DFAEN: Double-Order Knowledge Fusion and Attentional Encoding Network for Texture Recognition

Zhijing Yang[a,1], Shujian Lai[a,1], Xiaobin Hong[b,**], Yukai Shi[a], Yongqiang Cheng[c] and Chunmei Qing[d]

[a]School of Information Engineering, Guangdong University of Technology, Guangzhou, 510006, China
[b]School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510641, China
[c]Department of Computer Science and Technology, University of Hull, Hull, UK
[d]School of electronic and information engineeering, South China University of Technology, Guangzhou 510641, China

ARTICLE INFO

Keywords:

Deep learning

Information fusion

Attentional mechanism

Texture recognition

ABSTRACT

Recent studies have shown that deep convolutional neural networks (CNNs) have been successfully used for texture representation and recognition. One of the most successful texture recognition methods is the deep texture encoding network (DeepTEN), which has been shown to be effective. However, this network directly uses redundant CNN features with generality and ignores the role of multiorder information during the encoding and learning processes. To address these issues, this paper proposes a double-order knowledge fusion and attentional encoding network for texture recognition (DFAEN). First, crucial texture features are encoded by an embedded attention mechanism. Second, double-order modeling is implemented in the encoding and learning stage to make full use of convolution feature information with different orders, enabling the network to focus on and learn more texture domain information. Our method can stably and effectively perform end-to-end optimization. Evaluation experiments conducted on several widely used benchmark datasets (e.g., the FMD, MINC-2500, the DTD, KTH-TISP-2b, and GTOS-mobile) show that our method clearly demonstrates superior performance to that of competing approaches.

**Corresponding author.
[1]Zhijing Yang and Shujian Lai contributed equally in this work. (E-mail address: yzhj@gdut.edu.cn (Z. Yang), laishujian@163.com (S. Lai), mexbhong@scut.edu.cn (X. Hong), ykshi@gdut.edu.cn (Y. Shi), Y.Cheng@hull.ac.uk (Y. Cheng), qchm@scut.edu.cn (C. Qing).)

1. Introduction

Textures widely exist on the surfaces of all objects and form key feature information. Texture recognition is an important research topic in many computer vision tasks, including material recognition (Fekri-Ershad, 2020), image segmentation (Medeiros et al., 2016), scene analysis (Arandjelović et al., 2016), and facial recognition (Lee and Lee, 2016). For example, in industrial production scenarios, workers detect the quality of products by comparing their texture features (Uzen et al., 2021); in autonomous driving and navigation applications, it is necessary to analyze the textures of various real-world objects to precisely determine driving terrains and scenes (Xue et al., 2018); in medicine, medical images contain large amounts of texture information, and doctors diagnose patient lesions by observing the texture features of these images (Alfed and Khelifi, 2017). Notably, research on texture features helps humans understand many visual problems and promotes the solving of all kinds of computer vision problems.

The texture representation of a texture image is obtained by manual feature extraction or feature learning, usually producing a set of feature vectors. A classifier can use this texture representation for texture recognition. Thus, texture representation aims to extract important visual cues from an image to identify texture features. For example, the classic methods include gray-level cooccurrence matrices (GLCMs) (Haralick et al., 1973), local binary patterns (LBPs) (Ojala et al., 2002), vectors of locally aggregated descriptors (VLADs) (Jégou et al., 2010) and Fisher vectors (FVs) (Sánchez et al., 2013) based on second-order encoding. Recently, convolutional neural networks (CNNs) (Krizhevsky et al., 2012) have demonstrated their effectiveness in terms of many aspects and have become the main backbones of state-of-the-art texture recognition and classification methods. By combining a CNN pretrained on the ImageNet dataset with traditional methods (e.g., VLADs and FVs), the resulting texture recognition performance can be significantly improved. However, these methods have multistage pipelines, which typically include feature extraction modules, feature encoding modules, and support vector machine (SVM) classifiers. Each stage is

learned separately without global optimization. In classic image classification tasks based on deep learning, max pooling or average pooling is used to process convolutional feature information. The feature information processed by the pooling layer belongs to the category of first-order statistics. The first-order information becomes simple and fast based on the pooling layer and can be effectively used in classic image classification tasks. The first-order information extracted by a network usually refers to the overall characteristics of the input image, such as its contour, shape, and color information. The network realizes image recognition by learning the image feature information. However, in the texture recognition task, texture images belonging to different categories may behave very similarly in terms of features such as their contours, shapes, and colors. Single-order information cannot yield good recognition performance. Therefore, second-order information can be introduced into the utilized network. When the network uses single-order information to reach a performance bottleneck, second-order information can further improve the network performance. This second-order information contains the mutuality between texture image features, which can better capture the differences between these features. Therefore, Lin et al. (2018) proposed a bilinear CNN (BCNN) model for texture recognition. They modeled the second-order information of deep convolution features and achieved excellent performance on texture datasets in an end-to-end manner. However, the functions learned by a bilinear model are too general for learning domain-specific information. Since texture pictures hide many distinguishable visual texture details, they have inner similarity. For example, in the DTD (Cimpoi et al., 2014) dataset shown in Fig. 4(a), similar texture attributes appear in 3 different categories at the same time, and they are all dominated by the visual texture attributes of the grid. Hence, a bilinear model and deep learning fail to capture adequate and distinguishable visual texture signals for texture recognition. Later, to enable more effective network learning, Dai et al. (2017) integrated first-order convolution information on the basis of a BCNN. Recently, Zhang et al. (2017) combined the texture encoding layer (TEL) into a pretrained CNN model and proposed an end-to-end learning framework, the deep

texture encoding network (DeepTEN), to achieve state-of-the-art performance. They unified dictionary learning and residual encoding in a TEL, which further promoted the application of VLADs in texture recognition. However, their method only uses single-order texture encoding information, which fails to capture abundant distinguishable textural visual details for recognition purposes. In addition, during the feature extraction stage, if a pretrained general CNN model is directly deployed in the texture encoding phase, the network never focuses on the distinguishable visual features of the texture image. In other words, trivial convolution feature information is focused on the current recognition task. This negatively affects texture encoding and brings a drawback to the whole end-to-end training process.

To this end, this paper proposes a double-order knowledge fusion and attentional encoding network (DFAEN), which is different from previous methods, such as DeepTEN (Zhang et al., 2017), the BCNN (Lin et al., 2018), and FASON (Dai et al., 2017). These methods were modeled in a single CNN, essentially utilizing general CNN features for texture recognition. Our approach integrates the CNN feature learning stage and the encoding learning stage into a network. The purpose of the CNN feature learning stage is to learn crucial convolution features for encoding and learning, and the encoding and learning stage aims to obtain richer texture domain information for recognition. First, in the feature extraction stage before entering the TEK, we incorporate an attention mechanism into a pretraining CNN to build a frequency attention-based feature extraction network (FAEN). This network learns the importance levels of frequency feature channels based on a frequency channel index and therefore focuses on the important distinguishable convolution features for the current texture recognition task. Moreover, this paper proposes a double-order texture information encoding layer (DTEL). To focus on and learn more texture domain information to improve the performance of the classifier, we model double-order information in the encoding and learning stage. Compared with DeepTEN (Zhang et al., 2017), our architecture achieves more effective learning throughout its end-to-end training process. Additionally, our method makes full use of first- and second-order texture encoding information and clearly captures

more texture details. Experiments show that the proposed method obtains promising results on state-of-the-art benchmark datasets (e.g., material datasets, texture datasets, and outdoor scene datasets) that are superior to those of competing technical methods.

Our work contributions can be summarized in the following four items.

- A deep fusion network is designed to integrate the crucial feature learning and double-order information encoding and learning stages into an end-to-end network. Each component benefits from the flow of the gradient during backpropagation and can learn and adjust in a mutually reinforcing manner.
- An effective feature representation learning method is developed by combining an attention mechanism with encoding and learning to concentrate on texture features.
- To capture texture information from different orders to improve the performance of the developed classifier, the double-order information is modeled in the encoding and learning stage to make the network focus on and learn richer texture domain information.
- Our approach achieves excellent experimental results on 5 challenging datasets (the FMD (Sharan et al., 2013, 2009), the DTD (Cimpoi et al., 2014), MINC-2500 (Bell et al., 2015), KTH-T2b (Caputo et al., 2005), and GTOS-mobile (Xue et al., 2018)).

