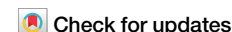


<https://doi.org/10.1038/s40494-025-01813-9>

HUNet: hierarchical universal network for multi-type ancient Chinese character recognition

Zhaojiang Wang^{1,2}, Chu Zhang², Qing Lang², Lianwen Jin³ & Hengnian Qi²

Ancient Chinese characters preserved in ancient books, calligraphy paintings, stones, and bronzes rubbings exhibit diverse morphological structures distinct from modern scripts, requiring specialized paleographic expertise for accurate recognition. However, few experts can recognize multiple character types simultaneously. This study constructed the Multi-Type Ancient Chinese Character Recognition (MTACCR) dataset and proposed HUNet, a Hierarchical Universal Network that employs a parameter-sharing architecture to achieve efficient recognition of diverse ancient character types through multi-stage feature extraction and fusion. The experimental results show that compared to other efficient models evaluated, the HUNet series achieves higher recognition performance with similar parameter counts (or maintains lower parameter counts at equivalent recognition performance) while sustaining high computational throughput. In addition, after training on MTACCR, HUNet's Top-1 accuracy on the MACR test set improved by 9.64%. Therefore, HUNet can be easily deployed on different devices and platforms for efficiently recognizing multiple types of ancient Chinese characters.

Chinese characters, as one of the oldest and most widely distributed writing systems in the world, have played a pivotal role in the development of Chinese civilization and exerted a profound influence on global culture. Over approximately 6000 years of evolution, Chinese characters have transitioned from pictographic forms such as Oracle Bone Script and Jinwen to the Great Seal Script, Small Seal Script, Clerical Script, Cursive Script, Running Script, and Regular Script, reflecting the impact of cultural, social, and technological advancements¹. These eight script types are regarded as the most representative forms of ancient Chinese characters^{2–6}. Today, diverse types of ancient Chinese characters with distinct structural and stylistic features are preserved in various mediums, including ancient books, calligraphy paintings, stone inscriptions, and bronze rubbings. Accurately recognizing and deciphering these characters not only facilitates the restoration and preservation of ancient documents and artworks but also enhances our understanding of ancient Chinese culture, philosophy, history, and language.

With the advancement of deep learning technologies, Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) have been widely applied to the recognition of various Chinese character styles and scripts. However, existing studies often focus on recognizing ancient Chinese characters from specific scripts or document types, such as Oracle Bone Script recognition^{7–10}, bamboo and silk manuscript recognition¹¹, and ancient book recognition¹²

(primarily focusing on Traditional Chinese characters). While single-script or single-document-type recognition allows for optimization tailored to the specific structural and stylistic features of the target script, it inherently suffers from significant limitations. Ancient Chinese characters exhibit substantial variations across different historical periods and regions, making it challenging for a single model to adapt to all types. For instance, Oracle Bone Script and Seal Script differ markedly in structure and form. When encountering new scripts or document types, it is often necessary to collect additional data and retrain the model, which increases costs and limits the scalability of the recognition system. Moreover, single-script recognition systems fail to meet the diverse demands of real-world applications, particularly in scenarios requiring the processing of multiple scripts and document types, thereby significantly restricting their applicability. For example, the renowned Chinese paintings *Dwelling in the Fuchun Mountains* and *Night Revels of Han Xizai* (Fig. 1) contain numerous inscriptions and seals written in different scripts. Accurately recognizing these inscriptions and seals not only aids in understanding the historical inheritance and evolution of the artworks but also reveals how historical figures or artists appreciated and evaluated these works, thereby enriching our understanding of their cultural context. Although some researchers have attempted to use CNNs for calligraphy character recognition^{13,14}, these studies are

¹College of Humanities, Shanghai Normal University, Shanghai, China. ²College of Information Engineering, Huzhou University, Huzhou, Zhejiang, China. ³School of Electronics and Information Engineering, South China University of Technology, Guangzhou, China. ✉ e-mail: qihengnian@foxmail.com

Fig. 1 | The inscriptions and seals in the famous Chinese paintings “The Dwelling in the Fuchun Mountains” and “The Night Banquet of Han Xizai”. These inscriptions and seals are written in different types of Chinese characters. Accurately identifying the inscriptions and seals in these artworks helps in better understanding the historical context, the artist’s creative intent, and the cultural significance embedded in the paintings. Images are adapted from Wikimedia Commons: *The Dwelling in the Fuchun Mountains* and *The Night Banquet of Han Xizai*.



Fig. 2 | Different types of ancient Chinese characters in stone tablets, plaques, couplets, and banners. They are widely found in famous mountains, temples, Taoist temples, public and private halls, and places of interest, and are written in a variety of types of Chinese characters, including seal script, clerical script, and cursive script.



typically limited by small-scale datasets, which restrict the number of recognizable characters.

Beyond calligraphy and ancient texts, multiple types of ancient Chinese characters are also widely found in stone tablets, plaques, couplets, and banners located in famous mountains, temples, Taoist temples, public and private halls, ruins, and scenic spots (Fig. 2). However, the recognition of ancient Chinese characters in natural scenes presents significant challenges due to the diversity of character types, the vast number of characters, and the complexity of scene distributions. Additional factors, such as limited data availability, large class scales, varied scenes, intricate backgrounds, high intra-class variance, and inter-class similarity, further complicate research on multi-type ancient Chinese character recognition, resulting in sub-optimal recognition performance. To address these challenges, Wang et al. constructed the Multi-Scene Ancient Chinese Character Recognition (MACR) dataset and improved recognition performance through techniques such as ensemble learning, class-center alignment, and domain adaptation feature alignment^{15,16}. While the MACR dataset provides a valuable resource for multi-scene ancient Chinese character recognition, its scale remains relatively small compared to other large-scale image datasets, limiting its potential to enhance model performance in multi-scene recognition tasks.

To address the aforementioned issues, this study constructs the Multi-Type Ancient Chinese Character Recognition (MTACCR) dataset based on the three-level character list and its appendix from the *Table of General Standard Chinese Characters* (通用规范汉字表; Tōngyòng Guīfàn Hànzì Biǎo). This table comprises 8105 Chinese characters categorized into three levels (3500 Level-1 characters, 3000 Level-2 characters, and 1605 Level-3 characters), along with the *Comparison Table of Standard, Traditional, and Variant Chinese Characters* (containing 3120 standard characters and their corresponding traditional and variant forms). The constructed dataset covers multiple types of ancient character images for 7874 Chinese characters, totaling over 9 million samples with rich diversity, including both original scanned images and segmented glyph images. The ancient character images used for dataset construction were primarily collected from ancient calligraphic character queries and retrieval websites, curated open-source

ancient character recognition datasets, and data augmentation techniques. Compared to existing open-source datasets, the MTACCR dataset achieves significant expansion and enrichment in terms of character count, types of ancient Chinese characters, dataset scale, and image diversity.

Designing a model for the recognition of multi-type ancient Chinese characters presents significant challenges due to the pronounced differences in shape structures and writing styles across character types, as well as the scale and diversity of the MTACCR dataset, which spans multiple historical periods and script styles. Addressing these challenges requires a careful balance between feature representation capability and computational efficiency. To begin with, the model must implement multi-scale feature fusion to effectively capture and unify the morphological characteristics of characters from different scripts, time periods, and stylistic variations. Furthermore, it must remain robust against structural degradation, such as missing strokes, ink bleeding, or erosion, which are common in historical documents. In addition, efficient parameterization is essential to accommodate the large number of character categories and the million-level training samples. A lightweight inference architecture is also necessary to support real-time processing with high throughput. Finally, the model must ensure hardware compatibility, allowing seamless deployment on scanners and mobile devices. Despite the stylistic divergence among ancient Chinese character types, they universally adhere to a hierarchical composition principle—primitive strokes coalesce into radicals, which subsequently form complete glyphs through spatial-configurational rules. This intrinsic structural regularity provides a unified topological prior for cross-type analysis. In light of these insights and following the principles of lightweight and efficient network design¹⁷, we propose HUNet (Hierarchical Universal Network) for the universal recognition of multi-type ancient Chinese characters.

HUNet adopts a hierarchical network architecture and improves the utilization efficiency of model parameters by focusing on specific types of features of Chinese characters at different stages. Specifically, the local stage emphasizes the local shapes and geometric features of the characters; the intermediate interaction stage focuses on key positions and strokes, such as the relationships between stroke connections and critical local structural

regions; and the global stage concentrates on the spatial structural relationships of the characters. With this design, HUNet achieves a balance between high training efficiency and inference speed while better modeling the different types of features of Chinese characters. Through comparative and ablation experiments, we validate the superior performance of HUNet on the MTACCR and MACR datasets for the task of recognizing various types of ancient Chinese characters. Experimental results demonstrate that the HUNet model outperforms several efficient CNN and ViT models on the MTACCR dataset (Fig. 3), while maintaining a low parameter count and high throughput. On the MACT test set¹⁵ (Fig. 4), the HUNet model trained on the MTACCR dataset achieves a 9.64% increase in Top-1 accuracy. HUNet achieves high recognition performance while maintaining a lightweight design, making it suitable for deployment on mainstream edge devices and mobile devices, with fast inference speed. Its highly scalable architecture allows for the integration of optimization modules at different stages to further enhance performance based on specific objectives. Additionally, experimental results on the DeepLontar¹⁸ and ALPUB_v2¹⁹ datasets (both for non-Chinese ancient script recognition) demonstrate that

HUNet can also be effectively extended to non-Chinese ancient text recognition tasks.

Methods

Constructing the MTACCR dataset

We constructed the MTACCR dataset primarily by referring to the *Table of General Standard Chinese Characters* compiled and issued by the Ministry of Education of China and the National Language Commission. The character table (<https://github.com/cdty/digital-table-of-general-standard-chinese-characters>) contains a total of 8105 characters, divided into three levels with 3500, 3000, and 1605 characters, respectively. Based on the characters included in this table, we collected and organized various types of ancient Chinese character images corresponding to each character from multiple data sources. We employed two primary methods for collecting single-character images of various types of Chinese characters. The first method involves downloading single-character images from established websites such as Han Dian (<https://www.zdic.net/>), Calligraphy Dictionary (<https://www.shufazidian.com/>), Yi Guan Calligraphy (<https://web.ygsf.com/>), and Calligraphy Masters (<https://www.sfds.cn/>). These images have been rigorously reviewed by domain experts and platform administrators, with further credibility established through long-term, high-volume user adoption and endorsement. The second method involves gathering images from publicly available datasets, including the Oracle Bone Script dataset and the Calligraphy Recognition dataset. For the MTACCR dataset construction, we primarily utilized the HWOBC⁷ dataset and two open-source calligraphy character recognition datasets (<https://github.com/zhuojg/chinese-calligraphy-dataset>, <https://github.com/kirosc/chinese-calligraphy-dataset>).

