Describing Textures in the Wild

Mircea Cimpoi
University of Oxford
mircea@robots.ox.ac.uk

Subhransu Maji
Toyota Technological Institute
smaji@ttic.edu

Iasonas Kokkinos
École Centrale Paris / INRIA-Saclay
iasonas.kokkinos@ecp.fr

Sammy Mohamed
Stony Brook
sammy.mohamed@stonybrook.edu

Andrea Vedaldi
University of Oxford
vedaldi@robots.ox.ac.uk

Abstract

Patterns and textures are key characteristics of many natural objects: a shirt can be striped, the wings of a butterfly can be veined, and the skin of an animal can be scaly. Aiming at supporting this dimension in image understanding, we address the problem of describing textures *with semantic attributes. We identify a vocabulary of forty-seven texture terms and use them to describe a large dataset of patterns collected "in the wild". The resulting* Describable Textures Dataset *(DTD) is a basis to seek the best representation for recognizing describable texture attributes in images. We port from object recognition to texture recognition the Improved Fisher Vector (IFV) and Deep Convolutional-network Activation Features (DeCAF), and show that surprisingly, they both outperform specialized texture descriptors not only on our problem, but also in established material recognition datasets. We also show that our describable attributes are excellent texture descriptors, transferring between datasets and tasks; in particular, combined with IFV and DeCAF, they significantly outperform the state-of-the-art by more than 10% on both FMD and KTH-TIPS-2b benchmarks. We also demonstrate that they produce intuitive descriptions of materials and Internet images.*

1. Introduction

Recently *visual attributes* have raised significant interest in the community [6, 12, 19, 27]. A "visual attribute" is a property of an object that can be measured visually and has a semantic connotation, such as the *shape* of a hat or the *color* of a ball. Attributes allow characterizing objects in far greater detail than a category label and are therefore the key to several advanced applications, including understanding complex queries in *semantic search*, learning about objects from *textual description*, and accounting for the content of images in great detail. Textural properties have an important role in object descriptions, particularly for those objects that are best qualified by a pattern, such as a shirt or the wing of a bird or a butterfly as illustrated in Fig. 1. Nevertheless, so far the attributes of textures have been investigated only tangentially. In this paper we address the question of whether there exists a "universal" set of attributes that can describe a wide range of texture patterns, whether these can be reliably estimated from images, and for what tasks they are useful.

Figure 1: Both the man-made and the natural world are an abundant source of richly textured objects. The textures of objects shown above can be described (in no particular order) as *dotted*, *striped*, *chequered*, *cracked*, *swirly*, *honeycombed*, and *scaly*. We aim at identifying these attributes automatically and generating descriptions based on them.

The study of perceptual attributes of textures has a long history starting from pre-attentive aspects and grouping [17], to coarse high-level attributes [1, 2, 35], to some recent work aimed at discovering such attributes by automatically mining descriptions of images from the Internet [3, 13]. However, the texture attributes investigated so far are rather few or too generic for a detailed description most "real world" patterns. Our work is motivated by the one of Bhusan et al. [5] who studied the relationship between commonly used English words and the perceptual properties of textures, identifying a set of words sufficient to describing a wide variety of texture patterns. While they study the psychological aspects of texture perception, the focus of this paper is the challenge of estimating such properties from images automatically.

Our **first contribution** is to select a subset of 47 *describable texture attributes*, based on the work of Bhusan et al., that capture a wide variety of visual properties of textures and to introduce a corresponding *describable tex-*

ture dataset consisting of 5,640 texture images *jointly* annotated with the 47 attributes (Sect. 2). In an effort to support directly real world applications, and inspired by datasets such as *ImageNet* [10] and the *Flickr Material Dataset* (FMD) [32], our images are captured "in the wild" by downloading them from the Internet rather than collecting them in a laboratory. We also address the practical issue of crowd-sourcing this large set of joint annotations efficiently accounting for the co-occurrence statistics of attributes and for the appearance of the textures (Sect. 2.1).

