npj | heritage science

Article

https://doi.org/10.1038/s40494-025-02066-2

Dual-stream multi-layer cross encoding network for texture analysis of architectural heritage elements

Check for updates

Xiaochun Xu[1], Bin Li[2] ✉ & Q.M.Jonathan Wu[3]

Texture provides valuable insights into building materials, structure, style, and historical context. However, traditional deep learning features struggle to address architectural textures due to complex inter-class similarities and intra-class variations. To overcome these challenges, this paper proposes a Dual-stream Multi-layer Cross Encoding Network (DMCE-Net). DMCE-Net treats deep feature maps from different layers as experts, each focusing on specific texture attributes. It includes two complementary encoding streams: the intra-layer encoding stream efficiently captures diverse texture perspectives from individual layers through multi-attribute joint encoding, while the inter-layer encoding stream facilitates mutual interaction and knowledge integration across layers using a cross-layer binary encoding mechanism. By leveraging collaborative interactions between both streams, DMCE-Net effectively models and represents complex texture attributes of architectural heritage elements. Extensive experimental evaluations on architectural heritage datasets and three texture databases demonstrate that DMCE-Net achieves superior performance compared to existing deep learning methods and handcrafted features, providing reliable texture representations.

Architectural heritage[1] is a vital component of human culture and history, and its preservation, particularly through digital means, is of utmost importance. These structures embody profound cultural, historical, and artistic values. By applying texture analysis to architectural images, we can not only provide quantitative support for academic research but also offer the public a deeper and more nuanced understanding of their historical significance[2]. Texture[3] is a crucial visual feature of building surfaces, offering key insights into a structure's materials, design, style, and even its historical context. For instance, surface features such as bricks, stone, wood, and carvings—commonly found in ancient buildings—reflect distinct craftsmanship and techniques. Texture[4] analysis not only aids in extracting detailed surface information but also helps in identifying various architectural elements and their stylistic characteristics. Moreover, this process contributes to the intelligent processing of large-scale datasets. Related intelligent applications[5,6], such as visual question-answering systems[7,8] and cross-modal fusion[9] can assist non-experts in accurately classifying and recognizing architectural heritage. Ultimately, these technologies provide more refined and effective tools for the preservation of cultural heritage.

As illustrated in Fig. 1, visual texture[10] refers to the structured patterns on an object's surface, which may appear either regular or random. These patterns result from the complex interplay of surface details, inherent structures, lighting conditions, and material properties, conveying vital information about the object's composition, shape, and structural characteristics. The surfaces of architectural heritage structures are particularly rich in intricate texture details. Robust texture analysis methods are crucial for effectively addressing the fundamental challenges posed by inter-class similarity and intra-class variability in the classification of architectural heritage elements (AHE). However, designing effective texture features remains a significant challenge in image analysis, as texture images often exhibit a wide variety of complex, both regular and irregular, patterns. Moreover, these textures are highly sensitive to changes in imaging conditions, including illumination, scale, rotation, viewpoint, and blur. To tackle these issues, a wide range of texture recognition methods have been proposed and applied across diverse domains in recent years. Broadly, these methods can be divided into two main categories: those based on handcrafted features, developed with deep expertise from domain specialists, and those that utilize deep features learned automatically by deep neural networks.

The binary pattern method[11,12] captures local texture features by analysing the difference between neighboring pixels and the central pixel. As one of the most successful handcrafted texture descriptors, it offers several advantages, including a simple theoretical implementation, low feature dimensionality, and robustness to monotonic illumination changes. In recent years, deep neural networks have exhibited remarkable performance

[1]School of Computer and Big Data, Minjiang University, Fuzhou, China. [2]College of Computer and Information Science, Fujian Agriculture and Forestry University, Fuzhou, China. [3]University of Windsor, Windsor, Canada. ✉e-mail: libin@fafu.edu.cn

Fig. 1 | Texture images of architectural heritage elements. This figure presents representative texture image samples, illustrating the visual diversity and material complexity of architectural heritage elements.

advantages in image classification. Consequently, researchers have increasingly explored their applications in texture attribute analysis within the domain of texture classification. Specifically, Zhai et al.[13] proposed a deep structure-revealed network (DSR-Net) which leveraged the spatial dependencies captured by the deep network as a structural representation for texture recognition. Khwaja et al.[14] proposed a LM trained multi-layer perceptron neural network structure optimization algorithms for defective texture classification. A multi-scale boosting feature encoding network[15] was proposed to address the challenge of scale variation and improve texture recognition accuracy. Literature[16] analysed the impact of global pooling measurements and developed RANKGP-CNN, demonstrating that layers at different depths could provide high-quality texture information. Wu et al.[17] designed a trimmed texture convolutional neural network for automatic texture exemplar extraction across a variety of texture objects, scales, and regularities under diverse conditions.

Recently, researchers have started integrating the strengths of hand-crafted features into neural networks, leading to the development of a series of advanced hybrid frameworks. Lee et al.[18] combined the advantages of hand-crafted feature preprocessing and shallow neural networks to develop ILBPSDNet for real-time character recognition. Reference[19] analysed deep convolutional neural networks to extract precise and robust latent features based on entropy for texture representation and plant species identification. Florindo et al.[20] proposed ELMP-Net, which constructs a two-layer mapping and uses the learned parameters to transform the original image, achieving competitive performance compared to state-of-the-art approaches. The author also designed fractal pooling[21], which uses fractal dimension of the feature map to capture more complex, multiscale, and non-linear relationships. In addition, the tailored design of modules for texture attributes has further broadened the potential applications of deep networks in texture analysis. Chen et al.[22] proposed deep tracing patterns that trace features generated along the convolutional layers, achieving a highly discriminative and robust global feature representation for texture descriptors. Lyra et al.[23] employed a hierarchical application of deep filter bank modules combined with Fisher vector pooling, proposing a multilevel pooling scheme for texture classification and Brazilian plant species identification. Pavlopoulos et al.[24] designed a fuzzy neural network classifier that enhances the effectiveness of texture feature analysis in ultrasonic liver images. Saihood et al.[25] proposed a guided attention-based fusion method for lung nodule classification by leveraging multi-orientation local texture features, enabling the extraction of more fine-grained discriminative information within the nodule volume. This approach not only improves classification performance but also provides strong support for the practical needs of medical experts. Reference[26] proposed a coarse-to-fine contrastive self-supervised learning framework for pixel-level texture analysis of low-resolution remote sensing images. Reference[27] proposed a dual-branch hybrid encoding embedded network to extract diverse pathological texture features, enabling efficient classification of histopathological images. Moinuddin et al.[28] designed a texture-compensated multi-resolution convolutional neural network to prevent over-smoothing and preserve both structural and textural information. Reference[29] proposed a structure-aware infrared and visible image fusion network, which effectively mitigates the loss of texture details in the fused images. It is evident that deep network frameworks demonstrate remarkable performance advantages in the field of texture analysis, thereby offering a solid theoretical foundation for their application in related domains.

In recent years, the application of texture feature analysis[30–32] has become increasingly widespread in the study of ancient relics. Andreetto et al.[33] proposed an automatic 3D modeling approach for textured cultural heritage objects to address the cost challenges associated with cultural heritage modeling. Reference[34] developed deterioration identification models for stone cultural heritage based on hyperspectral image texture features to distinguish between various types of deterioration. Earl et al.[35] presented the archaeological applications of polynomial texture mapping, a technique that allows for the recording and representation of subtle surface details. Reference[36] trained convolutional neural network (CNN) on Cypriot-built heritage to classify the architectural style of built heritage in 3D. Fan et al.[37] proposed to combine attached textual descriptions to perform intangible cultural heritage image classification. Li et al.[38] proposed LBCapsNet to extract features from images of porcelain artifact fragments, effectively handling unique textures and providing technical support for the digital preservation and restoration of cultural heritage. Clearly, texture analysis holds substantial potential in the classification of AHE. With the rapid advancements in image processing technology and deep learning, researchers can gain deeper insights into the evolution, structural characteristics, and possible damage of buildings by classifying and analysing images of architectural heritage, thereby offering a scientific foundation for conservation and restoration efforts.

This paper presents the design of a deep network framework aimed at capturing multi-view texture attribute information to address the challenges of subtle inter-class differences and significant intra-class variations in texture analysis. The proposed framework is applied to the classification of AHE. Specifically, we introduce the Dual-stream Multi-layer Cross Encoding Network (DMCE-Net), with the following key contributions:

1. To effectively address the inherent challenges of high inter-class similarity and significant intra-class variability in AHE, the proposed DMCE-Net introduces two complementary feature encoding streams: an intra-layer encoding stream and an inter-layer encoding stream.

2. The intra-layer encoding stream is dedicated to capturing multi-level texture information from the perspective of individual expert layers and achieves efficient fusion of diverse texture representations through a novel multi-attribute joint encoding strategy.
3. The inter-layer encoding stream focuses on facilitating mutual learning among texture cues across different expert layers and introduces a cross-layer binary encoding mechanism to enable effective integration of multi-level texture perspectives
4. Extensive experiments on AHE dataset and three challenging texture databases demonstrate that the proposed MCBE-Net achieves a compact and highly discriminative texture representation. Furthermore, it surpasses state-of-the-art methods in classification performance, exhibiting particularly remarkable advantages in AHE dataset.

The remainder of the paper is organized as follows: Section "Methods" introduces the related methods, including the basics of CNN, the architecture and strategy of the proposed DMCE-Net. Results are given in Section "Results". The discussion is drawn in Section "Discussion".

Methods

Review of CNN

Classical deep neural networks tackle complex problems by alternately connecting convolutional and pooling layers followed by one or more fully connected layers at the end. They learn features by training network parameters on large datasets. The core concept is to extract local features from the image using convolution operations and progressively capture higher-level, more abstract features through the cascade of multiple convolutional layers, thereby enabling effective image understanding and classification.

The convolutional layer is the core building block of a neural network. It extracts local features from the input image using convolutional kernels, followed by element-wise nonlinear activation. The activation of the ith feature map in the lth layer can be expressed as:

$$F_i^l = \sum_j g(w_{i,j}^l * F_j^{l-1} + b_i^l) \tag{1}$$

where $*$ is the convolutional operation. $F_j^{l-1} \in \mathbb{R}^{p\times q}$ is the output of $(l-1)^{th}$ layer, and it is also the input to the l^{th} layer. $w_{i,j}^l$ is the i^{th} convolutional kernel of the l^{th} layer. b_i^l is the corresponding bias. $g(g)$ is nonlinear activation function.

Then the feature map vector of the lth layer is represented as:

$$\mathbf{F}^l = g(\mathbf{w}^l * \mathbf{F}^{l-1} + \mathbf{b}^l) \tag{2}$$

where $\mathbf{w}^l$ is a set of convolutional kernels in the l^{th} layer. $\mathbf{F}^{l-1}$ is the feature map vector of the $(l-1)^{th}$ layer, and $\mathbf{b}^l$ is the corresponding bias vector of the l^{th} layer.

Typically, a pooling layer is applied after the convolutional layer. Pooling operations are a crucial local mechanism in neural networks, helping to reduce overfitting and enhance generalization. By effectively aggregating local information, they preserve the invariance of deep features to translation and slight local distortions. The pooling operation is represented as:

$$Z_i^l = P_p(F_i^l) \tag{3}$$

where Z_i^l is the i^{th} pooling feature map of the l^{th} layer. $P_p(g)$ is the type of pooling operation, such as max, average, and spatial pyramid pooling.

As illustrated in Fig. 2, CNNs extract deep features of an image in a hierarchical, layer-by-layer manner. The convolutional operations in the lower layers capture relatively simple, low-level features, while deeper layers progressively capture more abstract and semantic information. Unlike general images, texture images have distinct characteristics. On one hand, most of the recognizable features in texture images are low-level features derived from basic pixel information. On the other hand, textures, as crucial elements reflecting the local structure and surface patterns of an image, often display contradictory properties, such as both regularity and randomness, making their underlying attributes difficult to capture. This creates challenges for neural networks designed for general image processing when tasked with texture analysis.

Considering the unique nature of texture images, this paper proposes a deep network framework with adaptability to texture attributes, motivated by the following key factors:

1. In neural networks, lower-layer convolutional kernels typically capture relatively simple image attributes, and these basic low-level features are essential for texture recognition. Traditional networks that rely solely on deep features are not well-suited for texture analysis. Therefore, designing effective methods to properly harness the key texture information contained in shallow feature maps is crucial for the texture recognition tasks.
2. Although classical pooling schemes are effective local operations, they are not well-suited for texture attributes that depend on pixel-level information. The traditional Local Binary Pattern (LBP) approach offers an efficient technique for capturing texture features. Therefore, integrating the concept of binary patterns into the information processing of deep feature maps is crucial for effectively capturing texture attributes within deep learning frameworks.
3. The neural network framework is capable of extracting texture features at various levels. Convolution operations in the lower layers capture relatively simple texture features, while deeper layers use higher-level convolutional kernels to combine and abstract more complex texture patterns. In recent years, the concepts of dual-stream and cross-layer encoding have attracted widespread attention in computer vision fields such as object tracking[39,40] due to their powerful capability of representing multi-level abstract features. However, these approaches have been scarcely explored in the field of texture analysis. Texture is both a local perceptual feature and a global property. Therefore, achieving dual-stream feature learning across different regions and levels, while considering both the fine details and overall patterns of textures, is crucial for improving the recognition of complex textures.

Intra-layer encoding stream

In deep neural networks, convolutional kernels at different depths capture feature attributes at progressively higher levels of abstraction. Shallow feature maps primarily represent simple low-level features, while deeper feature maps extract increasingly abstract and intricate texture patterns.

To preserve feature attributes as comprehensively as possible, this paper extracts K layers of feature map tensors from the deep network framework (e.g., AlexNet[41], VGGNet[42]), each denoted as $\mathbf{F}^k.k = 1, 2, \cdots, K$ is the index of the mapping layer. Here, we treat the deep feature maps from different layers as domain experts, each possessing unique preferences for specific feature perspectives and contributing discriminative texture cues from various depth levels. Taking VGG19 as an example, we extract feature maps from $K = 4$ layers of VGG-VD19 (i.e., "conv2-2", "conv3-4", "conv4-4", and "conv5-4"), denoted as follows: $\mathbf{F}^1 \in \mathbb{R}^{112\times112\times128}$ $\mathbf{F}^2 \in \mathbb{R}^{56\times56\times512}$ $\mathbf{F}^3 \in \mathbb{R}^{28\times28\times512}$ and $\mathbf{F}^4 \in \mathbb{R}^{14\times14\times512}$. It is evident that the spatial dimensions of different convolutional layers vary, which presents a challenge for the encoding mechanisms proposed in this paper.

