TexFusionNet: An Ensemble of Deep CNN Feature for Texture Classification

Swalpa Kumar Roy, Shiv Ram Dubey, Bhabatosh Chanda, Bidyut B. Chaudhuri and Dipak Kumar Ghosh

Abstract The texture classification from images is one of the important problems in pattern recognition. Several hand-designed methods have been proposed in last few decades for this problem. Nowadays, it is observed that the convolutional neural networks (CNN) perform extremely well for the classification task mainly over object and scene images. This improved performance of CNN is caused by the availability of huge amount of images in object and scene databases such as ImageNet. Still, the focus of CNN in texture classification is limited due to non-availability of large texture image data sets. Thus, the trained CNN over Imagenet database is used for texture classification by fine-tuning the last few layers of the network. In this paper, a fused CNN (TexFusionNet) is proposed for texture classification by fusing the last representation layer of widely adapted AlexNet and VGG16. On the top of the fused layer, a fully connected layer is used to generate the class score. The categorical cross-entropy loss is used to generate the error during training, which is used to train the added layer after the fusion layer. The results are computed over several well-known Brodatz, CURET, and KTH-TIPS texture data sets and compared with the state-of-the-art texture classification methods. The experimental results confirm outstanding performance of the proposed TexFusionNet architecture for texture classification.
Keywords Convolutional neural network (CNN) · Deep learning · Texture classification · Fusion

1 Introduction

One of the underlying problems in pattern recognition is the automatic texture image classification. The texture classification has huge applications in different areas such as content-based image retrieval, ground analysis through satellite imagery, biomedical and industrial inspection, etc. Though the human is good recognizer to identify the texture in a real scenario, the definition of texture is still ambiguous [1]. Despite attempts in the past few decades, the texture classification problem is still challenging. The common practice in texture classification is to first compute the suitable features and then use those features with some classifier for training and classification. Numerous texture feature descriptors have been discovered by various researchers [2]. The main three desired properties of any texture feature descriptor are robustness to deal with the intraclass variability, discriminative to distinguish between different categories, and low dimensionality to reduce the processing time.

In earlier days, the popular texture classification methods are based on the statistical features such as co-occurrence matrix [3] and Markov random fields [4]. After that, there was an era of filtering-based approaches where the images were converted into feature vectors by applying the bank of filters such as wavelet filters [5], Gabor filters [6], etc. Later on, the macro-pattern-based approaches as local binary pattern (LBP) [7] are proved to be discriminative and robust for texture analysis such. The LBP became one of the state-of-the-art approach for feature extraction due to its simplicity and local relationship representation ability. Several variants of LBP are investigated for different applications such as biomedical image analysis [8–10], face recognition [11, 12], image retrieval [13, 14], pedestrian detection [15], local patch matching [16], etc. Other LBP-based feature descriptors are also proposed for texture classification such as completed local binary pattern (CLBP) [17], LBP variance (LBPV) [18], binary rotation invariant and noise-tolerant descriptor (BRINT) [19], fractal weighted local binary pattern local (FWLBP) [20], complete dual cross pattern (CDCP) [21], jet pattern (LJP) [22], local directional zig-zag pattern (LDZP) [23], local morphological pattern (LMP) [24], etc. These hand-crafted feature descriptors generally perform well in controlled conditions in practice. This is due to the lack of robustness against different types of geometric and photometric variations in a single descriptor.

Recently, the deep learning methodology has changed the research direction in machine learning, computer vision, natural language processing, and pattern recognition area. It learns from massive amount of data and improves the performance with great margin [25]. The first and revolutionary work using deep learning in computer vision was witnessed in 2012 by Krizhevsky et al. [26] for image classification task over a largest and challenging ImageNet database [27] by winning the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). Later on, this system
is named as AlexNet which is basically a convolutional neural network (CNN) with seven learning layers. After AlexNet, several CNN architectures were proposed for image classification such as VGG16 (16 learning layers) in 2014 by Simonyan and Zisserman of Oxford University [28], GoogleNet (22 learning layers) in 2014 by Szegedy et al. of Google [29], ResNet (152 learning layers) in 2015 by He et al. of Microsoft Research [30], etc. A trend to use more number of learning layers in CNN is observed over the years after 2012. The CNN-based approaches have also shown very exciting performance in other problems such as object detection (Faster R-CNN [31]), image segmentation (Mask R-CNN [32]), biomedical image analysis (Colon cancer detection [33]), etc.

