features from oracle bone inscriptions, and the graph convolutional block

models the topological structure of oracle bone inscriptions to aggregate contextual information. Furthermore, it constructs a feature pyramid to compute information at different scales, resulting in a more comprehensive feature representation. We utilize global average pooling layers to transform the feature maps $F_i \in R^{C \times W \times H}$ obtained from each feature extraction block into $F_j \in R^{C \times 1 \times 1}$ to aggregate global features for each channel. The residual gated recurrent unit is employed to adaptively fuse local feature information and global contextual information of oracle bone inscription. Finally, we use two fully connected layers as a classifier for the ultimate prediction. We will provide a detailed explanation of the composition and design principles of each block later.

Convolutional feature extraction block

The convolutional feature extraction block is a convolutional neural network used for extracting local features from oracle bone inscriptions. It mainly consists of a normal convolutional block and two residual convolutional blocks, as shown in Fig. 5. The normal convolutional block consists of two convolutional layers with the same number of parameters, and the kernel size is set to 3×3 , and both the stride and padding are set to 1. The residual connection was first introduced in [28], and it helps alleviate problems such as gradient vanishing and weight matrix degradation, which are bee learning process of neural networks. The residual convolutional block can be simply summarized as $y = F(x, w) + x$, x and y represent the input and output, respectively, while F and w represent the convolutional neural network and its weight parameters, respectively. The residual convolutional block consists of a main branch and a residual branch.

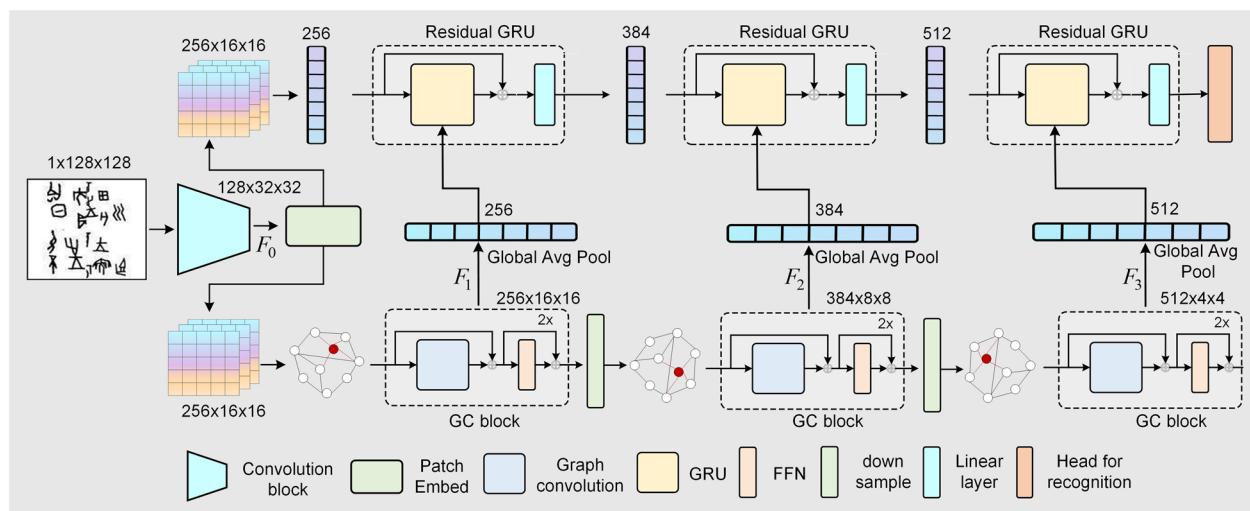


Fig. 4 Structure diagram of R-GNN

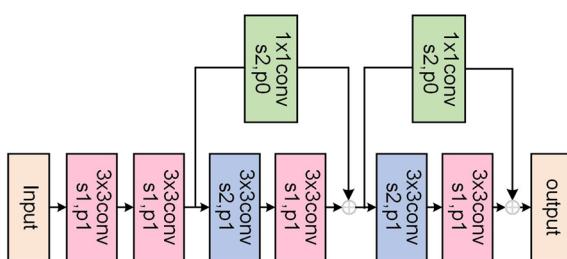


Fig. 5 Schematic diagram of the convolutional feature extraction block. Where s is the size of the convolution kernel when it slides over an input image or feature map, p is the addition of extra pixels around an input image or feature map

The main branch comprises a convolutional layer with a kernel size of 3×3 , a stride of 2, and padding of 1, followed by another convolutional layer with a kernel size of 3×3 , a stride of 1, and padding of 1. The residual branch adjusts the number of channels in the original feature map by using a convolutional layer with a kernel size of 1×1 and a stride of 2, which helps align the feature maps. The aforementioned convolutional layers are followed by a batch normalization layer and a non-linear activation function (GeLU). Eventually, we can extract a feature map F_0 with a size of $128 \times 32 \times 32$, and F_0 will serve as the initial input for the residual gated recurrent unit.