To address this, bilinear interpolation is applied for down-sampling, ensuring that feature maps from different layers are unified to a consistent spatial resolution, denoted as s×s. For VGG19, s is set to 14. Furthermore, to reduce both computational complexity and feature dimensionality, an expert-level average reduction strategy is employed, as shown in Fig. 3. Following this operation, the number of feature maps for each layer is reduced and denoted as N. This reduction is applied to the feature maps corresponding to each expert perspective. The resulting processed feature maps can be expressed as: $\widehat{\mathbf{F}}^1 \in \mathbb{R}^{s\times s\times N}$,$\widehat{\mathbf{F}}^2 \in \mathbb{R}^{s\times s\times N}$,$\widehat{\mathbf{F}}^3 \in \mathbb{R}^{s\times s\times N}$ and $\widehat{\mathbf{F}}^4 \in\in \mathbb{R}^{s\times s\times N}$.

To ensure that the network effectively captures the texture cues perceived by each expert perspective layer, this paper introduces the intra-layer encoding stream. In this stream, each layer's feature map undergoes independent intra-layer binary encoding strategy based on the local mean,

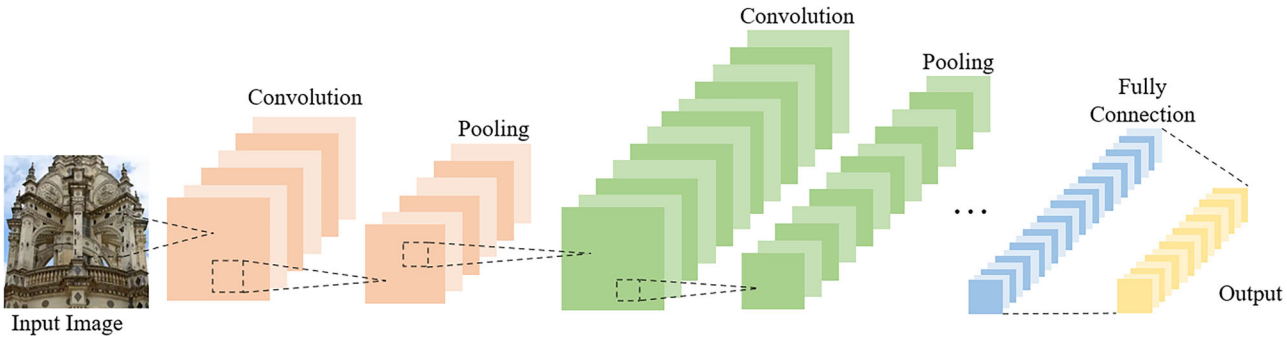

Fig. 2 | Typical architecture of CNN. This figure illustrates the standard architecture of a convolutional neural network (CNN), widely adopted for image classification. The model consists of an input layer, a sequence of convolutional and pooling layers, and one or more fully connected layers leading to the final classification output.

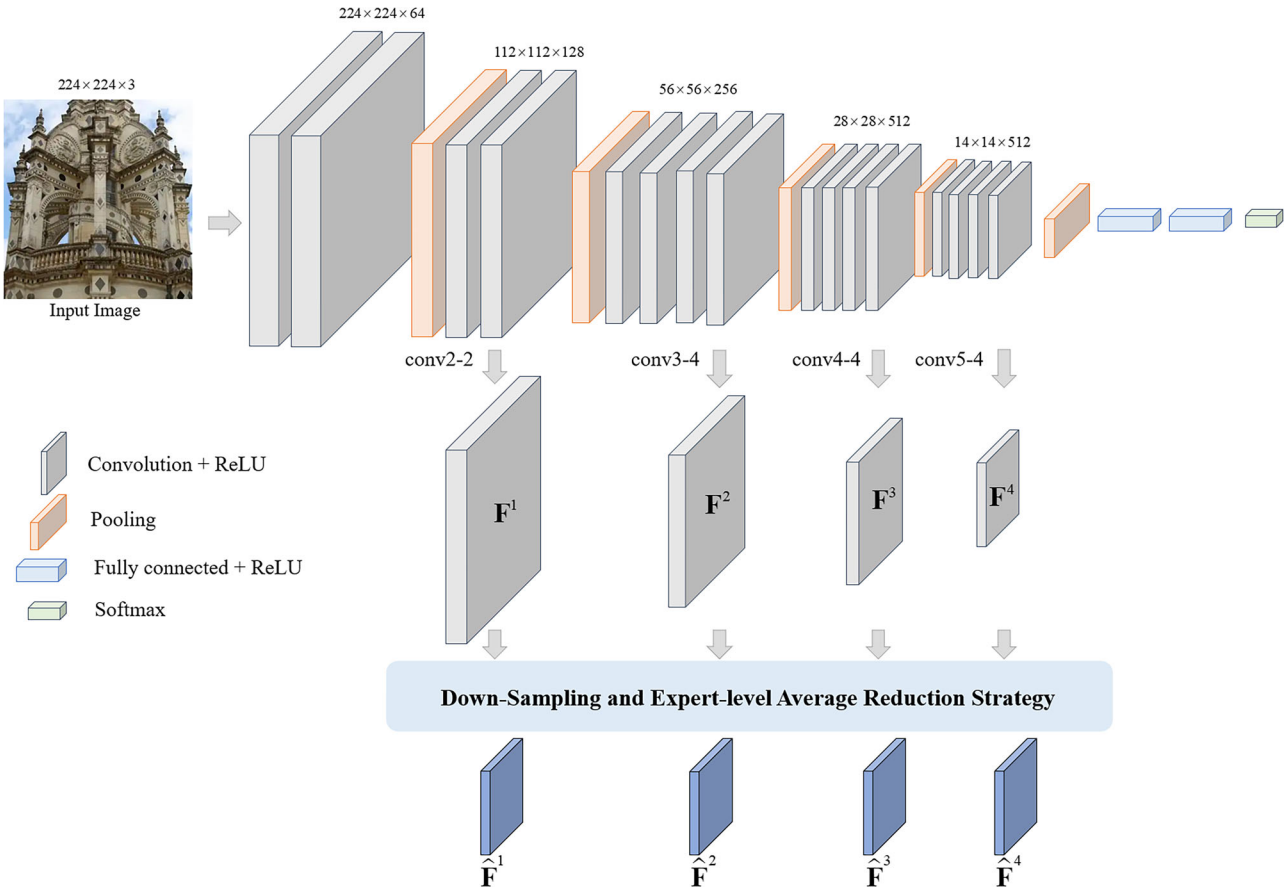

Fig. 3 | The illustration of Expert Perspective Feature Mapping Layer. This figure illustrates the basic structure of the Expert Perspective Feature Mapping Layer. The gray blocks represent Convolution followed by ReLU activation, the orange blocks indicate Pooling operations, the light blue blocks denote Fully Connected layers followed by ReLU, the green block corresponds to the Softmax layer, and the dark blue blocks represent the resulting Expert Perspective Feature Mappings.

allowing for the extraction of local texture attributes. Notably, this proposed strategy offers a simple yet efficient means of representing local texture information. It is applied to each local feature map $\widehat{F}_i^k$ in layer$k, k = 1, 2, 3, 4$, and get the intra-layer sign binary mapping $Intra\_SM_i^k$, formulated as follows:

$$Intra\_SM_i^k = \text{sign}(\widehat{F}_i^k - \text{mean}(\widehat{F}_i^k)) \tag{4}$$

where $mean(g)$ is the adaptive encoding threshold, is equal to the mean of all points in the local feature map $\widehat{F}_i^k$. It is worth noting that in this study, the mean value is consistently used as the threshold for binary encoding. The main reason lies in its statistical significance - the mean serves as a global measure of central tendency, making it a suitable reference for partitioning when the distribution of texture attributes is approximately symmetrical. This choice also offers the advantages of simplicity in implementation and computational efficiency. And sign(g) is the sign function, defined as

$$\text{sign}(x) = \begin{cases} 1, x > 0 \\ 0, \text{otherwise} \end{cases} \tag{5}$$

Then for the layer $k, k = 1, 2, \cdots, K$, we can get four independent intra expert perspectives sign mapping tensor, denoted as

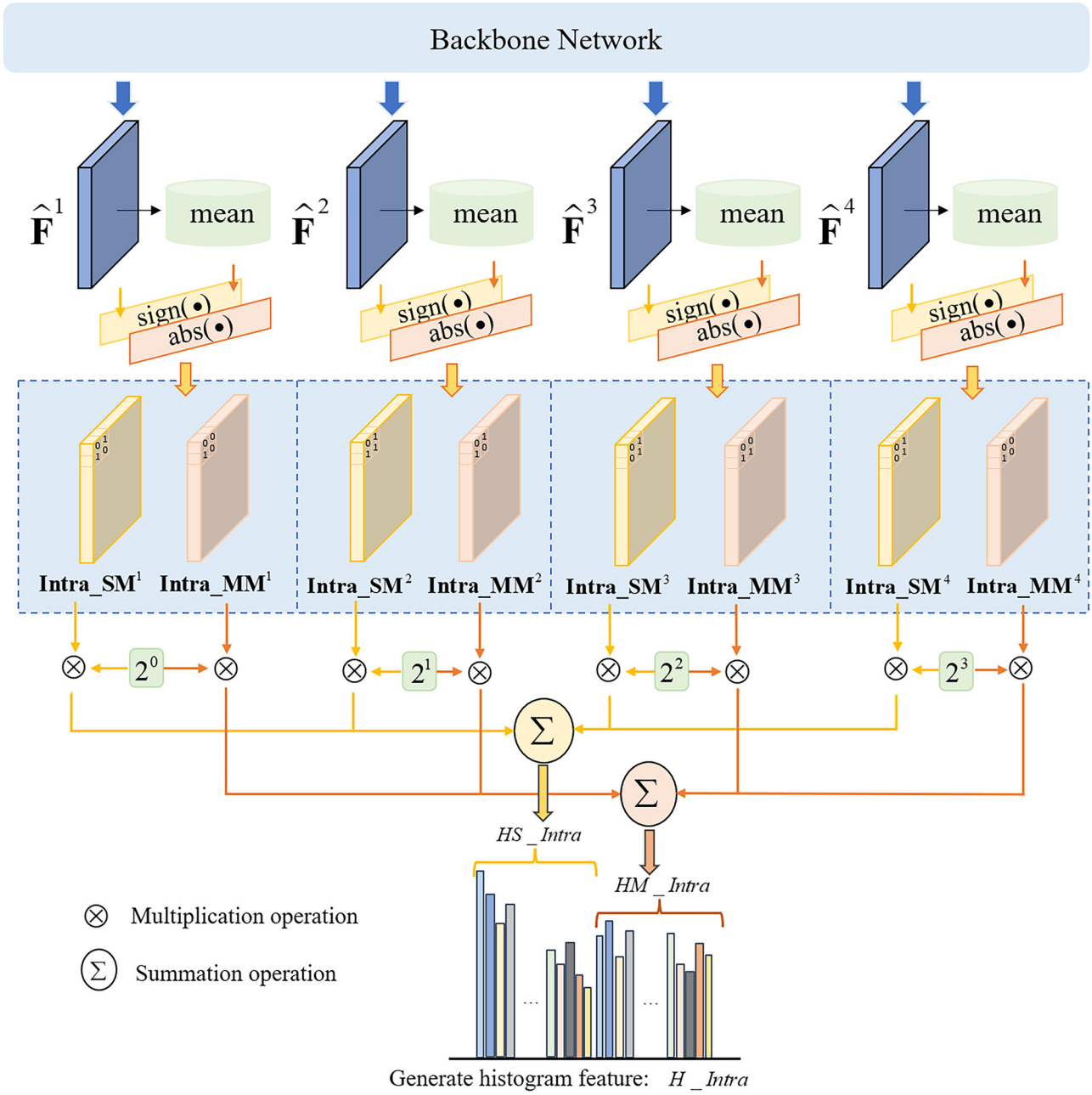

Fig. 4 | The illustration of Intra-layer Encoding Stream. This figure presents the core structure of the Intra-layer Encoding Stream. First, each convolutional layer's feature map is independently processed using an intra-layer binary encoding strategy based on local mean values, aiming to capture fine-grained local texture patterns. Second, this process yields four independent intra expert perspectives sign and magnitude mapping tensors. Last, we concatenate the intra-layer sign and magnitude encoding feature vectors to obtain the final feature vector.

IntraSM$^k, k = 1, 2, \cdots, K$. To enable the effective fusion of multiple independent sign mappings, this paper proposes a novel multi-attribute joint encoding strategy, as illustrated in Fig. 4. This strategy re-encodes the sign mappings derived from different expert layers into a unified, integrated sign pattern code. To facilitate histogram-based feature statistics, the resulting sign pattern code is further converted into its decimal form, computed as follows:

$$Intra\_SMP_i = \sum_{k=1}^{K} 2^{k-1} gIntra\_SM_i^k = 2^0 gIntra\_SM_i^1 + 2^1 gIntra\_SM_i^2 + \cdots + 2^{K-1} gIntra\_SM_i^K \quad (6)$$

It is worth noting that, for histogram representation, formula (6) converts the binary-coded *Intra_SM* into its decimal form*Intra_SMP*, where the weight 2^{k-1} is primarily used to facilitate the conversion from binary code to decimal value.