The hierarchical representation of in deep convolution neural networks (CNN) is a key and universal characteristic which is directly formed from the data set and utilized to classify the images. As a universal representation, deep CNNs have shown their recognition power. However, the lack of geometric invariance of globally used CNN activations has limited robustness for recognition. Due to lack of large-scale texture databases, the deep learning based work is not witnessed much for texture classification. Recently, some researchers used the CNN-based methods for texture classification using transfer learning [34, 35]. The transfer learning is a way to use the already trained CNN model for small-scale databases by considering the same weights and fine-tuning the last few layers. Filter banks are also used in the architecture of convolutional neural networks named as texture CNN and applied for classification in texture databases [36]. Cimpoi et al. proposed the Fisher Vector pooling of a CNN filter bank (FV-CNN) for texture recognition and segmentation [37]. Liu et al. [38] have done a performance analysis over LBP and deep texture descriptors in terms of the robustness against different geometric and photometric changes such as rotation, illumination, scale, etc. There is little penetration into the behavior and internal operation of the network although deep CNN models significantly progressed. In 2013, ScatNet was introduced by Mallat et al. [39, 40] where no learning process is needed and convolution filters are predefined as wavelet. Inspired by ScatNet Chan et al. [41] have proposed PCANet, a simple network of deep learning, where network is formed by cascading of multistage principle component analysis (PCA), histogram pooling, and binary hashing. Chan et al. also proposed RandNet [41], a simple variation of PCANet where the cascaded filters are randomly selected and no learning is required. Nowadays, use of more and more complex networks becomes a major trend in deep CNN research community. However, a powerful computer with large memory and GPUs are needed to train the networks.

Most of the existing CNN-based models for texture classification are based on the single model architecture. In this paper, we leverage the power of multiple CNN architectures and design a fused convolutional neural network model for texture classification named as the TexFusionNet. The proposed TexFusionNet architecture first fuses the last representation layer of AlexNet and VGG16 and then uses the fully connected layer and softmax layer on top of that for fine-tuning and classification. To evaluate the classification performance, the experiments are conducted on benchmark Brodatz, CUReT, and KTH-TIPS texture databases in support of proposed TexFusionNet architecture.
The rest of the paper is structured as follows: Sect. 2 presents the proposed TexFusionNet architecture; Sect. 3 depicts the experimental results and analysis; and Sect. 4 provides the conclusion.

## 2 Proposed TexFusionNet

The proposed TexFusionNet architecture for texture classification is illustrated in Fig. 1 by fusing two CNN models. The TexFusionNet is composed of the AlexNet [26] and VGG16 [28] architectures of image classification. The AlexNet and VGG16 architectures are the state-of-the-art methods and widely adapted for classification task. The fusion between these models is performed at the last representation layer. The last representation layer is referred to as the last layer of these models after removing the class score layer. In the original AlexNet and VGG16, the last layer is class score layer for 1000 classes of ImageNet database which is removed in our architecture. We could merge well at last representation layer because both AlexNet and VGG16 models produce the same dimensional features (i.e., 4096 dimensional) at the last layer. Moreover, the weights of filters from first layer to last representation layer in both AlexNet and VGG16 are transferred from the pretrained AlexNet and VGG16 models, respectively. Let us consider $I$ as the input color texture image of resolution $m \times n$, $alex$ is a function representing the combination of convolutional layers, max-pooling layers, and fully connected layers from first layer to last representation layer of AlexNet model, and similarly $vgg16$ is a function for VGG16 model. Note that the input resolutions for AlexNet and VGG16 are $227 \times 227$ and