The raw Chinese character images obtained using the two aforementioned methods can be categorized into two distinct types. The first type is binarized glyph images, which feature white backgrounds and black characters generated through grayscale conversion, denoising, and binarization. Their simple structure facilitates high accuracy in Chinese character recognition tasks. The second type is color images, obtained by scanning handwritten, carved, or rubbed characters. These images depict characters on diverse materials, such as paper, bamboo, wood, stone inscriptions, or rubbings, and introduce greater complexity due to noise and varied backgrounds, making recognition more challenging. Figure 5 displays two different types of image samples corresponding to the Chinese character “國” (country) in ancient scripts from the image dataset. For each type of ancient script, the first row presents binarized black-and-white glyph images, while the second row shows the scanned raw color images.

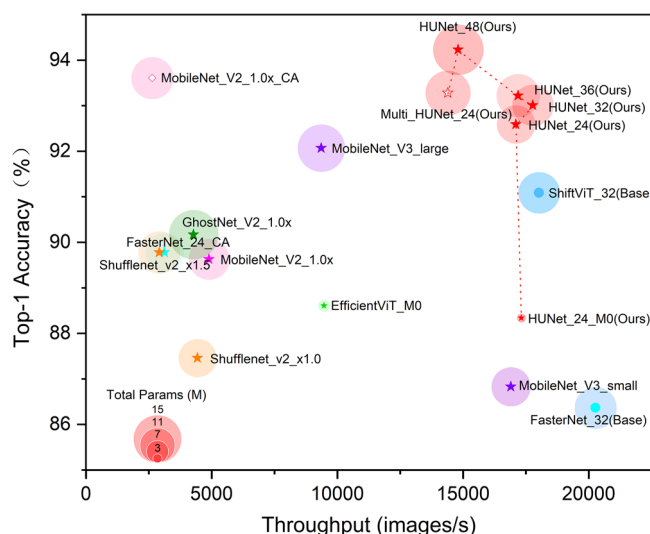


Fig. 3 | Comparison of Top-1 accuracy, speed, and number of parameters between HUNet (Ours) and other efficient CNN and ViT models. Tested on the Test_1 set of MTACCR using an Nvidia RTX 3090 GPU.

Fig. 4 | Comparison of Top-1 accuracy of different methods and models on the MACR test set. HUNet (Ours) and other efficient CNNs and ViTs are trained using the MTACCR training set.

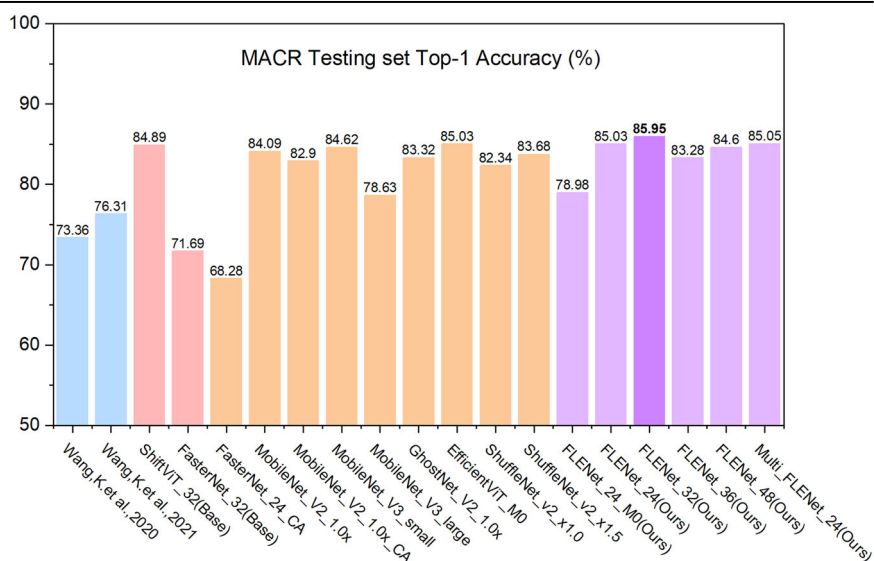




Fig. 5 | The two main types of images in the MTACCR dataset: color images and binarized glyph images. The color images are those obtained directly after scanning and segmentation. The binarized glyph images are generated through grayscale

conversion, denoising, and binarization. The individual Chinese character images in the figure are sourced from the internet (<https://web.ygsf.com/>).

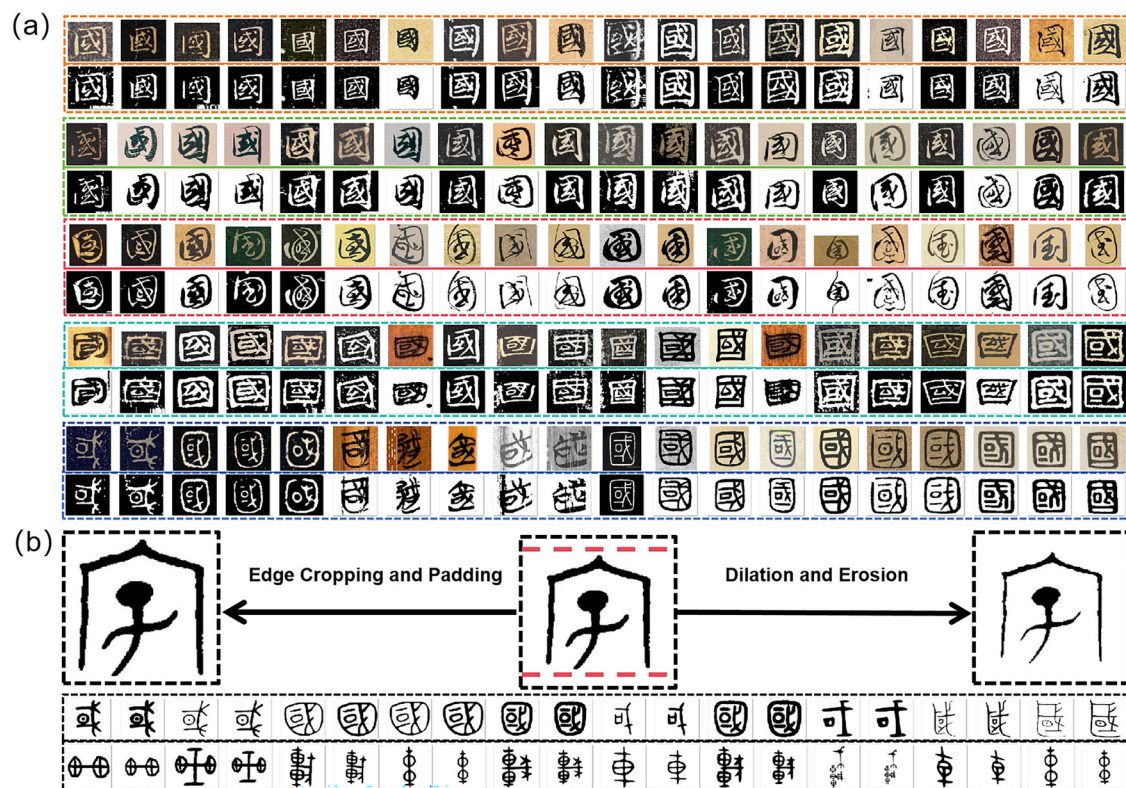


Fig. 6 | The preprocessed and offline-augmented ancient Chinese character images. **a** Shows the raw color images and the preprocessed images, while **(b)** presents the key preprocessing steps for the binarized glyph images and the results

after data augmentation processing. The individual Chinese character images in the figure are sourced from the internet (<https://web.ygsf.com/>).

We then performed additional preprocessing and offline data augmentation on both types of images. The raw color images, containing character carriers of diverse materials (e.g., paper, bamboo slips, stone artifacts, bronze ware, and oracle bones) with significant chromatic variations, first underwent adaptive threshold binarization and denoising to

eliminate background interference and normalize character color variations, followed by median filtering for noise suppression (Fig. 6a). Subsequently, all images were standardized to 256×256 pixels through aspect-ratio-preserving scaling with edge padding. For the ancient seal script characters (primarily small seal script and pre-Qin ancient scripts) that

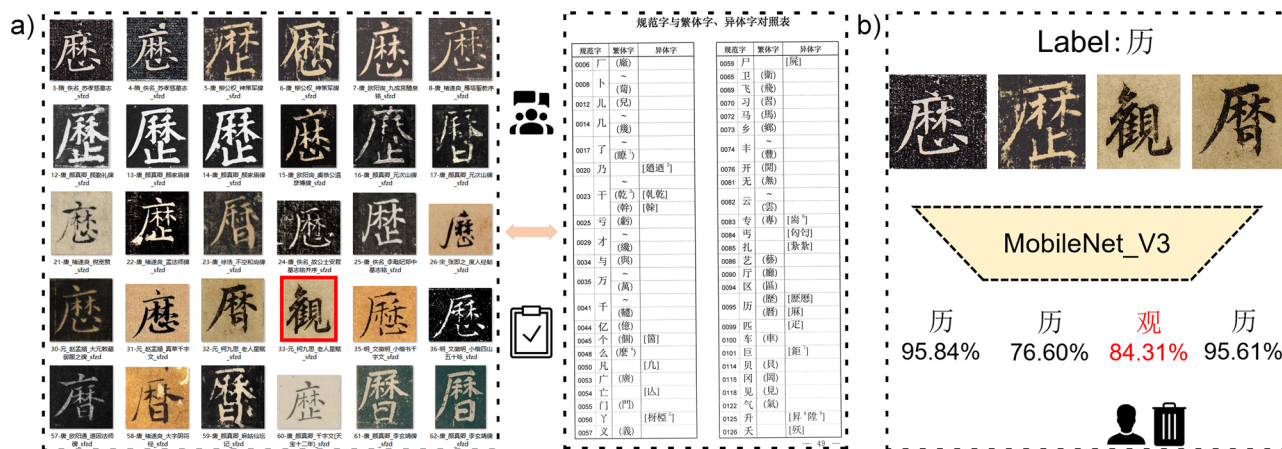


Fig. 7 | The two noise data filtering workflows: a manual screening, and b model-based automated screening. The individual Chinese character images in the figure are sourced from the internet (<https://www.shufazidian.com/>).