Furthermore, we utilize the histogram of $Intra\_SMP_i$ to construct feature histograms $H(Intra\_SMP_i)$ and concatenate the histograms of I channels to form the intra-layer sign encoding feature vector, represented as:

$$HS\_Intra = [H(Intra\_SMP_1), H(Intra\_SMP_2), ..., \mathrm{H(Intra\_SMP_I)}] \quad (7)$$

Inspired by Completed Local Binary Pattern (CLBP)[12], in addition to the intra-expert perspectives sign mapping tensor, we also design an intra-expert perspectives magnitude mapping tensor to provide complementary

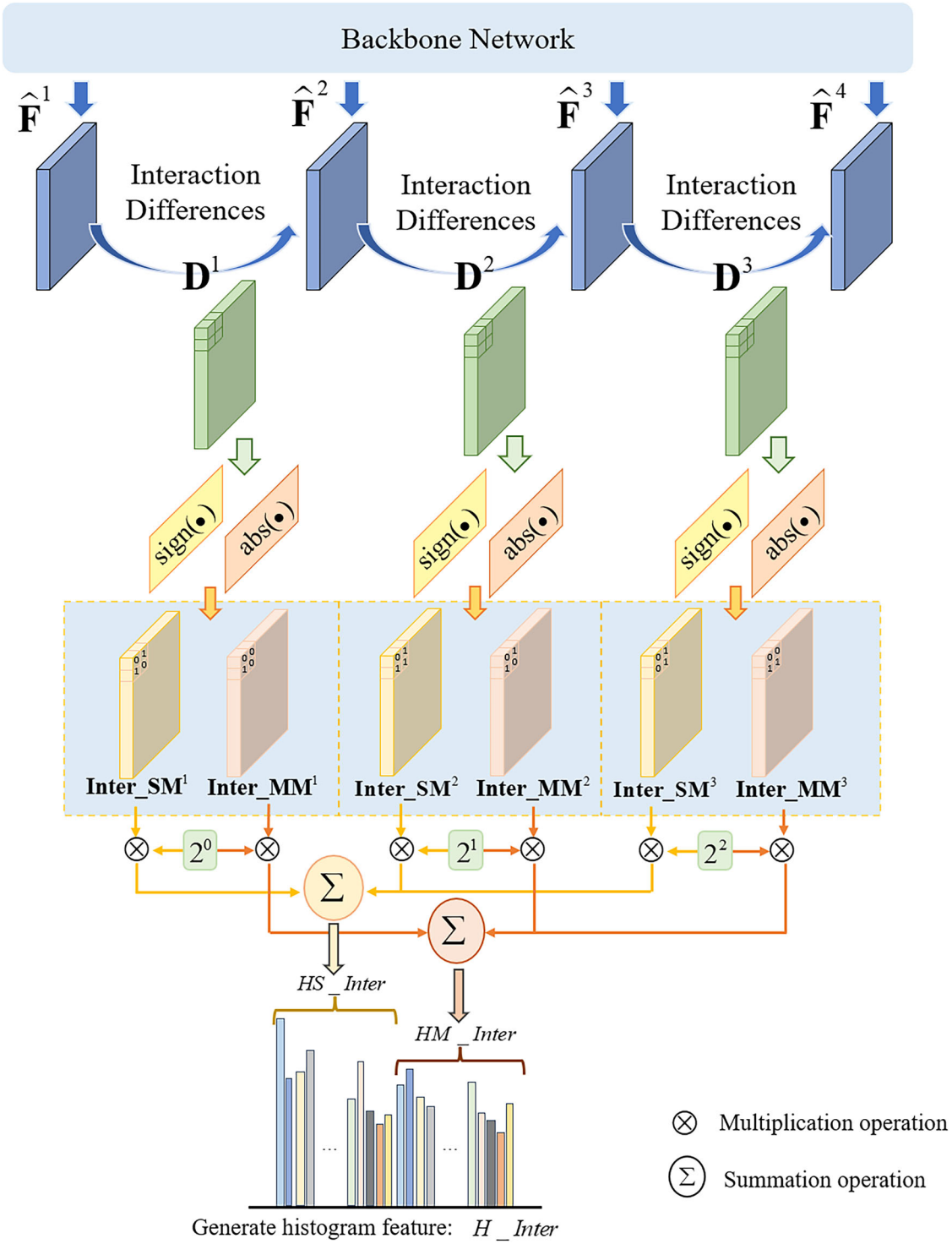

Fig. 5 | The illustration of inter-layer encoding stream. This figure presents the core structure of the inter-layer encoding stream. First, the interaction differences between adjacent expert perspective layers are computed to capture cross-layer feature variations. Then, a cross-layer binary encoding mechanism is applied to generate the corresponding inter-layer sign and magnitude pattern codes, which are used to construct the inter-layer sign and magnitude encoding feature vectors. Finally, these feature vectors are concatenated to form the final inter-layer feature representation.

local texture information. Specifically, the intra-layer magnitude difference mapping $Intra\_MD_i^k$ is defined as follows:

$$Intra\_MD_i^k = abs(\hat{F}_i^k - mean(\hat{F}_i^k)) \tag{8}$$

where $abs(g)$ is the absolute value operation.

Furthermore, to obtain the binary encoding, the mean of $Intra\_MD_i^k$ is used as the encoding threshold for$Intra_MD_i^k$, which is defined as follows:

$$Intra\_MM_i^k = t(Intra\_MD_i^k - mean(Intra\_MD_i^k)) \tag{9}$$

where $t(g)$ is the binarization threshold function, defined as $t(x-c)=\begin{cases}1, x>c\\ 0, otherwise\end{cases}$.

Similar to the previous step, the resulting magnitude pattern code is also converted into its decimal form, which is computed as follows:

$$Intra\_MMP_i = \sum_{k=1}^{K} 2^{k-1} gIntra\_MM_i^k = 2^0 gIntra\_MM_i^1 + 2^1 gIntra\_MM_i^2 + \cdots + 2^{K-1} gIntra\_MM_i^K \quad (10)$$

Furthermore, we compute the histogram of $Intra\_MMP_i$ to construct feature histograms $H(Intra\_MMP_i)$, and concatenate the histograms across I channels to form the intra-layer magnitude encoding feature vector, denoted as:

$$HM\_Intra = [H(Intra\_MMP_1), H(Intra\_MMP_2), ..., \mathrm{H(Intra\_MMP_I)}] \quad (11)$$

It is noteworthy that histogram fusion is adopted for feature vector construction due to its statistical integration capability at the distribution level. By aggregating histograms obtained from multiple binary encoding patterns, the fusion process effectively balances local variations and mitigates the bias associated with any single encoding scheme. Consequently, the resulting feature distribution becomes more representative and robust, reducing the impact of sample-specific randomness and enhancing the discriminative power across diverse samples.

As shown in Fig. 4, we concatenate the intra-layer sign and magnitude encoding feature vectors to obtain the final feature vector, denoted as:

$$H\_Intra = [HS\_Intra, HM\_Intra] \quad (12)$$

Intra-layer encoding stream

To facilitate mutual learning and interaction of key texture attributes across different expert layers, this paper introduces the inter-layer encoding stream. In contrast to the intra-layer encoding stream, it effectively captures hierarchical representations and models cross-layer dependencies within deep feature mappings. Specifically, the inter-layer encoding stream incorporates a cross-layer binary encoding mechanism, which leverages binary mutual encoding between different expert layers to extract inter-layer binary mappings.

We begin by defining the interaction differences between adjacent expert view layers, denoted as $D_i^k = \widehat{F}_i^k - \widehat{F}_i^{k+1}$. Based on the sign of these interaction differences, a binary mutual sign encoding is then constructed, as formulated below:

$$Inter\_SM_i^k = \mathrm{sign(D_i^k)} = \mathrm{sign}(\widehat{\mathrm{F}}_\mathrm{i}^\mathrm{k} - \widehat{\mathrm{F}}_\mathrm{i}^{\mathrm{k+1}}) \quad (13)$$

where $k, k=1,2,\cdots,K$, $\widehat{F}_i^k$ and $\widehat{F}_i^{k+1}$ are the adjacent different expert mapping layers. sign(g) is the sign function, defined in formula (5). $Inter\_SM_i^k$ is the resulting inter-layer binary mapping.

As shown in Fig. 5, if four layers of deep feature mappings $\widehat{F}_i^k, k=1,2,3,4$ are selected, they will yield three inter-layer binary mapping

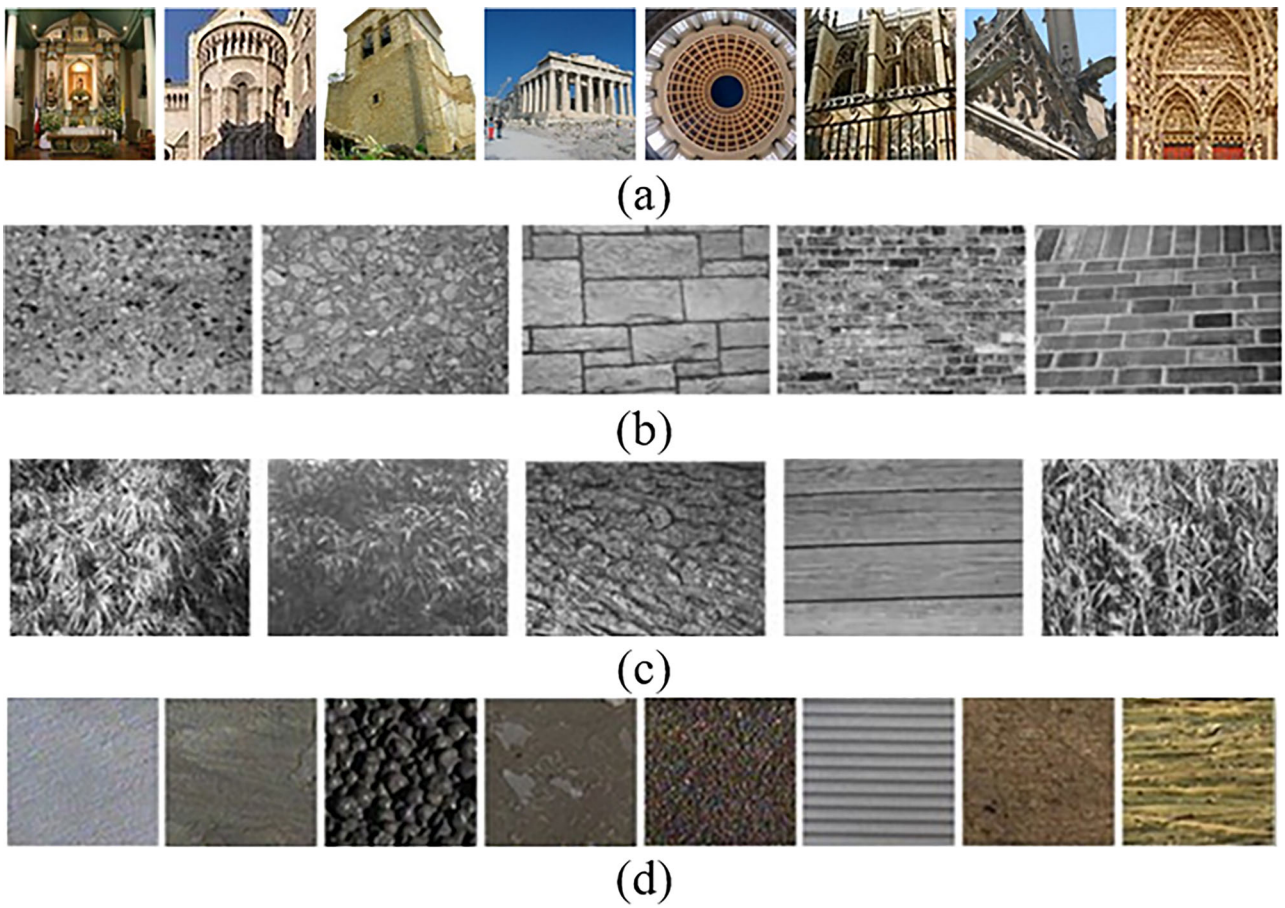

Fig. 6 | The texture samples of four databases. This figure presents representative texture image samples from the four datasets employed in this study. **a** It shows samples from the AHE database, containing diverse architectural heritage textures with varying material properties and surface patterns. **b** It displays samples from the UIUC database, which includes textures captured under uncontrolled imaging conditions with significant variations in scale and orientation. **c** It presents samples from the UMD database, characterized by complex environmental conditions. **d** It shows samples from the CUReT database, featuring textures affected by specular reflections and self-shadowing.

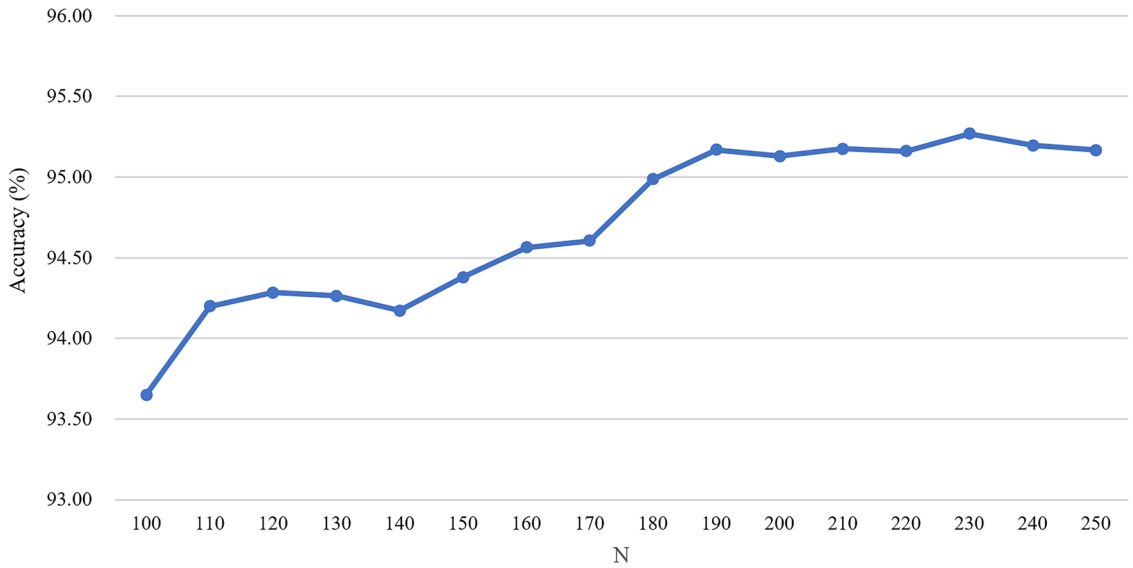

Fig. 7 | The impact of *N* on the classification performance of DMCE-Net on the AHE database. This figure illustrates how varying the number of aggregated channels (N) influences the classification accuracy of DMCE-Net on the AHE database.

InterBM^k, $k = 1, 2, 3$. To achieve more compact and effective cross-layer feature dependencies, the cross-layer binary encoding mechanism re-encodes the generated inter-layer sign binary mappings into a multi-bit binary code, denoted as

$$\begin{aligned} Inter\_SMP_i &= \sum_{k=1}^{K-1} 2^{k-2} \bullet Inter\_SM_i^{k-1} \\ &= 2^0 \bullet Inter\_SM_i^1 + 2^1 \bullet Inter\_SM_i^2 + \cdots + 2^{K-2} \bullet Inter\_SM_i^{K-1} \end{aligned} \quad (14)$$

Furthermore, similar to intra-layer encoding stream, we employ the histogram of $Inter\_SMP_i$ to build feature histograms $H(Inter\_SMP_i)$ and concatenate the histograms across I channels to construct the inter-layer sign encoding feature vector, described as:

$$HS\_Inter = [H(Inter\_SMP_1), H(Inter\_SMP_2), \ldots, \mathrm{H(Inter\_SMP_I)}] \quad (15)$$

Analogous to the intra-layer encoding stream, a binary mutual magnitude encoding is devised to capture complementary information derived from the magnitude differences between adjacent layers. Specifically, the inter-layer magnitude difference mapping is defined as follows

$$Inter\_MD_i^k = abs(\widehat{F}_i^k - \widehat{F}_i^{k+1}) \quad (16)$$

Then the inter-layer magnitude difference mapping is defined as:

$$Inter\_MM_i^k = t(Inter\_MD_i^k - mean(Inter\_MD_i^k)) \quad (17)$$

where $t(g)$ is the binarization threshold function. Similar to the previous step, the corresponding inter magnitude pattern code is also converted into its decimal form, computed as:

$$\begin{aligned} Inter\_MMP_i &= \sum_{k=1}^{K} 2^{k-1} \bullet Inter\_MM_i^k \\ &= 2^0 gInter\_MM_i^1 + 2^1 gInter\_MM_i^2 + \cdots + 2^{K-1} gInter\_MM_i^K \end{aligned} \quad (18)$$

Furthermore, we compute the histogram of $Inter\_MMP_i$ to construct feature histograms $H(Inter\_MMP_i)$, and concatenate the histograms across I channels to form the inter-layer magnitude encoding feature vector, denoted as:

$$HM\_Inter = [H(Inter\_MMP_1), H(Inter\_MMP_2), \ldots, \mathrm{H(Inter\_MMP_I)}] \quad (19)$$

As shown in Fig. 5, we concatenate the inter-layer sign and magnitude encoding feature vectors to obtain the final feature vector, denoted as:

$$H\_Inter = [HS\_Inter, HM\_Inter] \quad (20)$$

Table 1 | The overall classification accuracy (%) on the AHE database using different backbone network

Backbone network	DMCE-Net	Intra-layer encoding stream	Inter-layer encoding stream
VGG-VD19	95.27 ± 0.004	95.05 ± 0.004	92.93 ± 0.005
VGG-VD16	95.36 ± 0.004	95.16 ± 0.004	93.84 ± 0.005
AlexNet	93.99 ± 0.004	93.91 ± 0.005	92.95 ± 0.005

Dual-stream multi-layer cross encoding network

To capture reliable texture cues and distinguish between inter-class similarity and intra-class diversity, this paper proposes a DMCE-Net. It treats feature maps from different layers as expert representations, where the intra-layer stream extracts texture attributes from individual experts, and the inter-layer stream captures joint features through cross-layer interaction. This framework enhances both the discriminative power and interpretability of deep texture representations in complex visual scenes.