Residual gated recurrent unit

Gated recurrent unit incorporates a gating mechanism designed to capture short-distance dependencies in sequential data. This enables the model to better comprehend the relationships between adjacent time steps in a sequence, facilitating effective feature fusion. The residual gated recurrent unit makes the final font prediction by integrating the local feature (F_0) with multi-scale contextual information (F_1 , F_2 , and F_3). To obtain feature vectors for each feature map, we apply global average pooling to each feature map to obtain a feature vector of dimension c (the number of channels in the feature map). This serves as the final input for the residual gated recurrent unit. Additionally, we have added a fully connected layer after each residual gated recurrent unit to achieve feature transformation and alignment. The GRU is a type of recurrent neural network and it can address issues such as vanishing gradients in long-term memory and backpropagation. The expression for GRU can be represented as follows:

$$\begin{aligned} r_t &= \sigma(W_r x_t + U_r f_{t-1} + b_r), \\ z_t &= \sigma(W_z x_t + U_z f_{t-1} + b_z), \\ h_t &= \tanh(W_h x_t + U_h (r_t \odot f_{t-1}) + b_h), \\ f_t &= z_t \odot f_{t-1} + (1 - z_t) \odot h_t, \end{aligned} \quad (1)$$

where x_t represents the feature vector of the feature map, while f_t stands for the global contextual information at time step t . The symbols r_t and z_t denote the reset gate and update gate, respectively, with σ representing the sigmoid activation function. The parameters W_r , W_z , and W_h signify the learnable parameters during the training process. The GRU can also be globally expressed as $f_t = \text{GRU}(x_t, f_{t-1})$. Furthermore, we enhance the GRU by introducing an additional residual branch and incorporating more feature transformations to enhance feature diversity, as illustrated below:

$$\begin{aligned} f_t &= \text{GeLU}(W_t(\text{GRU}(x_t, f_{t-1}) + f_{t-1})) + b_t, \\ f^0 &= \text{GAP}(F_0), \end{aligned} \quad (2)$$

where W_t represents the learnable parameters during training, and GAP represents the global average pooling operation.

Graph convolutional feature extraction block

The graph convolutional feature extraction block is a graph convolutional neural network designed to extract global contextual information. We partition the obtained feature map F_0 into 256 blocks using a 2×2 window and transform it into a 256-dimensional feature vector x_i . These feature vectors can be regarded as a set of unordered nodes, denoted as $V = \{v_1, v_2, \dots, v_{256}\}$. Subsequently, we employ the k -nearest neighbors approach to identify the k Euclidean closest neighbors $N(v_i)$ of node v_i , and for all $v_j \in N(v_i)$, we add an edge directed from v_j to v_i . Finally, we can obtain a graph $G = (V, E)$, where V represents all the nodes and E represents all the edges. The advantage of the graph lies in its ability to flexibly model complex objects in images, effectively capturing intricate topological structures, and enabling long-range information interactions. The graph convolutional feature extraction block consists primarily of a graph convolutional layer and a multi-layer perceptron. The graph convolutional layer facilitates information interaction between nodes by aggregating the features of neighboring nodes, subsequently refining the representation of each node. The graph convolution operation can be represented as follows:

$$\begin{aligned} \mathcal{G}^{out} &= f(\mathcal{G}^{in}, W) \\ &= \text{Updatae}(\text{Aggregate}(\mathcal{G}^{in}, W_{agg}), W_{update}), \end{aligned} \quad (3)$$

where W_{agg} and W_{update} represent the learnable parameters in the aggregation and update operations, respectively.

Specifically, the aggregation operation computes the node representation by aggregating features from neighboring nodes, while the update operation further

integrates the aggregated features to obtain the updated node representation. We choose edgeConv [29] as the graph convolution operation in our approach, and its advantage lies in the effective learning of global spatial information. The edgeConv operation can be represented as follows:

$$x'_i = \max_{j:(i,j) \in \mathcal{E}} W_{agg}(x_i, x_j \mid j \in \mathcal{N}(x_i)), \quad (4)$$

where x_j represents the neighboring node of x_i , and x'_i is the updated node representation of x_i . W_{agg} denotes the learnable parameters in the network.

We further designed a multi-head aggregation operation for edgeConv based on the approach in [30]. In order to aggregate features of neighboring nodes from different subspaces to enhance feature diversity, we initially partition the node feature x_i into h heads. These heads can individually aggregate neighboring node features with different weight parameters. Finally, all the heads can be concatenated together to form the ultimate aggregated feature, as shown below:

$$x'_i = \max_{j:(i,j) \in \mathcal{E}} \left(\text{head}^1 W_{agg}^1, \dots, \text{head}^h W_{agg}^h \right). \quad (5)$$

The graph convolution operation described above can be represented as $X^{\text{out}} = \text{GraphConv}(X^{\text{in}})$. Additionally, we utilize a fully connected layer before and after the graph convolution operation to map node features to the same feature subspace, and incorporate residual connections to alleviate the oversmoothing. The graph convolution operation in this paper is ultimately represented as follows:

$$Y = \text{GeLU}(\text{GraphConv}(XW_{\text{in}}))W_{\text{out}} + X, \quad (6)$$

where X and Y respectively denote the inputs and outputs of the graph convolution operation, and W_{in} and W_{out} represent the learnable parameters of the fully connected layer.

