Deep Multiple-Attribute-Perceived Network for Real-world Texture Recognition

Wei Zhai[1*], Yang Cao[1*], Jing Zhang[2] , Zheng-Jun Zha[1†]
[1]Department of Automation, University of Science and Technology of China
[2]UBTECH Sydney Artificial Intelligence Centre, The University of Sydney
wzhai056@mail.ustc.edu.cn, forrest,zhazj@ustc.edu.cn, jing.zhang1@sydney.edu.au

Abstract

Texture recognition is a challenging visual task as multiple perceptual attributes may be perceived from the same texture image when combined with different spatial context. Some recent works building upon Convolutional Neural Network (CNN) incorporate feature encoding with orderless aggregating to provide invariance to spatial layouts. However, these existing methods ignore visual texture attributes, which are important cues for describing the real-world texture images, resulting in incomplete description and inaccurate recognition. To address this problem, we propose a novel deep Multiple-Attribute-Perceived Network (MAP-Net) by progressively learning visual texture attributes in a mutually reinforced manner. Specifically, a multi-branch network architecture is devised, in which cascaded global contexts are learned by introducing similarity constraint at each branch, and leveraged as guidance of spatial feature encoding at next branch through an attribute transfer scheme. To enhance the modeling capability of spatial transformation, a deformable pooling strategy is introduced to augment the spatial sampling with adaptive offsets to the global context, leading to perceive new visual attributes. An attribute fusion module is then introduced to jointly utilize the perceived visual attributes and the abstracted semantic concepts at each branch. Experimental results on the five most challenging texture recognition datasets have demonstrated the superiority of the proposed model against the state-of-the-arts.

1. Introduction

Texture refers to the fundamental microstructure of natural images and a preattentive visual cue of human perception [34, 10]. Since textural properties summarize not only the fine-scale details (e.g. leaf veins) but also describe the rough semantic concepts (e.g. leaf), they forms an important mid-level image representation for natural scene understanding. The remarkable robustness of texture representation to local image deformations, variable illumination and occlusions, as well as its describable attributes for characterizing object, make it beneficial for many applications, such as image retrieval, terrain classification and industrial visual inspection [19, 35, 27].

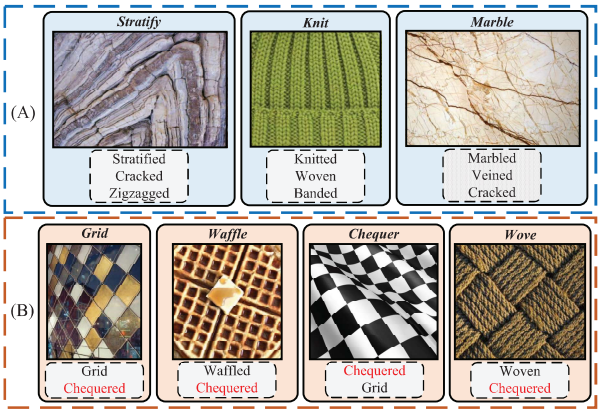

Figure 1. The part (A) shows that textures usually have multiple visual texture attributes. The part (B) shows that the images with different concepts may have the same visual texture attribute. The word above the image represents the concept to which the image belongs, and the words below the image represent the visual texture attributes contained in the image. All of the above images are from DTD datasets [2]

Texture recognition has been widely studied in the past years [19]. The classic approaches, such as texton histograms [6, 16, 22, 33], or bag-of-words [4, 15] usually apply a set of handcrafted filter banks to transform texture image into local features, and then aggregate them into a global representation. Recently, significant progresses on texture recognition have been achieved by deep convolutional neural networks (DCNNs). M. Cimpoi et al. [3] firstly introduce pre-trained CNNs into texture feature encoding for robust representation. Lin et al. [18] use a bilinear module instead of fully connected layers to capture the second-order variation of texture feature. To achieve spa-

*co-first author
†corresponding author

tial invariant representation, Zhang et al. [37] present Deep Texture Encoding Network (DeepTEN) that integrates dictionary learning and residual encoding into CNN to form an end-to-end texture recognition network. By leveraging DeepTEN as a texture encoding layer, J. Xue et al. [35] further present a Deep Encoding Pooling Network (DEP) to capture the orderless texture details together with local spatial information for ground terrain recognition.

Although the existing texture recognition methods work well on providing invariance to spatial context, there remain gaps when dealing with textures in the wild. The major reason is that the underlying visual attributes of texture images are not well exploited in recognition. Visual textures attributes play an important role in object descriptions, particularly for those objects that are best characterized by a pattern with semantic concept, such as a banded shirt or a striped zebra. Seeking for the best texture representation with describable texture attributes, M. Cimpoi et al. [2] design a public benchmark, DTD dataset, in which 47 texture terms are identified from the cognitive aspects of texture perception and used to describe a large dataset of patterns collected in the wild.

Different from the problem of object recognition, visual attributes and sematic descriptions of textures are highly inter-correlated but do not imply each other. As shown in Fig. 1, one texture description may include a combination of multiple visual attributes (e.g. marble can be marbled, veined and cracked at the same time). Meanwhile, one attribute may imply multiple texture description (e.g. Chequered may equally apply to grid or marble). This textural property makes the network tend to focus on the statistically dominant attributes while ignoring other visual attributes that are equally discriminative, if we just port from general object recognition CNN framework to texture recognition.