Our architecture systematically integrates complementary feature representations through hierarchical feature integration. Specifically, we construct a comprehensive final feature vector FH by concatenating the intra-layer encoding feature vector H_Intra and the inter-layer encoding feature vector H_Inter, formulated as:

$$FH = [H\_Intra, H\_Inter] \quad (21)$$

As shown in Fig. 4, for $K = 4$, the intra-layer encoding stream generates 4-bit binary codes using a multi-attribute joint encoding strategy, resulting in an intra-layer encoding feature vector with a dimensionality of $32N$. Meanwhile, as shown in Fig. 5, the inter-layer

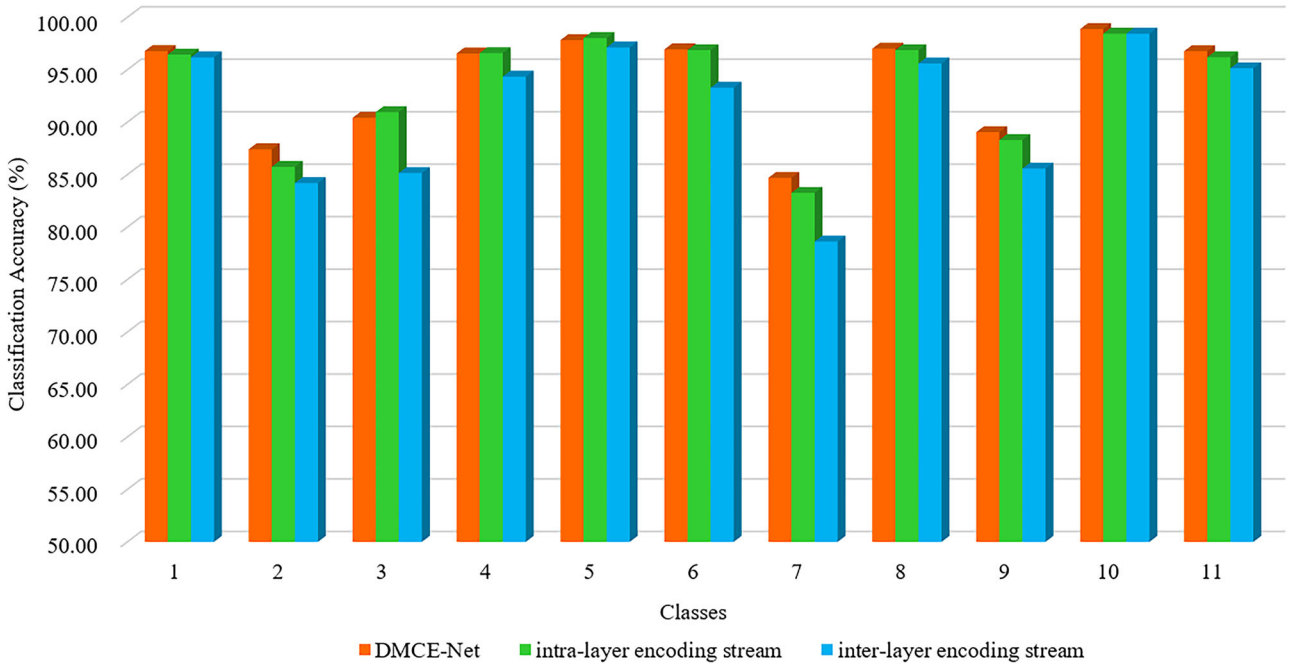

Fig. 8 | Category-wise classification results on the AHE database using VGG-VD19 as the backbone network. This figure presents the classification accuracy for each texture category in the AHE database, obtained using DMCE-Net with VGG-VD19 as the backbone network. Red, green, and blue bar represent the classification results of DMCE-Net, the intra-layer encoding stream, and the inter-layer encoding stream, respectively.

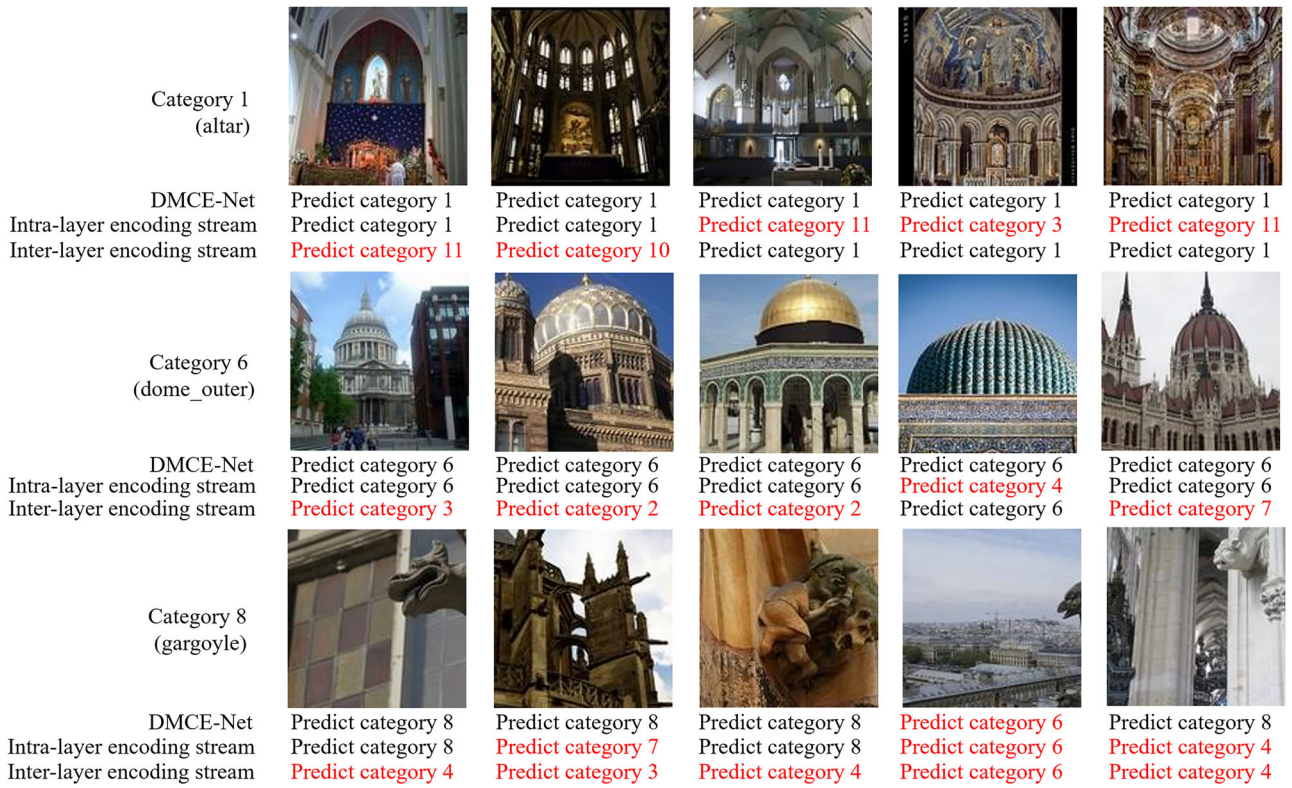

Fig. 9 | The classification results of representation images on the AHE database using VGG-VD19 as the backbone network. This figure presents the classification results of representative samples. The red text in the figure indicates misclassified results, while the black text represents correctly classified results.

encoding stream produces 3-bit binary codes via a cross-layer binary encoding mechanism, yielding an inter-layer encoding feature vector with a dimensionality of 16N. Therefore, the final feature vector has a total dimensionality of 48N. This clearly indicates that the system parameter N, determined by the expert-level average reduction strategy, directly influences the output feature dimensionality of DMCE-Net. A detailed analysis of N is provided in Section "Effect of parameters setting" of this paper.

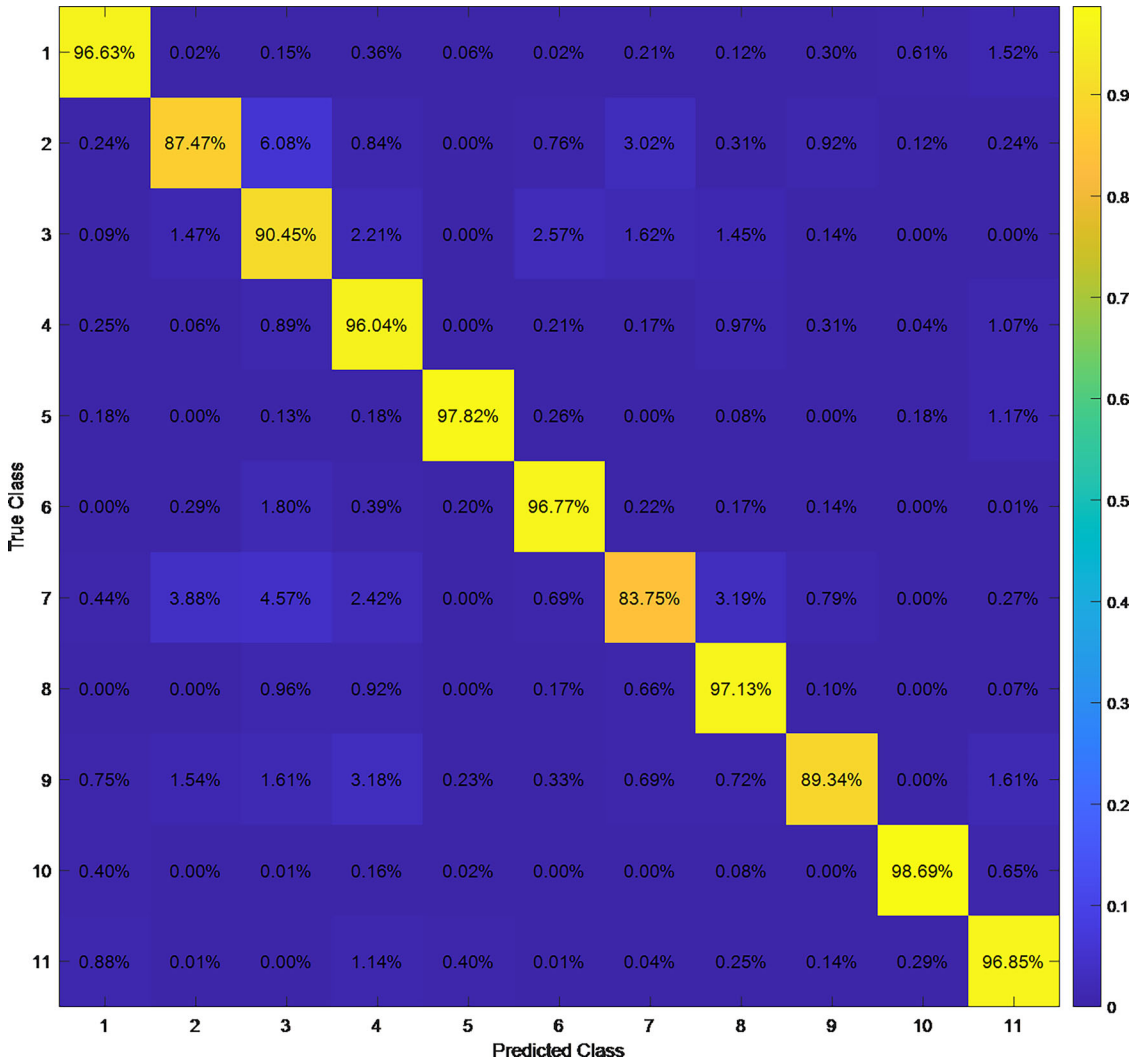

Fig. 10 | The confusion matrix of DMCE-Net with VGG-VD19 on the AHE database. This figure presents the confusion matrix of the proposed DMCE-Net using VGG-VD19 as the backbone network on the AHE database. The results are averaged over 50 random splits to ensure statistical stability and robustness. The matrix illustrates the model's classification accuracy across all categories, where diagonal elements represent correct predictions and off-diagonal elements indicate misclassifications.

The technical advantages of the proposed DMCE-Net can be outlined as follows:

1. The proposed DMCE-Net regards feature maps from different layers of a deep network as domain experts, each offering a distinct specialized perspective. The dual-stream framework is designed to not only preserve the independence of expert-specific features but also promote mutual learning and interaction of texture attributes across hierarchical levels. This design effectively addresses the challenges posed by inter-class similarity and intra-class variability in texture representation.
2. In the intra-layer encoding stream, the proposed intra-layer binary encoding strategy effectively harnesses the advantages of binary encoding to capture fine-grained local texture attributes. Moreover, the multi-attribute joint encoding strategy enables efficient fusion of features from diverse expert perspectives, thereby enhancing the overall expressiveness and robustness of texture representation.
3. In the inter-layer encoding stream, the proposed cross-layer binary encoding mechanism cleverly facilitates mutual learning and interaction of texture attributes across different expert perspective layers, offering robust technical support for the unified representation of cross-layer features.

In conclusion, the dual-stream mechanism of DMCE-Net facilitates dual-stream learning across regions and layers, striking a balance between fine details and overall patterns, and ensuring the collaborative optimization of multi-level texture features. This offers a reliable technical approach for the representation and recognition of complex textures.