Fig. 8 | After balanced sampling, different types of Chinese character samples for the character “国” (country) in Test_1. The file name abbreviations ks (kǎishū), xs (xíngshū), cs (cǎoshū), ls (lìshū), and zs (zhuànshū) represent regular script (楷书),

running script (行书), cursive script (草书), clerical script (隶书), and seal script (篆书) respectively. The individual Chinese character images in the figure are sourced from the internet (<https://www.sdfs.cn/>).

exhibited limited data volume and diversity, we specifically implemented morphological augmentation techniques (Fig. 6b). These included erosion and dilation operations to modulate stroke thickness, along with edge cropping and padding to alter character spatial distribution—effectively enhancing data diversity while preserving essential morphological features. This pipeline ensures consistent data representations for model training while improving seal script generalization.

The two types of image data collected from different sources contain noisy samples where certain images are clearly mismatched with their labels. During dataset construction, we addressed noisy samples in the two types of image data through manual and automated filtering approaches. First, human reviewers manually eliminated obvious labeling errors by cross-referencing the Table of General Standard Chinese Characters. Second, we trained a MobileNet_V3 model on the verified data to automatically detect mismatches. Using a confidence threshold of 0.8, the model flagged high-confidence (>0.8) but incorrectly labeled samples for secondary verification before removal. Figure 7 illustrates this dual-stage noise filtering workflow, which ensured data quality while preserving valid character variations. These samples were further verified before removal. Figure 7 illustrates the workflow of the two noise-filtering strategies.

To thoroughly evaluate the generalization performance of the model, the MTACCR dataset is divided into training and test sets drawn from distinct data sources. The images from publicly available datasets and those collected from [Han Dian](#), [Calligraphy Dictionary](#), and [Yi Guan Calligraphy](#) websites are of relatively high quality, making them suitable for use as base data for training and data augmentation. To mitigate data distribution differences, the color images from these sources are preprocessed and

converted to binarized black-and-white images. On the other hand, images sourced from the [Calligraphy Master](#) website exhibit a greater diversity in terms of image size, type, and quality. Some of these images suffer from blurring, watermarks, wear, and erosion, presenting a unique challenge. However, this variety allows for a more comprehensive assessment of the model's generalization ability. Consequently, the images from the [Calligraphy Master](#) website are used for model validation and testing.

To address the inherent imbalance in sample size and character type distribution within the raw dataset from the Calligraphy Master website, this study employs a multi-round balanced sampling strategy to construct test sets, replacing conventional single-test-set approaches. This methodology effectively mitigates data skewness by generating multiple balanced subsets, enabling comprehensive evaluation of model generalization performance across different data distributions while avoiding potential assessment bias inherent to single test sets. The implementation involves five rounds of systematic independent sampling from the source test data, with each round ensuring 20 samples per Chinese character per font type (or all available samples if fewer than 20 exist). This stratified sampling approach yields five statistically independent and balanced test sets exhibiting controlled progressive reduction in both character variety and sample size. Figure 8 displays representative samples of the Chinese character “国” (meaning nation or country) from the first test set. Figure 9a, b illustrate the sample quantity distribution across characters in both the MTACCR training and test sets using stacked area charts. Notably, the training set exhibits a pronounced long-tail distribution, while the five balanced test sets demonstrate significantly reduced imbalance factors. Figure 9c, d visualize the proportion

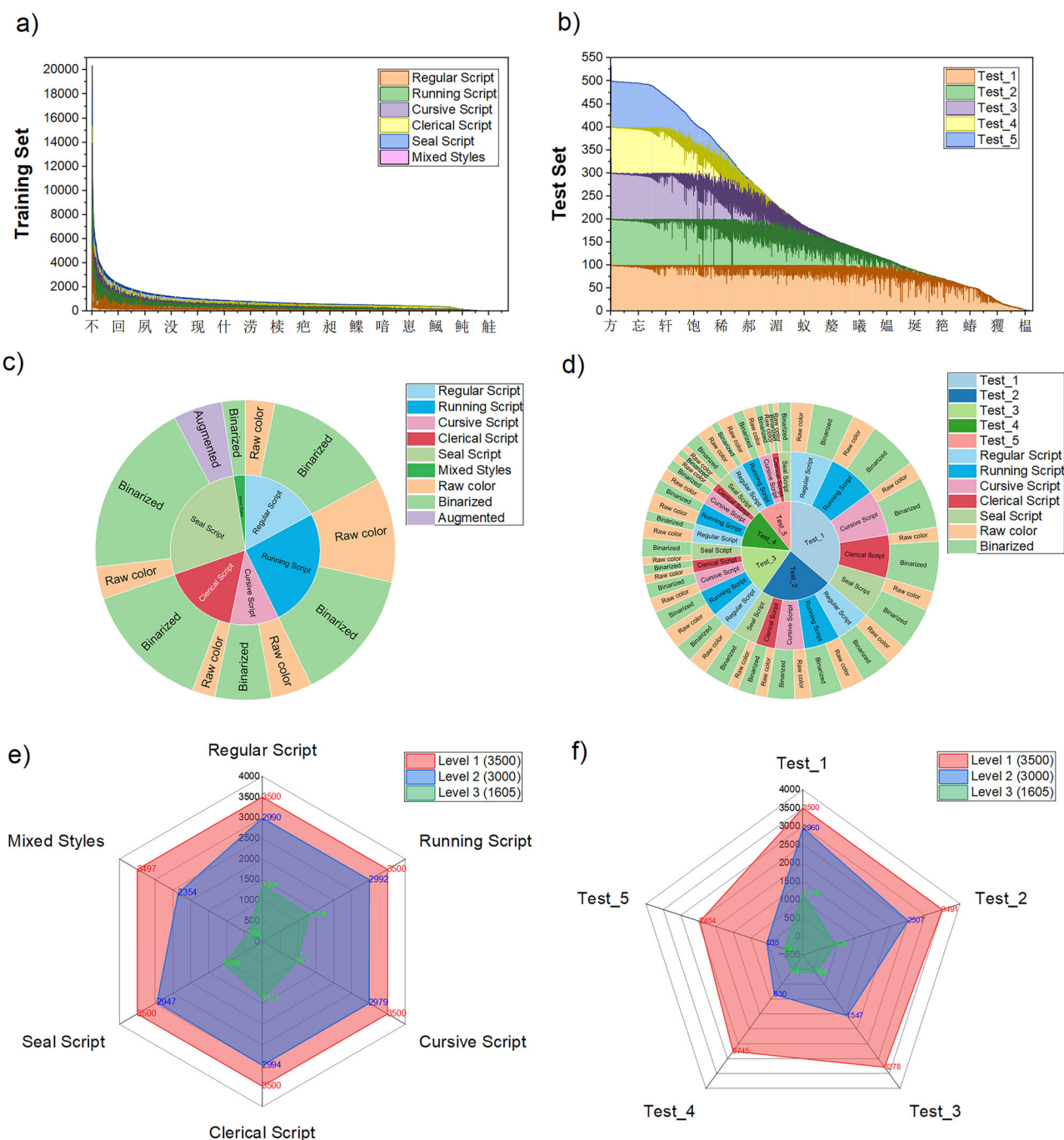


Fig. 9 | Detailed statistics of training and test sets in the MTACCR dataset. **a, b** Sorted sample size distribution of characters in training and test sets. **c, d** Proportion of two image types in training and test sets (Raw color: binarized raw color images). **e, f** Character count distribution across three levels of Chinese characters in the dataset.

of different font types and image types in the training and test sets using sunburst charts. Figure 9e, f display the number of characters from the three levels of the *Table of General Standard Chinese Characters* included in the training and test sets. The “Mixed Styles” in the training set refers to a subset of mixed image samples that were not categorized by script type. Table 1 summarizes the MTACCR dataset, including character counts, sample sizes, class imbalance metrics (max/min ratio, mean, standard deviation), and data sources.

GridNoise data augmentation method

To enhance the model’s ability to resist noise, this study employs a GridMask-inspired data augmentation method²⁰ on the binarized glyph images in the MTACCR training set. This method is referred to as

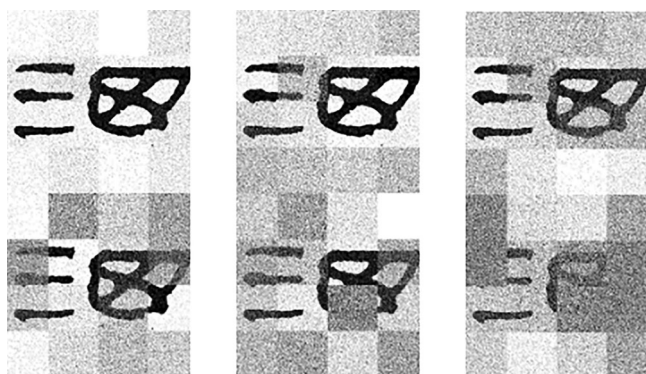
GridNoise augmentation. GridNoise simulates the erosion and wear of Chinese characters in natural environments. It divides the character image into grid regions and randomly adds speckle and salt-and-pepper noise. This augmentation method effectively enhances the model’s robustness to noise and interference. Figure 10 illustrates images after applying noise augmentation with varying noise levels to the grid regions of a Chinese character. GridNoise is applied during model training with a certain probability.

Macro network architecture

The proposed HUNet consists of three main components. The first component employs the FasterNet Block²¹, based on partial convolution (PConv), for local low-level feature extraction. The second component uses

Table 1 | Division and statistics of the training and test sets in the MTACCR dataset

MTACCR dataset	Category	Data volume	Imbalance factor			Source
			max/min	mean	Std deviation	
Training Set	7874	7,423,379	10,178	942	1069	Other data sources
Test Sets	Test_1	7638	600,844	100	78	Calligraphy Masters
	Test_2	6432	390,337	100	60	
	Test_3	4926	275,788	100	55	
	Test_4	3629	214,612	100	59	
	Test_5	3023	177,632	100	58	

**Fig. 10 | Single-character images generated using the GridNoise data augmentation method.** The probability of adding noise to an image increases sequentially.

an efficient channel attention module (ECA)²² for feature interaction and enhancement. The third component integrates the Shift Block²³, which uses partial shift operations to extract global high-level features and semantic information. Additionally, a ResLT Block²⁴ is incorporated after the HUNet backbone as an optimization module. This module enhances HUNet's feature extraction capabilities and balances recognition performance across different categories. We named the model to which we added the ResLT Block Multi_HUNet.