Our **second contribution** is to identify a *gold standard texture representation* that achieves state-of-the-art recognition of the describable texture attributes in challenging real-world conditions. Texture classification has been widely studied in the context of recognizing materials supported by datasets such as *CUReT* [9], *UIUC* [20], *UMD* [42], *Outex* [25], *Drexel Texture Database* [26], and *KTH-TIPS* [7, 15]. These datasets address material recognition under variable occlusion, viewpoint, and illumination and have motivated the creation of a large number of specialized texture representations that are invariant or robust to these factors [21, 25, 38, 39]. In contrast, generic object recognition features such as SIFT were shown to work the best for material recognition in FMD, which, like DTD, was collected "in the wild". Our findings are similar, but we also find that Fisher vectors [28] computed on SIFT features and certain color features, as well as generic deep features such as DeCAF [11], can significantly boost performance. Surprisingly, these descriptors outperform specialized state-of-the-art texture representations not only in recognizing our describable attributes, but also in a variety of datasets for material recognition, achieving an accuracy of 65.5% on FMD and 76.2% on KTH-TIPS2-b (Sect. 3, 4.1).

Our **third contribution** consists in several *applications* of the proposed describable attributes. These can serve a complimentary role for recognition and description in domains where the material is not-important or is known ahead of time, such as fabrics or wallpapers. However, can these attributes improve other texture analysis tasks such as material recognition? We answer this question in the affirmative in a series of experiments on the challenging FMD and KTH datasets. We show that estimates of these properties when used a features can boost recognition rates even more for material classification achieving an accuracy of 55.9% on FMD and 71.2% on KTH when used alone as a 47 dimensional feature, and 67.1% on FMD and 77.3% on KTH when combined with SIFT, simple color descriptors, and deep convolutional network features (Sect. 4.2). *These represent more than an absolute gain of 10% in accuracy over previous state-of the-art.* Furthermore, these attributes are easy to describe and can serve as intuitive dimensions to explore large collections of texture patterns – for example product catalogs (wallpapers or bedding sets) or material datasets. We present several such visualizations in the paper (Sect. 4.3).

2. The describable texture dataset

This section introduces the *Describable Textures Dataset* (DTD), a collection of real-world texture images annotated with one or more adjectives selected in a vocabulary of 47 English words. These adjectives, or *describable texture attributes*, are illustrated in Fig. 2 and include words such as *banded*, *cobwebbed*, *freckled*, *knitted*, and *zigzagged*.

DTD investigates the problem of **texture description**, intended as the recognition of describable texture attributes. This problem differs from the one of *material recognition* considered in existing datasets such as CUReT, KTH, and FMD. While describable attributes are correlated with materials, attributes do not imply materials (*e.g*. *veined* may equally apply to leaves or marble) and materials do not imply attributes (not all marbles are *veined*). Describable attributes can be *combined* to create rich descriptions (Fig. 3; marble can be *veined*, *stratified* and *cracked* at the same time), whereas a typical assumption is that textures are made of a single material. Describable attributes are *subjective* properties that depend on the imaged object as well as on human judgements, whereas materials are objective. In short, attributes capture properties of textures *beyond* materials, supporting human-centric tasks where describing textures is important. At the same time, they will be shown to be helpful in material recognition too (Sect. 3.2 and 4.2).

DTD contains **textures in the wild**, *i.e*. texture images extracted from the web rather than begin captured or generated in a controlled setting. Textures fill the images, so we can study the problem of texture description independently of texture segmentation. With 5,640 such images, this dataset aims at supporting real-world applications were the recognition of texture properties is a key component. Collecting images from the Internet is a common approach in categorization and object recognition, and was adopted in material recognition in FMD. This choice trades-off the systematic sampling of illumination and viewpoint variations existing in datasets such as CUReT, KTH-TIPS, Outex, and Drexel datasets for a representation of real-world variations, shortening the gap with applications. Furthermore, the invariance of describable attributes is not an intrinsic property as for materials, but it reflects invariance in the human judgements, which should be captured empirically.