![Diagram](image)

**Fig. 1** Proposed TexFusionNet architecture for texture classification
224 × 224, respectively. So, the input image I is converted into \( I_{\text{alex}} \) and \( I_{\text{vgg}} \) having the resolution of 227 × 227 and 224 × 224 for AlexNet and VGG16, respectively, as follows:

\[
I_{\text{alex}} = \tau(I, [227, 227]) \tag{1}
\]

\[
I_{\text{vgg}} = \tau(I, [224, 224]) \tag{2}
\]

where \( \tau(I, [\beta, \beta]) \) is a transformation function to resize any image I into the resolution of \( (\beta \times \beta) \).

Let us consider \( \Phi_{\text{alex}} \) and \( \Phi_{\text{vgg}} \) represent the features or values of last representation layer of AlexNet and VGG16 models, respectively. The \( \Phi_{\text{alex}} \) and \( \Phi_{\text{vgg}} \) are defined as

\[
\Phi_{\text{alex}} = \text{alex}(I_{\text{alex}}) \tag{3}
\]

\[
\Phi_{\text{vgg}} = \text{vgg}(I_{\text{vgg}}) \tag{4}
\]

where the dimensions of \( \Phi_{\text{alex}} \) and \( \Phi_{\text{vgg}} \) representation features are \( D_{\text{alex}} \) and \( D_{\text{vgg}} \), respectively, with \( D_{\text{alex}} = D_{\text{vgg}} \).

In the proposed work, the \( \Phi_{\text{alex}} \) and \( \Phi_{\text{vgg}} \) texture features of input image I using AlexNet and VGG16 models are fused by addition to produce a combined texture feature representation denoted as \( \Phi_{\text{fused}} \) and defined as follows:

\[
\Phi_{\text{fused}}(i) = \Phi_{\text{alex}}(i) + \Phi_{\text{vgg}}(i) \quad \forall i \in [1, D] \tag{5}
\]

where \( D \) is the dimension of the fused feature \( \Phi_{\text{fused}} \) with \( D = D_{\text{alex}} = D_{\text{vgg}} \). \( \Phi_{\text{fused}}(i) \) is the \( i \)th element of fused feature \( \Phi_{\text{fused}} \) while \( \Phi_{\text{alex}}(i) \) and \( \Phi_{\text{vgg}}(i) \) are the \( i \)th elements of input features \( \Phi_{\text{alex}} \) and \( \Phi_{\text{vgg}} \), respectively.

The computed fused features \( \Phi_{\text{fused}} \) are considered as the input to a fully connected layer which produces \( n \) number of outputs as the class scores for \( n \) classes of any texture database (see Fig. 1). Suppose \( S_j \mid j \in [1, n] \) represents the class score for \( j \)th class of the texture database, where \( n \) is the number of classes. Mathematically, the class score \( S_j \) for \( j \)th class is defined as

\[
S_j = \sum_{k=1}^{D} w_{k,j} \times \Phi_{\text{fused}}(k) \tag{6}
\]

where \( j \in [1, n] \) and \( w_{k,j} \) is a weight connecting \( k \)th element of feature map \( \Phi_{\text{fused}} \) to \( j \)th class score.

During training the pretrained weights of first to last representational layer of both AlexNet and VGG16 are freeze (i.e., not trained over texture database) to utilize the already trained layers. The last fully connected layer (mapping from \( D \) to \( n \)) is trained by computing the categorical cross-entropy loss over the class scores (5) and
backpropagating to last added layer only. Suppose $c$ is the ground truth class and $S$ is the computed class scores for image $I$. Then the categorical cross-entropy loss ($L_I$) for input image $I$ is defined as follows:

$$L_I = -\log \left( \frac{e^{S_c}}{\sum_{j=1}^{n} e^{S_j}} \right) = -S_c + \log \sum_{j=1}^{n} e^{S_j}. \quad (7)$$

The training loss is computed batchwise and the weights of last added fully connected layer are updated in the opposite direction of its gradients.