The overall architecture of HUNet is presented in Fig. 11. Similar to the network architecture of Next-ViT²⁵, HUNet adopts a hierarchical structure with four stages for feature extraction, where the feature map sizes at each stage are $\{1/2, 1/4, 1/8, 1/16\}$ of the input image size. The input image size is fixed at 128×128 . Each stage begins with an embedding or merging layer, which uses a $\text{Conv2D}(2, 2)$ operation for spatial downsampling and channel expansion. With each successive stage, the feature map size is halved, while the number of channels doubles. The first two stages focus on local feature extraction and consist of multiple stacked FasterNet Blocks²¹. Since the feature maps in these stages are larger, fewer layers are stacked to reduce memory access and computational overhead. The last two stages focus on global feature extraction and consist of multiple stacked Shift Blocks²³. An ECA module²² is inserted as an interaction layer between the second and third stages to enhance feature interaction. To improve the overall recognition performance and optimize the performance on both head and tail classes, a ResLT Block²⁴ is inserted as an independent optimization neck module between the backbone network and the classification head. This hierarchical and modular design enables adaptive adjustment of embedding dimensions and layer depths according to task complexity and deployment requirements. The overall architecture of HUNet is simple and clear, while offering scalability for model size adjustments and extensions based on dataset scale and characteristics. At each stage of HUNet, feature extraction for MTACCR images is performed by stacking the basic building blocks. The values of $\{L1, L2, L3, L4\} = \{2, 2, 6, 2\}$ represent the number of layers stacked at each stage.

Local feature extraction stage

To improve the efficiency of convolution operations while reducing parameters and computational complexity, specialized convolution operations and attention mechanisms are introduced. For local feature extraction, we adopt the FasterNet Block²¹, which combines Partial Convolution (PConv) and Pointwise Convolution (PWConv) to extract low-level local features. The upper part of Fig. 12 illustrates the working principle of PConv, where regular convolution is applied only to a subset of input channels for spatial feature extraction, leaving other channels unchanged. Compared to standard convolution, PConv reduces redundant computations and memory accesses, enabling faster inference. While PConv has lower FLOPs (floating-point operations) than regular convolution, it achieves higher FLOPs (floating-point operations per second) compared to DWConv and GConv.

To fully utilize all channel information, PWConv is applied after PConv to enable effective channel-wise feature fusion, with BatchNorm (BN) and ReLU activation layers inserted between the PWConv layers. A residual connection is introduced to reuse input features, forming an inverted residual block that expands the number of channels in intermediate layers, with the channel expansion rate set to 4 in our experiments. The lower part of Fig. 12 shows the detailed structure of the FasterNet block. In the early stages of the model, FasterNet Blocks are stacked to efficiently extract local features. After this stage, the complete contour features of Chinese characters are captured.

Intermediate feature interaction stage

In the local stage, the model extracts low-level local features from the image, where the spatial resolution of the feature maps progressively decreases while the channel dimension increases. Recent studies have demonstrated that both channel and spatial attention mechanisms can significantly enhance the performance of convolutional neural networks. However, complex attention modules increase the model's computational complexity and training time. ECA is a simple yet efficient channel attention module that performs cross-channel interaction using 1D convolutions, avoiding dimensionality reduction while significantly improving performance and reducing computational overhead²². To enhance the feature extraction capabilities of the HUNet model, we introduce an ECA module (with a kernel size of 3) between the local and global stages to strengthen high-quality features and suppress low-quality ones.

Global feature extraction stage

Although Convolutional Neural Networks (CNNs) exhibit exceptional performance in capturing local features, their ability to capture global features is relatively weak. As a result, in tasks requiring global context information, CNNs may not achieve optimal performance. However, during the global stage, we still hope the model can learn higher-level semantic features, such as the spatial relationships and combinatory patterns of components within a Chinese character. Inspired by ShiftViT²³, we utilize the computationally efficient, parameter-free Partial Shift operation (Fig. 13 Upper) and Shift Block (Fig. 13 Lower) for spatial feature extraction and semantic modeling. This simple shift operation enhances feature representation without adding parameters or computational overhead. Moreover, the

Fig. 11 | Macro network architecture of HUNet. H and W denote the height and width of the input image and C denotes the number of channels of the feature map. (l_1, l_2, l_3, l_4) denote the number of layers of different modules stacked at each stage. The ResTL Block is inserted as an optional optimization module after the backbone network of HUNet.

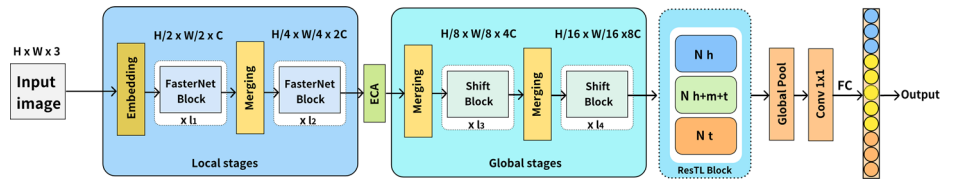


Fig. 12 | Detailed structure of partial convolution operation and FasterNet block. h , w , and c are the height, width, and number of channels of the intermediate feature maps. PConv and Conv denote the partial convolution and ordinary convolution operations. BN is the batch normalization, and ReLU is the activation function.

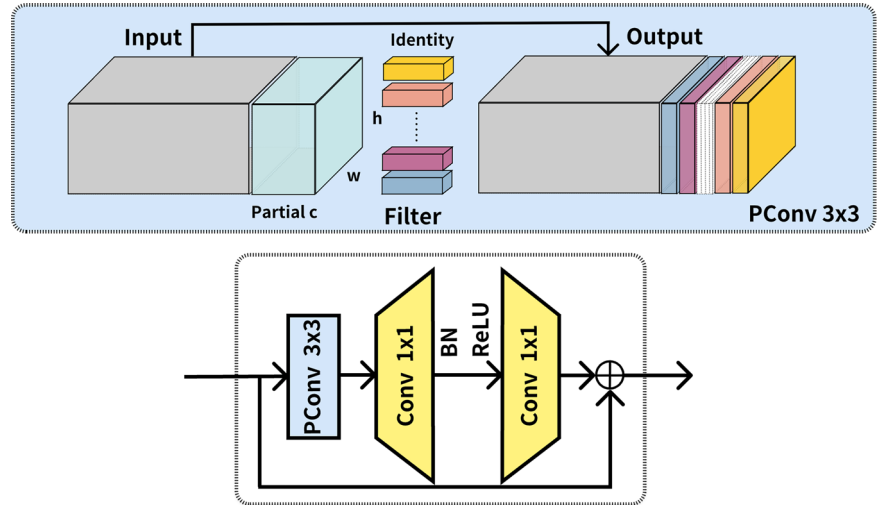
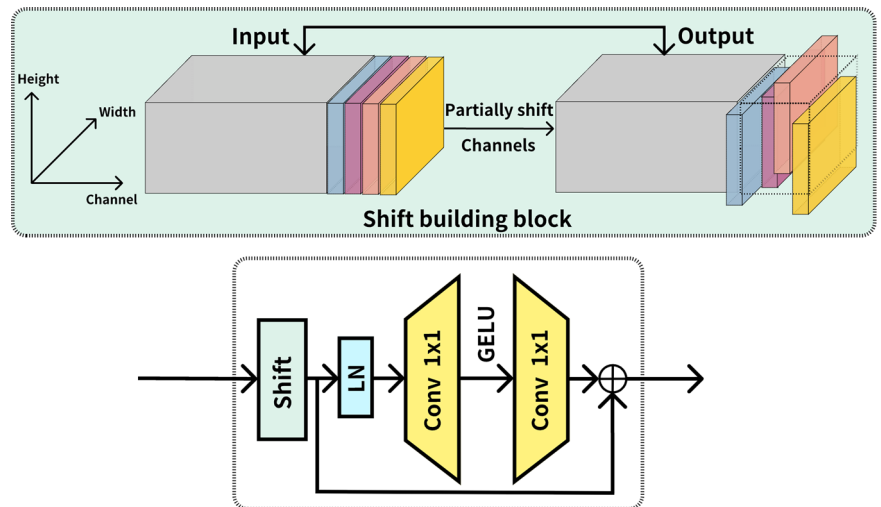


Fig. 13 | Detailed structure of the Partial Shift operation and Shift Block. Conv 1×1 is the 1×1 convolution, LN is the layer normalization, and GELU is the activation function.



Partial Shift operation indirectly helps the model learn the combinatorial relationships between components within a Chinese character.

Although CNNs possess translation invariance, they still struggle to handle cases where the feature map undergoes large shifts in arbitrary directions. Figure 14 illustrates the changes in the spatial locations of feature maps after the Shift operation. Unlike direct translation of the input image, the Partial Shift operation facilitates data augmentation within the model. This operation improves the model's robustness, particularly for scenarios involving slight positional shifts or partial occlusions in Chinese characters. For instance, when a Chinese character undergoes small spatial shifts or partial occlusions, humans can still recognize it. Similarly, the Partial Shift operation enhances the model's capacity to handle such variations.

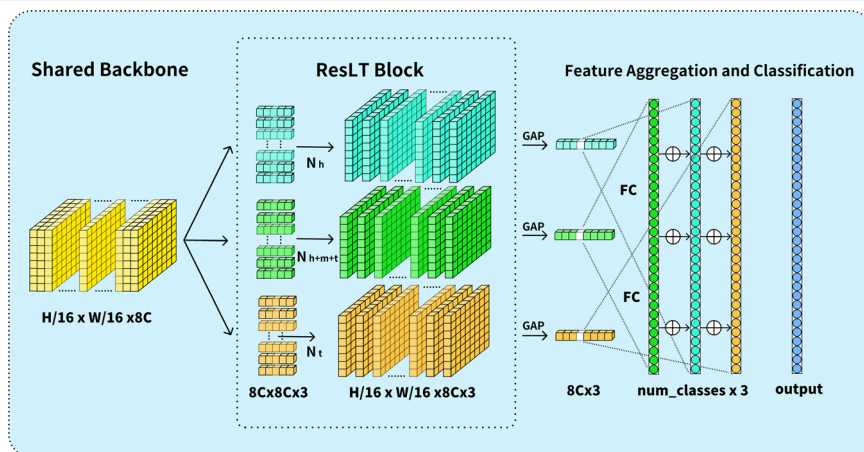
Long-Tail class balancing module

The training set of the MTACCR dataset exhibits characteristics of a long-tail distribution, while the test set is composed of multiple relatively balanced subsets obtained through repeated balanced sampling from an independent data source. Previous studies on long-tail visual recognition and imbalanced data classification have pointed out that recognition models generally perform worse on medium and tail classes compared to head classes^{26,27}. However, the model exhibits distinctly different behavior when evaluated on the MTACCR test set. The HUNet₂₄ test results show that the model's recognition performance on the middle classes is generally higher than the head and tail classes. Upon reviewing the data in the test set and the test results, we find that the head category has more diverse data samples. This diversity manifests in three key aspects: (1) complex character backgrounds;

Fig. 14 | Spatial location changes of feature maps in the global stage of HUNet_24. After the Shift operation, the intermediate feature maps are successively translated by several pixels to the right, left, down, and up.