Results

Experimental setup and database

In our experiments, we employ three widely-used CNN pre-trained frameworks—AlexNet[41], VGG-VD16[42], and VGG-VD19[42]—to extract deep feature maps. For AlexNet, feature maps are obtained from three convolutional layers: conv3, conv4, and conv5, with the input images resized to $227 \times 227 \times 3$. For VGG-VD16, we utilize intermediate convolutional layers conv2-2, conv3-3, conv4-3, and conv5-3 to capture deep features. Similarly, for VGG-VD19, feature maps are extracted from four layers: conv2-2, conv3-4, conv4-4, and conv5-4. The input images for both VGG-VD16 and VGG-VD19 are resized to $224 \times 224 \times 3$. All experiments are conducted on a desktop running MATLAB 2024, equipped with a 2.6 GHz CPU and 64 GB of RAM, without GPU acceleration. All experiments in this study employ an SVM classifier. Specifically, we use a linear-kernel SVM based on the LIBSVM[43] library with default parameter settings.

As shown in Fig. 6, the experiments in this section involve one challenging AHE dataset and three competitive texture classification datasets: AHE[44], UIUC[45], UMD[46], and CUReT[47] database.

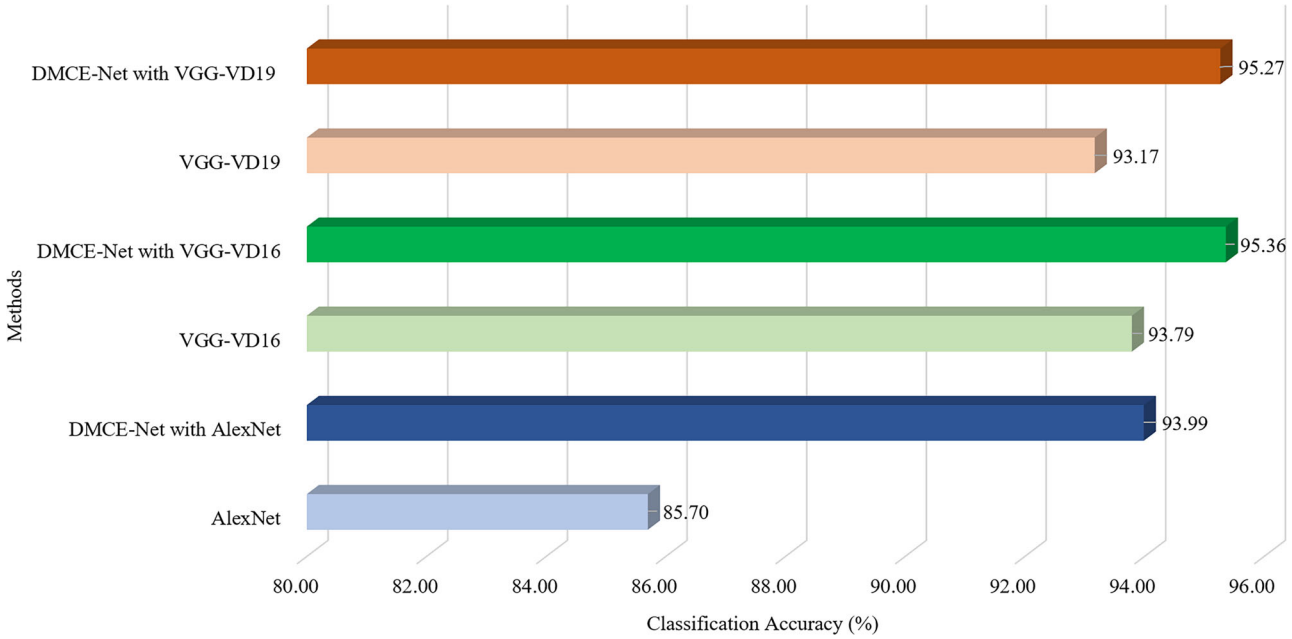

Fig. 11 | Comparative analysis of the proposed DMCE-Net and its backbone counterparts on the AHE database. This figure presents a comparative analysis between the proposed DMCE-Net and its corresponding backbone networks on the AHE database. DMCE-Net consistently outperforms the baseline networks, demonstrating its enhanced capability in capturing fine-grained texture attributes and improving overall classification accuracy.

Table 2 | Classification results of different feature components DMCE-Net with VGG on the AHE Dataset

Category	DMCE-Net	Intra-layer encoding stream	Intra-layer sign encoding stream	Intra-layer magnitude encoding stream	Inter-layer encoding stream	Inter-layer sign encoding stream	Inter-layer magnitude encoding stream
1	96.63 ± 0.013	96.46 ± 0.013	94.72 ± 0.017	96.05 ± 0.015	95.92 ± 0.014	94.53 ± 0.015	95.68 ± 0.013
2	87.47 ± 0.032	86.33 ± 0.036	82.14 ± 0.042	82.88 ± 0.037	83.84 ± 0.038	77.14 ± 0.039	83.41 ± 0.040
3	90.45 ± 0.018	90.92 ± 0.019	87.58 ± 0.022	87.69 ± 0.025	84.88 ± 0.020	79.62 ± 0.027	83.18 ± 0.021
4	96.04 ± 0.009	96.22 ± 0.008	93.57 ± 0.012	94.91 ± 0.011	93.65 ± 0.013	90.87 ± 0.014	93.34 ± 0.013
5	97.82 ± 0.013	98.07 ± 0.011	98.15 ± 0.013	97.82 ± 0.013	97.14 ± 0.015	95.48 ± 0.019	97.37 ± 0.014
6	96.77 ± 0.016	96.80 ± 0.014	95.45 ± 0.014	95.49 ± 0.014	92.92 ± 0.016	91.66 ± 0.021	92.20 ± 0.019
7	83.75 ± 0.046	82.30 ± 0.043	77.68 ± 0.050	77.16 ± 0.044	78.27 ± 0.045	70.35 ± 0.048	77.63 ± 0.038
8	97.10 ± 0.010	97.15 ± 0.010	95.62 ± 0.012	96.18 ± 0.012	95.63 ± 0.012	93.54 ± 0.014	95.42 ± 0.012
9	89.34 ± 0.045	88.39 ± 0.044	86.62 ± 0.045	85.41 ± 0.049	85.93 ± 0.047	81.67 ± 0.058	85.84 ± 0.048
10	98.69 ± 0.008	98.43 ± 0.009	97.80 ± 0.009	97.86 ± 0.011	98.25 ± 0.007	97.50 ± 0.009	98.41 ± 0.008
11	96.85 ± 0.013	96.19 ± 0.014	94.79 ± 0.016	95.68 ± 0.015	95.51 ± 0.013	94.39 ± 0.015	95.12 ± 0.016
Total	95.27 ± 0.004	95.05 ± 0.004	93.02 ± 0.005	93.53 ± 0.006	92.93 ± 0.005	90.08 ± 0.006	92.30 ± 0.006

The AHE dataset[44] is a specialized collection designed for the recognition and classification of architectural components found in cultural heritage sites. It contains high-resolution images of various elements—such as columns, arches, windows, cornices, and decorative motifs—captured from a wide range of historical buildings. The dataset poses significant challenges due to variations in scale, viewpoint, illumination, occlusion, weathering, and stylistic diversity across different architectural periods and regions. AHE provides a valuable benchmark for developing and evaluating computer vision algorithms aimed at the visual understanding and preservation of historical structures. The UIUC database[45] is a widely adopted benchmark for evaluating texture classification algorithms. It comprises 25 texture categories, each containing 40 grayscale images with a resolution of 640 × 480 pixels. The images exhibit considerable variations in scale, viewpoint, illumination, and minor deformations, making the dataset well-suited for assessing the robustness and generalization capability of texture analysis methods. The UMD texture dataset[46] comprises 25 texture categories, each containing 40 high-resolution images of 1280 × 960 pixels captured under diverse viewpoints, scales, and lighting conditions. The textures feature pronounced 3D structures, non-uniform patterns, and natural appearances, making the dataset particularly well-suited for evaluating the performance of algorithms in real-world texture recognition tasks. The CUReT database[47] comprises images from 61 texture categories, each containing 92 samples of size 200 × 200 pixels, captured under diverse combinations of illumination directions and viewing angles. The textures span a wide range of real-world materials—including fabric, metal, wood, and stone—and exhibit substantial variations in reflectance properties and surface geometry.

In our experiments, 80% of the samples were randomly selected for training, and the remaining 20% were used for testing. The final classification accuracy was computed as the average over 50 independently randomized dataset splits. Notably, the proposed method leverages pre-trained deep backbone networks without the need for parameter fine-tuning or

Table 3 | Classification results of DMCE-Net with different number of layers on the AHE Dataset

Category	DMCE-Net (k = 1,2,3,4)	DMCE-Net (k = 1,2,3)	DMCE-Net (k = 2,3)
1	96.63 ± 0.013	96.25 ± 0.017	94.79 ± 0.017
2	87.47 ± 0.032	83.69 ± 0.035	77.29 ± 0.048
3	90.45 ± 0.018	86.65 ± 0.026	76.62 ± 0.025
4	96.04 ± 0.009	94.26 ± 0.011	90.16 ± 0.014
5	97.82 ± 0.013	96.07 ± 0.017	94.36 ± 0.020
6	96.77 ± 0.016	93.61 ± 0.017	89.00 ± 0.021
7	83.75 ± 0.046	76.20 ± 0.042	70.05 ± 0.058
8	97.10 ± 0.010	95.64 ± 0.011	93.13 ± 0.017
9	89.34 ± 0.045	86.39 ± 0.042	81.08 ± 0.038
10	98.69 ± 0.008	98.33 ± 0.011	97.67 ± 0.012
11	96.85 ± 0.013	95.98 ± 0.012	93.97 ± 0.017
total	95.27 ± 0.004	93.04 ± 0.005	89.20 ± 0.007

GPU acceleration. This characteristic results in low computational overhead, thereby enhancing the method's applicability, scalability, and ease of deployment in resource-constrained environments.

Effect of parameters setting

Before performing the intra-layer and inter-layer encoding stream, the proposed DMCE-Net normalizes feature maps from different depth levels to a unified size of $s \times s \times N$ through down-sampling and an expert-level average reduction strategy. For AlexNet, $s = 13$; for VGG-VD16 and VGG-VD19, $s = 14$. The parameter N is empirically determined through preliminary experiments. Taking VGG-VD19 as an example, Fig. 8 illustrates the impact of N on the classification performance of DMCE-Net on the AHE database.

As shown in Fig. 7, the classification accuracy increases significantly as the value of N rises from 100 to 190. When N exceeds 190, the performance improvement gradually becomes marginal, reaching its peak at $N = 230$. Since the value of N directly determines the final feature dimensionality, we fix N at 230 in this study, considering both classification performance and computational cost. Taking VGG-VD19 as an example, the feature dimension of intra-layer and inter-layer encoding feature vector are $32N = 7360$ and $16N = 3680$, respectively. Consequently, the final feature vector of DMCE-Net has a total dimensionality of 11040.

Ablation study on AHE database

To validate the effectiveness of the proposed DMCE-Net, as well as its two individual encoding streams—the Intra-layer and inter-layer encoding streams—this section evaluates both the overall classification performance and the per-category results across various AHE types on the AHE database, using three deep backbone networks: VGG-VD19, VGG-VD16, and AlexNet.

Table 1 summarizes the overall classification accuracy of DMCE-Net and its two individual encoding streams employing different backbone networks on the AHE database. For each class, 80% of the sample images were used for training, with the remaining 20% reserved for testing. The reported results represent the mean classification accuracy and variance computed over 50 random dataset splits. As shown in Table 1, DMCE-Net achieves significantly higher classification accuracy compared to its two individual encoding streams. Moreover, the intra-layer encoding stream outperforms the inter-layer encoding stream by a notable margin. For instance, when using VGG-VD19 as the backbone network, DMCE-Net achieves a classification accuracy of 95.27%, which is 0.22% and 2.34% higher than that of the intra-layer and inter-layer encoding streams, respectively.

Table 4 | Category-wise classification accuracy (%) of DMCE-Net with different backbone networks on the UIUC dataset

Category	DMCE-Net with AlexNet	DMCE-Net with VGG-VD16	DMCE-Net with VGG-VD19
1	95.50 ± 0.07	100.00 ± 0.00	100.00 ± 0.00
2	97.25 ± 0.05	98.50 ± 0.04	99.00 ± 0.03
3	93.00 ± 0.09	100.00 ± 0.00	99.50 ± 0.02
4	99.75 ± 0.01	100.00 ± 0.00	100.00 ± 0.00
5	94.50 ± 0.08	100.00 ± 0.00	100.00 ± 0.00
6	90.25 ± 0.11	96.50 ± 0.06	97.75 ± 0.05
7	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
8	95.75 ± 0.06	100.00 ± 0.00	100.00 ± 0.00
9	94.50 ± 0.07	100.00 ± 0.00	100.00 ± 0.00
10	97.00 ± 0.05	99.75 ± 0.02	97.50 ± 0.05
11	97.75 ± 0.07	100.00 ± 0.00	99.50 ± 0.02
12	95.50 ± 0.07	98.25 ± 0.04	100.00 ± 0.00
13	98.25 ± 0.04	99.50 ± 0.02	97.75 ± 0.05
14	93.00 ± 0.09	96.25 ± 0.06	99.00 ± 0.03
15	97.50 ± 0.05	100.00 ± 0.00	100.00 ± 0.00
16	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
17	99.00 ± 0.03	100.00 ± 0.00	99.75 ± 0.02
18	98.50 ± 0.04	99.50 ± 0.02	99.75 ± 0.02
19	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
20	99.50 ± 0.02	100.00 ± 0.00	100.00 ± 0.00
21	99.00 ± 0.04	99.00 ± 0.03	100.00 ± 0.00
22	97.25 ± 0.05	100.00 ± 0.00	100.00 ± 0.00
23	93.75 ± 0.10	99.50 ± 0.02	98.25 ± 0.05
24	97.50 ± 0.05	100.00 ± 0.00	99.75 ± 0.02
25	99.50 ± 0.02	100.00 ± 0.00	100.00 ± 0.00

Figure 8 illustrates the classification accuracy of DMCE-Net and its two individual encoding streams for each category in the AHE database, using VGG-VD19 as the backbone network. As shown, DMCE-Net shows generally higher classification accuracy across most AHE types. Specifically, for category 10 (stained glass), DMCE-Net achieves the highest classification accuracy of 98.87%, outperforming both the intra-layer and inter-layer encoding streams by 0.43%. For the most challenging category, category 7 (flying buttress), DMCE-Net surpasses the intra-layer and inter-layer encoding streams by 1.43% and 6.07%, respectively. Figure 9 illustrates representative visual classification results for samples from distinct categories. It can be observed that DMCE-Net tends to outperform the individual intra-layer and inter-layer encoding streams, particularly in handling complex architectural textures. For example, in Class 8 (gargoyle), the second and fourth samples exhibit pronounced variations in illumination and viewpoint. In these challenging scenarios, relying solely on either the intra-layer encoding stream or the inter-layer encoding stream result in less reliable feature discrimination. By contrast, DMCE-Net, leveraging the complementary strengths of both encoding strategies, achieves correct classification. Notably, for the fourth sample in Class 8 (gargoyle), which contains only a minimal region of discriminative texture features and a large proportion of irrelevant background, none of the three methods yields correct classification. This result highlights a limitation of DMCE-Net when confronted with samples characterized by extensive background interference and extremely sparse target features.