We have further augmented the graph convolution operation with residual connection and feed-forward network, aiming to enhance feature representation capabilities and alleviate over-smoothing. The entire graph convolutional feature extraction module is presented as follows:

$$Z = \text{GeLU}(YW_i)W_j + Y, \quad (7)$$

where W_i and W_j represent the learnable parameters of the feed-forward network. We apply a convolutional layer after each group of graph convolutional feature extraction modules to construct a feature pyramid. This process allows us to obtain multi-scale feature maps F_1 ,

F_2 , and F_3 , which will serve as inputs to the residual gated recurrent unit.

Experiment and discussion

Experimental environment

The proposed R-GNN network is implemented using the Python programming language and the PyTorch framework. The hardware configuration of the operating platform includes an Nvidia GeForce RTX 2080 SUPER GPU with 8 GB of memory. We employed the AdamW optimizer [31] for training the R-GNN, with default momentum parameters of $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The initial learning rate was set to 5×10^{-4} , with a minimum learning rate of 5×10^{-7} . The learning rate decay strategy was implemented using Cosine Annealing. The weight decay coefficient was set to 0.01, the batch size to 16, and a total of 100 epochs were conducted.

Measure indicators

The primary metric of concern for oracle bone inscription experts is the accuracy of the model in recognizing oracle bone inscription fonts. Therefore, we use Top-1 accuracy as a key performance evaluation metric. In addition, recall, precision, F1-score, and mAP are employed as supplementary evaluation metrics. The value of mAP represents the average performance of the model across the entire image database. A higher mAP value indicates that the model is more capable of accurately retrieving relevant images. These metrics are computed utilizing the confusion matrix. The symbols in the confusion matrix are defined as follows: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Top-1 accuracy can be expressed as follows:

$$\text{Top-1 accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (8)$$

Utilize Eq. (9) for computing the precision rate, which represents the ratio of accurately predicted samples (true positives) among those predicted as correct, reflecting the accuracy of the model predictions.

$$\text{Precision} = \frac{TP}{TP + FP}. \quad (9)$$

Utilize Eq. (10) for computing the recall rate, representing the count of correctly predicted true values among the positive samples, to reflect the comprehensiveness of the model predictions.

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (10)$$

The formula for calculating the F1 score for each category is illustrated in Eq. (11). The F1 score addresses the

Table 1 Classification results of oracle bone inscription fonts using different methods

Method	Top-1	Precision	Recall	F1-score
ResNet-34 [28]	0.804	0.790	0.784	0.787
GR-RNN(vertical) [16]	0.852	0.832	0.831	0.832
GR-RNN(horizontal) [16]	0.857	0.844	0.834	0.839
FragNet-16 [15]	0.833	0.816	0.818	0.817
FragNet-32 [15]	0.845	0.837	0.855	0.845
FragNet-64 [15]	0.847	0.856	0.837	0.846
Patch [17]	0.832	0.829	0.796	0.812
SA-Net [17]	0.860	0.854	0.852	0.853
MSRF-Net [17]	0.836	0.838	0.805	0.821
HAMVisContexNN [19]	0.810	0.805	0.792	0.798
HAMVisContexNN+WIDNN+Bridge [19]	0.822	0.836	0.801	0.818
WriterNet [20]	0.847	0.850	0.847	0.848
Ours	0.882	0.881	0.879	0.880

The best results are marked in bold black

trade-off between precision and recall, with higher values indicating better performance.

$$F1\text{-score} = 2 \frac{Precision \times Recall}{Precision + Recall}. \quad (11)$$

Comparative experiments

Font classification of oracle bone inscriptions

We compared our approach with existing font classification methods using evaluation metrics such as Top-1 accuracy, Precision, Recall, and F1-score. Table 1 presents the results of the comparative experiments. Additionally, we showcase the predictions of our model on several oracle bone images, as shown in Fig. 6. Quantitative analysis reveals that the proposed R-GNN model significantly outperforms other competing methods. Our model achieves a top-performing Top-1 accuracy of 88.2%, surpassing SA-Net by 2.2%. Regarding the

more comprehensive evaluation metric, F1-score, our model attains an F1-score of 88.0%, also surpassing SA-Net by 2.7%. Furthermore, the baseline model ResNet-34 exhibited the poorest performance in the experiments. This suggests that careful network architecture design is indeed necessary to achieve satisfactory results in oracle bone inscription font classification. In comparison to ResNet-34, the proposed R-GNN model achieves a higher Top-1 accuracy by 7.8% and a higher F1-score by 9.3%. We can conclude that the proposed R-GNN achieves superior performance in oracle bone inscription font classification compared to other competing methods, demonstrating the superiority of our approach.