The rest of this paper is structured as follows. We first introduce the previously developed texture description methods and discuss the related works in Section 2. Then, Section 3 elaborates on the design of our architecture and introduces each submodule. In Section 4, we explain and analyze the experimental results obtained on state-of-the-art benchmark texture datasets to illustrate the effectiveness of our method. Finally, the conclusion is summarized in Section 5.

2. Related Work

Due to the subtle differences in texture patterns caused by changes in brightness, rotation, scaling, etc., such differences are usually manifested as large intraclass appearance and small interclass appearance differences. This requires the constructed texture representation to have strong robustness, and therefore, texture recognition is more difficult than traditional image recognition. The key to the texture recognition task is to design a powerful texture descriptor. Therefore, in recent decades, texture descriptors have been a focus of texture recognition task research. Early methods included GLCMs, LBPs, filter bank responses (Leung and Malik, 2001) and other hand-designed descriptors. The bag-of-words (BoWs) method (Sivic et al., 2005) learns a dictionary offline, encodes the observed features, and finally learns the appropriate classification. Later, powerful texture descriptors such as VLADs, FVs, 2D LBPs (Cerkezi and Topal, 2020), and LDTPs (El khadiri et al., 2018) further improved the performance achieved on texture recognition tasks.

Some recent work has shown (Chen et al., 2021; He et al., 2016; Chen et al., 2020) that CNNs have considerable performance in computer vision tasks. CNNs can learn efficacious features from images, which can be effectively used for image recognition. Cimpoi et al. (2015) combined an FV with a CNN to achieve improved texture classification performance. Similarly, Arandjelović et al. (2016) integrated a VLAD with a CNN and effectively applied their method in scene recognition. Lin et al. (2018) built a powerful image representation by performing second-order modeling in a deep learning framework. However, the high-dimensional bilinear feature (Lin et al., 2018) produced by their model requires much memory to be implemented. Later, Lin et al. (2018) improved the bilinear model and proposed compact bilinear pooling (Gao et al., 2016) by using randomization (RM) and tensor sketching (TS) to reduce the dimensionality of the bilinear model. The FASON (Dai et al., 2017) proposed by Dai et al. further enhanced the integration of deep features. The authors combined first- and second-order information into an existing deep neural network framework and proved that both the first- and second-order information are beneficial for

texture classification and recognition (Lin et al., 2018). Tschannerl et al. (2019) and Chen et al. (2019) achieved improved texture classification performance through multifeature fusion. The locality-aware coding layer proposed by Bu et al. Bu et al. (2019) proved the effectiveness of the sparsity and locality of a pretrained CNN in texture recognition. These methods (Lin et al., 2018; Gao et al., 2016; Dai et al., 2017; Bu et al., 2019; Tschannerl et al., 2019; Chen et al., 2019) combine deep learning and texture description methods or increase the effectiveness of linear combinations of deep network features to achieve performance improvements. They merely adapted general-purpose CNNs to texture recognition, but they fail to learn texture domain information.

To enable a network to learn the feature information in the texture domain, DeepTEN, proposed by Zhang et al., integrates dictionary learning and residual encoding in a TEL, constructing an end-to-end framework (Zhang et al., 2017). DeepTEN has achieved improved classification accuracy on multiple texture datasets and has now become the state-of-the-art benchmark for texture recognition. Later, Xue et al. (2018) proposed a deep encoding and pooling (DEP) network, which combines texture encoding information with local spatial information for outdoor scene recognition. The DEP network has also achieved better recognition performance on other texture datasets (including outdoor scene datasets). However, these methods (Zhang et al., 2017; Xue et al., 2018) have two drawbacks. First, they only capture first-order texture domain information. Second, the redundancy of the features extracted from the CNN stream is ignored. Traditional attention mechanism-based models, such as Senet (Hu et al., 2020), typically use global average pooling (GAP) to obtain feature weight functions. Nevertheless, this process ignored the effects of other frequency components. However, Refs. (Qin et al., 2021; Fujieda et al., 2017) proved that conventional CNNs only use the low-frequency parts of features. Fujieda et al. (2017) integrated spectrum analysis into a CNN, which proved that low-frequency and high-frequency components both contribute to texture recognition. At the same time, Rippel et al. (2015) modeled a CNN in the spectral domain and demonstrated powerful representation

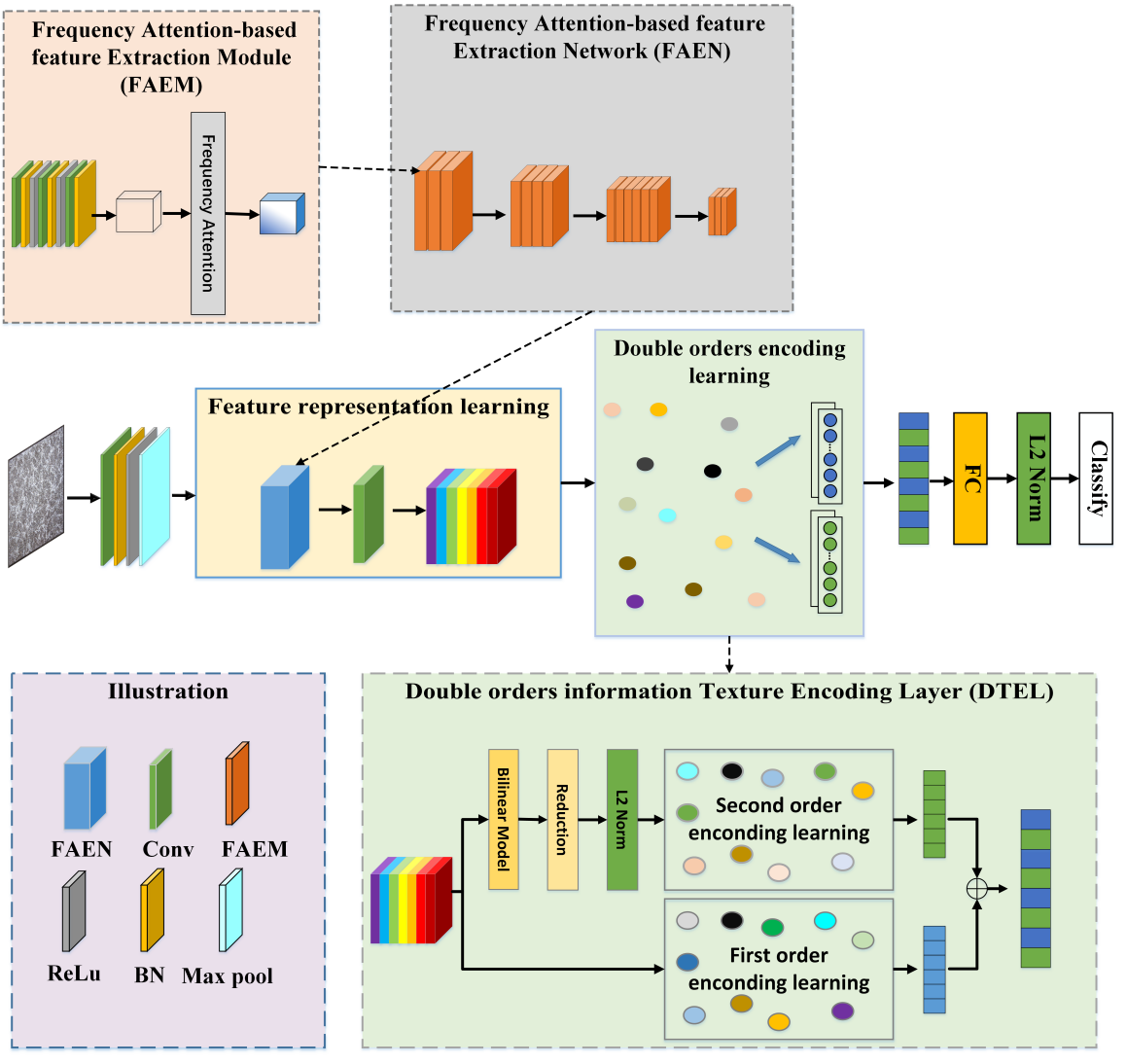


Fig. 1. The architecture of our proposed DFAEN method. The network captures the important features of the input image and performs texture encoding based on its double-order information. More specifically, the model concatenates the obtained first- and second-order texture encoding features, and then the proposed model generates predictions through the classification layer.

and efficient calculation approaches. To make full use of the spectrum information in each part, we introduce a frequency attention mechanism (Qin et al., 2021). A discrete cosine transform (DCT) is combined into the attention mechanism network, and all useful frequency domain components are added to the network for joint training. First, we integrate the frequency analysis-based attention mechanism into the feature extraction network for texture recognition, as this approach can filter out convolutional feature information that is

useless for encoding and make the network focus on crucial texture convolution features. Second, we apply the first- and second-order convolutional feature information into the encoding and learning stage to learn double-order texture domain features.