To further investigate the classification performance of the proposed model, Fig. 10 presents the confusion matrix obtained by averaging the results of 50 random experiments on the AHE dataset. Overall, the distribution of the confusion matrix exhibits generally good classification

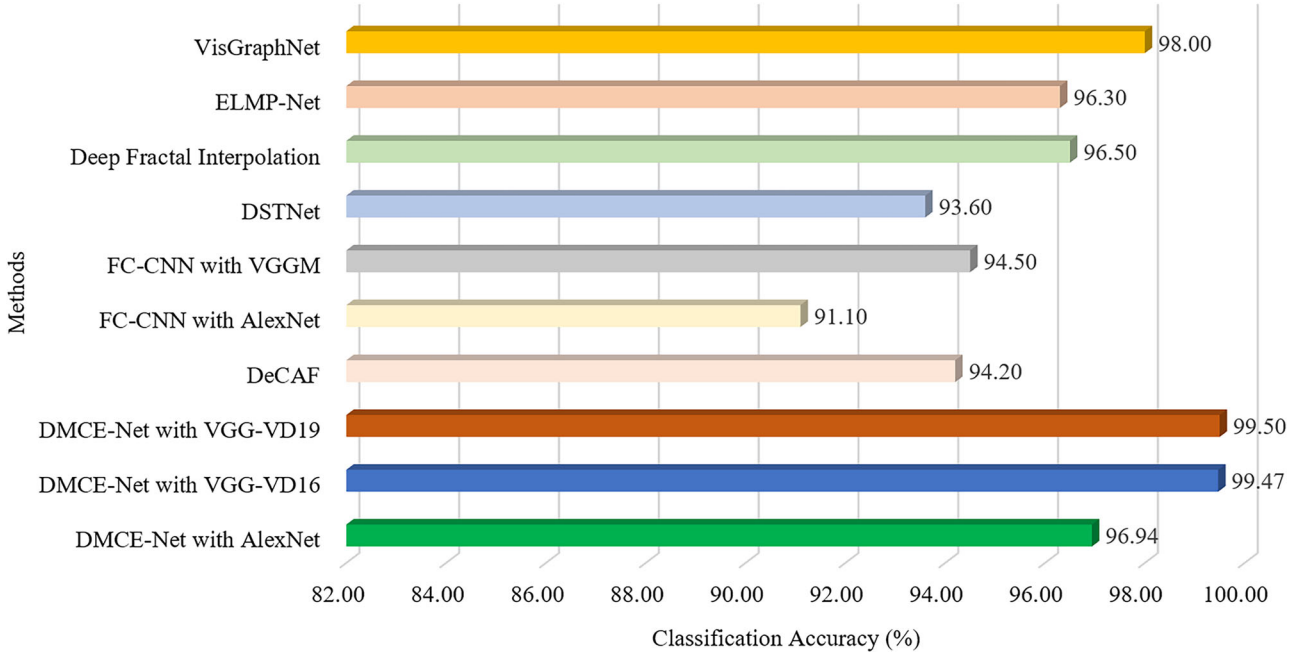

Fig. 12 | Comparison between DMCE-Net with various backbones and state-of-the-art methods on the UIUC dataset. This figure compares the classification performance of the proposed DMCE-Net, implemented with different backbone networks, against several state-of-the-art texture classification methods on the UIUC dataset.

Table 5 | Category-wise classification accuracy (%) of DMCE-Net with different backbone networks on the UMD dataset

Category	DMCE-Net with AlexNet	DMCE-Net with VGG-VD16	DMCE-Net with VGG-VD19
1	97.50 ± 0.05	100.00 ± 0.00	100.00 ± 0.00
2	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
3	97.75 ± 0.05	99.75 ± 0.02	96.75 ± 0.06
4	100.00 ± 0.00	98.75 ± 0.04	100.00 ± 0.00
5	99.75 ± 0.02	99.75 ± 0.02	100.00 ± 0.00
6	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
7	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
8	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
9	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
10	99.00 ± 0.05	99.00 ± 0.05	98.50 ± 0.06
11	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
12	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
13	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
14	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
15	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
16	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
17	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
18	98.50 ± 0.04	100.00 ± 0.00	100.00 ± 0.00
19	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
20	96.50 ± 0.07	100.00 ± 0.00	100.00 ± 0.00
21	97.75 ± 0.05	100.00 ± 0.00	100.00 ± 0.00
22	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
23	98.75 ± 0.04	100.00 ± 0.00	100.00 ± 0.00
24	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
25	98.75 ± 0.04	100.00 ± 0.00	100.00 ± 0.00

performance across most categories, as indicated by the dominance of diagonal elements. This suggests that the model is able to capture and distinguish discriminative features among different classes. However, certain categories exhibit notable confusion, particularly Category 2 (87.47%) and Category 7 (83.75%), whose classification performance is slightly inferior compared to other categories. On one hand, the misclassification rates of Category 2 and Category 7 are relatively high. Specifically, 6.08% of the Category 2 samples were misclassified as Category 3, and 3.03% as Category 7. Similarly, Category 7 showed 3.88% misclassification into Category 2, and 4.57% into Category 3. These results indicate that the model encounters challenges in discriminating between these categories. Further, among all samples predicted as Category 7, 3.02% actually belong to Category 2. This bilateral misclassification highlights the overlap in feature distributions between Category 2 and Category 7, which may lead to ambiguous decision boundaries and increased risk of cross-category errors.

Figure 11 compares the classification performance of the proposed DMCE-Net with its original backbone networks on the AHE dataset. As observed, DMCE-Net shows consistent performance gains over its backbone counterparts. Specifically, DMCE-Net with VGG-VD19 outperforms VGG-VD19 by 2.10%, while DMCE-Net with VGG-VD16 and AlexNet show gains of 1.57% and 8.29%, respectively. These results demonstrate the effectiveness of the proposed DMCE framework leverages texture cues from different expert view layers via the Intra-layer and Inter-layer encoding streams. Furthermore, the integration of the multi-attribute joint encoding strategy and the cross-layer binary encoding mechanism offers a robust approach for enhancing the representation and utilization of deep texture features.

To further investigate the impact of different feature components on classification performance, Table 2 presents the results of ablation experiments based on various encoding streams. As shown in the table, DMCE-Net achieves the highest average classification accuracy of 95.27% across all categories, achieving higher accuracy than each individual encoding stream or feature component. This result suggests that the dual-stream mechanism and multi-feature representation contribute to enhancing the overall model performance.

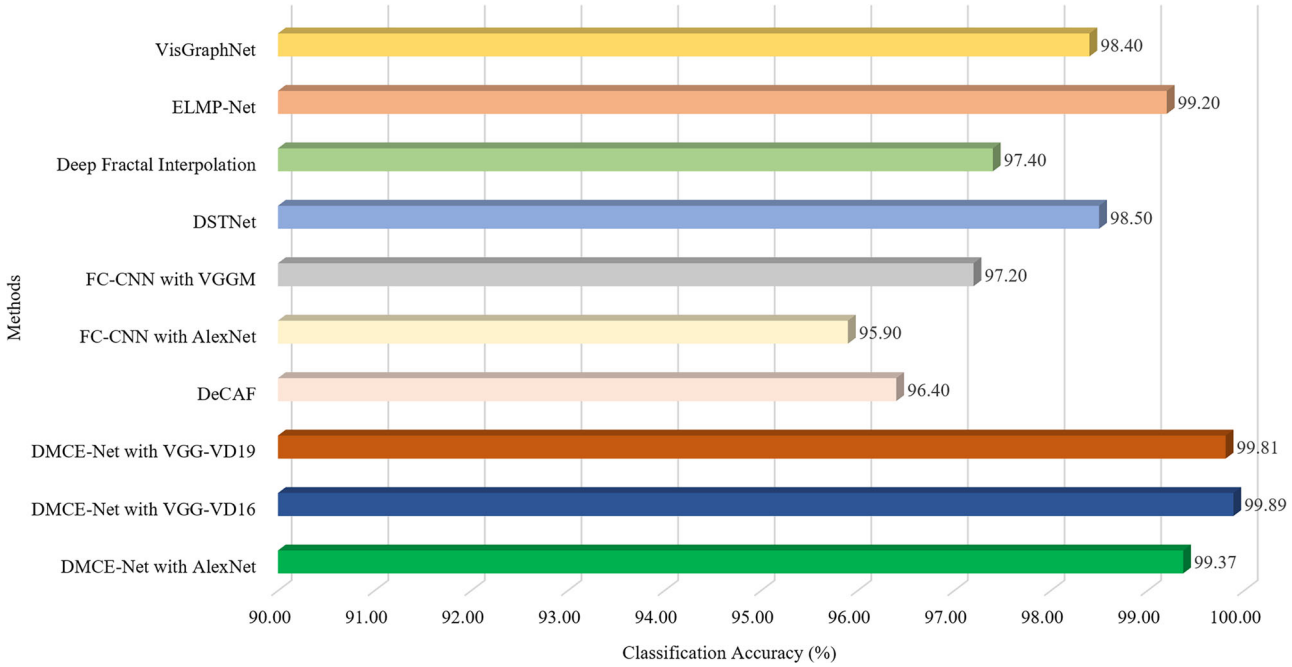

Fig. 13 | Comparison between DMCE-Net with various backbones and state-of-the-art methods on the UMD dataset. This figure presents a comparative evaluation of the proposed DMCE-Net with different backbone networks against several state-of-the-art texture classification methods on the UMD dataset.

The Intra-layer Sign encoding stream and Magnitude encoding stream yield classification accuracies of 93.02% and 93.53%, respectively, both lower than that of the full Intra-layer encoding stream. Similarly, the Inter-layer Sign encoding stream and Inter-layer Magnitude encoding stream achieve 90.08% and 92.30%, respectively, which are also relatively lower than the inter-layer encoding stream. These findings suggest that using Sign or Magnitude features independently may lead to incomplete feature representations, while their joint encoding is more effective in capturing discriminative information. Furthermore, the inter-layer encoding stream achieves an average accuracy of 92.93%, slightly lower than the Intra-layer encoding stream, indicating that the cross-layer encoding serves as a complementary feature extractor but may have limited standalone effectiveness.

In summary, the complete DMCE-Net model, by jointly leveraging Intra-layer and inter-layer encoding streams as well as integrating both Sign and Magnitude features, achieves the highest classification accuracy among the tested configurations. The ablation results demonstrate that individual feature branches generally underperform compared to the combined encoding strategy, particularly, the performance degradation is more pronounced when using Sign or Magnitude features alone, emphasizing the critical role of comprehensive multi-feature integration and the cooperative advantage brought by the dual-stream design.

To investigate the impact of the number of encoding layers on classification performance, Table 3 reports results on the AHE dataset using different sets of convolutional layers as feature encoding sources: k = 1,2,3,4 ; k = 1,2,3 and k = 2,3. The full configuration incorporating four layers achieves the highest average accuracy of 95.27%, showing improved performance over the three-layer (93.04%) and two-layer (89.20%) variants. This clearly underscores the benefit of multi-layer encoding feature integration, where the combination of both shallow and deep representations contributes to improved discriminative power. A class-wise analysis reveals a general decline in accuracy as fewer layers are included in the encoding, with Categories 2, 3, 6, and 7 exhibiting particularly notable drops. For example, the accuracy for Category 3 decreases from 90.45% (four-layer) to 86.65% (three-layer), and further to 76.62% (two-layer). A more severe drop is observed for Category 7, which falls from 83.75% to 70.05%, suggesting that deeper-layer features are especially critical for classes characterized by high intra-class variability or inter-class similarity.

In summary, these results provide evidence for the effectiveness of the complete four-layer DMCE-Net configuration, which enables more comprehensive capture of hierarchical texture attributes ranging from fine-grained local patterns to high-level semantic abstractions. Such multi-level encoding is shown to be essential for achieving robust and accurate classification across diverse texture categories.

Performance comparison on different texture datasets

The UIUC dataset poses significant challenges for texture classification due to its uncontrolled acquisition conditions. Each class exhibits considerable variations in scale, orientation, illumination, and viewpoint, while inter-class similarity in visual patterns and structural appearance leads to limited discriminative cues. Table 4 reports the classification results of DMCE-Net using different backbone networks on the UIUC dataset, averaged over 50 random splits.

As shown, DMCE-Net with AlexNet performs slightly below its VGG-VD16 and VGG-VD19 counterparts. Specifically, the DMCE-Net with AlexNet achieves perfect classification on three classes (Classes 7, 16, and 19), while the DMCE-Net with VGG-VD16 and VGG-VD19 correctly classify 16 and 14 classes, respectively. The poorest performance for DMCE-Net with AlexNet is observed on Classes 3 and 14. In contrast, DMCE-Net with VGG-VD16 and VGG-VD19 achieves 100% and 99.50% accuracy on Category 3, respectively. For Category 14, the DMCE-Net with VGG-VD16 and VGG-VD19 outperform the DMCE-Net with AlexNet by 3.25% and 6.00%, respectively. Similar trends are observed across other categories in the dataset. The primary advantage stems from the architectural depth of VGG-VD16 and VGG-VD19, which contain 16 and 19 weight layers, respectively—substantially deeper than the 8-layer structure of AlexNet. This increased depth facilitates the extraction of more complex and abstract hierarchical features, which is essential for capturing fine-grained texture patterns. Moreover, in contrast to AlexNet's use of large convolutional kernels (e.g., 11×11 and 5×5), the VGG networks adopt uniform 3×3 kernels throughout. This design enables finer localization of texture details and better preservation of discriminative micro-patterns, thereby improving overall texture classification performance.