Font retrieval of oracle bone inscriptions

We performed font retrieval on the testing set of oracle bone inscriptions to assess the robustness of different methods. Font retrieval of oracle bone inscriptions

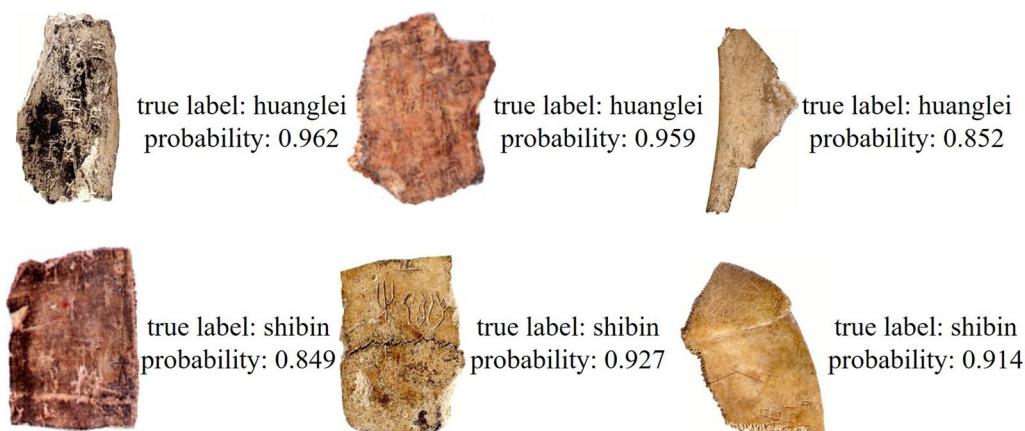


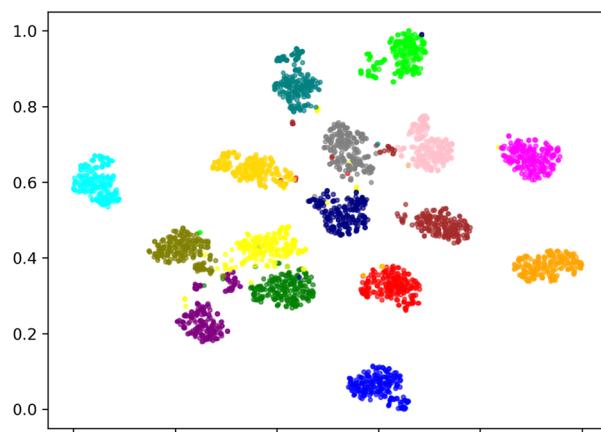
Fig. 6 Predictions of R-GNN on oracle bone images

Table 2 Retrieval results of oracle bone inscription fonts using different methods

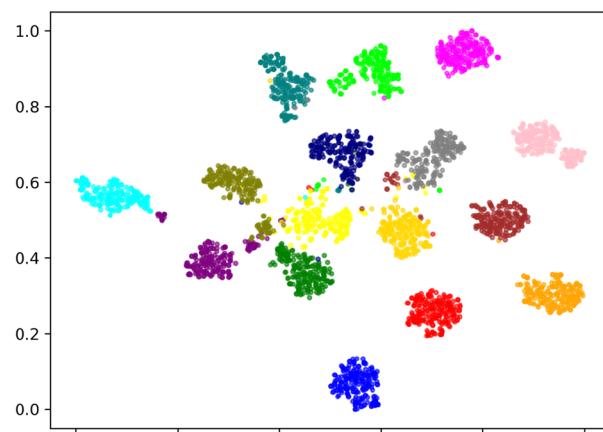
Method	Top-1 accuracy	mAP
ResNet-34 [28]	0.880	0.519
GR-RNN(vertical) [16]	0.960	0.656
GR-RNN(horizontal) [16]	0.962	0.658
FragNet-16 [15]	0.928	0.561
FragNet-32 [15]	0.944	0.585
FragNet-64 [15]	0.952	0.621
Patch [17]	0.875	0.555
SA-Net [17]	0.947	0.603
MSRF-Net [17]	0.891	0.561
HAMVisContexNN [19]	0.890	0.525
HAMVisContexNN+WIDNN+Bridge [19]	0.908	0.538
WriterINet [20]	0.950	0.619
Ours	0.980	0.705

The best results are marked in bold black

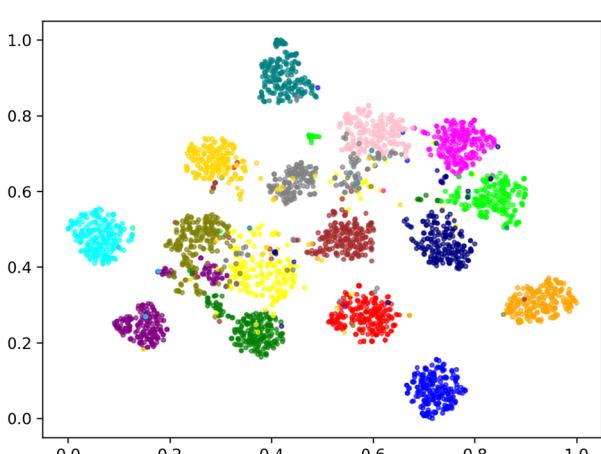
is highly similar to font classification, aiming to identify samples that belong to the same font as the query. If each query corresponds to only one true sample, this can be considered as oracle bone inscription font classification. We removed the classifiers of all methods to extract the feature representations of each sample in the testing set. By randomly selecting a sample from the testing set as a query, we conducted oracle bone inscription font retrieval experiments. We utilized mean average precision (mAP) and Top-1 accuracy as evaluation metrics for oracle bone inscription font retrieval. Table 2 presents the results of oracle bone inscription font retrieval using the proposed R-GNN model in comparison with other competing methods. The experimental results indicate that the proposed R-GNN outperforms other competing methods. Our method achieves a Top-1 accuracy of 98.0%, surpassing GR-RNN (horizontal) by 1.8%. As for the comprehensive evaluation metric mAP, our model