To address this problem, this paper proposes a novel deep Multiple-Attribute-Perceived Network (MAP-net) by progressively learning visual texture attributes in a mutually reinforced manner. We find that there are inherent correlations between the spatial contexts corresponding to multiple visual attributes of texture images. Therefore, we devise a multi-branch network architecture to take advantage of the correlation between attributes for texture recognition. Specifically, the cascaded global contexts are learned by introducing similarity constraint at each branch, and leveraged as guidance for spatial feature encoding at next branch through an attribute transfer scheme. Moreover, a deformable pooling strategy is introduced to achieve adaptive spatial sampling based on global context, leading to perceive new visual attributes without additional supervision. An attribute fusion module is then introduced to jointly utilize the perceived visual attributes and the abstracted semantic concepts at each branch. The resultant network shows excellent performance not only for describing texture dataset DTD, but also for general material databases (FMD [30, 29], GTOS [36] and GTOS-mobile [35]).

Our contributions are summarized as follows:

(1) This paper presents a novel Deep Multiple-Attribute-Perceived Network (MAP-Net) by progressively learning visual texture attributes in a mutually reinforced manner, providing a complete description for real-world texture image.

(2) This paper proposes a multi-branch network architecture with an attribute transfer scheme, in which cascaded global contexts are learned by introducing similarity constraint at each branch, and leveraged as guidance for spatial feature encoding at next branch, leading to perception of multiple visual attributes.

(3) Experimental results on the five most challenging texture recognition datasets have demonstrated the superiority of the proposed model against the state-of-the-arts.

2. Related Work

Texture recognition has been a topic of intensive research in the fields of computer vision due to its important role in a wide variety of applications. Here we only present the research most relevant to the work of this paper. Please refer to [19] for a comprehensive review of texture recognition.

The research of texture representation traditionally embraces three problems including feature extraction, feature encoding and feature aggregation. Typical local features used in texture representation include various filter banks [17, 32], gray level co-occurrence matrix [12], LBP [24], SIFT [20] and HOG [9]. Given the extracted texture features from an image, some studies such as Bag-of-Words model (BoWs) [5], Vector of Locally Aggregated Descriptors (VLAD) [13] and Fisher Vector [25] are presented to map local features to texton dictionaries, resulting in feature coding vectors. Then a global feature representation is produced by aggregating the coded feature vectors, and used as the basis for recognition. Many approaches such as Nearest Neighbor Classifier (NNC), Support Vector Machines (SVM), neural networks, and random forests are candidates for texture recognition.

Recently, some deep leaning based texture representation methods have been proposed by employing pretrained CNN models or performing finetuning for specific tasks. The previous CNN based method usually combines the CNN feature extraction with the BoWs method. For example, FVCNN [3] and LFV [31] directly use the FV method to encode features extracted by the deep network. The advance of these existing methods is the powerful basic feature representation (e.g. CNN and SIFT) and the ability of features coding (e.g. FV-based method). Furthermore, some methods [18, 11, 8] introduce orderless bilinear pooling module to polarize strong response and weak response features. DeepTEN [37] uses the classic idea of residual

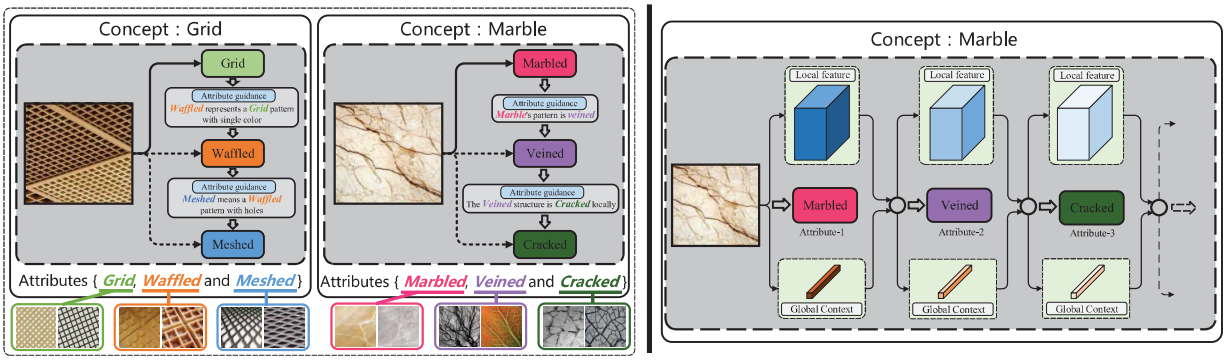

Figure 2. The part (a) shows that a texture image has multiple visual texture attributes, and there are correlations among these attributes. The part (b) shows the multiple visual attributes learning process which is summarized by our motivation.

dictionary learning to encode feature, embedding the coding layer into the network for end-to-end training. DEP [35] uses DeepTEN as a texture encoding layer to capture orderless texture details and local spatial feature.

Different from the existing texture recognition methods, this paper aims to perceive multiple visual attributes to achieve optimal real-world texture representation by leaming cascaded global context as guidance. Technically, a Deep Multiple-Attribute-Perceived Network (MAP-Net) with cascaded multi-branch architecture is proposed to progressively learn visual texture attributes in a mutually reinforced manner. In particular, our work seems to be similar to that of multi-label learning [38], which is to specify multiple labels (visual attributes) for an input texture image. However, different from the multi-label learning that each training sample is associated with several labels, each texture image has only one attribute label as supervision, which means that our work is more challenging. To address this problem, we utilize the inherent correlations between the attributes of the same texture for perception of multiple visual attributes.

3. Multiple-Attribute-Perceived Network

3.1. Motivation

The human visual system represents image patches as textures, which not only summarizes the fundamental microstructure of natural image but also the perceptual attributes from pre-attentive aspects [34, 10]. This gives us an inspiration: a good texture representation should summarize the granularity and repetitive patterns of object surfaces, as well as the describable attributes for characterizing object. However, it is a challenging task to take visual attributes of textures into account for recognition, since multiple perceptual attributes may be identified from the same texture image when combined with different spatial context for perceiving.