3. Methodology

In this section, we introduce the framework of our proposed DFAEN. Our architecture is shown in Fig. 1. First, we introduce the FAEN with multispectral channel attention, and then we describe the texture encoding module (DTEL) that combines the obtained first- and second-order feature information.

3.1. FAEN

In the feature extraction stage, the features extracted by the CNN usually contain much redundant feature information. The essential purpose of encoding is to compress these high-dimensional features and try to use low-dimensional disordered features for representation. Therefore, it is critical to prescreen the high-dimensional space before performing encoding. Feature screening aims to strengthen important features while suppressing unimportant features. We propose an FAEN that is suitable for texture recognition. We add a frequency attention mechanism to the texture encoding and learning module, as shown in Fig. 1. To provide context knowledge, we briefly introduce the derivation process of the frequency attention mechanism (Qin et al., 2021).

As shown in Fig. 2., the frequency attention mechanism is based on a DCT for frequency attention analysis. Suppose that the one-dimensional input signal is $x_i = \{0, 1, ..., T-1\}$; the corresponding one-dimensional DCT transformation can be defined as:

$$\begin{aligned} f_p &= \Sigma_{i=0}^{T-1} x_i cos(\frac{\pi p}{T}(i+\frac{1}{2})), \\ s.t.\ p &\in \{0, 1, ..., T-1\}. \end{aligned} \tag{1}$$

where $f \in \mathbb{R}^T$ is the DCT coefficient or spectrum. From Eq. (1), DCT can be viewed as the weighted sum of inputs. Furthermore, we can generalize this process to a 2D DCT:

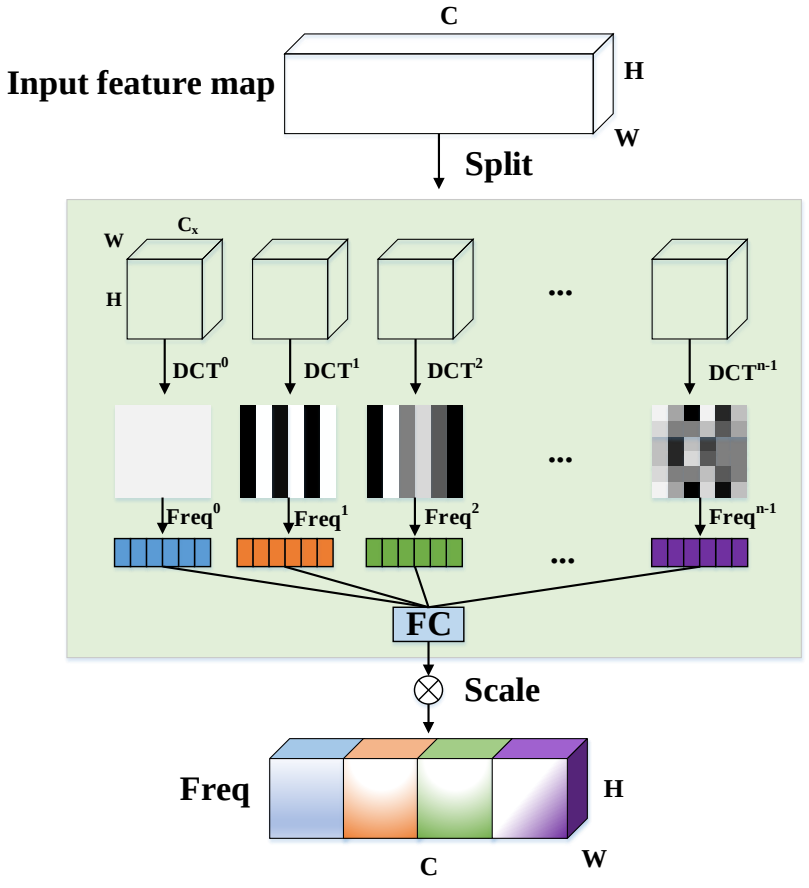


Fig. 2. Illustration of the frequency attention module. We can see that the frequency attention mechanism takes full advantage of the information contained in multiple frequency components by using a 2D DCT.

$$f_{h,w}^{2d} = \Sigma_{i=0}^{H-1}\Sigma_{j=0}^{W-1} x_{i,j}^{2d} cos(\frac{\pi h}{H}(i+\frac{1}{2}))cos(\frac{\pi w}{W}(j+\frac{1}{2})),$$
$$s.t.\ h \in \{0,1,...,H-1\}, \omega \in \{0,1,...,W-1\}. \tag{2}$$

where $f^{2d} \in \mathbb{R}^{H\times W}$ is the two-dimensional DCT spectrum and $x^{2d} \in \mathbb{R}^{H\times W}$ represents the two-dimensional input signal (referring to the feature map in the deep network). The positions of the pixels in the image are denoted by i and j. H and W are the height and width of x^{2d}, respectively. h and w are the frequencies of the basic cosine function; supposing that h and w in Eq. (2) are 0, we have:

$$
\begin{aligned}
f_{0,0}^{2d} &= \Sigma_{i=0}^{H-1}\Sigma_{j=0}^{W-1} x_{i,j}^{2d} cos(\frac{0}{H}(i+\frac{1}{2}))cos(\frac{0}{W}(j+\frac{1}{2})), \\
&= \Sigma_{i=0}^{H-1}\Sigma_{j=0}^{W-1} x_{i,j}^{2d}, \\
&= GAP(x^{2d})HW.
\end{aligned} \tag{3}
$$

In Eq. (3), $f_{0,0}^{2d}$ is proportional to GAP and is the component of the 2D DCT with the lowest frequency. However, the traditional channel attention mechanism utilizes the information with the lowest frequency in a GAP manner, and other frequency components and information are discarded. Therefore, we can add more frequency components to the channel attention mechanism by using the 2D DCT. For ease of understanding, we use B to represent the weight component of the 2D DCT:

$$
B_{h,w}^{i,j} = cos(\frac{\pi h}{H}(i+\frac{1}{2}))cos(\frac{\pi w}{W}(j+\frac{1}{2})). \tag{4}
$$

B is a constant in Eq. (4). Typically, the inverse 2D DCT is defined as:

$$
\begin{aligned}
x_{i,j}^{2d} &= \Sigma_{i=0}^{H-1}\Sigma_{j=0}^{W-1} f_{h,w}^{2d} cos(\frac{\pi h}{H}(i+\frac{1}{2}))cos(\frac{\pi w}{W}(j+\frac{1}{2})), \\
&= f_{0,0}^{2d}B_{0,0}^{i,j} + f_{0,1}^{2d}B_{0,1}^{i,j} + ... + f_{H-1,W-1}^{2d}B_{H-1,W-1}^{i,j} \\
&= GAP(x^{2d})HW B_{0,0}^{i,j} + f_{0,1}^{2d}B_{0,1}^{i,j} + ... + f_{H-1,W-1}^{2d}B_{H-1,W-1}^{i,j} \\
&s.t.\ h \in \{0,1,...,H-1\}, \omega \in \{0,1,...,W-1\}.
\end{aligned} \tag{5}
$$

By observing Eq. (5), we find that the input image can be represented as a combination of different frequency components. To integrate other frequency components into the channel attention mechanism, the input $X \in \mathbb{R}^{C\times H\times W}$ is divided into n parts according to their channels; these parts are denoted as $[X^0, X^1, ..., X^{n-1}]$, where $X^i \in R^{C_x\times H\times W}$, $i \in \{0,1,...,n-1\}$, and $C_x = \frac{C}{n}$. Note that n must be divisible by C. Therefore, a 2D DCT frequency component is assigned to each part as the preprocessing result of the channel attention mechanism.

$$
\begin{aligned}
Freq^i &= 2DDCT^{u,v}(X^i), \\
&= \Sigma_{i=0}^{H-1}\Sigma_{j=0}^{W-1} X^i_{:,h,w} B^{u,v}_{h,w} \\
&s.t.\ i \in \{0, 1, ..., n-1\},
\end{aligned} \tag{6}
$$

where $[u, v]$ represents the 2D index of the frequency component X^i and $Freq^i \in \mathbb{R}^{C_x}$ is the preprocessed frequency component of the i^{th} original feature map. Subsequently, as shown in Fig. 2, we can connect the various frequency components. In this way, we have:

$$
Freq = cat([Freq^0, Freq^1, ..., Freq^{n-1}]), \tag{7}
$$

where the output $Freq \in \mathbb{R}^C$ is a multifrequency component and cat represents the concatenation operation. The final framework can be written as:

$$
\mathcal{F} = sigmoid(fc(Freq)). \tag{8}
$$

It can be seen that the network learns important frequency features by decomposing image features into different frequency components. According to Ref. (Qin et al., 2021), we uniformly set $n = 16$. As shown in Fig. 1, we apply a frequency attention mechanism after each residual block. The redundant texture convolution feature with a size of $C \times H \times W$ is transmitted through the frequency attention mechanism, and the core texture convolution feature with a size of $C \times H \times W$ is the final output.