Fig. 15 | Detailed structure of the ResLT Block. N_h , N_{h+m+t} and N_t represent three different branches. N_h is optimized using only Chinese character images of the head class, N_t is optimized using only Chinese character images of the tail class, and N_{h+m+t} is optimized using all Chinese character images. GAP stands for Global Average Pooling, and FC stands for Fully Connected layer.



(2) inclusion of multiple Chinese character types and their variants; (3) each character type contains various writing styles. On the other hand, due to the limited training data for the original tail classes (less than 20 samples), the model's recognition performance on certain tail classes is lower than that on the head and medium classes.

Inspired by the residual fusion mechanism in the ResLT Block²⁴, we add three independent branches to the HUNet head, optimized with image data from the full, head, and tail classes. Both branch losses and fusion losses jointly supervise the model during training. The final output and prediction are obtained by aggregating the outputs from these branches. The ResLT Block (Fig. 15) is integrated after the HUNet backbone to enhance recognition performance. In the head of the backbone network, group convolutions with a kernel size of 1×1 and 3 groups are used to extract features for the full class, head class, and tail class, respectively. The improved model is referred to as Multi_HUNet. During the training of Multi_HUNet, the loss function is computed as follows:

$$\mathcal{L}_{fusion} = \mathcal{J}(\mathcal{N}_{h+m+t}(X) + \mathcal{N}_h(X) + \mathcal{N}_t(X), Y) \quad (1)$$

$$\mathcal{L}_{branch} = \sum_{i \in \{h+m+t, h, t\}} \mathcal{J}(N_i(SX_i), SY_i) \quad (2)$$

$$\mathcal{L}_{all} = (1 - \alpha)\mathcal{L}_{fusion} + \alpha\mathcal{L}_{branch} \quad (3)$$

where (X, Y) denote the images and labels in a mini batch of data. (SX_{h+m+t}, SY_{h+m+t}) is equivalent to (X, Y) and contains images from all classes. (SX_h, SY_h) is a subset of (X, Y) , containing only images and labels from the head class. Similarly, (SX_t, SY_t) is a subset of (X, Y) , containing only images and labels from the tail class. $\mathcal{J}$ represents the cross-entropy loss, and α is a hyperparameter (set to 0.9 in the experiments). The fusion loss term $\mathcal{L}_{fusion}$ in Eq. 1 optimizes for all Chinese character classes by adding the outputs from branches $\mathcal{N}_h$ and $\mathcal{N}_t$ to the main branch output $\mathcal{N}_{h+m+t}$, with the fused output used for inference, where the final result is obtained by summing the outputs of the three branches. $\mathcal{L}_{branch}$ in Eq. 2 is the sum of three independent losses, each specialized for optimization of the full, head, and tail classes. $\mathcal{L}_{all}$ in Eq. 3 is the final loss, obtained by the weighted sum of $\mathcal{L}_{fusion}$ and $\mathcal{L}_{branch}$, with the hyperparameter α controlling the weight of each loss term.

Table 2 | Performance comparison of HUNet with other efficient CNNs and ViTs on the MTACCR Test_1 set

Model	Top-1 (%)	Params (M)	Feature_dim	Total depths	Throughput (images/s)		
					GPU	CPU	ONNX
ShiftViT_32(Base) ²³	91.09	12.81 (2.43)	1280	12 (2,2,6,2)	<u>18,022</u>	47.1	42.2
FasterNet_32(Base) ²¹	86.37	12.95 (2.56)	1280	12 (2,2,6,2)	20,268	51.6	141.8
FasterNet_24_CA ^{21,29}	89.78	11.92 (1.54)	1280	12 (2,2,6,2)	3124	20.7	—
MobileNet_V2_1.0x ³⁴	89.63	12.61 (2.22)	1280	11	4893	30	93.7
MobileNet_V2_1.0x_CA ^{29,34}	93.61	13.05 (2.67)	1280	11	2638	21.8	—
MobileNet_V3_small ²⁸	86.83	12.05 (1.67)	1280	11	16,904	111.8	<u>322.6</u>
MobileNet_V3_large ²⁸	92.07	14.59 (4.20)	1280	15	9361	46.5	90.8
GhostNet_V2_1.0x ³⁵	90.17	15.26 (4.88)	1280	16	4279	27.2	80.1
ShuffleNet_v2_x1.0 ³⁶	87.46	11.75 (1.38)	1280	16	4434	63.7	249
ShuffleNet_v2_x1.5 ³⁶	89.78	13.04 (2.66)	1280	16	2917	55	158.5
EfficientViT_M0 ³⁷	88.61	3.72 (2.16)	192	6 (1,2,3)	9471	49.2	330.3
HUNet_24_M0(Ours)	88.34	2.75 (1.19)	192(24 × 8)	12 (2,2,6,2)	17,318	<u>64.9</u>	117.7
HUNet_24(Ours)	92.59	11.82 (1.43)	1280	12 (2,2,6,2)	17,103	50.8	108.1
HUNet_32(Ours)	93.01	12.82 (2.43)	1280	12 (2,2,6,2)	17,777	39.1	81.4
HUNet_36(Ours)	93.22	13.42 (3.03)	1280	12 (2,2,6,2)	17,192	40.2	73.6
HUNet_48(Ours)	94.23	15.60 (5.22)	1280	12 (2,2,6,2)	14,806	32.7	53.9
Multi_HUNet_24(Ours)	<u>93.28</u>	13.89 (3.51)	1280	12 (2,2,6,2)	14,395	43.6	—

GPU Throughput and CPU Throughput are tested on the Nvidia RTX 3090 GPU and the Intel(R) Core (TM) i9-10900K CPU @ 3.70 GHz CPU, respectively. A higher Throughput indicates faster inference speed. The values in parentheses for Params represent the parameter size of the backbone network, and the numbers in the model names correspond to the number of channels C in the initial embedded feature maps.

Boldface indicates the best performance in the test set, while underlined text denotes the second-best results.

Results

Experimental settings

To comprehensively evaluate the feature extraction and representation learning capabilities of HUNet, we train the model using the standard Cross-Entropy Loss (CE-loss) and compare its performance with several state-of-the-art lightweight network architectures. The HUNet model is trained on the Training Set of the MTACCR dataset, with validation and testing performed on the Test_1 to Test_5 subsets of MTACCR (see Methods for details) and the MACR test set (see Supplementary Materials for details). The HUNet model is implemented using PyTorch 1.12.1, while the baseline models are implemented using their official open-source code to ensure fairness. All models are trained for 20 epochs on an Nvidia RTX 3090 GPU, utilizing the AdamW optimizer and a Cosine learning rate scheduler. The batch size is set to 256, and input images are resized to either 128×128 or 224×224 pixels. The initial learning rate is set to 1×10^{-3} , with betas values of (0.9, 0.999) and a weight decay of 1×10^{-2} . To enhance model robustness, we employ data augmentation techniques, including RandomRotation within a range of $(-10^\circ, 10^\circ)$, GaussianBlur with kernel sizes ranging from 5 to 41, and GridNoise (see Methods for details), each applied with a probability of 0.1. Model complexity metrics, including the number of parameters (Params), floating point operations (Flops), and memory read/write (MemR+W) are calculated using the torchstat package. Additionally, we evaluate the throughput of the model on different hardware platforms. GPU throughput measurements were conducted using an NVIDIA GeForce RTX 3090, using the largest feasible batch size on the Test_1 set. For the CPU and ONNX runtime, throughput is measured on an Intel(R) Core(TM) i9-10900K CPU @ 3.70 GHz processor, with a batch size of 1 (averaged over 5000 runs). To further validate the generalization capability of HUNet, we test the trained model on the MACR test set¹⁵. Notably, the MTACCR training set consists of binarized black-and-white images, while the MACR test set contains colored images. To ensure consistency in data distribution, we also apply an adaptive threshold binarization algorithm (OTSU) to the MACR test set before resizing the images to either 128×128 or 224×224 pixels for testing.

Comparison experiment results on MTACCR and MACR

To evaluate the performance of the proposed HUNet model, we conducted comprehensive comparison experiments with state-of-the-art efficient CNN and ViT models on multiple test sets of the MTACCR dataset and the MACR test set. The experimental results, summarized in Tables 2, 3, demonstrate that HUNet achieves an optimal balance between accuracy and inference speed across various evaluation settings.

The HUNet architecture incorporates two core modules: the FasterNet Block and the Shift Block, derived from FasterNet²¹ and ShiftViT²³, respectively. Compared to FasterNet_32 and ShiftViT_32, HUNet_32 achieves a 6.73% and 1.92% improvement in Top-1 accuracy on the MTACCR Test_1 set, while maintaining comparable parameter counts and throughput. Furthermore, under identical parameter scales and testing conditions, the HUNet series demonstrates a consistent performance advantage over other evaluated efficient CNN and ViT models while maintaining computational efficiency. Specifically, HUNet achieves superior Top-1 accuracy compared to models with comparable parameter counts. Among them, HUNet_24/32/36 exhibits GPU throughput performance second only to two benchmark models (ShiftViT_32 and FasterNet_32). Notably, high GPU throughput typically correlates with reduced training duration and accelerated iteration speed, particularly when handling large-scale training datasets. Multi_HUNet is a model that adds the ResLT Block²⁴ optimization module after the HUNet backbone network, further improving its recognition performance compared to the original model.