Our proposed method is based on the observation that: 1) there are occurrences between the visual attributes. For example, the attribute *grid* tends to co-exist with *waffled*, 2) there are also correlations between the image contexts corresponding to attributes. As illustrated in Fig. 2 (a), the texture with description of 'grid' contains three attributes: *Grid*, *Waffled* and *Meshed*, and the texture with description of 'marble' contains three attributes: *Marbled*, *Veined* and *Cracked*. There are dependencies between these attributes, e.g. *Waffled* represents a grid pattern with single color, and 'Mesh' means a *waffled* pattern with holes. When we observe the typical patterns of these attributes, we can find that the dependencies are also reflected in their corresponding image context. It is mainly manifested in two aspects: 1) these patterns have similar texture primitives, i.e., textons or repetitive patterns, 2) the locations and scales of the corresponding image contexts are correlated. For example, the attributes of *Marbled*, *Veined* and *Cracked* correspond to the image contexts at coarse-to-fine multiple scales. And the locations of image contexts related to *Grid*, *Waffled* and *Meshed* are somewhat coincidence.

Based on the above observations, we propose to utilize the correlation of image contexts corresponding to multiple attributes for texture recognition. As shown in Fig. 2 (b), we combine local spatial encoded features related to texture details with global context features corresponding to visual attributes to generate texture representation. To take advantage of dependencies between these attributes, we devise a cascaded multi-branch architecture with an attribute transfer scheme, in which cascaded global context are leaned and leveraged as guidance to perceive multiple visual attributes.

3.2. Architecture

In this paper, we propose a novel deep Multiple-Attribute-Perceived Network (MAPnet) for real-world texture recognition by progressively learning visual texture at-

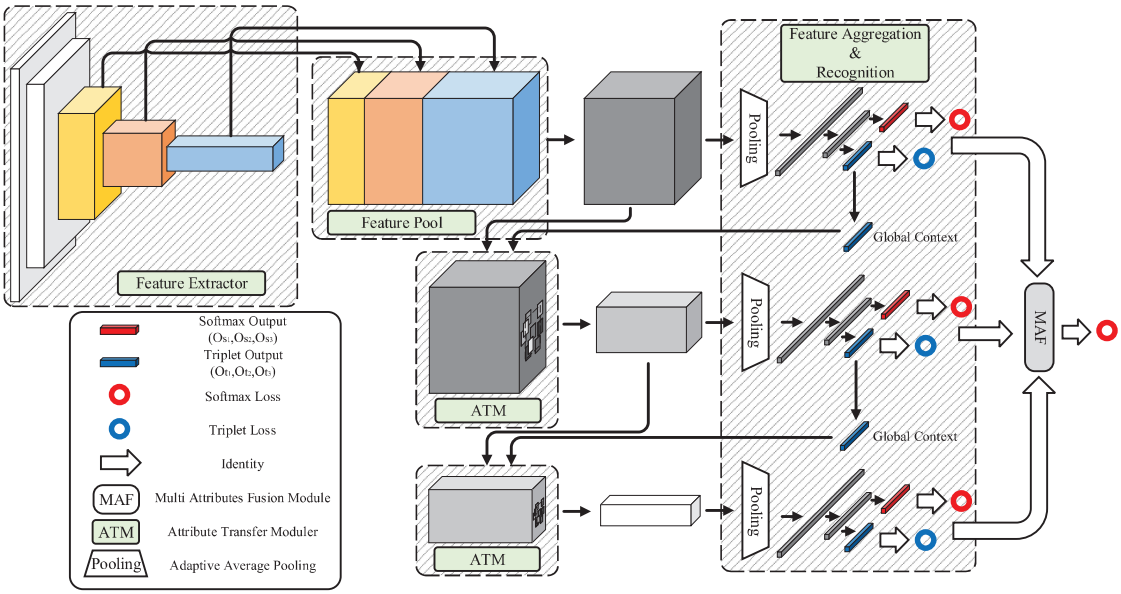

Figure 3. Architecture of our proposed MAP-net. MAP-net is a cascading structure, in which first branch is divided into three parts: feature extractor, feature pool, and feature aggregation. The attribute transformation layer is used to connect between adjacent branches. Finally, the results of each branch are merged through a multi attributes fusion module. To get a better description of attributes, a similarity constraint is introduced. A deformable pooling strategy with adaptive offsets to global context is introduced in ATM to enhance the modeling capability of spatial transformation.

tributes in a mutually reinforced manner. The overall architecture of our proposed method is illustrated in Fig. 3. Specifically, a multi-branch CNN is proposed, in which cascaded global contexts are learned by introducing similarity constraint at each branch, and leveraged as guidance for spatial feature encoding at next branch through an attribute transfer scheme. To perceive new visual attributes at each branch, a deformable pooling strategy with adaptive offsets to global context is introduced to enhance the modeling capability of spatial transformation. The perceived visual attributes and the abstracted semantic concepts at each branch are jointly utilized by an attribute fusion module, achieving a comprehensive recognition.