3.2. Bilinear models

We use a bilinear model (Lin et al., 2018) to obtain the second-order statistics of texture images. As shown in Fig. 3, the bilinear model uses two identical low-rank convolution features to perform an outer product operation and then gathers local features to obtain an image description vector. Given an input image I, we assume that $F \in \mathbb{R}^{w \times h \times c}$ is the feature

extracted by two deep CNNs from I. The bilinear feature $B \in \mathbb{R}^{c \times c}$ is given by:

$$B(F) = \Sigma_{i=1}^{w} \Sigma_{j=1}^{h} F_{i,j} F_{i,j}^{T}. \tag{9}$$

The output bilinear matrix B captures the correlations between feature channels, which

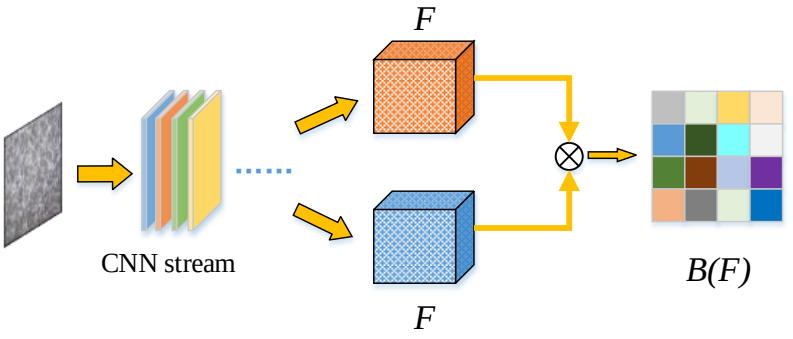


Fig. 3. Structure of the BCNN model. The input texture image passes through two identical CNN streams to produce a feature map F. The outer product of the matrix is used to obtain the bilinear matrix $B(F)$.

belong to the second-order feature information statistics. The first- and second-order information also contribute greatly to texture recognition. First-order information is essential for capturing the general features of an image. Furthermore, the first-order information is optimized through backpropagation in an end-to-end manner. Li et al. (2017) showed that second-order statistical information is helpful for image recognition. The second-order information brings more discriminative information to the texture classifier.

3.3. Dimensionality reduction

The high-dimensional bilinear features demand expensive memory and consumption costs, and they are difficult to embed in the existing end-to-end deep neural network frameworks with respect to these costs. To train our network efficiently, we use the dimensionality reduction method. We reduce the dimensionality of a high-dimensional bilinear matrix of size $c_h = c \times c$ to a low-dimensional feature vector c_l.

Suppose that we are given two randomly sampled mapping vectors $h_k \in N^{c_h}$ and $s_k \in \{+1, -1\}^{c_h}$, where $h_k(i)$ is drawn uniformly from $\{1, 2, ..., c_l\}$, $s_k(i)$ is drawn uniformly from $\{+1, -1\}$, $i = 1, 2, ..., c_h$, and $k = 1, 2$. Then, we can define the sketch function as:

$$\varphi(x, h, s) = \{(V)_1, (V)_2, ..., (V)_{c_l}\}, \tag{10}$$

where $(V)_j = \Sigma_{t:h(t)=j} s(t)x(t)$ and $x \in \mathbb{R}^{c_h}$ is a higher-dimensional bilinear feature vector.

The resulting lower-dimensional vector is defined as:

$$\gamma(x) = FFT^{-1}(FFT(\varphi(x, h_1, s_1))oFFT(\varphi(x, h_2, s_2))). \tag{11}$$

where FFT represents the fast Fourier transform, FFT^{-1} represents the inverse FFT, and o represents elementwise multiplication. We treat a bilinear matrix of size $c_h = c \times c$ as a vector $\mathcal{X} \in \mathbb{R}^{c_h}$ for convenience. After performing dimensionality reduction, this matrix becomes a low-dimensional vector $\gamma \in \mathbb{R}^{c_l}$, where $c_l << c_h$. In all experiments, we reduce the original bilinear dimensionality to $c_l = 512$ dimensions.

3.4. Double-order texture information encoding layer

3.4.1. TEL

The TEL encodes deep convolutional features to capture encoding information from the texture domain. Suppose that we are given a set of N texture descriptors $Y = \{y_1, y_2, ..., y_N\}$ and then given the codebook $C = \{c_1, c_2, ..., c_M\}$ to be learned, where M is the number of D-dimensional codewords. The residual vector is defined as $r_{im} = y_i - c_m$, where $i = 1, 2, ..., N$, $m = 1, 2, ..., M$. Moreover, we assign a weight w_{im} to each codeword c_m. The weight w_{im} is assigned by y_i and can be learned. Therefore, given the assigned codewords and inputs, the residual vector encoding for each codeword c_m is:

$$e_m = \Sigma_{i=1}^{N} w_{im} r_{im}, \tag{12}$$

where e_m is the coded output and w_{im} is the weight assigned to the codeword, which is obtained by the following formula:

$$w_{im} = \frac{exp(-s_m \| r_{im} \|^2)}{\Sigma_{j=1}^{M} exp(-s_j \| r_{ij} \|^2)}. \tag{13}$$

where s denotes learnable smoothing factors. This process takes the ambiguity of the codeword into account and solves the situation in which the model itself is a nondifferentiable problem. The TEL encodes the convolution feature information derived from the CNN and outputs a fixed-length encoding vector $E = \{e_1, e_2, ..., e_M\}$.

3.4.2. Double-order information fusion for texture encoding

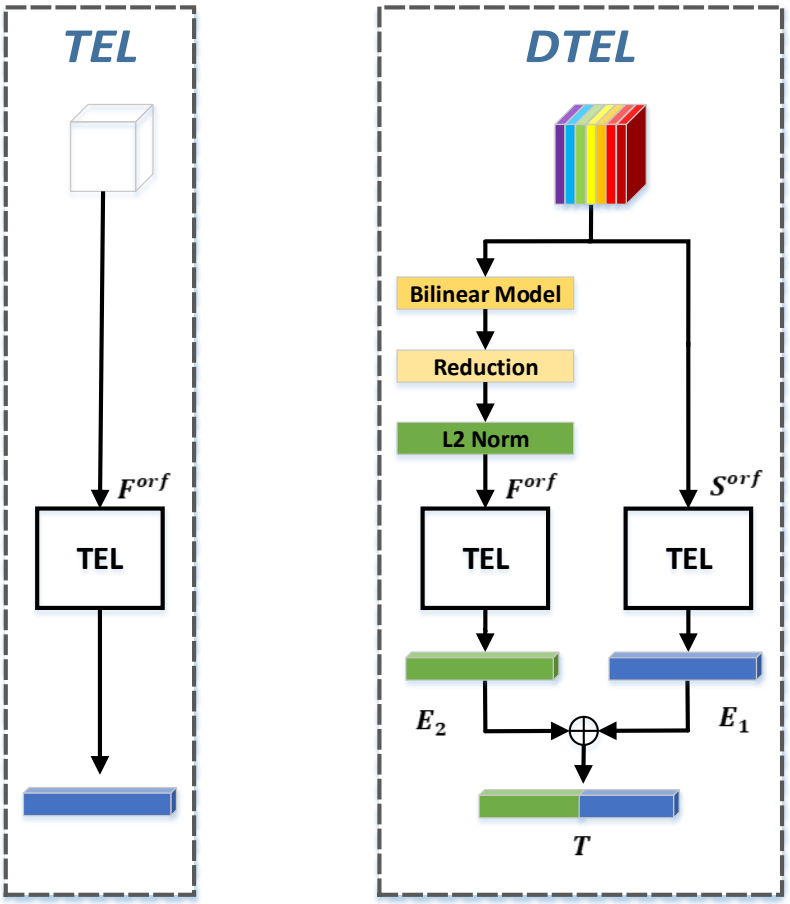


Fig. 4. The left side shows the TEL model, and the right side presents our DTEL model. F^{orf} and S^{orf} represent first-order and second-order convolution features, respectively. E_1 and E_2 represent first-order and second-order texture encoding information. T is the output obtained after the aggregation of E_1 and E_2.