DTD is designed as a **public benchmark**, following the standard practice of providing 10 preset splits into equally-sized training, validation and test subsets for easier algorithm comparison (these splits are used in all the experiments in the paper). DTD is publicly available on the web at `http://www.robots.ox.ac.uk/~vgg/data/dtd/`, along with standardized code for evaluation and reproducing the results in Sect. 4.

Figure 2: The 47 texture words in the **describable texture dataset** introduced in this paper. Two examples of each attribute are shown to illustrate the significant amount of variability in the data.

Related work. Apart from material datasets, there have been numerous attempts at collecting attributes of textures at a smaller scale, or in controlled settings. Our work is related to the work of [24], where they analysed images in the Outex dataset [25] using a subset of the attributes we consider; differently from them, we demonstrate that our DTD attributes generalize to new datasets, for example by helping to establish state-of-the-art performance in material recognition.

2.1. Dataset design and collection

This section discusses how DTD was designed and collected, including: selecting the 47 attributes, finding at least 120 representative images for each attribute, and collecting all the attribute labels for each image in the dataset.

Selecting the describable attributes. Psychological experiments suggest that, while there are a few hundred words that people commonly use to describe textures, this vocabulary is redundant and can be reduced to a much smaller number of representative words. Our starting point is the list of 98 words identified by Bhusan, Rao and Lohse [5]. Their seminal work aimed to achieve for texture recognition the same that color words have achieved for describing color spaces [4]. However, their work mainly focuses on the cognitive aspects of texture perception, including perceptual similarity and the identification of directions of perceptual texture variability. Since we are interested in the visual aspects of texture, we ignored words such as "corrugated" that are more related to surface shape properties, and words such as "messy" that do not necessarily correspond to visual features. After this screening phase we analysed the remaining words and merged similar ones such as "coiled", "spiralled" and "corkscrewed" into a single term. This resulted in a set of 47 words, illustrated in Fig. 2.

Bootstrapping the key images. Given the 47 attributes, the next step was collecting a sufficient number (120) of example images representative of each attribute. A very large initial pool of about a hundred-thousand images was downloaded from Google and Flickr by entering the attributes and related terms as search queries. Then Amazon Mechanical Turk (AMT) was used to remove low resolution, poor quality, watermarked images, or images that were not almost entirely filled with a texture. Next, detailed annotation instructions were created for each of the 47 attributes, including a dictionary definition of each concept and examples of correct and incorrect matches. Votes from three AMT annotators were collected for the candidate images of each attribute and a shortlist of about 200 highly-voted images was further manually checked by the authors to eliminate residual errors. The result was a selection of 120 *key representative images* for each attribute.

Sequential join annotations. So far only the key attribute of each image is known while any of the remaining 46 attributes may apply as well. Exhaustively collecting annotations for 46 attributes and 5,640 texture images is fairly expensive. To reduce this cost we propose to exploit the correlation and sparsity of the attribute occurrences (Fig. 3). For each attribute q, twelve key images are annotated exhaustively and used to estimate the probability $p(q'|q)$ that *another* attribute q' could co-exist with q. Then for the remaining key images of attribute q, only annotations for attributes q' with non negligible probability – in practice 4 or 5 – are collected, assuming that the attributes would not apply. This procedure occasionally misses attribute annotations; Fig. 3 evaluates attribute recall by 12-fold cross-validation on the 12 exhaustive annotations for a fixed budget of collecting 10 annotations per image (instead of 47).

A further refinement is to suggest which attributes q' to annotate not just based on q, but also based on the appearance of an image ℓ_i. This was done by using the attribute classifier learned in Sect. 4; after Platt's calibration [30] on an held-out test set, the classifier score $c_{q'}(\ell_i) \in \mathbb{R}$ is