Cascaded global context. Cascaded global contexts related to visual attributes are learned by a multi-branch CNN model. The first branch of our proposed model can be divided into three parts: feature extractor, feature pool, and feature aggregation. Similar to [37], we also use Resnet50 as our feature extractor. To capture the comprehensive local spacial feature, we concatenate different levels of features together as a feature pool to capture more comprehensive local features. We upsample different feature maps to obtaion the same resolution as the original feature map via bilinear interpolation. To decrease the computational cost, a 1×1 convolution layer is used to reduce the channel to 2048. To adapt to local features with different sizes, we adopt adaptive average pooling to achieve orderless aggregation. Finally, the predicted semantic concepts and global context are obtained through the non-linear transformation by full connection layer. Furthermore, to get a better description of attributes, we introduce similarity constraints into the feature space to make the intra-class tighter and the inter-class more dispersed by using triplet loss [28]. A triplet consists of three images, denoted as (d_i, p_i, n_i), where $i \epsilon N$, N is the number of triplets, d_i is the reference image from a specific class, p_i an image from the same class, and n_i an image from a different class. The triplet loss is defined as follows:

$$E_{triplet}(d,p,n,m) = \frac{1}{2N}\sum_{i=1}^{N} max\left\{0, D(d_i,p_i) - D(d_i,n_i) + m\right\}, \quad (1)$$

where $D(d_i,p_i) = \|f_{triplet}(d_i) - f_{triplet}(p_i)\|_2^2$ and $D(d_i,n_i) = \|f_{triplet}(d_i) - f_{triplet}(n_i)\|_2^2$, $f_{triplet}$ is one of the output of the fully connected layer for measuring distances in feature space, $\|\cdot\|_2^2$ is the squared Euclidean distance between two l_2-normalized vectors, m is a certain margin $m > 0$. Theoretically, triplet loss can effectively constrain the intra-class relationship with the inter-class relationship.

Combining the advances of softmax loss and triplet loss, we jointly optimize two kinds of losses by a multi-task learning strategy. Here the two kinds of losses are integrated

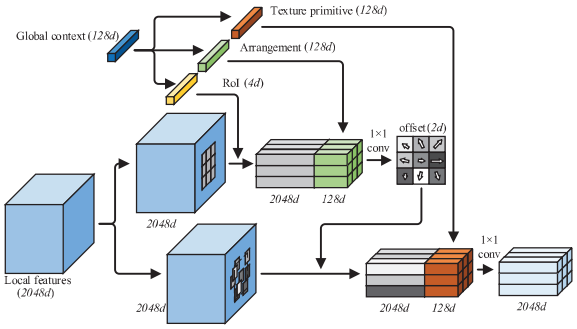

Figure 4. Attribute Transfer Module. A deformable pooling strategy with adaptive offsets to global context is introduced in attribute transfer module to enhance the modeling capability of spatial transformation.

though linear weighed summation:

$$E = (1-\lambda)E_{softmax} + \lambda E_{triplet}(d,p,n,m), \quad (2)$$

where $E_{softmax} = \sum_{i\epsilon\{d,p,n\}} E_{softmax}(i,l_i)$, where λ is the balance weight. We jointly optimizing softmax loss and triplet loss, which not only improvs the representation capability of CNN, but also resolves the issue of slow convergence when only using the triplet loss.

Attribute transfer module. Through an attribute transfer module, the cascaded global context corresponding to visual attribute are leveraged as guidance of spatial feature encoding at the next branch [26]. To enhance the modeling capability of spatial transformation, a deformable pooling strategy is introduced to augment the spatial sampling with adaptive offsets to the global context, leading to perceive new visual attributes. Here, we define the global contexts that imply the correlation between multiple attributes as region of interest (RoI), spatial arrangement and texture primitive, respectively. The structure of the attribute transfer module is shown in Fig. 4. First, we transform global context information into RoI vector which is four-dimensional vector containing size w, h ($w \times h$) and a top-left corner $\mathbf{p}_0$, arrangement vector and texture primitives vector through three fully connection layers. Then we use the arrangement vector to guide the offset generation, which can be expressed as follows:

$$\mathbf{\Delta p} = \sigma(b + W\begin{bmatrix} repeat(\mathbf{a}) \\ \mathbf{x} \end{bmatrix}), \quad (3)$$

Here $\mathbf{\Delta p}$ is the offset, $\mathbf{x}$ is the input feature map and $\mathbf{a}$ is arrangement vector, $repeat$ is stacking operation which aim to transfer vector feature to feature block, W is a weight matrix ($1 \times 1 \times (2048+128)$), and b is a bias. Both W and b are learnable. To ensure the integrity of the local feature, we only focus on spatial location in deformable ROI pooling, Thus the form of offset is $w \times h \times 2$ in this paper. The

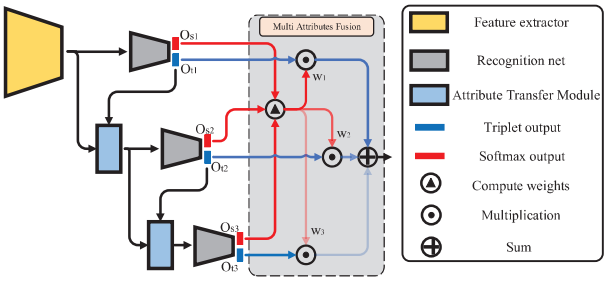

Figure 5. Multi Attributes Fusion Module. Multi attributes fusion module uses the semantic concepts learned by each branch to obtain a weight vector to re-weight the attribute descriptions perceived.

pooling process can be expressed as follows:

$$y(i,j) = \mathbf{x}(\mathbf{p_0} + \mathbf{p_{ij}} + \mathbf{\Delta p_{ij}}), \quad (4)$$

where y is the output of feature map, $\mathbf{p}$ is the original sampling points position. $\mathbf{p_n} + \mathbf{\Delta p_n}$ is the sampling points position after the offset is introduced, where $\mathbf{n}$ is the collection of original sampling points and $\mathbf{ij}\epsilon\mathbf{n}$. As the offset $\mathbf{\Delta p_n}$ is typically fractional, Eq. 4 is also implemented via bilinear interpolation as [7]. Then, we use the same structure to introduce texture primitive vector into a new spatial feature encoding.