Among the compared models, MobileNet_V3_small²⁸ exhibits superior CPU and ONNX model inference speed for models using the same input resolution. Notably, the integration of the Coordinate Attention (CA) module²⁹ into MobileNet_V2_1.0x_CA improves Top-1 accuracy by 3.98% with only a marginal increase in parameters. Similarly, FasterNet_24_CA achieves a 3.41% improvement in Top-1 accuracy over FasterNet_32, despite having fewer parameters. The CA module enhances feature extraction by performing attention operations along both channel and spatial dimensions. However, this improvement comes at the cost of

Table 3 | Performance comparison of HUNet with other Efficient CNNs and ViTs on all MTACCR test sets and the MACR test set

Model	Test_1	Test_2	Test_3	Test_4	Test_5	ZS	LS	XS	CS	KS	MACR
ShiftViT_32(Base) ²³	91.09	89.55	88.31	87.63	87.27	88.07	89.44	91.57	83.97	93.01	84.89
FasterNet_32(Base) ²¹	86.37	84.08	82.41	81.46	81.17	83.13	83.84	86.53	78.20	87.14	71.69
FasterNet_24_CA ^{21,22}	89.78	87.57	85.86	84.81	84.39	85.23	86.52	88.52	82.55	92.83	68.28
MobileNet_V2_1.0x ³⁴	89.63	87.94	86.78	86.09	85.79	85.8	86.96	90.76	83.44	91.04	84.09
MobileNet_V2_1.0x_CA ^{22,34}	93.61	92.57	91.76	91.39	91.36	<u>89.59</u>	<u>92.55</u>	94.58	88.95	95.84	82.90
MobileNet_V3_small ²⁸	86.83	84.50	82.37	81.08	80.45	83.59	85.48	85.36	77.14	88.57	84.62
MobileNet_V3_large ²⁸	92.07	90.53	89.23	88.65	88.36	88.44	89.98	92.47	86.09	93.83	78.63
GhostNet_V2_1.0x ³⁵	90.17	88.90	87.80	87.14	86.83	83.75	89.45	91.39	83.82	93.74	83.32
EfficientViT_M0 ³⁷	88.61	86.62	85.14	84.36	83.89	84.87	87.73	88.07	80.90	90.06	85.03
ShuffleNet_v2_x1.0 ³⁶	87.46	85.19	83.05	81.73	81.14	84.27	86.77	86.61	78.95	87.19	82.34
ShuffleNet_v2_x1.5 ³⁶	89.78	88.14	86.79	85.91	85.6	87.38	88.78	89.84	82.67	90.53	83.68
HUNet_24_M0(Ours)	88.34	87	85.79	85.11	84.73	82.34	87.46	94.16	79.99	93.35	78.98
HUNet_24(Ours)	92.59	91.81	91.37	91.11	91.07	89.1	91.58	94.16	87.70	95.43	85.03
HUNet_32(Ours)	92.55	92.02	91.66	91.53	91.59	88.18	91.56	94.64	88.22	96.05	85.95
HUNet_36(Ours)	93.22	<u>92.57</u>	<u>92.13</u>	<u>91.89</u>	<u>91.87</u>	89.58	92.30	94.63	<u>89.16</u>	<u>95.95</u>	83.28
HUNet_48(Ours)	94.23	93.71	93.41	93.25	93.26	90.74	93.55	95.67	90.80	96.79	84.60
Multi_HUNet_24(Ours)	93.28	92.38	91.76	91.5	91.36	89.47	91.43	<u>94.64</u>	89.08	95.84	<u>85.05</u>

In this table, ZS (篆书), LS (隶书), XS (行书), CS (草书), and KS (楷书) represent Seal Script (including Oracle Script, Bronze Script, Greater Seal Script, and Small Seal Script), Clerical Script, Running Script, Cursive Script, and Regular Script, respectively.

Boldface indicates the best performance in the test set, while underlined text denotes the second-best results.

Table 4 | Performance comparison of four HUNet_24 variants on the MTACCR test_1 set

Model	Top-1 (%)	Params (M)	Flops (M)	MemR + W (MB)	Input	Feature_dim	Total depths	Throughput (images/s)		
								GPU	CPU	ONNX
HUNet_24(Ours)	92.59	11.82 (1.43)	261	74.59	128	1280	12 (2,2,6,2)	17,103	50.8	108.1
HUNet_24_all_ECA	92.51	11.82 (1.44)	259	65.02	128	1280	12 (2,2,6,2)	14,148	48.2	71.3
HUNet_24_no_ECA	91.96	11.81 (1.43)	258	65.02	128	1280	12 (2,2,6,2)	14,235	52.1	76.4
HUNet_24_local_ECA	<u>92.55</u>	11.81 (1.43)	258	65.02	128	1280	12 (2,2,6,2)	13,742	50.6	75.6
HUNet_24_global_ECA	92.15	11.81 (1.43)	258	65.02	128	1280	12 (2,2,6,2)	13,680	50.8	75.7

Boldface indicates the best performance in the test set, while underlined text denotes the second-best results.

reduced inference speed and potential compatibility issues during ONNX model conversion.

To further assess the generalization capability of HUNet, we evaluated its performance on MTACCR Test_2 to Test_5 and the MACR test set. As shown in Table 3, HUNet consistently maintains excellent performance across all test sets, demonstrating superior generalization and scalability compared to other efficient models. For instance, HUNet's Top-1 accuracy remains stable across the five MTACCR test sets, and increasing the model's width progressively enhances its recognition performance. Meanwhile, MobileNet_V2_1.0x equipped with the CA module demonstrates competitive advantages across five test sets and diverse Chinese character recognition tasks, indicating that this module effectively enhances the model's feature extraction capability.

Compared to the previously reported highest Top-1 accuracy of 76.31%¹⁶, most lightweight efficient networks trained on our MTACCR dataset achieve significant improvements on the MACR test set (Fig. 4). This underscores the critical role of high-quality data in supervised deep learning tasks. Notably, HUNet_32 achieves a 9.64% improvement in Top-1 accuracy without relying on complex algorithms, a result primarily attributed to the larger dataset size and effective data augmentation techniques. Among the various types of ancient Chinese characters, the recognition accuracy for cursive and seal scripts remains the lowest. This is mainly due to the inherent complexity of seal script characters and the

stroke connections and reductions in cursive script, which pose significant challenges for recognition.

Ablation experiment results

To maintain the simplicity of the HUNet architecture while minimizing computational complexity, we integrated a lightweight Efficient Channel Attention (ECA) module²² exclusively in the intermediate stages of local and global feature extraction. The ECA module enhances channel interactions within feature maps and suppresses noise through weighted operations. To evaluate the effectiveness of the ECA module, we designed four variants of HUNet_24: HUNet_24_all_ECA (ECA applied in both local and global stages), HUNet_24_no_ECA (no ECA), HUNet_24_local_ECA (ECA applied only in the local stage), and HUNet_24_global_ECA (ECA applied only in the global stage).

All variants were trained under identical settings as HUNet_24. The kernel size of the ECA module was adaptively determined based on the channel count of the feature maps, following the configuration in ECA-Net²². Table 4 summarizes the details and experimental results for HUNet_24 and its variants. Due to the low parameter and computational overhead of the ECA module, the variants exhibited negligible changes in Params, FLOPs, and MemR+W, with throughput remaining stable across GPU, CPU, and ONNX platforms. In terms of performance, incorporating the ECA module improved recognition accuracy on the Test_1 set.

Table 5 | Performance comparison of four HUNet_24 network depth expansion on the MTACCR test_1 set

Model	Top-1 (%)	Params (M)	Flops (M)	MemR + W (MB)	Input	Feature_dim	Total depths	Throughput (images/s)		
								GPU	CPU	ONNX
HUNet_24	92.59	11.82 (1.43)	258.62	65.02	128	1280	12 (2,2,6,2)	17,103	50.8	108.1
HUNet_24	92.59	11.87 (1.48)	339.42	77.02	128	1280	16 (4,4,6,2)	13,796	34.8	71.7
HUNet_24	93.38	12.56 (2.18)	1000	117.55	224	1280	16 (2,2,8,4)	11,556	17.5	52.5

Boldface indicates the best performance in the test set, while underlined text denotes the second-best results.

Table 6 | Performance of HUNet on the EVOBC dataset (total character categories and sample size in parentheses)

Dataset	Type	Categories	Images	Model	Top-1 (%)
EVOBC	Oracle Bone Characters	7061 (13714)	186,600 (229,170)	HUNet_24	76.31
	Bronze Inscriptions			HUNet_32	77.16
	Small Seal Script			HUNet_48	77.34
	Spring and Autumn period Characters				
	Warring States period Characters				
	Clerical Script				

However, excessive use of the ECA module did not yield significant performance gains. The most notable improvements were observed when the ECA module was applied to the local feature extraction stage or between the local and global stages. These results highlight the critical role of high-quality local feature extraction in enhancing the model's overall performance.

Increasing the depth of a network enables it to learn more complex patterns from data, thereby improving its ability to handle intricate tasks. Conversely, increasing the width of the network enhances its capacity to capture diverse features at each layer, boosting its representational power and recognition accuracy. In the HUNet series models, the primary distinction lies in their width, which refers to the number of channels in the feature maps at each stage. Experimental results demonstrate that increasing the model's width can improve performance to a certain extent. Beyond increasing width, expanding the depth of the model is another effective strategy to enhance its capacity. To compare the impact of these two approaches, we applied two depth expansion strategies to HUNet_24. The first strategy increases the depth of the Local stage, while the second strategy increases the depth of the Global stage. Table 5 provides detailed information and experimental results for HUNet_24 and its depth-expanded variants. In the global stage of HUNet, each additional Shift Block shifts the feature map by one pixel in different directions. The results reveal that expanding the depth of the local stage increased Params, FLOPs, and Throughput but did not significantly improve performance. In contrast, expanding the depth of the global stage, coupled with an increase in input image size, enhanced recognition performance. However, this improvement came at the cost of increased FLOPs and reduced Throughput.

Analysis of incorrectly recognized samples

Although the HUNet architecture achieves state-of-the-art performance on multiple test sets of the MTACCR dataset, a significant number of samples remain misclassified. To further enhance the recognition performance of multi-type ancient Chinese characters and identify the bottlenecks limiting model performance, we conducted a detailed analysis of approximately 40,000 misclassified samples from HUNet_24 on the Test_1 set. Our analysis reveals that most misclassifications can be categorized into four primary types: special types of Chinese characters, blur and occlusion, shape similarity, and label noise. Representative examples of misclassified samples across these categories are provided in the Supplementary Materials.