Figure 12 illustrates the classification performance comparison between the proposed DMCE-Net and state-of-the-art methods (including VisGraphNet[48], ELMP-Net[20], Deep Fractal Interpolation[49], DSTNet[50],

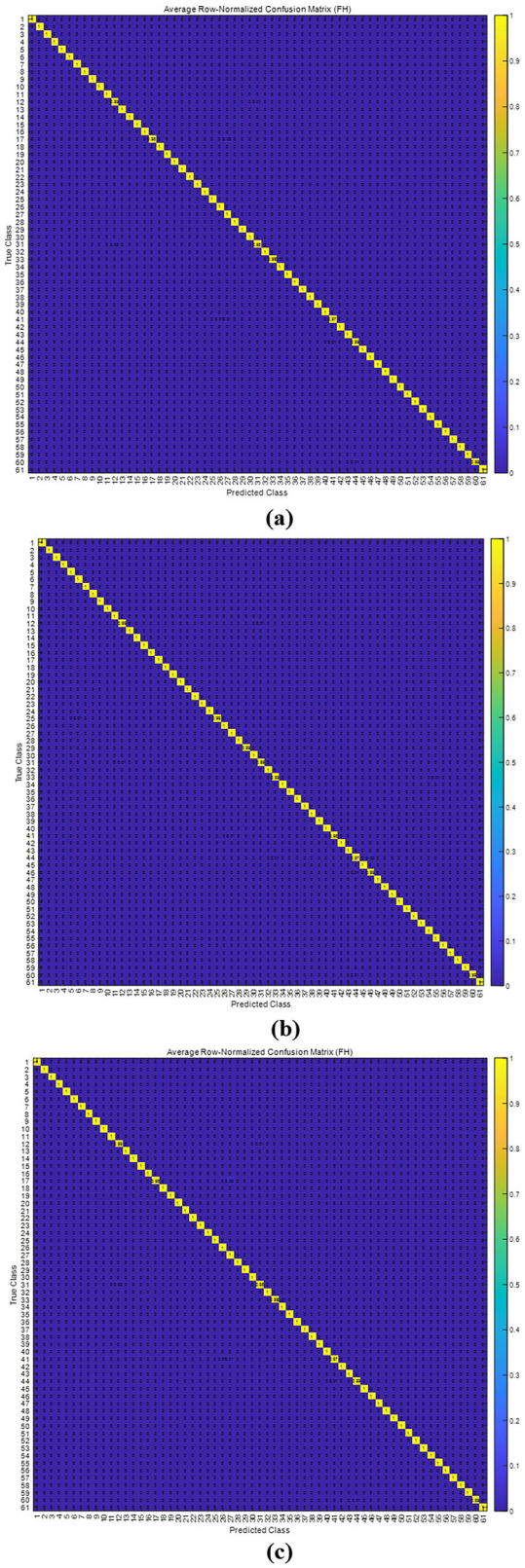

Fig. 14 | Confusion matrix of DMCE-Net with different backbones on the CUReT dataset. This figure shows the confusion matrices of the proposed DMCE-Net using three different backbone networks on the CUReT dataset. **a** It corresponds to DMCE-Net with AlexNet, **b** shows the results with VGG-VD16, and **c** presents the results with VGG-VD19.

DeCAF[51] and FC-CNN[52] with VGGM and AlexNet) on the UIUC dataset. As illustrated in Fig. 10, the proposed DMCE-Net with VGG-VD19 achieves the highest classification accuracy among the compared methods at 99.50%, exceeding the versions based on VGG-VD16 and AlexNet by 0.03% and 2.56%, respectively. Compared to FC-CNN with AlexNet, DMCE-Net with AlexNet achieves an improvement of 5.84%. Moreover, DMCE-Net generally outperforms several recently proposed deep networks for texture classification. Specifically, DMCE-Net with VGG-VD19 surpasses VisGrapNet, ELMP-Net, DSTNet, and DeCAF by 1.50%, 3.20%, 5.90%, and 5.30%, respectively. These results clearly demonstrate the effectiveness of the proposed DMCE-Net in capturing discriminative texture features, attributed to its dual-stream design incorporating intra-layer and inter-layer encoding streams.

The UMD texture classification dataset comprises texture images captured under complex real-world imaging conditions, and is specifically designed to comprehensively evaluate the robustness of texture classification methods with respect to sampling environments, as well as their scale, viewpoint, and illumination invariance. Table 5 summarizes the classification performance of the proposed DMCE-Net using different backbone networks across 25 texture categories in the UMD dataset. As observed, DMCE-Net with AlexNet successfully classified 16 categories, with its lowest accuracy recorded at 97.50% in Category 1. In comparison, DMCE-Net with VGG-VD16 and DMCE-Net with VGG-VD19 both achieved 100% classification accuracy in Category 1. Moreover, DMCE-Net with VGG-VD16 correctly classified 21 categories, whereas DMCE-Net with VGG-VD19 failed to achieve perfect classification in only Category 3 (96.75%) and Category 10 (98.50%). These results clearly indicate that DMCE-Net with VGG-VD19 and VGG-VD16 delivers superior classification performance on the UMD dataset compared to its counterpart based on AlexNet.

Figure 13 compares the classification performance of the proposed DMCE-Net with state-of-the-art methods reported in the literature. As illustrated in Fig. 13, DMCE-Net with VGG-VD16 achieved the best classification accuracy among the compared methods at 99.89%, slightly outperforming DMCE-Net with VGG-VD19, and surpassing DMCE-Net with AlexNet by 0.52%. Moreover, DMCE-Net with AlexNet improved upon FC-CNN with AlexNet by 3.47% in classification accuracy. Compared to several recently proposed deep learning methods for texture image classification, the proposed DMCE-Net shows a measurable performance advantage. For instance, DMCE-Net with VGG-VD16 outperforms VisGraphNet, ELMP-Net, DSTNet, and DeCAF by 1.49%, 0.69%, 1.39%, and 3.49%, respectively. These results clearly indicate that the proposed DMCE-Net effectively handles the complex imaging condition variations present in the UMD dataset and generally outperforms several state-of-the-art deep learning approaches in terms of classification accuracy. This can be primarily attributed to the Dual-stream architecture of the proposed DMCE-Net, which not only maintains the independence of expert-specific features but also fosters mutual learning and interaction of texture attributes across different hierarchical levels. This design effectively addresses the complex imaging challenges posed by inter-class similarity and intra-class variability in texture representation.

To further evaluate the effectiveness of the proposed DMCE-Net in handling complex texture classification tasks, comprehensive experiments were conducted on the CUReT dataset. In addition to its diverse and challenging imaging conditions, CUReT includes a substantial number of images affected by specular reflections and self-shadowing, which introduce additional complexity and serve as rigorous benchmarks for assessing the robustness of texture analysis approaches.

Figure 14 presents the confusion matrix of DMCE-Net with different backbone networks on the CUReT dataset. As illustrated in the figure, DMCE-Net shows generally good classification performance across the majority of texture categories in the CUReT dataset. Even for samples

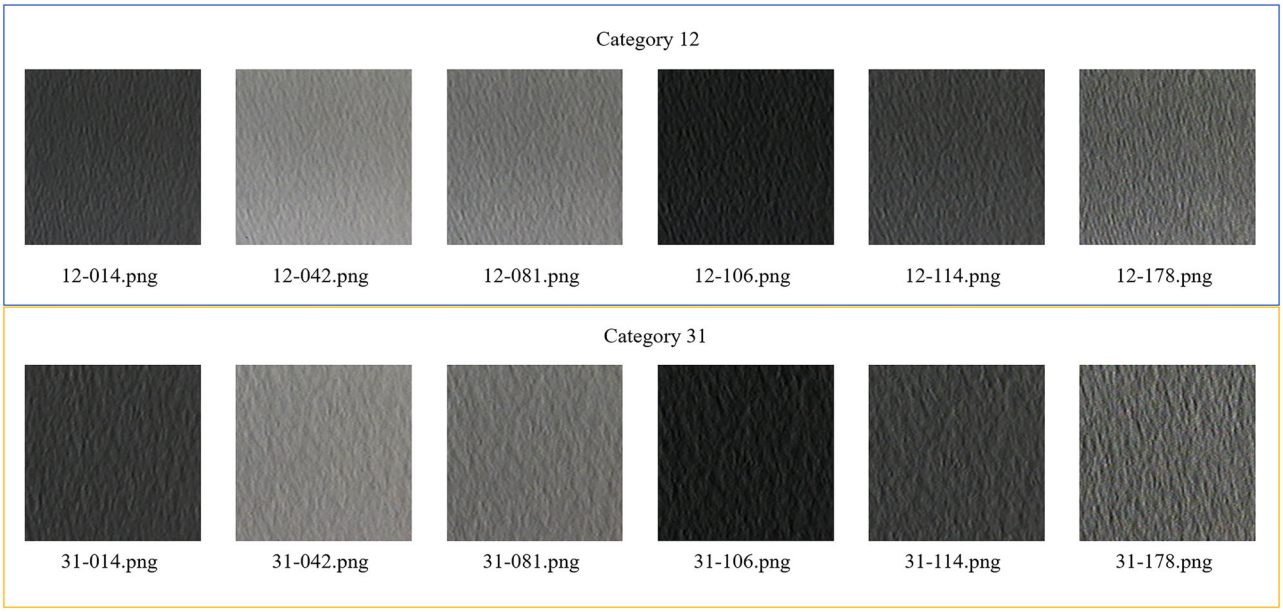

Fig. 15 | Some sample instances from Category 12 and Category 31 of the CUReT dataset. This figure displays representative image samples from Category 12 and Category 31 of the CUReT dataset. These two categories are selected to illustrate the visual similarity that often leads to misclassification.

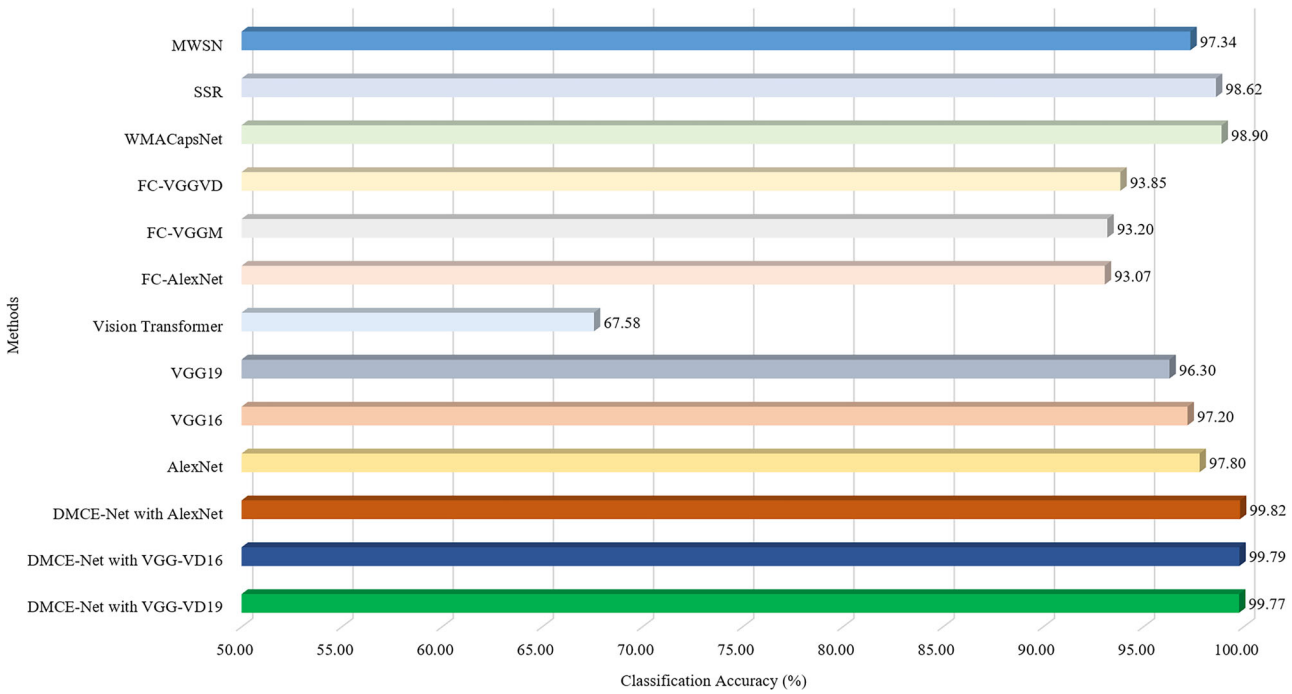

Fig. 16 | Comparison between DMCE-Net with various backbones and state-of-the-art methods on the CUReT dataset. This figure presents a comparative analysis of the proposed DMCE-Net, implemented with different backbone networks, against several state-of-the-art texture classification methods on the CUReT dataset.

exhibiting high inter-class similarity, the model achieves superior classification accuracy. In particular, for Category 12, DMCE-Net with all three backbone networks (i.e., AlexNet, VGG-VD16, and VGG-VD19) achieves a classification accuracy of up to 99%. However, each of them misclassifies 1% of the samples as Category 31. Similarly, the classification accuracies for Category 31 are 98%, 99%, and 98% for DMCE-Net with AlexNet, VGG-VD16, and VGG-VD19, respectively, with 2%, 1%, and 2% of the samples being incorrectly classified as Category 12. Figure 15 presents example samples from both Category 12 and Category 31, which visually exhibit extremely high inter-class similarity. A similar observation can be made for Category 61 and Category 44. These findings highlight the substantial challenge posed by the extreme inter-class similarity and intra-class variability inherent in texture image classification. The proposed DMCE-Net appears to mitigate this issue by leveraging complementary information derived from expert-specific feature mappings at different depths through intra-layer and inter-layer encoding streams. This design enables more efficient capture of local texture details and allows the model to maintain robust classification performance even under conditions of high inter-class similarity and intra-class variation.

Table 6 | Classification accuracy (%) and average running time (s) on the CUReT dataset

Method	Classification accuracy	Average running time
LBP	95.96	0.0470
CLBP	97.00	0.0720
SWOBP	97.06	0.2470
MCLBP+M	98.51	1.5600
FC-AlexNet	93.07	0.0175
FC-VGGM	93.20	0.0231
FC-VGGVD	93.85	0.0348
DMCE-Net with AlexNet	99.82	0.0393
DMCE-Net with VGG-VD16	99.79	0.1478
DMCE-Net with VGG-VD19	99.77	0.1636

Figure 16 presents a comparison between the highest classification accuracy previously reported in the literature on the CUReT dataset (including MWSN[53], SSR[54], WMACapsNet[55], FC-VGGVD[52], FC-VGGM[52], FC-AlexNet[52] and Vision Transformer[56]) and the accuracy achieved by the proposed DMCE-Net. As shown in Fig. 16, DMCE-Net with AlexNet achieves the highest classification accuracy among the compared methods on the CUReT dataset, reaching 99.82%, which constitutes a 2.02% improvement over the baseline AlexNet. Similarly, DMCE-Net with VGG-16 outperforms the original VGG-VD16 by 2.59%, while DMCE-Net with VGG-VD19 delivers a 3.47% performance gain compared to its corresponding backbone. These results demonstrate that the proposed DMCE-Net provides an enhancement in texture representation by leveraging mutual learning and cross-encoding mechanisms among expert-level feature mappings at different network depths. In addition to improvements over its own backbones, DMCE-Net also exhibits clear performance advantages compared with other state-of-the-art deep learning methods. For instance, the classification accuracy of DMCE-Net with AlexNet surpasses that of the Vision Transformer by 32.24%. This performance gap is primarily due to the fact that Vision Transformers are better suited for modeling global relationships, whereas texture recognition typically relies on strong local structural awareness. This underscores our choice of convolutional neural networks as the foundational backbone in the proposed framework. The proposed DMCE-Net with AlexNet also surpasses FC-AlexNet by 6.75% in classification accuracy. Moreover, it outperforms MWSN, SSR, and WMACapsNet by 2.48%, 1.20%, and 0.92%, respectively. These experimental provide additional evidence supporting the effectiveness of DMCE-Net's architectural design. Through the integration of intra-layer and inter-layer encoding streams, the model facilitates mutual learning and joint representation across deep expert mappings. By incorporating the intra-layer binary encoding strategy, multi-attribute joint encoding strategy, and cross-layer binary encoding mechanism, DMCE-Net enables comprehensive and fine-grained learning of diverse texture attributes. Consequently, it contributes to improved discriminative capacity, indicating that it can serve as a promising framework for texture analysis and related applications.