To capture the second-order texture encoding information, we introduce the second-order feature information to the TEL. The second-order feature information captures the pairwise reciprocity between the feature channels, which is equivalent to increasing the number of feature combinations and enriching the feature representation information. Therefore, texture encoding is performed on the second-order feature information, as this technique can capture more distinguishable texture feature information. We design a new TEL (the DTEL). The double-order convolution features are subjected to texture encoding to obtain first- and second-order texture encoding information, as shown in Fig. 4. The double-order convolution features are processed with the DTEL layer, and the obtained first- and second-order texture encoding vectors are E_1 and E_2, respectively. The sizes of E_1 and E_2 are both $D \times M$. We combine E_1 and E_2 and define the information output after this combination as T:

$$T = E_1 \oplus E_2. \tag{14}$$

where T represents the texture encoding vector output by the DTEL with a size of $2D \times M$, and $\oplus$ represents the vector concatenation operation.

3.5. DFAEN

The DFAEN is shown in Fig. 1. First, the feature extractor adopted by the DFAEN is different from those of other deep network-based texture recognition methods (Zhang et al., 2017; Xue et al., 2018). The frequency attention mechanism (Xue et al., 2018) is inserted into the network (pretrained on ImageNet) to form the FAEN. The FAEN is used to extract and learn important feature information. Due to the high dimensionality of the feature map output by the FAEN, we use a 1×1 convolutional layer to reduce the number of feature map channels. Then, we insert the DTEL after the 1×1 convolutional layer and set the number of codebooks in the DTEL to $C = 128$. The double-order convolution information obtained by the FAEN is passed through the DTEL to obtain first- and second-order texture encoding

information. L_2 normalization is used to normalize the output of the DTEL:

$$\phi(x) = \frac{sign(x)\sqrt{|x|}}{||sign(x)\sqrt{|x|}||_2}. \tag{15}$$

Then, we aggregate the first- and second-order texture encoding information. Finally, classification is realized through a fully connected layer and a nonlinear classification layer. Our framework can be trained in an end-to-end manner, and all parameters can be backpropagated and optimized together.

4. Experimental Results and Analysis

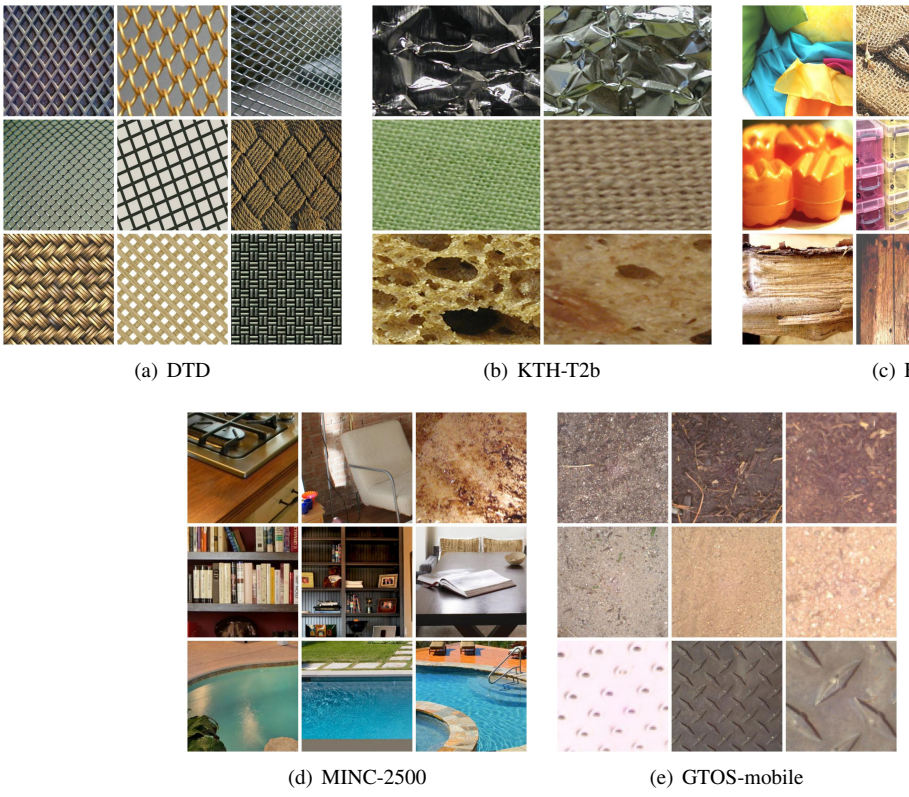
In this section, we evaluate the performance of our architecture and compare it with the latest methods on several texture datasets. To perform an equal comparison, we use the same training and test settings for all experiments.

4.1. Datasets and evaluation

We evaluate our model on 5 public texture datasets, including the Describable Texture Dataset (DTD) (Cimpoi et al., 2014), the KTH-TISP-2b dataset (KTH-T2b) (Caputo et al., 2005), the Flickr Material Dataset (FMD) (Sharan et al., 2013, 2009), the Materials in Context Database (MINC-2500) (Bell et al., 2015), and the Ground Terrain in Outdoor Scenes dataset (GTOS-mobile) (Xue et al., 2018), whose sample images are shown in Fig. 5.

The DTD is a widely used texture recognition benchmark. It contains a total of 5640 images spanning 47 categories, and each category has 80 samples for training and 40 samples for testing. The pictures in the database all come from online collections, which are series of images with texture and visual attributes. The main challenge with the DTD dataset is its large intraclass differences. The dataset is randomly divided into 10 splits, and we report the mean accuracy achieved on these 10 splits.

KTH-T2b is a texture material dataset that contains 11 categories and a total of 4752 images. Under laboratory control conditions, material instance images are collected with



(a) DTD (b) KTH-T2b (c) FMD

(d) MINC-2500 (e) GTOS-mobile

Fig. 5. Sample images obtained from various texture datasets, including the DTD, the FMD, MINC-2500, GTOS-mobile and KTH-T2b. The rows in each grid map fall into the same category.

different postures, illuminations, and scales. We randomly split the dataset samples into 50% for training and 50% for testing. The reported accuracy is the average obtained on 10 splits.

The FMD is a material dataset that contains 10 categories; each category has 100 samples, for a total of 1000 images. The samples are collected from the Internet, and each sample comes from different material. The FMD is different from a general texture dataset because it pays more attention to the materials of the objects, such as plastic, glass, and fiber. We use the provided dataset to split the training dataset and testing dataset. A total of 990 images are used for training, and 110 images are used for testing. We report the average accuracy obtained on 10 splits.

Similar to the FMD, MINC is also a material dataset, but the scale of MINC is obviously larger than that of the FMD. In this work, we use the publicly available MINC-2500 subset

of MINC. MINC-2500 contains 23 categories and a total of 57,500 samples. We follow the evaluation setup in Ref. (Zhang et al., 2017) for the large number of samples in the MINC-2500 dataset. The dataset is randomly divided into training and test sets (2 splits). The reported accuracy is the average obtained on these 2 splits.

GTOS is a ground terrain dataset for outdoor scenes. We use the extended GTOS-mobile dataset provided by (Xue et al., 2018). The GTOS-mobile dataset is collected from the GTOS dataset by mobile phones, and it contains 31 categories. The evaluation settings are consistent with those of the MINC-2500 dataset.