Multi attributes fusion. To make full use of the attribute representations under different branches, a multi attributes fusion module is proposed. Fig. 5 shows the structure of the multi attributes fusion module. We consider the difference between the semantic concepts learned by each branch. Through a nonlinear transformation (fully connection layer), we obtain a weight vector to re-weight the attribute descriptions perceived by each branch. Finally, a comprehensive texture feature representation is obtained by summation. This process can be expressed as:

$$W = softmax(f_w(cat(O_{t1}, O_{t2}, O_{t3}))), \quad (5)$$

where W is the weight vector, $f_w(\cdot)$ is the fully connection layer for getting weight vector, $cat(\cdot)$ is the concatenate operation, $softmax(\cdot)$ is the softmax function, $O_{t.}$ is the triplet output of three branches. The final result is obtained by weighting fusion: $r = f_r(\sum_{i\epsilon\{1,2,3\}} w_i O_{s_i})$, where r is the fusion result, f_r is the fully connection layer for getting final recognition result. $O_{s.}$ is the semantic concept prediction output of three branches. The loss of fusion result is: $E_{fusion} = \sum_{r\epsilon\{d,p,n\}} E_{softmax}(r,l)$, where d is the reference image from a specific class, p is an image from the same class, and n is an image from a different class, E_{fusion} is the loss of fusion result, $E_{softmax}(\cdot)$ is the softmax cross entropy loss, l is the true label. Combining this fusion loss, we present the loss function of our MAP-net:

$$E_{total} = \alpha E_{fusion} + \sum_{i\epsilon\{1,2,3\}} \beta_i E_i, \quad (6)$$

where E_{total} is the total loss of MA-net, $E.$ is the loss of each branch ($softmax + triplet$), α is the weight of fusion loss, $\beta.$ is the weight of each branch. In this paper, we set $\alpha : \sum_{\cdot} \beta(\cdot) = 1 : 1$.

4. Experiments

In this section, we evaluate the proposed method on five texture/material datasets and then analyze the effectiveness of the method.

4.1. Datasets and implementation

Experimental data. We evaluate our model on five most challenge representative texture/material recognition datasets. Describable Texture Database (DTD) [2] is an attribute-based texture dataset contains 47 texture categories with a total of 5640 images. Flickr Material Dataset (FMD) [30, 29] consists of 10 material categories, each of which contains 100 images. KTH-TISP2b (KTH) [1, 23] dataset contains 11 material categories with a total of 4752 images. Ground Terrain in Outdoor Scenes (GTOS) [36] is a dataset of ground materials in outdoor scene with 40 categories. GTOS-mobile [37] is collected from GTOS dataset by mobile phone, which consists of 31 material classes. For DTD, FMD and KTH datasets, we randomly divide each dataset into 10 splits and report the mean accuracy across splits. For GTOS and GTOS-mobile datasets, the evaluation is based on provided train-test splits. Similar to [37], the result accuracy $mean \pm std\%$ are reported. The results on DTD, FMD, KTH and GTOS datasets are based on 5-time statistics, and the results on GTOS-mobile datasets are averaged over 2 runs.

Implementation details. We implement our model with PyTorch, two TITAN Xp GPUs are used for training and testing. Resnet-50 is used as feature extractor, in which *res3d*, *res4f* and *res5c* are used as the source of feature pool, the feature maps channels from feature pool are reduced to 2048 by 1 convolutional layer. Our model is trained using stochastic gradient descent with mini-batch size 256. We use Nesterov momentum with a weight of 0.9 without dampening, and a weight decay of 10^{-4}. Our model is trained for 30 epochs with a learning rate of 10^{-4}. To get the best result, we set $\alpha = 1.0$, $\beta_1 = 0.6$, $\beta_2 = 0.3$ and $\beta_3 = 0.1$, respectively. We generate *100k, 50k, 50k, 100k* and *100k* triplets as train samples in every fold on DTD, KTH-T2b, FMD, GTOS and GTOS-mobile, respectively. We set 128 as the output dimension of the triplet with margin 0.2. At the training phase, we set 0.6 learning rate for the feature extractor, set 1.0 learning rate for other networks. For all datasets, the training images are randomly cropped to $80\% \sim 100\%$ of the image areas, keeping the aspect ratio between $\frac{3}{4}$ and $\frac{4}{3}$. The training images are resized to 256×256. 50% images are chosen to horizontal flips and vertical flips. At the testing phase, we use the resolution of 224×224 for all datasets.

Table 1. Setting an appropriate loss weight λ in the similarity constrained loss is important. 'SC' denotes the similarity constraint loss. The baseline is MAP-net only use softmax loss. Empirically, $\lambda = 0.2$ yields the best performance.

	DTD	GTOS
$\lambda = 0$ (*without SC*)	$75.0_{\pm 1.2}$	$83.8_{\pm 2.7}$
$\lambda = 0.2$	$\mathbf{76.1_{\pm 0.6}}$	$\mathbf{84.7_{\pm 2.2}}$
$\lambda = 0.4$	$75.2_{\pm 0.9}$	$83.5_{\pm 2.5}$
$\lambda = 0.6$	$74.6_{\pm 1.0}$	$82.9_{\pm 2.5}$

Table 2. Ablation study on DTD and GTOS-mobile datasets. 'F-P' is Feature Pool. 'SC' is Similarity Constraint. 'MAF' Multi Attributes Fusion. N represents the number of cascades.