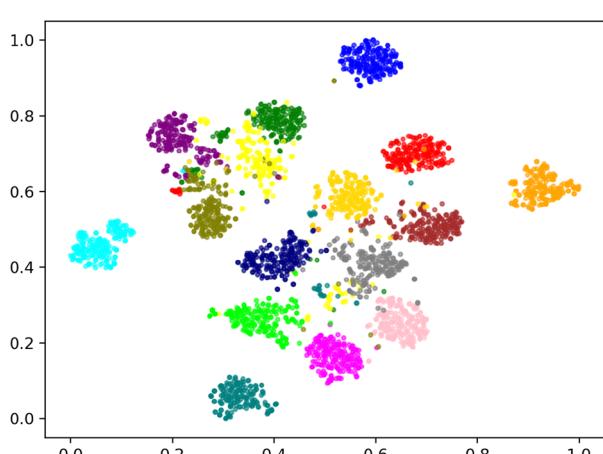
(a) ours



(b) GR-RNN(horizontal)



(c) GR-RNN(vertical)



(d) FragNet-64

Fig. 7 Scatter plots of features on the test dataset using different methods, where each color represents a different font

reaches 70.5%, which is 4.7% higher than that of GR-RNN (horizontal). Additionally, SA-Net, which demonstrates strong performance in oracle bone inscription font classification, shows relatively lower performance in font retrieval. This observation indicates that the robustness of deep feature representation of SA-Net is comparatively weaker, underscoring the significance of font retrieval experiments in model evaluation. Additionally, we visualized the feature representations of the top-performing four models on the testing set samples and employed t-SNE to project the feature vectors into a 2D space, as depicted in Fig. 7. The results suggest that the proposed approach effectively separates various categories of oracle bone inscription fonts in a uniform manner, with feature representations of samples within the same category being closely clustered. This further indicates that R-GNN is capable of learning robust and effective feature representations.

Ablation experiments

The graph convolutional feature extraction block and the residual gated recurrent unit are important components of R-GNN. The graph convolutional feature extraction module is used to model the topological structure and global contextual information of oracle bone inscriptions at multiple scales, while the residual gated recurrent unit is employed to effectively integrate the local features and global contextual information of oracle bone inscriptions. We validate the effectiveness of these proposed blocks through ablation experiments. The number of neighboring nodes, denoted as k , is a hyperparameter that controls the range of feature aggregation. In our case, we set k to be 9. The multi-head aggregation operation allows the aggregation of features from different subsets of neighboring nodes. We initially set the number of heads to be 1. We employ the proposed R-GNN for oracle bone inscriptions font classification and utilize the Top-1 accuracy, a metric of primary concern to oracle bone experts, as an evaluation criterion to validate the effectiveness of various blocks within R-GNN. Table 3 presents the results of ablation experiments. We can observe that the performance of oracle bone inscription

font classification using only the graph convolution feature extraction blocks is relatively poorer. However, it surpasses the majority of the fully convolutional neural networks in Table 1. This suggests the effectiveness of the graph convolution feature extraction blocks. By further introducing the GRU to integrate local features and global context information, and incorporating residual connections, the performance of oracle bone inscription font classification is further enhanced. This indicates the effectiveness of using the residual gated recurrent unit (GRU). In summary, the design approach of extracting oracle bone inscription font features through graph convolution operations and integrating features using residual gated recurrent units proves to be effective. Ultimately, the R-GNN achieves a Top-1 accuracy of 87.7%.

Performance with different numbers of GC blocks

We compared the impact of different numbers of blocks in each group of graph convolutional feature extraction blocks on the classification results of oracle bone inscriptions fonts. We conducted classification experiments on oracle bone inscription fonts by varying the number of blocks in each group of graph convolutional feature extraction blocks. Top-1 accuracy, precision, recall, and F1-score were used as evaluation metrics. Table 4 presents the corresponding experimental results. The experimental results indicate that, under the constraint of a similar number of parameters in the model, the best performance is achieved when every two graph convolutional feature extraction blocks form a group. The Top-1 accuracy reaches 88.2%, and the F1-score reaches 88.0%. We believe that having too few graph convolutional feature extraction blocks can lead to insufficient modeling of oracle bone inscription font features, while having too many blocks can lead to a decrease in feature diversity, thereby affecting classification performance.

Performance of different graph convolution variants

We compared the performance of four different variants of graph convolution in oracle bone inscription font classification, including EdgeConv [30], GIN [32], Graph-SAGE [33], and Max-Relative GraphConv [34]. We also considered the impact of hyperparameter k and the number of heads in multi-head aggregation on the classification performance of oracle bone inscription fonts. Figure 8 presents the corresponding experimental results. The experimental results indicate that all four different graph convolution variants achieve optimal classification performance when used with $k = 9$ and the number of heads for the three groups of graph convolution feature extraction blocks is 1, 2, and 1, respectively. Furthermore, using EdgeConv in R-GNN yielded the best classification performance, achieving a Top-1 accuracy of

Table 3 Ablation results of the graph convolutional feature extraction block and the residual gated recurrent unit in R-GNN