4.2. Implementation details

We choose the VGG19 (Simonyan and Zisserman, 2015), Resnet50 (He et al., 2016), Inceptionv4 (Szegedy et al., 2017), and DenseNet161 (Huang et al., 2017) networks pretrained (Fang et al., 2021) on ImageNet as the backbones. During the training procedure, we use the stochastic gradient descent optimizer with a minimum batch size of 64. Fine-tuning is adopted during training, and the learning rate starts from 0.001 and is divided by 10 when the training error plateaus. We use a weight decay of 0.0001 and a momentum of 0.9. In all experiments, we resize each input to 224 x 224, with a 50% chance of horizontal flipping, for evaluation purposes. The proposed framework is implemented in Pytorch 1.6 (Paszke et al., 2019) and on an NVIDIA RTX 2080Ti GPU.

4.3. Comparisons with state-of-the-art methods

In this section, to evaluate the effectiveness of our method, we compare it with several state-of-the-art approaches, including the BCNN (Lin et al., 2018), DeepTEN, and the DEP network. These methods are based on end-to-end training. The results of all methods are obtained under the same experimental conditions. We evaluate the performance of our method on the aforementioned datasets: the FMD (Sharan et al., 2013, 2009), the MINC-2500 (Bell et al., 2015) material dataset, the DTD (Cimpoi et al., 2014), the KTH-T2b (Caputo et al., 2005) texture dataset and the GTOS (Xue et al., 2018) outdoor ground scene dataset.

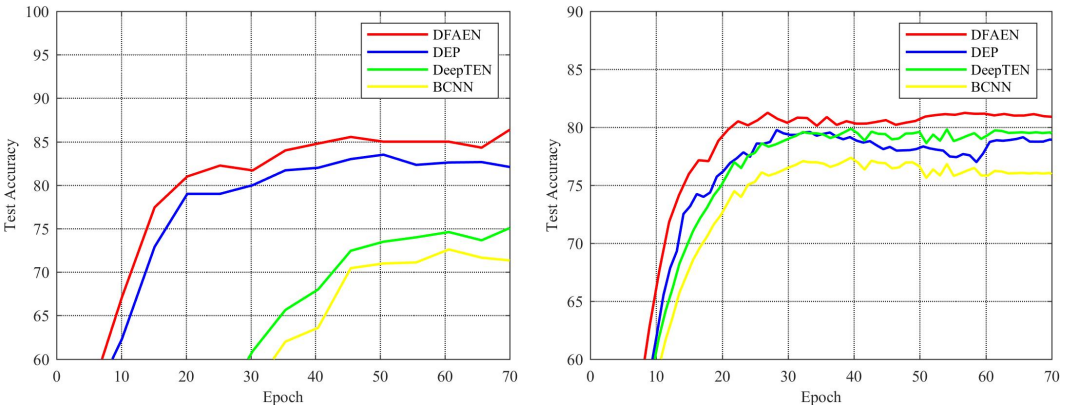
Table 1
Comparison with state-of-the-art methods on the FMD, MINC-2500, the DTD, KTH and GTOS-mobile. The top-1 test accuracies are reported in the table. The results of the BCNN, locality-aware coding, BPCA, HL, and residual pooling methods are cited from (Lin et al., 2018), (Gao et al., 2016), (Dong et al., 2020), (Peeples et al., 2021), and (Mao et al., 2021) respectively. The results of the FASON, DeepTEN, and DEP methods are obtained by running them under our experimental conditions.

Methods	Backbone	FMD	MINC-2500	DTD	KTH	GTOS-mobile
BCNN (Lin et al., 2018)	Resnet50	80.70%	-	69.60%	75.10%	-
FASON (Dai et al., 2017)	VGG19	82.10%	-	72.50%	-	-
DeepTEN (Zhang et al., 2017)	VGG19	81.20%	76.61%	60.77%	-	-
DeepTEN (Zhang et al., 2017)	Resnet50	82.40%	79.20%	70.53%	85.45%	84.30%
DEP (Xue et al., 2018)	VGG19	-	77.70%	68.00%	-	-
DEP (Xue et al., 2018)	Resnet50	84.80%	79.86%	71.90%	84.53%	85.67%
Locality-aware coding (Bu et al., 2019)	Resnet50	82.40%	-	71.10%	76.90%	-
BPCA (Dong et al., 2020)	VGG16	-	-	70.02%	-	-
HL (Peeples et al., 2021)	Resnet50	-	82.42%	71.98%	-	79.75%
Residual pooling (Mao et al., 2021)	Resnet101	83.12%	-	72.93%	-	-
DFAEN(Ours)	VGG19	83.40%	76.47%	68.50%	-	-
DFAEN(Ours)	Resnet50	86.90%	81.66%	73.20%	86.34%	86.87%
DFAEN(Ours)	Inceptionv4	85.80%	82.05%	75.18%	84.42%	84.40%
DFAEN(Ours)	Densenet161	**87.60%**	**83.55%**	**76.10%**	**86.57%**	**86.90%**

In Table 1, we present the overall results yielded by our method in comparison with those of other methods.

First, we evaluate the effectiveness of our method on the FMD and the MINC-2500 dataset. We compare it with the state-of-the-art BCNN, DEP, and DeepTEN methods. For a fair comparison, we use the same training/testing set divisions and the same learning hyperparameters for each of the respective datasets. For the FMD, we conduct training with a fixed learning rate of 0.001. For the MINC-2500 dataset, the learning rate starts from 0.001 and is divided by 10 when the error plateaus. Fig. 6 shows the accuracy comparison results for each method tested on the FMD and the MINC-2500 dataset. The figure shows that our DFAEN method is significantly better than the DEP and DeepTEN approaches. In

the early stage of training, our method has a slightly worse test effect than the DEP method because the network has not yet been fully trained. After training for 20 epochs, our DFAEN method consistently performs better than the other two methods. It is worth noting that our architecture has better stability and robustness. It can be seen from Fig. 6 that both the DEP network and DeepTEN have different degrees of overfitting in the later stages of training, while our method is very stable.

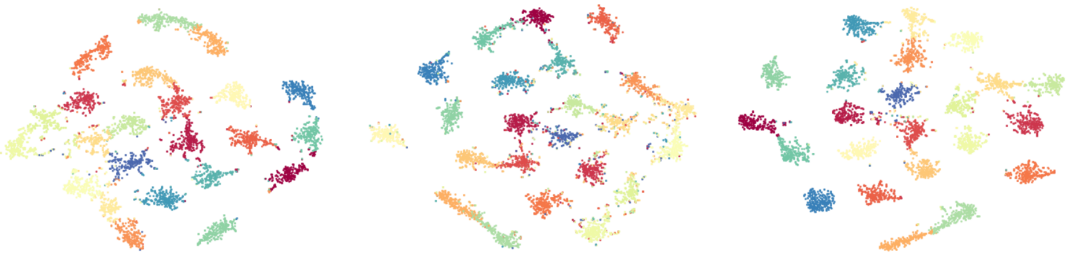


(a) FMD (b) MINC-2500

Fig. 6. Comparison of test accuracy between our method, DEP, DeepTEN and BCNN methods on FMD and MINC-2500 datasets.

Material recognition. Our method is compared with several state-of-the-art methods on the FMD and the MINC-2500 dataset. Table 1 shows that our method achieves the best performance on the DenseNet161 backbone. Our method is significantly superior to the BCNN. In particular, the results of our best model are 2.8% and 3.15% higher than those of the DEP benchmark and 5.2% and 3.8% higher than those of the DeepTEN benchmark on the FMD and the MINC-2500 dataset, respectively. On the FMD, our method is also better than locality-aware coding and residual pooling. Moreover, our method outperforms HL by 0.59% on the MINC-2500 dataset. The results show that our architecture is more competitive. Intuitively, we perform a visual analysis among the DEP network, DeepTEN and our model to clarify the level of superiority. We use Barnes-Hut t-distributed stochastic neighborhood embedding (t-SNE) (Van Der Maaten, 2014) to visualize the features obtained before the

classification layers of DeepTEN, the DEP network, and our DFAEN. We use the MINC-2500 test set for the visualization experiments. As shown in Fig. 7, our method separates different classes further and makes the same classes more clustered. The visualization results show that our method can better identify and classify materials.



(a) DeepTEN (b) DEP (c) DFAEN(Our)

Fig. 7. Barnes-Hut t-SNE (Van Der Maaten (2014)) visualizations produced by the three models (DeepTEN is on the left, the DEP network is in the middle, and our DFAEN is on the right) on the MINC-2500 dataset. We take the test set data of MINC-2500 to extract features and perform visualization experiments before reach the three model classification layers. We can clearly see that DFAEN can produce better classification results.