Model	N	FP	SC	MAF	DTD	GTOS
Resnet50	1				$68.9_{\pm 1.2}$	$76.0_{\pm 2.8}$
	2				$71.1_{\pm 1.1}$	$78.6_{\pm 2.8}$
Baseline	3				$72.1_{\pm 1.0}$	$80.0_{\pm 2.8}$
	3	✓			$73.9_{\pm 1.1}$	$81.8_{\pm 2.7}$
	3		✓		$74.1_{\pm 0.6}$	$81.4_{\pm 2.3}$
	3			✓	$74.5_{\pm 1.1}$	$81.5_{\pm 2.8}$
	3	✓	✓		$75.6_{\pm 0.6}$	$82.8_{\pm 2.2}$
	3		✓	✓	$75.2_{\pm 0.7}$	$82.1_{\pm 2.2}$
	3	✓		✓	$75.0_{\pm 1.2}$	$83.8_{\pm 2.7}$
MAP-net	3	✓	✓	✓	$\mathbf{76.1_{\pm 0.6}}$	$\mathbf{84.7_{\pm 2.2}}$

4.2. Ablation study

To evaluate MAP-net, we conduct experiments with several settings, including Feature Pool (FP), Similarity Constrained loss (SC) and Multi Attributes Fusion module (MAF).

Similarity Constraint Loss. In this section, we explore the effectiveness of similarity constraint loss for MAP-net. We set the similarity constraint loss (SC) weight λ between 0 and 1 and show the results in Table 1. It is worth noting that when the similarity constraint loss is not used, we choose the output of the penultimate layer of the fully connected layer as the global context information, and the other parameters are unchanged. For DTD and GTOS, adding the similarity constraint loss, $\lambda = 0.2$ yields the best performance. It outperforms the baseline with an improvement of $1.1\%/0.9\%$ in terms of mean accuracy. As shown in Table 2, The performance is improved by $2.0\%/1.4\%$ (Baseline $\rightarrow$ Baseline+SC) and $1.1\%/0.9\%$ (Baseline+FP+MAF $\rightarrow$ MA-net). It indicated that taking into account the similarity constraint is helpful for texture representation.

Number of Cascades. To study the effect of cascading numbers, we set the number of branches *N=1,2,3* (*N=1* is general Resnet50), respectively. We perform the experiments on DTD, KTH-T2b and FMD datasets. As shown in

Table 3. The contribution of each branch to the final result. Where $branch\_1 + branch\_2 + branch\_3 = 100\%$.

	$branch\_1$	$branch\_2$	$branch\_3$
DTD	78.3	17.1	4.6
KTH-T2b	80.6	13.6	5.8
FMD	88.7	7.9	3.4
GTOS	88.9	8.8	2.3
GTOS-mobile	90.1	6.8	3.1

Table 2. From the overall point of view, '*N=1*' < '*N=2*' < '*N=3*'. It shows that cascade structure helps to improve the final recognition result. Each additional branch would make the final result better. It proves that using cascaded structures can get more potential discriminating basis. For DTD dataset, the performance is $68.9\% \rightarrow 71.1\% \rightarrow 72.1\%$. For GTOS dataset, the performance is $76.0\% \rightarrow 78.6\% \rightarrow 80.0\%$. On the whole, the growth rate of the final result decreases with the number of branches increasing. It shows that with the increasing of branches, the representation ability of network is nearly saturated.

Feature Pool. We evaluate the effectiveness of feature pool. As shown in Table 2, The performance is improved by $1.8\%/1.3\%$ (Baseline $\rightarrow$ Baseline+FP) and $0.9\%/2.6\%$ (Baseline+SC+MAF $\rightarrow$ MAP-net). The feature pool combines different levels of features, which provided a good basic representation to the next feature encoding module and feature aggregation.

Multi Attributes Fusion. In this part, we study the effect of the multi attributes fusion module. The multi attributes fusion module is designed to provide a fusion strategy for the final recognition result. This strategy takes into account the contribution of different attributes and adaptively fusing. As shown in Table 2, The performance is improved by $2.4\%/1.5\%$ (Baseline $\rightarrow$ Baseline+MAF), $1.1\%/0.7\%$ and $0.5\%/1.9\%$ (Baseline+SC+MAF $\rightarrow$ MAP-net). It is proved that the multi attributes fusion module had a good adaptive weight selection effects.

4.3. Performance Analysis.

To explore the contribution of each branch to the final fusion result, we choose all the samples predicted correctly in test set (n-fold average). And the highest weighted branch calculated by the multi attributes fusion module is chosen as major recognition evidence. We compute the distribution of the major recognition evidence of the test set. As shown in Table 3, we find that the contribution of '$branch\_1$' to the five datasets is the largest, respectively. And the contributions of the other two branch are $17.1\% \rightarrow 4.6\%$, $15.6\% \rightarrow 3.8\%$, $5.9\% \rightarrow 1.4\%$, $8.8\% \rightarrow 0.3\%$ and $5.8\% \rightarrow 1.1\%$, respectively. It confirms that even if '$branch\_1$' plays an important role, the contributions of the other two branches can not be ignored.

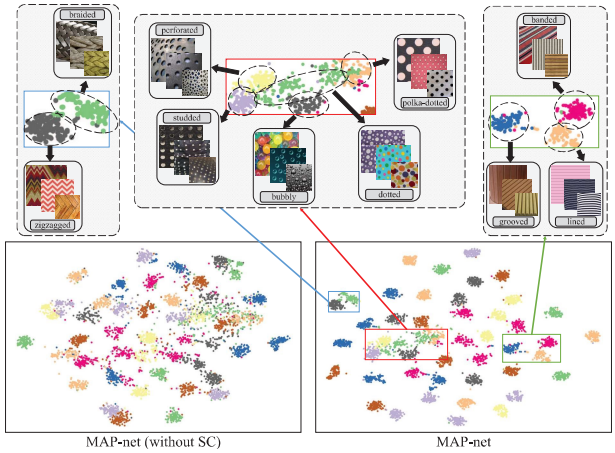

Figure 6. Visualization of the extracted features of MAP-net (without MC) and MAP-net after dimension reduced by t-SNE [21]. Here we use Resnet50 as the backbone and train two models on DTD dataset.