GC block	GRU	Attention	LSTM	R-GRU	Top-1
✓	✗	✗	✗	✗	0.855
✓	✗	✓	✗	✗	0.871
✓	✗	✗	✓	✗	0.869
✓	✓	✗	✗	✗	0.874
✓	✗	✗	✗	✓	0.877

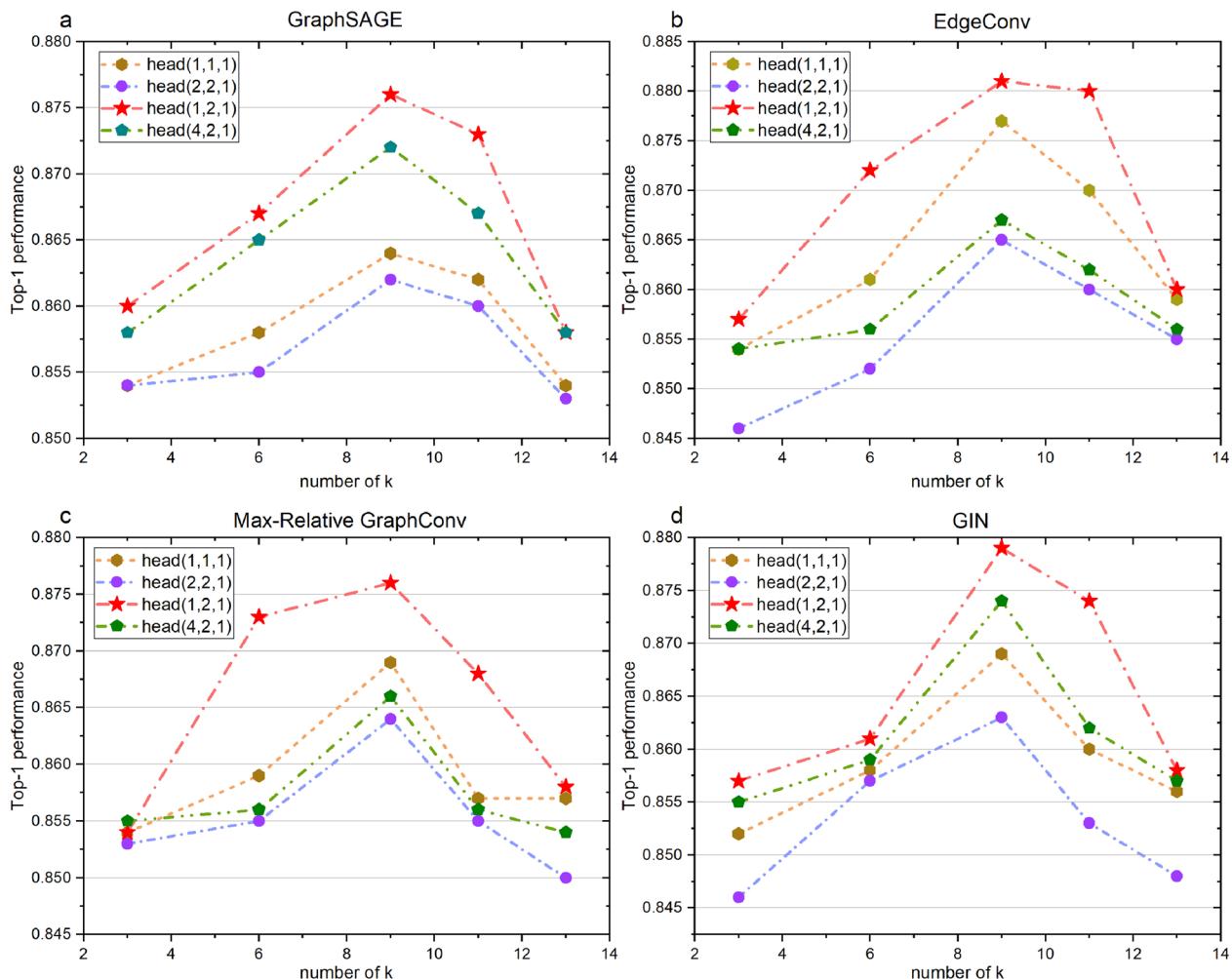


Fig. 8 Results of different graph convolution variants and different hyperparameters in oracle bone inscription fonts classification

Table 4 The experimental results of varying the number of blocks in each group of graph convolutional feature extraction blocks

Num of blocks	#Params	Top-1	Precision	Recall	F1-score
[1, 2, 4]	21.14M	0.873	0.882	0.854	0.868
[2, 4, 1]	21.05M	0.870	0.862	0.856	0.859
[4, 2, 1]	21.14M	0.868	0.843	0.850	0.847
[2, 2, 2]	20.62M	0.882	0.881	0.879	0.880

The best results are marked in bold black

88.2%. We believe that having too few neighboring nodes (k) can lead to insufficient interaction between nodes, while having too many neighboring nodes can result in oversmoothing and impact feature diversity. Through experiments, we can conclude that the optimal

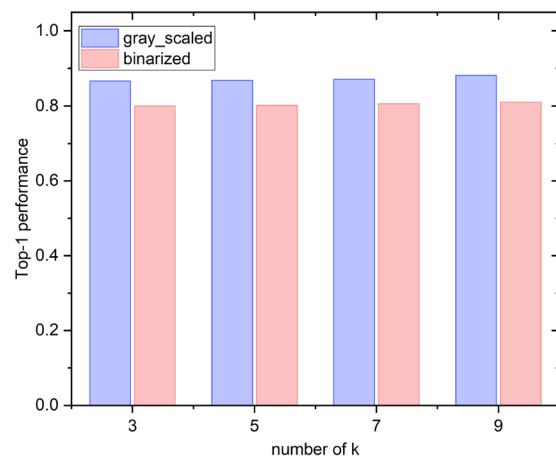
performance for oracle bone inscriptions font classification is achieved when $k = 9$.