Texture recognition. In addition to the material datasets, we also evaluate the performance of our method on the DTD and the KTH-T2b texture dataset. Table 1 shows the experimental results obtained by the state-of-the-art methods and our method. The results show that our method is superior to the state-of-the-art methods on these two texture datasets. On the DTD, the test accuracy of our method is 3.7% higher than that of the residual pooling method, 4.1% higher than that of HL, 6.1% higher than that of the BPCA method, and 5.0% higher than that of the locality-aware coding method. On the KTH-T2b dataset, the classification accuracy of the DEP network is lower than that of DeepTEN. This is caused by the insufficiency of the training data in the KTH-T2b dataset. However, our method still achieves 1.1%, 11%, and 9.6% improvements over DeepTEN, the BCNN, and locality-aware coding, respectively. This justifies the effectiveness of our method in terms of texture recognition.

Table 2
Comparison of our method with the DeepTEN method, which only uses first-order information for texture encoding, under different settings. We experiment on the DTD and the MINC-2500 dataset separately. DTEL means that only the first- and second-order information is fused, and DTEL+FAEN means that the FAEN is inserted on the basis of fusion.

Method	DTD	MINC-2500
	Accuracy	Accuracy
DeepTEN	71.98%	79.20%
resnet50 + DTEL	72.71%	81.30%
FAEN + DTEL	**73.24%**	**81.66%**

Terrain recognition. To further evaluate the effectiveness of our method, we experiment on the GTOS dataset and compare it with the popular state-of-the-art methods. As shown in Table 1, our model is better than DeepTEN and the DEP network under the same experimental settings. Our model achieves a 1.2% improvement over the DEP network, which is specialized for outdoor terrain recognition, and a 7.15% improvement over the HL method. Overall, our method exhibits promising generalization ability for texture recognition and classification tasks.

4.4. Ablation study

In this section, we further analyze and verify the effectiveness of each module in our architecture. Table 2 shows the performance improvement provided by each module on the DTD and the MINC-2500 dataset.

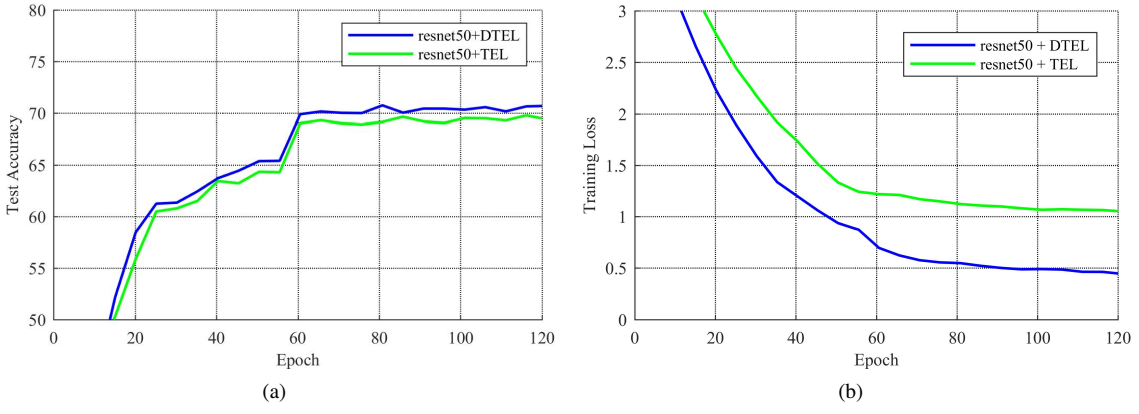


Fig. 8. Comparison between the learning curves of the DTEL and TEL on the DTD.

The effectiveness of the DTEL. We evaluate the effectiveness of our proposed DTEL module on the DTD. We compare it with the DeepTEN method, which only uses single-order information. Consistent with the previous description, we use the same experimental setup to ensure a fair comparison. We use a learning rate of 0.01 at the beginning and divide it by 10 when the error plateaus. We only observe the results for the first 120 epochs. Fig. 8(a) and Fig. 8(b) show the learning comparison curves of the test accuracy and the training loss, respectively. We sketch the training loss curve in Fig. 8. The comparison shows that our model can achieve a lower training error and a higher test accuracy while exhibiting better robustness. The Resnet50 + DTEL method in the figure represents our network architecture. Our proposed DTEL integrates double-order information into the TEL. Resnet50 + TEL is the network architecture in DeepTEN, in which the TEL uses only single-order information. As shown in Fig. 8, after performing sufficient learning and training, our architecture (Resnet50 + DTEL) is significantly better than that utilizing only the single-order information (Resnet50 + TEL). Our experiments show that our proposed insertion of double-order information into the TEL is effective.

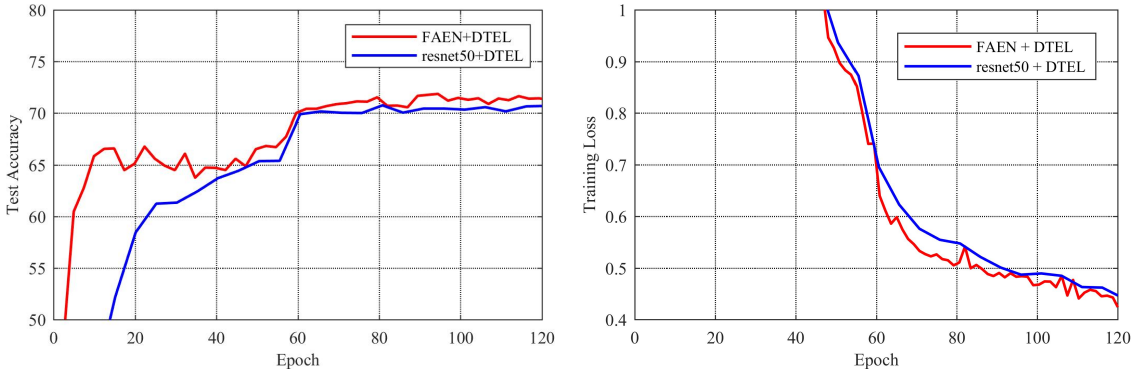


Fig. 9. Comparison between the learning curves obtained on the DTD after adding the FAEN.

The effectiveness of the FAEN for texture encoding. To further verify the effectiveness of our architecture, we perform a diagnostic analysis on the FAEN module. To ensure the consistency of the experiment, all the settings in this experiment are consistent with those

in the experiment on the effectiveness of the DTEL. As shown in the learning comparison curves in Fig. 9(a) and Fig. 9(b), after inserting the FAEN, the entire network converges better, and the recognition performance is further improved.

5. Conclusion

We propose a novel end-to-end network architecture for texture recognition. This architecture fuses the second-order convolution feature information in the encoding layer to obtain richer discriminative texture encoding information. Second, our proposed method uses frequency attention to filter the feature information obtained before the texture encoding stage. Experimental results show that both of our proposed modules yield significant texture recognition improvements. In addition, compared with the state-of-the-art methods, our architecture achieves better recognition accuracy on the FMD, MINC-2500, the DTD, KTH, and GTOS-mobile. Specifically, our method achieves performance improvements of approximately 3% on the DTD and the MINC-2500 dataset and improvements of approximately 1% on the other datasets.

Acknowledgements

This work is supported by the Research and Development Project in Key Areas of Guangdong Province (2018B010109004), the Science and Technology Project of Guangdong Province (no. 2019A050513011), the Guangzhou Science and Technology Plan Project (no. 202002030386), and the Guangdong Provincial Key Laboratory of Photonics Information Technology (no. 2020B121201011).

References

Alfed, N., Khelifi, F. (2017). Bagged textural and color features for melanoma skin cancer detection in dermoscopic and standard images. Expert Systems with Applications, 90, 101–110. https://doi.org/10.1016/j.eswa.2017.08.010.

Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J. (2016). Netvlad: Cnn architecture for weakly supervised place recognition, https://arxiv.org/abs/1511.07247.

Bell, S., Upchurch, P., Snavely, N., Bala, K. (2015). Material recognition in the wild with the materials in context database. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3479–3487. https://doi.org/10.1109/CVPR.2015.7298970.