Table 6 presents a comparison of the classification accuracy and average running time between representative binary pattern–based methods (including LBP[57], CLBP[12], SWOBP[58], and MCLBP+M[59]) and deep learning–based approaches. As observed, the proposed DMCE-Net consistently achieves higher classification accuracy than both categories of methods. In terms of computational efficiency, the feature extraction time of DMCE-Net depends on the choice of backbone network. When employing the lightweight AlexNet backbone, DMCE-Net achieves shorter running time than all binary pattern–based methods, while remaining slightly slower than FC-AlexNet, FC-VGGM, and FC-VGGVD. These results indicate that DMCE-Net is capable of delivering competitive computational efficiency while maintaining superior classification performance.

Discussion

To address the challenges of complex inter-class similarity and intra-class variability commonly encountered in architectural textures, this paper proposes a DMCE-Net. The proposed framework treats different feature mapping layers within the deep backbone as expert modules, each capturing distinct domain-specific representations. A dual-stream architecture is introduced, wherein the intra-layer encoding stream facilitates the joint representation of independently learned expert knowledge, while the inter-layer encoding stream enables mutual learning and integration across these expert modules. Comprehensive experiments conducted on the AHE dataset demonstrate that DMCE-Net effectively addresses the intricate texture classification tasks associated with architectural relic surfaces. In addition, evaluation on three benchmark texture classification datasets further validates the robustness of DMCE-Net in handling significant inter-class similarity and intra-class variation across diverse texture scenarios. The integration of a dual-stream architecture, multi-attribute joint encoding strategy, and cross-layer binary encoding mechanism provides a strong theoretical foundation for the effective capture of deep texture attributes. However, this result also highlights a limitation of DMCE-Net when confronted with samples characterized by extensive background interference and extremely sparse target features, which may lead to reduced discriminative power. The demonstrated capability of DMCE-Net in texture analysis has the potential to significantly advance various texture classification applications. Future work will focus on extending this framework to broader domains, such as remote sensing and fine-grained visual recognition.

Data availability

The datasets analyzed during the current study are publicly available at the following sources: The AHE dataset is available at https://old.datahub.io/dataset/architectural-heritage-elements-image-dataset. The UIUC dataset can be accessed at https://slazebni.cs.illinois.edu. The UMD dataset is available at https://users.umiacs.umd.edu/~fer/High-resolution-data-base/hr_database.htm. The CUReT dataset is available at https://www.cs.columbia.edu/CAVE/software/curet/.

Received: 19 April 2025; Accepted: 22 September 2025;
Published online: 06 October 2025

References

1. Croce, V., Caroti, G., Piemonte, A., De Luca, L. & Véron, P. H-BIM and artificial intelligence: classification of architectural heritage for semi-automatic scan-to-BIM reconstruction. *Sensors* **23**, 2497 (2023).
2. Siountri, K. & Anagnostopoulos, C. N. The classification of cultural heritage buildings in athens using deep learning techniques. *Heritage* **6**, 3673–3705 (2023).
3. Pintus, R., Pal, K., Yang, Y., Weyrich, T., Gobbetti, E. & Rushmeier, H. A survey of geometric analysis in cultural heritage. *Comput. Graph. Forum.* **35**, 4–31 (2016).
4. Grilli, E. & Remondino, F. Classification of 3D digital heritage. *Remote Sens.* **11**, 847 (2019).
5. Chowdhury, S. & Soni, B. R.-V. Q. A. A robust visual question answering model. *Knowl. Based Syst.* **309**, 112827 (2025).
6. Chowdhury, S. & Soni, B. Beyond words: ESC-Net revolutionizes VQA by elevating visual features and defying language priors. *Comput. Intell.* **40**, e70010 (2024).
7. Chowdhury, S. & Soni, B. Envqa: Improving visual question answering model by enriching the visual feature. *Eng. Appl. Artif. Intell.* **142**, 109948 (2025).
8. Chowdhury, S. & Soni, B. Handling language prior and compositional reasoning issues in Visual Question Answering system. *Neurocomputing* **635**, 129906 (2025).

9. Zhang, X., Xu, C., Fan, G., Hua, Z., Li, J. & Zhou, J. FSCMF: a dual-branch frequency-spatial joint perception cross-modality network for visible and infrared image fusion. *Neurocomputing* **641**, 130376 (2025).
10. Bianconi, F., Fernández, A., Smeraldi, F. & Pascoletti, G. Colour and texture descriptors for visual recognition: a historical overview. *J. Imaging* **7**, 245 (2021).
11. Liu, L., Fieguth, P., Guo, Y., Wang, X. & Pietikäinen, M. Local binary features for texture classification: taxonomy and experimental study. *Pattern Recognit.* **62**, 135–160 (2017).
12. Guo, Z., Zhang, L. & Zhang, D. A completed modeling of local binary pattern operator for texture classification. *IEEE Trans. Image Process.* **19**, 1657–1663 (2010).
13. Zhai, W., Cao, Y., Zha, Z. J., Xie, H. & Wu, F. Deep structure-revealed network for texture recognition. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11010–11019 (IEEE, 2020).
14. Mohammed, K. M. C. & Prasad, G. Defective texture classification using optimized neural network structure. *Pattern Recognit. Lett.* **135**, 228–236 (2020).
15. Song, K., Yang, H. & Yin, Z. Multi-scale boosting feature encoding network for texture recognition. *IEEE Trans. Circuits Syst. Video Technol.* **31**, 4269–4282 (2021).
16. Condori, R. H. & Bruno, O. M. Analysis of activation maps through global pooling measurements for texture classification. *Inf. Sci.* **555**, 260–279 (2021).
17. Wu, H., Yan, W., Li, P. & Wen, Z. Deep texture exemplar extraction based on trimmed T-CNN. *IEEE Trans. Multimedia* **23**, 4502–4514 (2020).
18. Lee, S. H., Yu, W. F. & Yang, C. S. ILBPSDNet: based on improved local binary pattern shallow deep convolutional neural network for character recognition. *IET Image Process.* **16**, 669–680 (2022).
19. Florindo, J. B. Renyi entropy analysis of a deep convolutional representation for texture recognition. *Appl. Soft Comput.* **149**, 110974 (2023).
20. Florindo, J. B., Backes, A. R. & Neckel, A. ELMP-Net: The successive application of a randomized local transform for texture classification. *Pattern Recognit.* **153**, 110499 (2024).
21. Florindo, J. B. Fractal pooling: a new strategy for texture recognition using convolutional neural networks. *Expert Syst. Appl.* **243**, 122978 (2024).
22. Chen, Z., Quan, Y., Xu, R., Jin, L. & Xu, Y. Enhancing texture representation with deep tracing pattern encoding. *Pattern Recognit.* **146**, 109959 (2024).
23. Lyra, L. O., Fabris, A. E. & Florindo, J. B. A multilevel pooling scheme in convolutional neural networks for texture image recognition. *Appl. Soft Comput.* **152**, 111282 (2024).
24. Pavlopoulos, S., Kyriacou, E., Koutsouris, D., Blekas, K., Stafylopatis, A. & Zoumpoulis, P. Fuzzy neural network-based texture analysis of ultrasonic images. *IEEE Eng. Med. Biol. Mag.* **19**, 39–47 (2000).
25. Saihood, A., Karshenas, H. & Naghsh-Nilchi, A. R. Multi-Orientation local texture features for guided attention-based fusion in lung nodule classification. *IEEE Access* **11**, 17555–17568 (2023).
26. Yang, M., Jiao, L., Liu, F., Hou, B., Yang, S., Zhang, Y. & Wang, J. Coarse-to-fine contrastive self-supervised feature learning for land-cover classification in SAR images with limited labeled data. *IEEE Trans. Image Process.* **31**, 6502–6516 (2022).
27. Li, M., Hu, Z., Qiu, S., Zhou, C., Weng, J., Dong, Q. & Zhou, M. Dual-branch hybrid encoding embedded network for histopathology image classification. *Phys. Med. Biol.* **68**, 195002 (2023).
28. Moinuddin, M., Khan, S., Alsaggaf, A. U., Abdulaal, M. J., Al-Saggaf, U. M. & Ye, J. C. Medical ultrasound image speckle reduction and resolution enhancement using texture compensated multi-resolution convolution neural network. *Front. Physiol.* **13**, 961571 (2022).
29. Lv, G., Gao, X., Dong, A., Wei, Z. & Cheng, J. SLFusion: a structure-aware infrared and visible image fusion network for low-light scenes. *IEEE Trans. Circuits Syst. Video Technol*. **early access,** 1–1 (2025).
30. Han, F., Wang, H., Zhang, G., Han, H., Song, B., Li, L., Moore, W., Lu, H., Zhao, H. & Liang, Z. Texture feature analysis for computer-aided diagnosis on pulmonary nodules. *J. Digit. Imaging* **28**, 99–115 (2015).
31. Câmara, A., de Almeida, A., Caçador, D. & Oliveira, J. Automated methods for image detection of cultural heritage: Overviews and perspectives. *Archaeol. Prospect.* **30**, 153–69 (2023).
32. Humeau-Heurtier, A. Texture feature extraction methods: a survey. *IEEE Access* **7**, 8975–9000 (2019).
33. Andreetto, M., Brusco, N. & Cortelazzo, G. M. Automatic 3D modeling of textured cultural heritage objects. *IEEE Trans. Image Process.* **13**, 354–69 (2004).
34. Li, X., Yang, H., Chen, C., Zhao, G. & Ni, J. Deterioration identification of stone cultural heritage based on hyperspectral image texture features. *J. Cult. Herit.* **69**, 57–66 (2024).
35. Earl, G., Martinez, K. & Malzbender, T. Archaeological applications of polynomial texture mapping: analysis, conservation and representation. *J. Archaeol. Sci.* **37**, 2040–50 (2010).
36. Artopoulos, G., Maslioukova, M. I., Zavou, C., Loizou, M., Deligiorgi, M. & Averkiou, M. An artificial neural network framework for classifying the style of cypriot hybrid examples of built heritage in 3D. *J. Cult. Herit.* **63**, 135–47 (2023).
37. Fan, T., Wang, H. & Deng, S. Intangible cultural heritage image classification with multimodal attention and hierarchical fusion. *Expert Syst. Appl.* **231**, 120555 (2023).
38. Li, R., Geng, G., Wang, X., Qin, Y., Liu, Y., Zhou, P. & Zhang, H. LBCapsNet: a lightweight balanced capsule framework for image classification of porcelain fragments. *Herit. Sci.* **12**, 133 (2024).
39. Gu, S., Ma, J., Hui, G., Xiao, Q. & Shi, W. STMT: Spatio-temporal memory transformer for multi-object tracking. *Appl. Intell.* **53**, 23426–23441 (2023).
40. Gu, S., Zhang, M., Xiao, Q. & Shi, W. Cascaded matching based on detection box area for multi-object tracking. *Knowl. Based Syst.* **299**, 112075 (2024).
41. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems* 25 (Curran Associates Inc., 2012).
42. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. Preprint at https://arxiv.org/abs/1409.1556 (2014).
43. Chang, C. C. & Lin, C. J. LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2**, 1–27 (2011).
44. Datahub. https://old.datahub.io/dataset/architectural-heritage-elements-image-dataset.
45. Lazebnik, S., Schmid, C. & Ponce, J. A sparse texture representation using local affine regions. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**, 1265–1278 (2005).
46. Xu, Y., Ji. H. & Fermuller, C. A projective invariant for textures. In *Proc. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2**, 1932–1939 (IEEE, 2006).
47. Dana, K. J., van Ginneken, B., Nayar, S. K. & Koenderink, J. J. Reflectance and texture of real-world surfaces. *ACM Trans. Graph.* **18**, 1–34 (1999).
48. Florindo, J. B., Lee, Y. S., Jun, K., Jeon, G. & Albertini, M. K. VisGraphNet: a complex network interpretation of convolutional neural features. *Inf. Sci.* **543**, 296–308 (2021).
49. Florindo, J. & Bruno, O. M. Using fractal interpolation over complex network modeling of deep texture representation. In *Proc. 2022 Eleventh International Conference on Image Processing Theory, Tools and Applications*. 1–5 (IEEE, 2022)
50. Florindo, J. B. DSTNet: Successive applications of the discrete Schroedinger transform for texture recognition. *Inf. Sci.* **507**, 356–364 (2020).
51. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S. & Vedaldi, A. Describing textures in the wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 3606–3613 (IEEE, 2014).

52. Cimpoi, M., Maji, S. & Vedaldi, A. Deep filter banks for texture recognition and segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 3828–3836 (IEEE, 2015).
53. Chak, W. H. & Saito, N. Monogenic wavelet scattering network for texture image classification. *JSIAM Lett.* **15**, 21–24 (2023).
54. Ribas, L. C., Scabini, L. F., Condori, R. H. & Bruno, O. M. Color-texture classification based on spatio-spectral complex network representations. *Phys. A Stat. Mech. Appl.* **635**, 129518 (2024).
55. Tao, Z., Wei, T. & Li, J. Wavelet multi-level attention capsule network for texture classification. *IEEE Signal Process. Lett.* **28**, 1215–1219 (2021).
56. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T. & Houlsby, N. An image is worth 16x16 words: transformers for image recognition at scale. Preprint at https://arxiv.org/abs/2010.11929 (2020).
57. Ojala, T., Pietikainen, M. & Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern. Anal. Mach. Intell.* **24**, 971–987 (2002).
58. Song, T., Feng, J., Wang, S. & Xie, Y. Spatially weighted order binary pattern for color texture classification. *Expert Syst. Appl.* **147**, 113167 (2020).
59. Shu, X., Song, Z., Shi, J., Huang, S. & Wu, X. J. Multiple channels local binary pattern for color texture representation and classification. *Signal Process. Image Commun.* **98**, 116392 (2021).

Acknowledgements

The paper is funded by Fujian Natural Science Foundation Project (2023J05243), Fashu Charity Foundation Donation Fund Research Special (No. MFK23003), Fujian Province Young and Middle-aged Teacher Education Research Project (JAT220312), and Minjiang University Scientific Research Promotion Fund (MJY22015).

Author contributions

Xiaochun Xu directly participated in the Funding acquisition, Methodology, and Writing – original draft. Bin Li participated in the Formal analysis, Resources, Validation and Writing – review & editing. Q.M.Jonathan Wu participated in the Resources, Supervision, and Writing – review & editing.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Bin Li.