Among the misclassifications, a major portion arises from special types of Chinese characters and their variant forms. In the MTACCR test set, there exist certain special types of characters, primarily scripts from the Spring and Autumn period and the Warring States period. The political independence of various feudal states during this era led to regional differentiation in writing systems, resulting in significant graphical variations for the same character across different territories. Moreover, these scripts were inscribed on diverse mediums such as bronze vessels, coins, bamboo slips, wooden tablets, silk fabrics, and jade or pottery artifacts. Due to their relatively short usage duration and limited circulation, historical preservation is sparse, leaving scarce extant glyphic materials. Consequently, the MTACCR training set exhibits a notable insufficiency in samples from the Spring and Autumn/Warring States scripts. This combination of diverse variant forms, pronounced glyphic discrepancies, heterogeneous's writing substrates, and scarce training samples collectively constitutes the primary reason for the model's lower recognition accuracy in this domain. EVOBC³⁰ is a dataset designed for studying the evolution of oracle bone scripts, incorporating authoritative texts and websites covering early ancient Chinese characters across six historical periods. We directly employed the HUNet model trained on the MTACCR dataset to evaluate 7,061 characters (totaling 186,600 samples) from EVOBC that are included in the *Table of General Standard Chinese Characters*. Test results in Table 6 demonstrate that the current HUNet model still has considerable room for improvement in recognizing early ancient characters with substantial morphological variations. On one hand, further expansion of the dataset is required; on the other hand, targeted optimization of the network architecture should be implemented based on enhancing the model's holistic feature extraction capabilities.

Image blur and occlusion also constitute major causes of recognition failures. Low image resolution, quality degradation due to suboptimal capturing devices, compression artifacts during transmission, or storage-induced corruption can severely compromise the structural integrity of character images. In such cases, even human recognition becomes challenging. Aging and physical wear further exacerbate the problem, leading to broken strokes, uneven stroke thickness, and blurred stroke contours. Without sufficient contextual information, convolutional neural networks and other advanced models struggle to accurately identify these characters. Unlike models that rely on dedicated noise-handling modules, HUNet maintains architectural simplicity and inference efficiency by adopting an implicit robustness strategy. However, under extreme noise and severe character degradation, its recognition performance may degrade significantly due to the lack of explicit structural restoration mechanisms and contextual modeling components.

Character shape similarity presents another significant challenge. Some Chinese characters, either within the same font or across different fonts, exhibit extremely similar shapes, leading to frequent misclassifications, particularly in the absence of contextual clues. This similarity poses a significant challenge for recognition, particularly in the absence of contextual cues. In the feature space, such characters tend to have closer distances and blurry decision boundaries, especially when the model is trained solely using cross-entropy loss without additional regularization or feature refinement techniques.

Finally, the MTACCR dataset contains instances where simplified and Traditional Chinese character forms are mixed, introducing label noise. For example, a Traditional character image may be incorrectly annotated with

Table 7 | Non-Chinese ancient script recognition dataset and model performance comparison

Dataset	Character Classes	Total Samples	Imbalance Factor (max/min)	Data Split	Macro-Averaged Accuracy (%)	
					HUNet_24	MobileNet_V3_small
DeepLontar	55	135,555	2943	6:4	91.67	71.09
ALPUB_v2	24	205,797	44.36	6:4	79.15	66.26

Boldface indicates the best performance in the test set, while underlined text denotes the second-best results.

Table 8 | Performance Comparison of HUNet Variants (Top-1 Accuracy on Test_1, Parameters, FLOPs, Memory Usage, and Storage Requirements)

Model	Top-1 (%)	Params (M)	Flops (M)	MemR + W (MB)	ONNX model storage space (MB)
HUNet_24_M0	88.34	2.75 (1.19)	252	40	10.7
HUNet_24	92.59	11.82 (1.43)	261	74.59	45.3
HUNet_32	93.01	12.82 (2.43)	454	88.17	49.1
HUNet_36	93.22	13.42 (3.03)	571	95.34	51.4
HUNet_48	94.23	15.60 (5.22)	1010	118.33	59.7

its Simplified counterpart, or vice versa. This inconsistency in labeling has adversely affected the model’s recognition performance, as it introduces ambiguity during training and evaluation. In addition, there are some samples with incorrect labels in the test set.

Recognition of non-Chinese ancient scripts

To validate the generalization capability of the HUNet model in cross-script ancient character recognition tasks, this study further employs two Non-Chinese ancient script benchmarks—the DeepLontar¹⁸ (a dataset for handwritten Balinese character detection and syllable recognition on Lontar manuscripts) and ALPUB_v2¹⁹ (a dataset for ancient Greek character recognition)—for model training and testing. The model was trained with a batch size of 512 over 100 epochs using 128 × 128 input images, employing the AdamW optimizer ($\text{lr} = 1\text{e-}4$, $\beta = (0.9, 0.99)$) with a two-phase learning rate schedule: 10-epoch linear warmup ($0.01 \rightarrow 1 \times \text{base rate}$) followed by 90-epoch cosine decay (to $1\text{e-}7$). Table 7 presents dataset characteristics and model performance comparisons (macro-averaged accuracy) between HUNet_24 and MobileNet_V3_small. In experiments on DeepLontar and ALPUB_v2, HUNet_24 outperformed MobileNet_V3_small in accuracy, especially on the more imbalanced DeepLontar dataset. This suggests that HUNet has the potential to be extended to non-Chinese ancient script recognition tasks, demonstrating certain competitive advantages.

Hardware requirements and deployment environment testing

One of the key objectives in designing and developing HUNet is to successfully deploy it on edge devices and mobile platforms. Based on the detailed metrics provided in Table 8, we further analyze the hardware requirements for deploying different HUNet variants. The lightest variant (HUNet_24_M0) requires only 2.75 M parameters and 252 M FLOPs, with a memory footprint of 40MB during inference. This makes it particularly suitable for resource-constrained devices like ARM Cortex-M7 MCUs. HUNet_24 (45.3MB ONNX) and HUNet_32 (49.1MB ONNX) can be deployed on mid-tier mobile SoCs (e.g., Qualcomm Snapdragon 7-series) and achieve real-time inference by leveraging GPU/DSP acceleration through frameworks such as TensorFlow Lite or ONNX Runtime. HUNet_48 requires 1010 M FLOPs and is therefore recommended for deployment on flagship devices (e.g., Snapdragon 8+ Gen 1 equipped with Hexagon 7th-generation DSP). We have deployed the HUNet_24 (ONNX model) on a server equipped with 4 vCPUs (AMD EPYC 7K62 48-Core Processor) and 8GB RAM. Using the LOCUST framework (simulating 100 peak users with a ramp-up rate of 10 users/second), we conducted a 10 min stress test. During this test, the CPU utilization reached nearly 100%. The test generated a total

of 14,167 requests, with an average response time of 2.69 s and a throughput of 24 requests per second.

Discussion

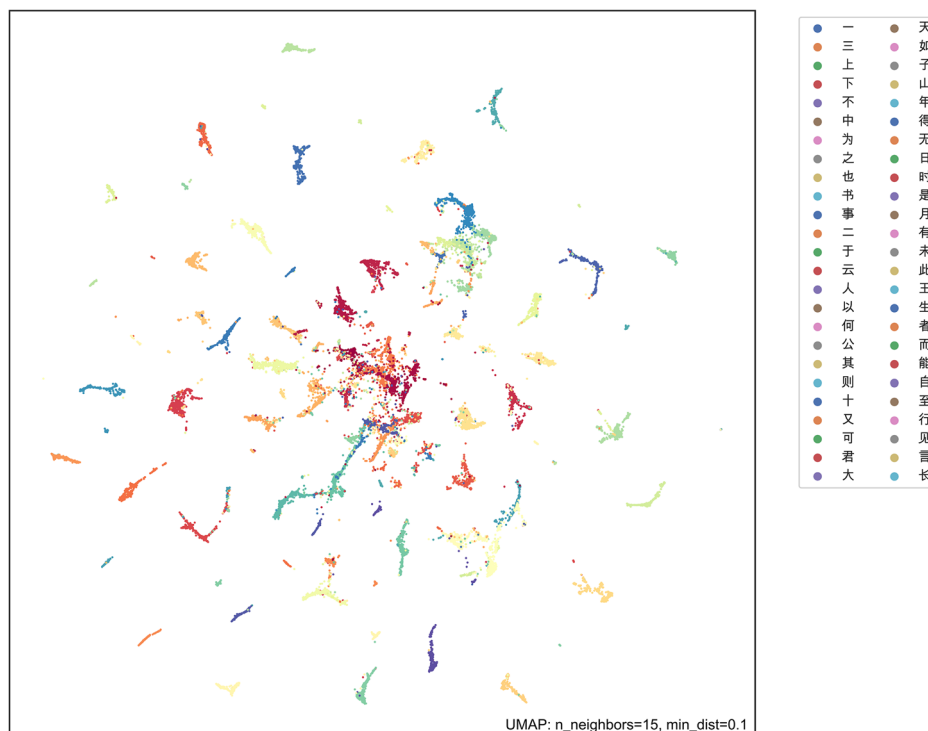
To address the practical need for recognizing diverse types of ancient Chinese characters, we construct the MTACCR dataset. It consists of one training set and five test sets, with 7,874 different characters covering most Level-1 and Level-2 characters from the *Universal Standard Chinese Character Table* and some Level-3 characters. MTACCR contains nearly 9 million annotated samples across different ancient script types, exhibiting significant diversity in both visual styles and writing mediums. Statistical analysis reveals that the training set follows a typical long-tailed distribution, while the five test sets maintain balanced sample quantities and script-type distributions for fair evaluation. Compared to existing open-source datasets, MTACCR expands the coverage of ancient character types and sample sizes, yet there remains room for improvement: (1) The MTACCR dataset currently covers a relatively limited number of Chinese characters. Notably, some characters found in oracle bone script, bronze script, and bamboo slip script fall outside the scope of the *Table of General Standard Chinese Characters*. Therefore, a potential improvement could involve consulting broader-coverage standards (such as the *Standardized Glyph Table for Ancient Book Printing*) as well as specialized academic resources like oracle bone script dictionaries and bronze script dictionaries to expand the character count and coverage range of the MTACCR dataset. (2) The MTACCR dataset can be expanded by integrating the latest open-source single-category ancient script recognition datasets, such as *HUST-OBC*³¹ (2024) and *DeepJiandu*³² (2025). (3) For character categories or variant forms with limited training samples, data augmentation techniques can be further employed to effectively expand the dataset. (4) The dataset is restricted to character-level annotations only, lacking information at the line or paragraph level. To improve this, future efforts could involve incorporating line and paragraph-level annotations, which would provide richer context for understanding the relationships between characters within larger text units.

We propose HUNet, a hierarchical universal network for recognizing multiple types of ancient Chinese characters. It uses a hierarchical multi-stage feature extraction strategy to capture ancient characters’ multi-scale structures and morphological features. HUNet has a concise architecture, balancing accuracy and speed. Experiments on MTACCR show that, compared to other efficient models evaluated, the HUNet series achieves higher recognition performance at comparable parameter levels (or comparable recognition performance with lower parameter counts) while maintaining high GPU throughput. Incorporating the ResLT module²⁴ into HUNet can further enhance the overall recognition performance of the

Fig. 16 | Comparison of saliency maps generated by FullGrad algorithm on different types of ancient Chinese characters using HUNet. Original character images are sourced from the internet (<https://web.ygsf.com/>).