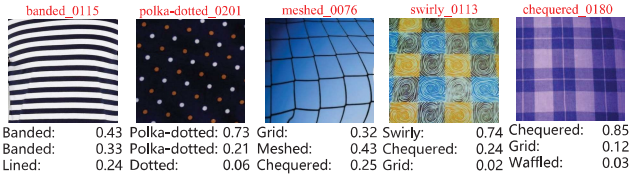

Figure 7. Attributes learned from different branches and their fusion weights. The order of results corresponds to that of branch.

For further analyzing the influence of similarity constraint loss for MAP-net, we preform a experiment, in which we extract features of baseline model MAP-net (without MC) and the final model MAP-net and visualize them in Fig. 6, The feature vector dimension is reduced by t-SNE [21]. Compared with MAP-net (without MC), MAP-net shows a better feature distribution that exhibits smaller intra-class differences and larger inter-class variations. It demonstrates that the features of jointly optimizing softmax loss and triplet loss are consistently much better separated than only using softmax loss and enable MAP-net to achieve a more discriminative feature representation.

In this paper, we use a novel deep Multiple-Attribute-Perceived Network (MAP-net) by progressively learning multiple visual texture attributes in a mutually reinforced manner which are contained in one image with the same semantic concept supervision. Fig. 7 shows the attributes learned from different branches, where the weights are obtained by multi attributes fusion module. In overall, the attributes mined by each branch are the same as human understanding, and there is a correlation between these attributes.

Table 4. Comparison with state-of-the-art. For a fair comparison, we experiment on MAP-net using different backbone (VGG-19, Resnet-18 and Resnet-50). When Resnet-18 is used as feature extractor, *res3b*, *res4b* and *res5b* are used as the source of feature pool, the feature maps channels from feature pool are transferred to 2048 with 1×1 convolutional layer. And when VGG-19 is used as feature extractor, *conv5_4*, *conv4_4* and *conv3_4* are used as the source of feature pool, the feature maps channels from feature pool are transferred to 2048 with 1 convolutional layer. Here 'VGGVD' is VGG-19.

Texture Dataset	Backbone	DTD		KTH-T2b		FMD		GTOS		GTOS-mobile	
Method		*mean*	*std*	*mean*	*std*	*mean*	*std*	*mean*	*std*	*mean*	*std*
FV-VGGVD [3]	VGGVD	72.3	1.0	75.4	1.5	79.8	1.8	77.1	–	–	–
FC-VGGVD [3]	VGGVD	62.9	0.8	81.8	2.5	77.4	1.8	–	–	–	–
B-CNN [18]	VGGVD	69.6	0.7	5.1	2.8	77.8	1.9	–	–	–	–
B-CNN [35]	Resnet18	–	–	–	–	–	–	–	–	75.43	–
LFV [31]	VGGVD	73.8	1.0	82.6	2.6	82.1	1.9	–	–	–	–
FASON [8]	VGGVD	72.3	0.6	76.5	2.3	–	–	–	–	–	–
DeepTEN [37]	Resnet50	69.6	–	82.0	3.3	80.2	0.9	84.5	2.9	–	–
DeepTEN [35]	Resnet18	–	–	–	–	–	–	–	–	76.12	–
DEP [35]	Resnet50	73.2	–	–	–	–	–	–	–	–	–
DEP [35]	Resnet18	–	–	–	–	–	–	–	–	82.18	–
DEP [35]	Resnet18	–	–	–	–	–	–	–	–	82.18	–
PN [14]	Resnet50	–	–	–	–	85.5	–	–	–	–	–
MAP-net	VGGVD	74.1	0.6	82.7	1.5	82.9	0.9	80.8	2.5	82.0	1.6
MAP-net	Resnet18	69.5	0.8	80.9	1.8	80.8	1.0	80.3	2.6	82.98	1.6
MAP-net	Resnet50	**76.1**	0.6	**84.5**	1.3	**85.7**	0.7	**84.7**	2.2	**86.64**	1.5

For image $chequered\_0180$, there are three attributes (*chequered*, *grid* and *waffled*) in this texture image, and there is a connection among the three attributes ($e.g.$ *grid* can be regarded as the whole structure of *chequered*, while *waffled* can be regarded as *chequered* with the same color).

4.4. Comparisons against state-of-the-arts.

We compare the performance of our method with several texture/material recognition methods on DTD, KTH-T2b, FMD, GTOS and GTOS-mobile datasets. All the methods are listed as follows: **F**isher **V**ector **CNN** (**FV-VGGVD**) [3]. General VGG-19 (**FC-VGGVD**) [3]. **B**ilinear-CNN (B-CNN) [18]. **F**irst **A**nd **S**econd **O**rder information fusion **N**etwork (**FASON**) [8]. **Deep T**exture **E**ncoding **N**etwork (**DeepTEN**) [37]. **L**ocally-Transferred **F**isher **V**ectors (**LFV**) [31]. **D**eep **E**ncoding **P**ooling Network (**DEP**) [35]. **P**ower **N**ormalizations Network (**PN**) [14]. Table 4 shows the final result. Under the same backbond, our method achieves the best performance among all texture/material recognition methods. For DTD, KTH-T2b and FMD datasets, our method is $0.3\%/0.1\%/0.2\%$ higher than the state-of-the-art method LFV (based VGGVD) and PN (based Resnet50). Our method achieves a more distinguishable feature representation based on end-to-end training without additional handcrafed features. For GTOS dataset, our method is 0.4% higher than the state-of-the-art method DeepTEN (based Resnet50) which is built by a deep texture encoding layer. For GTOS-mobile dataset, our method is 0.8% higher than the state-of-the-art method DEP (based Resnet18) which is benefited from the deep texture encoding layer and its association with orderless bilinear pooling. It proves that the global context information guides the local spatial features to perceive multiple visual texture attributes, resulting in a more comprehensive feature representation. Furthermore, the overall variance of our results are smaller than that of other methods. It demonstrates that our method is more stable than the compared methods.