Performance of different image input formats

We compared the impact of different image input formats on the classification performance of oracle bone inscriptions fonts, including gray-scale and binary images. The experimental results are shown in Fig. 9(b). Binarization is a common operation in image processing. We used the OTSU [35] thresholding method to obtain binary images of oracle bone inscriptions. Figure 9(a) displays several examples of gray-scale and binarized images. From the images, it can be observed that the binarized images lose some texture details and ink strokes. The experimental results indicate that the network trained on gray-scale images achieves better performance. Compared to using binarized images as the network input, the classification accuracy of using grayscale images as the network



(a) Examples of some gray-scale images and their corresponding binarized images.



(b) Results of different input image formats in oracle bone inscription font classification.

Fig. 9 Impact of different input image formats on network performance

input is on average higher by 8.0%. This suggests that the texture information present in oracle bone inscription images plays a crucial role in enabling the network to learn effective feature representations for oracle bone inscription fonts. Additionally, this also suggests that when employing computer-assisted classification of oracle bone inscription fonts, binarizing digital images can impact the final classification performance. This is because the loss of texture information due to binarization is crucial for the classification of oracle bone inscription fonts.

Conclusion

Computer-aided font classification of oracle bone inscriptions can automate the batch classification of oracle bone inscription fonts and provide essential reference for subsequent font-based clustering of oracle bone inscriptions. This holds significant value for researchers in the field of oracle bone inscriptions. In this paper, we introduce a pioneering deep learning

approach for the font classification of oracle bone inscriptions, namely the R-GNN network. This method effectively captures both the local fine-grained details and the global contextual information of oracle bone inscriptions. The proposed R-GNN outperforms other competitive methods in terms of performance on the OFCD dataset, as demonstrated by extensive experiments. This serves as strong evidence for the effectiveness of the R-GNN network. In future work, we aim to further enhance the Top-1 accuracy of our method. Additionally, we aspire to expand the application scope of our approach. For instance, the performance of applying the proposed method to the font classification of oracle bone inscription rubbings is still uncertain, and this is also a focus of our future work.

Abbreviations

CNN	Convolutional neural network
RNN	Recurrent neural network
GNN	Graph neural network
OFCD	Oracle font classification
GRU	Gated recurrent unit
GAN	Generative adversarial network
FFN	Feed-forward neural network
GeLU	Gaussian error linear unit
GAP	Global average pooling
mAP	Mean average precision

Acknowledgements

This work is supported by the Open Projects Program of Ministry of Education Key Laboratory for Intelligent Analysis and Security Governance of Ethnic Languages, Minzu University of China(No:202302), the Science and Technology Research Program of Chongqing Municipal Education Commission under Project(No: KJZD-K202200203), Natural Science Foundation of Chongqing CSTC(No:cstc2020jcyj-msxmX0876).

Author contributions

JY designed the study, conducted the experiments and discussions, and mainly wrote the article; SC provided overall guidance and supervision of the study and proposed an optimized protocol; BM provided experimental related datasets and assisted in article calibration; YM assisted in the query and sorting of the literature; WZ helped in the proofreading of the article; CZ assisted in the calibration of this article. All authors reviewed the manuscript.

Funding

None.

Availability of data and materials

The datasets used or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 20 August 2023 Accepted: 2 January 2024
Published online: 29 January 2024