Fig. 17 | UMAP feature downscaling visualization results for 50 Chinese character test samples. The 50 Chinese characters with the largest test sample size are selected.



model with only a slight increase in parameters. Additionally, tests on the MACR dataset show HUNet has strong feature representation and learning abilities. Figure 16 displays the saliency maps generated by the HUNet model using the FullGrad algorithm on ancient Chinese character images. The visualization demonstrates that the model not only focuses on the character shapes but also assigns higher attention to critical components and structural positions within the glyphs. These key regions and components play a vital role in ancient character recognition, particularly in cases where different characters share similar structural patterns (e.g., the bronze script forms of “王” (king) and “玉” (jade), or the small seal script forms of “月” (moon) and “肉” (meat/flesh) in Fig. 16b). Although HUNet demonstrates certain recognition performance advantages, its CPU and ONNX throughput do not show significant superiority. This indicates that there is still room for improvement in both the architectural design and practical implementation of the model. Specifically, the CPU inference speed of HUNet could be further improved by incorporating more efficient convolutional operators or adopting optimized implementations of existing modules.

Finally, we conducted a detailed analysis of the model’s misclassified samples and categorized them into four main types. These categories reveal the key challenges that persist in multi-type ancient Chinese character recognition tasks and provide clear directions for future improvements. Given the prevalent issues of noise and character degradation in ancient character images, the recognition

performance and robustness of HUNet could be further enhanced by incorporating dedicated, efficient feature extraction modules and noise-handling mechanisms. Additionally, in the experiments, we trained all models using the traditional cross-entropy loss function. After visualizing the 1280-dimensional features extracted by HUNet₂₄ using the UMAP dimensionality reduction algorithm³³ (Fig. 17), we observed that the feature distributions of some Chinese characters were relatively scattered or formed multiple class centers. This phenomenon, to some extent, degraded the recognition performance. This is due to significant structural differences in the glyphs of certain Chinese characters across different historical periods and types. Therefore, it is necessary to further improve the model’s recognition performance for special types of Chinese characters, variant forms, and shape-similar characters by optimizing the HUNet network architecture and training strategies.

In conclusion, this study constructs the MTACCR dataset, which encompasses various types of ancient Chinese characters. The dataset supports both character recognition and retrieval tasks, and can also be utilized for training multimodal large models to enhance their capability in ancient character recognition. Based on the structural characteristics of different ancient character types and the distributional properties of the dataset, we propose a lightweight network, HUNet. HUNet employs a multi-stage, hierarchical feature extraction mechanism to effectively model both the local details and global structures of ancient characters. It achieves

competitive recognition performance while maintaining a compact architecture with low parameter complexity. Although there is still significant room for improvement in its recognition capability, it strikes a good balance between accuracy and inference efficiency, demonstrating potential and feasibility for practical application deployment.

Data availability

The datasets generated and analyzed during the current study are available from the corresponding author upon reasonable request.

Code availability

The underlying code for this study is available in the GitHub repository and can be accessed via this link <https://github.com/1602353775/HUNet>.

Received: 11 February 2025; Accepted: 22 May 2025;

Published online: 19 June 2025

References

- Zhang, M. The ‘Chinese wisdom’ in Chinese characters: an analysis of the evolution of Chinese characters and their contemporary value. *Hanzi Cult.* **04**, 96–98 (2023).
- Liu, J. On the fonts and types of Chinese characters. *J. Shaoxing Univ. (Philos. Soc. Sci.)* **01**, 77–82 (2007).
- Zhang, H. A critical analysis of the development origins of Chinese character fonts. *J. Inn. Mong. Norm. Univ. (Philos. Soc. Sci.)* **03**, 92–94 (2007).
- Zhu, L. A discussion on the evolution of ancient Chinese character forms. *J. Mudanjiang Norm. Univ. (Philos. Soc. Sci.)* **01**, 95–96 (2015).
- Ma, X. A study on the evolution and development of Chinese character forms. *Hanzi Cult.* **11**, 1–3 (2022).
- Yang, Q. A review of Qiu Xigui’s “Outline of philology”: also discussing the evolution of Chinese character forms and simplified characters. *Hanzi Cult.* **24**, 4–5 (2022).
- Li, B. et al. HWOBC-A handwriting oracle bone character recognition database. *J. Phys. Conf. Ser.* **1651**, 012050 (2020).
- Wang, M., Deng, W. & Liu, C.-L. Unsupervised structure-texture separation network for oracle character recognition. *IEEE Trans. Image Process.* **31**, 3137–3150 (2022).
- Li, J. et al. Towards better long-tailed oracle character recognition with adversarial data augmentation. *Pattern Recognit.* **140**, 109534 (2023).
- Wang, M., Deng, W. & Su, S. Oracle character recognition using unsupervised discriminative consistency network. *Pattern Recognit.* **148**, 110180 (2024).
- Li, S., Zhou, C. & Wang, K. WA-Net: Wavelet integrated attention network for silk and bamboo character recognition. *Eng. Appl. Artif. Intell.* **140**, 109674 (2025).
- Xu, Y. et al. CASIA-AHCDB: a large-scale Chinese ancient handwritten characters database. In *Proc. 2019 International Conference on Document Analysis and Recognition (ICDAR)*, 793–798 (2019).
- Miao, Y., Liang, L., Ji, Y., Li, Z. & Li, G. Research on Chinese ancient characters image recognition method based on adaptive receptive field. *Soft Comput.* **26**, 8273–8282 (2022).
- Sun, J., Li, P. & Wu, X. Handwritten ancient Chinese character recognition algorithm based on improved Inception-ResNet and attention mechanism. In *Proc. 2022 IEEE 2nd International Conference on Software Engineering and Artificial Intelligence (SEAI)*, 31–35 (2022).
- Wang, K., Yi, Y., Liu, J., Lu, L. & Song, Y. Multi-scene ancient Chinese text recognition. *Neurocomput.* **377**, 64–72 (2020).
- Wang, K., Yi, Y., Tang, Z. & Peng, J. Multi-scene ancient Chinese text recognition with deep coupled alignments. *Appl. Soft Comput.* **108**, 107475 (2021).
- Zhang, J. et al. Rethinking mobile block for efficient attention-based models. In *Proc. 2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 1389–1400 (2023).
- Siahaan, D., Sutramiani, N. P., Suciati, N., Duija, I. N. & Darma, I. W. A. S. DeepLontar dataset for handwritten Balinese character detection and syllable recognition on Lontar manuscript. *Sci. Data* **9**, 761 (2022).
- Swindall, M. I. et al. Exploring learning approaches for ancient Greek character recognition with citizen science data. In *Proc. 2021 IEEE 17th International Conference on eScience (eScience)*, 128–137 (2021).
- Chen, P., Liu, S., Zhao, H. & Jia, J. GridMask data augmentation. Preprint at <https://arxiv.org/abs/2001.04086> (2020).
- Chen, J. et al. Run, don’t walk: chasing higher FLOPS for faster neural networks. In *Proc. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 12021–12031 (2023).
- Wang, Q. et al. ECA-Net: Efficient channel attention for deep convolutional neural networks. In *Proc. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11531–11539 (2020).
- Wang, G., Zhao, Y., Tang, C., Luo, C. & Zeng, W. When shift operation meets vision transformer: an extremely simple alternative to attention mechanism. *Proc. AAAI Conf. Artif. Intell.* **36**, 2423–2430 (2022).
- Cui, J., Liu, S., Tian, Z., Zhong, Z. & Jia, J. ResLT: residual learning for long-tailed recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**, 3695–3706 (2023).
- Li, J. et al. Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios. Preprint at <https://arxiv.org/abs/2207.05501> (2022).
- Liu, Z. et al. Open long-tailed recognition in a dynamic world. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**, 1836–1851 (2022).
- Zhang, Y., Kang, B., Hooi, B., Yan, S. & Feng, J. Deep long-tailed learning: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**, 10795–10816 (2023).
- Howard, A. G. et al. Searching for MobileNetV3. In *Proc. 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 1314–1324 (2019).
- Hou, Q., Zhou, D. & Feng, J. Coordinate attention for efficient mobile network design. In *Proc. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13708–13717 (2021).
- Guan, H. et al. An open dataset for the evolution of oracle bone characters: Evobc. Preprint at <https://arxiv.org/abs/2401.12467> (2024).
- Wang, P. et al. An open dataset for oracle bone character recognition and decipherment. *Sci. Data* **11**, 976 (2024).
- Liu, Y. et al. DeepJiandu dataset for character detection and recognition on Jiandu manuscript. *Sci. Data* **12**, 398 (2025).
- McInnes, L., Healy, J., Saul, N. & Grossberger, L. UMAP: Uniform Manifold Approximation and Projection. *J. Open Source Softw.* **3**, 861 (2018).
- Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A. & Chen, L.-C. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proc. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4510–4520 (2018).
- Tang, Y. et al. GhostNetV2: Enhance Cheap Operation with Long-Range Attention. *Adv. Neural Inf. Process. Syst.* **35**, 9969–9982 (2022).
- Ma, N., Zhang, X., Zheng, H.-T. & Sun, J. ShuffleNet V2: Practical guidelines for efficient CNN architecture design. *Comput. Vision-ECCV 2018*, 122–138 (2018).
- Liu, X. et al. EfficientViT: Memory efficient vision transformer with cascaded group attention. In *Proc. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14420–14430 (2023).

Acknowledgements

This study was funded by Chaomi S&T Company Cooperation Project (Grant number: HK16003) and Zhejiang Key R&D Plan (Grant number: 2017C03047). The funders played no role in study design, data collection, analysis and interpretation of data, or the writing of this manuscript.

Author contributions

Z.W. compiled the experimental dataset, designed the methodology, conducted core experiments, and drafted the initial manuscript. H.Q. conceptualized, designed, and led the study, defined the research objectives and methodology, oversaw the overall framework and research direction, and critically revised the manuscript. Q.L. and C.Z. reviewed and revised the manuscript. L.J. conducted a comprehensive evaluation of the manuscript's content, structure, and logical flow, reviewed the drafts iteratively, and provided critical suggestions for improvement. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s40494-025-01813-9>.

Correspondence and requests for materials should be addressed to Hengnian Qi.

Reprints and permissions information is available at <http://www.nature.com/reprints>

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

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2025