5. Conclusion

In this paper, we propose a novel deep Multiple-Attribute-Perceived Network (MAP-Net) by progressively learning visual texture attributes in a mutually reinforced manner. By considering the multiple visual texture properties that texture images have, we provide a more comprehensive description of real-world texture image. Specifically, a multi-branch network architecture with an attribute transfer scheme is proposed, in which cascaded global contexts are learned by introducing similarity constraint at each branch, and leveraged as guidance of spatial feature encoding at next branch, leading to perception of multiple visual attributes. Our MAP-net outperforms the state-of-the-art texture/material recognition methods on the five most challenging datasets: DTD, KTH-T2b, FMD, GTOS and GTOS-mobile. Leveraging our proposed texture recognition model to comprehensive scene understanding will be our future work.

Acknowledgments: This work was supported by the National Natural Science Foundation of China (NSFC) under Grants 61872327, 61472380, 61622211 and 61620106009 as well as the Fundamental Research Funds for the Central Universities under Grant WK2380000001 and WK2100100030.

References

[1] Barbara Caputo, Eric Hayman, and P Mallikarjuna. Class-specific material categorisation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1597–1604, 2005.

[2] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3606–3613, 2014.

[3] Mircea Cimpoi, Subhransu Maji, and Andrea Vedaldi. Deep filter banks for texture recognition and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3828–3836, 2015.

[4] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, pages 1–2, 2004.

[5] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, pages 1–2, 2004.

[6] Oana G Cula and Kristin J Dana. Compact representation of bidirectional texture functions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages I–I, 2001.

[7] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017.

[8] Xiyang Dai, Joe Yue-Hei Ng, and Larry S Davis. Fason: First and second order information fusion network for texture recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7352–7360, 2017.

[9] Dalal, Navneet, Triggs, and Bill. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005.

[10] Jeremy Freeman and Eero P Simoncelli. Metamers of the ventral stream. *Nature neuroscience*, page 1195, 2011.

[11] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 317–326, 2016.

[12] Robert M Haralick, Karthikeyan Shanmugam, et al. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, pages 610–621, 1973.

[13] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3304–3311, 2010.

[14] Piotr Koniusz, Hongguang Zhang, and Fatih Porikli. A deeper look at power normalizations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5774–5783, 2018.

[15] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2169–2178, 2006.

[16] Thomas Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision (IJCV)*, 43(1):29–44, 2001.

[17] Thomas Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision (IJCV)*, pages 29–44, 2001.

[18] Tsung-Yu Lin and Subhransu Maji. Visualizing and understanding deep texture representations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2791–2799, 2016.

[19] Li Liu, Jie Chen, Paul Fieguth, Guoying Zhao, Rama Chellappa, and Matti Pietikainen. From bow to cnn: Two decades of texture representation for texture classification. *International Journal of Computer Vision (IJCV)*, pages 1–36, 2018.

[20] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, pages 91–110, 2004.

[21] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR)*, pages 2579–2605, 2008.

[22] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision (IJCV)*, 43(1):7–27, 2001.

[23] P Mallikarjuna, Alireza Tavakoli Targhi, Mario Fritz, Eric Hayman, Barbara Caputo, and Jan-Olof Eklundh. The kth-tips2 database. *Computational Vision and Active Perception Laboratory (CVAP), Stockholm, Sweden*, 2006.

[24] Timo Ojala, Pietik, Matti Inen, and Topi. *Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns.* Springer Berlin Heidelberg, 2000.

[25] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *European Conference on Computer Vision (ECCV)*, pages 143–156, 2010.

[26] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Mirrorgan: Learning text-to-image generation by redescription. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1505–1514, 2019.

[27] Yong Rui, Thomas S Huang, and Shih-Fu Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation (JVCIR)*, pages 39–62, 1999.

[28] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[29] Lavanya Sharan, Ce Liu, Ruth Rosenholtz, and Edward H Adelson. Recognizing materials using perceptually inspired features. *International Journal of Computer Vision (IJCV)*, pages 348–371, 2013.

[30] Lavanya Sharan, Ruth Rosenholtz, and Edward Adelson. Material perception: What can you see in a brief glance? *Journal of Vision*, pages 784–784, 2009.

[31] Yang Song, Fan Zhang, Qing Li, Heng Huang, Lauren J O'Donnell, and Weidong Cai. Locally-transferred fisher vectors for texture classification. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4922–4930, 2017.

[32] Manik Varma and Andrew Zisserman. Texture classification: Are filter banks necessary? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages II–691, 2003.

[33] Manik Varma and Andrew Zisserman. A statistical approach to texture classification from single images. *International journal of computer vision (IJCV)*, pages 61–81, 2005.

[34] Thomas SA Wallis, Christina M Funke, Alexander S Ecker, Leon A Gatys, Felix A Wichmann, and Matthias Bethge. A parametric texture model based on deep convolutional features closely matches texture appearance for humans. *Journal of vision*, pages 5–5, 2017.

[35] Jia Xue, Hang Zhang, and Kristin Dana. Deep texture manifold for ground terrain recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 558–567, 2018.

[36] Jia Xue, Hang Zhang, Kristin J Dana, and Ko Nishino. Differential angular imaging for material recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6940–6949, 2017.

[37] Hang Zhang, Jia Xue, and Kristin Dana. Deep ten: Texture encoding network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 708–717, 2017.

[38] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, pages 1819–1837, 2014.