References

- Guo Z, Zhou Z, Liu B, Li L, Jiao Q, Huang C, Zhang J. An improved neural network model based on inception-v3 for oracle bone inscription character recognition. *Sci Program.* 2022;2022:1–8.
- Zhang C, Wang B, Chen K, Zong R, Mo B-f, Men Y, Almanidis G, Chen S, Zhang X. Data-Driven Oracle Bone Rejoining: A Dataset and Practical Self-Supervised Learning Scheme. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022;pp. 4482–4492.
- Gao F, Zhang J, Liu Y, Han Y. Image translation for oracle bone character interpretation. *Symmetry.* 2022;14:743.
- Gao W, Chen S, Zhang C, Mo B, Liu X. OBM-CNN: a new double-stream convolutional neural network for shield pattern segmentation in ancient oracle bones. *Appl Intell.* 2022;52:12241–57.
- Wang M, Deng W, Liu C-L. Unsupervised structure-texture separation network for oracle character recognition. *IEEE Trans Image Process.* 2022;31:3137–50.
- Wenjun Z, Benpeng S, Ruiqi F, Xihua P, Shanxiong C. EA-GAN: restoration of text in ancient Chinese books based on an example attention generative adversarial network. *Herit Sci.* 2023;11:42.
- Pan H, Chen S, Xiong H. A high-dimensional feature selection method based on modified gray wolf optimization. *Appl Soft Comput.* 2023;135: 110031.
- Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans Neural Networks.* 1994;5:157–66.
- Huang T. Classification and chronology of oracle bone inscriptions from the yin ruins. China: China Science Press; 1991. (in Chinese).
- Liu Y. Compilation and research of oracle bone inscriptions with the font named wuming. Beijing: Jindun Publishing Company; 2014. (in Chinese).
- Liu F. Compilation and research of yin xu village south series oracle bone inscriptions. Shanghai: Ancient Books Publishing House; 2014. (in Chinese).
- Mo B. Summary of the research on the font style of oracle bone inscriptions in yin ruins. *Chin Calligr.* 2019;23:178–83 (in Chinese).
- Wang Z, Yang J, Jin H, Shechtman E, Agarwala A, Brandt J, Huang TS. DeepFont: Identify Your Font from An Image. In: Proceedings of the 23rd ACM International Conference on Multimedia, 2015;pp. 451–459.
- Zhang Y-K, Zhang H, Liu Y-G, Yang Q, Liu C-L. Oracle Character recognition by nearest neighbor classification with deep metric learning. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), 2019;pp. 309–314.
- He S, Schomaker L. FragNet: writer identification using deep fragment networks. *IEEE Trans Inf Forensics Secur.* 2020;15:3013–22.
- He S, Schomaker L. GR-RNN: global-context residual recurrent neural networks for writer identification. *Pattern Recogn.* 2021;117: 107975.
- Srivastava A, Chanda S, Pal U. Exploiting multi-scale fusion, spatial attention and patch interaction techniques for text-independent writer identification. 2021. arXiv preprint [arXiv:2111.10605](https://arxiv.org/abs/2111.10605).
- Mohammadian M, Maleki N, Olsson T, Ahlgren F. Persis: A persian font recognition pipeline using convolutional neural networks. In: 2022 12th International Conference on Computer and Knowledge Engineering (ICKE), 2022;pp. 196–204. <https://doi.org/10.1109/ICKE57176.2022.9960037>.
- Wang Z-R, Du J. Fast writer adaptation with style extractor network for handwritten text recognition. *Neural Netw.* 2022;147:42–52. <https://doi.org/10.1016/j.neunet.2021.12.002>.
- Chahi A, El merabet Y, Ruichek Y, Touahni R. Writerinet: a multi-path deep CNN for offline text-independent writer identification. *Int J Doc Anal Recog.* 2022;26(2):89–107. <https://doi.org/10.1007/s10032-022-00418-3>.
- Yu Q, Yang Y, Liu F, Song Y-Z, Xiang T, Hospedales TM. Sketch-a-Net: a deep neural network that beats humans. *Int J Comput Vis.* 2017;122:411–25.
- Liu G. Oracle-Bone inscription recognition based on deep convolutional neural network. *J Comput.* 2018;13:1442–50.
- Huang S, Wang H, Liu Y, Shi X, Jin L. Obc306: A large-scale oracle bone character recognition dataset. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), 2019;pp. 681–688. <https://doi.org/10.1109/ICDAR.2019.00114>.
- Li K, Batjargal B, Maeda A. A prototypical network-based approach for low-resource font typeface feature extraction and utilization. *Data.* 2021;6:134.
- Zhang Y-K, Zhang H, Liu Y-G, Yang Q, Liu C-L. Oracle character recognition by nearest neighbor classification with deep metric learning. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), 2019;pp. 309–314.
- Li J, Wang Q-F, Zhang R, Huang K. Mix-up augmentation for oracle character recognition with imbalanced data distribution. In: 2021 International Conference on Document Analysis and Recognition (ICDAR), 2021;pp. 237–251.
- Li J, Wang Q-F, Huang K, Yang X, Zhang R, Goulermas JY. Towards better long-tailed oracle character recognition with adversarial data augmentation. *Pattern Recogn.* 2023;140: 109534.
- He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016;pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>.
- Wang Y, Sun Y, Liu Z, Sarma SE, Bronstein MM, Solomon JM. Dynamic graph CNN for learning on point clouds. *ACM Transact Graphics (TOG).* 2019;38:12.
- Han K, Wang Y, Guo J, Tang Y, Wu E. Vision gnn: an image is worth graph of nodes. *Adv Neural Inf Process Syst.* 2022;35:8291–303.
- Loshchilov I, Hutter F. Decoupled weight decay regularization. In: Proceedings of the 36th International Conference on Machine Learning; 2019.
- Xu K, Hu W, Leskovec J, Jegelka S. How powerful are graph neural networks? In: International Conference on Learning Representations; 2019.
- Hamilton W, Ying R, Leskovec J. Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems, 2017;pp. 1025–1035.
- Li G, Müller M, Thabet A, Ghanem B. Deepgcns: Can gcns go as deep as cnns? In: The IEEE International Conference on Computer Vision (ICCV); 2019.
- Otsu N. A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern.* 1979;9(1):62–6. <https://doi.org/10.1109/TSMC.1979.4310076>.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.