Once the added layer in TexFusionNet is trained over training texture image database, it is used as a trained classifier over test cases. At test time, the input to TexFusionNet is an image itself and the output is class scores. First, the features are computed automatically in intermediate layers by AlexNet and VGG16 separately then these features are fused to produce a combined feature map which is finally used as the input to the final fully connected layer to produce the class scores.

### 3 Experimental Results and Discussion

In this section, we examine and evaluate the power of features representation using proposed TexFusionNet which combines two pretrained CNN models for texture classification. We evaluate the effectiveness of proposed TexFusionNet CNN features on the following three publicly available texture databases: first the details of used Brodatz, CUREt, and KTH-TIPS texture databases are discussed in detail. Then the experimental setup and evaluation criteria are illustrated and finally the classification results are analyzed over texture databases using proposed TexFusionNet and compared with the state-of-the-art results.

#### 3.1 Databases Used

Three benchmark texture databases including Brodatz album [42], CUREt [43], and KTH-TIPS [44] are used in this paper to justify the improved performance of introduced TexFusionNet architecture. The presence of various categories, variable number of images in each category, and high degree of intraclass variations are some of the difficulties which make these databases very challenging. Brodatz [42] texture database is opted to facilitate a fair comparison with the state-of-the-art results [45]. There are 32 homogeneous texture categories in this database. Each image is partitioned into 25 nonoverlapping sub-regions of size $128 \times 128$, and each sub-image is downsampled to $64 \times 64$ pixels. The same subset of images of CUREt database [43] as used in [46] is also chosen in this paper. It contains 61 texture categories having the large intraclass variations with 92 images per category.
cropped into 200 × 200 pixels region. The sample images are shown in Fig. 2. The images are taken with varying illumination and viewing points with constant scale. The KTH-TIPS database [44] is extended by imaging new samples of ten CUREt textures as shown in Fig. 3. It contains texture images with three different poses, four illuminations, and nine different scales of size 200 × 200 and hence each class contains 81 samples.
3.2 Experimental Setup and Evaluation Criteria

We use Keras to implement the proposed TexFusionNet which is derived from two widely adapted AlexNet and VGG16 models by fusing at the last representation layer. As both of adapted AlexNet and VGG16 models require the predefined size of the input texture image, all images are resized to $227 \times 227 \times 3$ for AlexNet and $224 \times 224 \times 3$ for VGG16 where number of feature maps, kernel sizes, etc. are kept same. The mean subtraction preprocessing, a prior stage to computing CNN activation is used in the implementation where the pixel mean value is subtracted from RGB channels through the whole training set corresponding to each pixel. In TexFusionNet, the pretrained convolutional neural network is further trained using Adadelta optimizer where the base learning rate is 0.001 and the weight decay is 0.0006 during the performance evaluation using the proposed TexFusionNet. The classification performance of CNN feature is evaluated in terms of the classification accuracy using $K$-fold cross-validation test. The ROC is also measured to investigate the performance in terms of the True Positive Rate (TPR) and False Positive Rate (FPR).

3.3 Texture Classification Results

The training and testing texture classification results in terms of the loss and accuracy against the number of epochs using proposed TexFusionNet are depicted in Fig. 4 over Brodatz texture database. The first plot shows the loss versus epochs, whereas the second plot demonstrates the accuracy versus epochs. The similar results over CUReT and KTH-TIPS texture databases are illustrated in Figs. 5 and 6, respectively. It can be observed in the plots of loss and accuracy versus epochs that the loss is decreasing and accuracy is increasing over the epochs. Roughly within 10 epochs of

![Fig. 4 The performance of texture classification using the TexFusionNet method over Brodatz texture database, a Loss versus epochs and b Accuracy versus epochs](image-url)
training, the proposed TexFusionNet attains a very high classification result which depicts the benefit of using the pretrained weights of AlexNet and VGG16 models.