Bu, X., Wu, Y., Gao, Z., Jia, Y. (2019). Deep convolutional network with locality and sparsity constraints for texture classification. Pattern Recognition, 91, 34–46. https://doi.org/10.1016/j.patcog.2019.02.003.

Caputo, B., Hayman, E., Mallikarjuna, P. (2005). Class-specific material categorisation. Tenth IEEE International Conference on Computer Vision (ICCV'05), 2, 1597–1604. https://doi.org/10.1109/ICCV.2005.54.

Cerkezi, L., Topal, C. (2020). Towards more discriminative features for texture recognition. Pattern Recognition, 107, 107473. https://doi.org/10.1016/j.patcog.2020.107473.

Chen, J., Zhang, Y., Jiang, Y. (2019). Multi-features fusion classification method for texture image. The Journal of Engineering, 2019, 8834–8838. https://doi.org/10.1049/joe.2018.9118.

Chen, T., Lin, L., Hui, X., Chen, R., Wu, H. (2020). Knowledge-guided multi-label few-shot learning for general image recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44, 1371–1384. https://doi.org/10.1109/TPAMI.2020.3025814.

Chen, T., Pu, T., Wu, H., Xie, Y., Liu, L., Lin, L. (2021). Cross-domain facial expression recognition: A unified evaluation benchmark and adversarial graph learning. IEEE transactions on pattern analysis and machine intelligence, 1–1. https://doi.org/10.1109/TPAMI.2021.3131222.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A. (2014). Describing textures in the wild. IEEE Conference on Computer Vision and Pattern Recognition, 3606–3613. https://doi.org/10.1109/CVPR.2014.461.

Cimpoi, M., Maji, S., Vedaldi, A. (2015). Deep filter banks for texture recognition and segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3828–3836. https://doi.org/10.1109/CVPR.2015.7299007.

Dai, X., Ng, J.Y.H., Davis, L.S. (2017). Fason: First and second order information fusion network for texture recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 6100–6108. https://doi.org/10.1109/CVPR.2017.646.

Dong, X., Zhou, H., Dong, J. (2020). Texture classification using pair-wise difference pooling-based bilinear convolutional neural networks. IEEE Transactions on Image Processing, 29, 8776–8790. https://doi.org/10.1109/TIP.2020.3019185.

El khadiri, I., Chahi, A., El merabet, Y., Ruichek, Y., Touahni, R. (2018). Local directional ternary pattern: A new texture descriptor for texture classification. Computer Vision and Image Understanding, 169, 14–27. https://doi.org/10.1016/j.cviu.2018.01.004.

Fang, Z., Ren, J., Marshall, S., Zhao, H., Wang, S., Li, X. (2021). Topological optimization of the densenet with pretrained-weights inheritance and genetic channel selection. Pattern Recognition, 109, 107608. https://doi.org/10.1016/j.patcog.2020.107608.

Fekri-Ershad, S. (2020). Bark texture classification using improved local ternary patterns and multilayer neural network. Expert Systems with Applications, 158, 113509. https://doi.org/10.1016/j.eswa.2020.113509.

Fujieda, S., Takayama, K., Hachisuka, T. (2017). Wavelet convolutional neural networks for texture classification, https://arxiv.org/abs/1707.07394.

Gao, Y., Beijbom, O., Zhang, N., Darrell, T. (2016). Compact bilinear pooling. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 317–326. https://doi.org/10.1109/CVPR.2016.41.

Haralick, R.M., Shanmugam, K., Dinstein, I. (1973). Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics, SMC-3, 610–621. https://doi.org/10.1109/TSMC.1973.4309314.

He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. https://doi.org/10.1109/CVPR.2016.90.

Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E. (2020). Squeeze-and-excitation networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42, 2011–2023. https://doi.org/10.1109/TPAMI.2019.2913372.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q. (2017). Densely connected convolutional networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2261–2269. https://doi.org/10.1109/CVPR.2017.243.

Jégou, H., Douze, M., Schmid, C., Pérez, P. (2010). Aggregating local descriptors into a compact image representation. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 3304–3311. https://doi.org/10.1109/CVPR.2010.5540039.

Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097–1105. https://doi.org/10.1145/3065386.

Lee, S., Lee, C. (2016). Multiscale morphology based illumination normalization with enhanced local textures for face recognition. Expert Systems with Applications, 62, 347–357. https://doi.org/10.1016/j.eswa.2016.06.039.

Leung, T., Malik, J. (2001). Representing and recognizing the visual appearance of materials using three-dimensional textons. International journal of computer vision, 43, 29–44. https://doi.org/10.1023/A:1011126920638.

Li, P., Xie, J., Wang, Q., Zuo, W. (2017). Is second-order information helpful for large-scale visual recognition? Proceedings of the IEEE international conference on computer vision, 2070–2078. .

Lin, T.Y., RoyChowdhury, A., Maji, S. (2018). Bilinear convolutional neural networks for fine-grained visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40, 1309–1322. https://doi.org/10.1109/TPAMI.2017.2723400.

Mao, S., Rajan, D., Chia, L.T. (2021). Deep residual pooling network for texture recognition. Pattern Recognition, 112, 107817. https://doi.org/10.1016/j.patcog.2021.107817.

Medeiros, R., Scharcanski, J., Wong, A. (2016). Image segmentation via multi-scale stochastic regional texture appearance models. Computer Vision and Image Understanding, 142, 23–36. https://doi.org/10.1016/j.cviu.2015.06.001.

Ojala, T., Pietikainen, M., Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24, 971–987. https://doi.org/10.1109/TPAMI.2002.1017623.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32, 8026–8037. .

Peeples, J., Xu, W., Zare, A. (2021). Histogram layers for texture analysis. IEEE Transactions on Artificial Intelligence, 1–1. https://doi.org/10.1109/TAI.2021.3135804.

Qin, Z., Zhang, P., Wu, F., Li, X. (2021). Fcanet: Frequency channel attention networks, 783–792. https://doi.org/10.1109/ICCV48922.2021.00082.

Rippel, O., Snoek, J., Adams, R.P. (2015). Spectral representations for convolutional neural networks. Proceedings of the 28th International Conference on Neural Information Processing Systems, 2, 2449–2457. https://doi.org/10.5555/2969442.2969513.

Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J. (2013). Image classification with the fisher vector: Theory and practice. International journal of computer vision, 105, 222–245. https://doi.org/10.1007/s11263-013-0636-x.

Sharan, L., Liu, C., Rosenholtz, R., Adelson, E.H. (2013). Recognizing materials using perceptually inspired features. International journal of computer vision, 103, 348–371. https://doi.org/10.1007/s11263-013-0609-0.

Sharan, L., Rosenholtz, R., Adelson, E. (2009). Material perception: What can you see in a brief glance? Journal of Vision, 9, 784–784. https://doi.org/10.1167/9.8.784.

Simonyan, K., Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition, https://arxiv.org/abs/1409.1556.

Sivic, J., Russell, B., Efros, A., Zisserman, A., Freeman, W. (2005). Discovering objects and their location in images. Tenth IEEE International Conference on Computer Vision (ICCV'05), 1, 370–377 Vol. 1. https://doi.org/10.1109/ICCV.2005.77.

Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. Thirty-first AAAI conference on artificial intelligence, https://arxiv.org/abs/1602.07261.

Tschannerl, J., Ren, J., Yuen, P., Sun, G., Zhao, H., Yang, Z., Wang, Z., Marshall, S. (2019). Mimr-dgsa: Unsupervised hyperspectral band selection based on information theory and a modified discrete gravitational search algorithm. Information Fusion, 51, 189–200. https://doi.org/10.1016/j.inffus.2019.02.005.

Uzen, H., Turkoglu, M., Hanbay, D. (2021). Texture defect classification with multiple pooling and filter ensemble based on deep neural network. Expert Systems with Applications, 175, 114838. https://doi.org/10.1016/j.eswa.2021.114838.

Van Der Maaten, L. (2014). Accelerating t-sne using tree-based algorithms. Journal of Machine Learning Research, 15, 3221–3245. https://doi.org/10.5555/2627435.2697068.

Xue, J., Zhang, H., Dana, K. (2018). Deep texture manifold for ground terrain recognition. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 558–567. https://doi.org/10.1109/CVPR.2018.00065.

Zhang, H., Xue, J., Dana, K. (2017). Deep ten: Texture encoding network. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2896–2905. https://doi.org/10.1109/CVPR.2017.309.