The performance comparison of proposed TexFusionNet with state-of-the-art hand-crafted as well as deep learning based methods is also carried out. Table 1 summarized the classification results over Brodatz, CUReT, and KTH-TIPS databases for different methods. The results for hand-crafted descriptors such as LBPV [18], BRINT [19], CLBP [17], PRICOLBP [47], COALBP [48], CDCP [21], and RI-LBD [49] are compared in Table 1. The learning-based methods such as ScatNet [39, 40], PCANet [41], AlexNet [26], FV-VGGM [37], FV-VGGVD [37], and RandNet [41] are also compared in Table 1 with our method. It is observed from Table 1 that the proposed TexFusionNet outperforms the hand-crafted methods such as LBPV [18], BRINT [19], CDCP [21], RI-LBD [49], and also provides better classification performance compared to AlexNet [26], FV-VGGM [37], and other state-of-the-art deep learning based methods. Though the number of training samples is very less,
Table 1 Comparison of proposed TexFusionNet CNN feature with other variants of LBPs and state-of-the-art deep CNN methods in terms of the texture classification accuracy

<table>
<thead>
<tr>
<th>Hand-craft methods</th>
<th>Classification rates (%)</th>
<th>Deep CNN Methods</th>
<th>Classification rates (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Brodatz</td>
<td>KTH-TIPS</td>
<td>CUReT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LBPV [18]</td>
<td>93.80</td>
<td>95.50</td>
<td>94.00</td>
</tr>
<tr>
<td>BRINT [19]</td>
<td>95.22</td>
<td>97.75</td>
<td>97.06</td>
</tr>
<tr>
<td>CLBP [17]</td>
<td>94.80</td>
<td>97.19</td>
<td>97.40</td>
</tr>
<tr>
<td>PRICOLBP [47]</td>
<td>96.90</td>
<td>98.40</td>
<td>98.40</td>
</tr>
<tr>
<td>COALBP[48]</td>
<td>94.20</td>
<td>97.00</td>
<td>98.00</td>
</tr>
<tr>
<td>CDCP [21]</td>
<td>97.20</td>
<td>97.90</td>
<td>—</td>
</tr>
<tr>
<td>RI-LRD [49]</td>
<td>97.80</td>
<td>99.30</td>
<td>98.60</td>
</tr>
</tbody>
</table>

Fig. 7 The area under the curve (AUC) indicates the probability that a model provides classification scores between 0 and 1 ranks a randomly chosen true sample larger than a randomly chosen false sample

the proposed TexFusionNet-based CNN features achieve the outstanding average classification accuracy of 98.76%, 100%, and 99.76% for Brodatz, KTH-TIPS, and CUReT test suites, respectively.

To further visualize the performance of the proposed TexFusionNet in terms of the receiver operating characteristics (ROC), the area under curve (AUC) is measured and depicted in Fig. 7 over Brodatz, CUReT, and KTH-TIPS databases. The true positive rate (TPR) and false positive rate (FPR) values are plotted along y- and x-axes, respectively. It is observed from Fig. 7 that the performance of proposed model is reasonable over each texture database and proposed CNN feature achieves AUC values of 98.32%, 99.61% and 99.39%, respectively for Brodatz, KTH-TIPS, and CUReT test suites.
4 Conclusion

In this paper, a TexFusionNet model is proposed for the texture classification. The TexFusionNet used the existing pretrained CNN models of AlexNet and VGG16 and fused at last representation layer after removing the original class layer of these networks. The fusion is performed by adding the last representation layer of both networks. A fully connected layer is placed over the fusion layer to generate the class scores which is used to generate the loss during training with categorical cross-entropy loss function. After training, the fused model is used for the testing over texture images in classification framework. Three benchmark Brodatz, CURET, and KTH-TIPS texture databases are used to judge the performance of proposed model. The results are compared with the state-of-the-art hand-crafted and learning-based methods. The experimental results suggest that the introduced TexFusionNet outperforms the hand-crafted methods and also shows very promising performance as compared to the deep learning based methods.

References