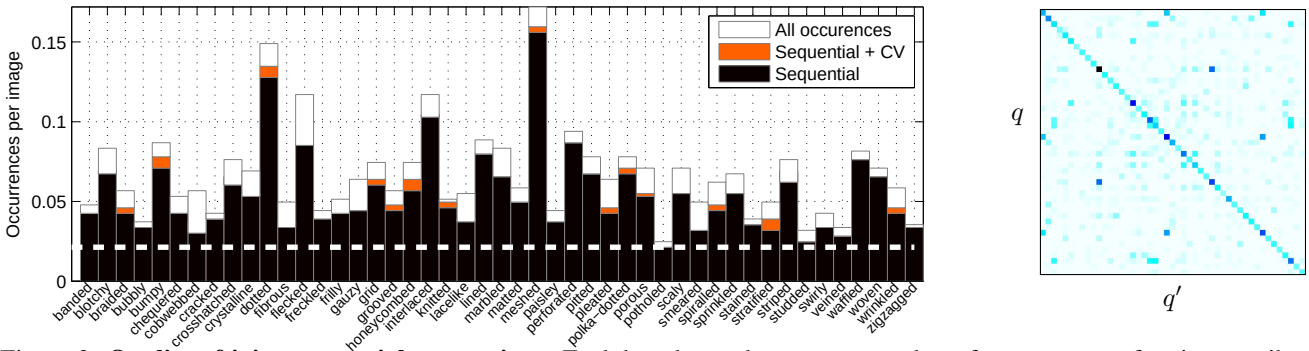

Figure 3: **Quality of joint sequential annotations.** Each bar shows the average number of occurrences of a given attribute in a DTD image. The horizontal dashed line corresponds to a frequency of 1/47, the minimum given the design of DTD (Sect. 2.1). The black portion of each bar is the amount of attributes discovered by the sequential procedure, using only 10 annotations per image (about one fifth of the effort required for exhaustive annotation). The orange portion shows the additional recall obtained by integrating CV in the process. **Right: co-occurrence of attributes.** The matrix shows the joint probability $p(q, q')$ of two attributes occurring together (rows and columns are sorted in the same way as the left image).

transformed in a probability $p(q'|\ell_i) = \sigma(c_{q'}(\ell))$ where $\sigma(z) = 1/(1 + e^{-z})$ is the sigmoid function. By construction, Platt's calibration reflects the prior probability $p(q') \approx p_0 = 1/47$ of q' on the validation set. To reflect the probability $p(q'|q)$ instead, the score is adjusted as

$$p(q'|\ell_i, q) \propto \sigma(c_{q'}(\ell_i)) \times \frac{p(q'|q)}{1 - p(q'|q)} \times \frac{1 - p_0}{p_0}$$

and used to find which attributes to annotated for each image. As shown in Fig. 3, for a fixed annotation budged this method increases attribute recall. Overall, with roughly 10 annotations per images it was possible to recover of all the attributes for at least 75% of the images, and miss one out of four (on average) for another 20% while keeping the annotation cost to a reasonable level.

3. Texture representations

Given the DTD dataset developed in Sect. 2, this section moves on to the problem of designing a system that can automatically recognize the attributes of textures. Given a texture image ℓ the first step is to compute a *representation* $\phi(\ell) \in \mathbb{R}^d$ of the image; the second step is to use a classifier such as a Support Vector Machine (SVM) $\langle \mathbf{w}, \phi(\ell) \rangle$ to score how strongly the image ℓ matches a given perceptual category. We propose two such representations: a gold-standard low-level texture descriptor based on the improved Fisher Vector or DeCAF features (Sect. 3.1) and a mid-level texture descriptor consisting of the describable attributes themselves (Sect. 3.2), discussed in detail in Sect. 4.

3.1. Texture descriptors

This section describes two texture descriptors that we port to texture from the object recognition: the *Improved Fisher Vector* (IFV) [29] and the *Deep Convolutional Activation Feature* (DeCAF) [11]. Differently from popular specialized texture descriptors, both representation are tuned for object recognition. We were therefore somewhat surprised to discover that these off-the-shelf methods surpass by a large margin the state-of-the-art in several texture analysis tasks (Sect. 4.1).

IFV. Given an image ℓ, the *Fisher Vector* (FV) formulation of [28] starts by extracting local SIFT [22] descriptors $\{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$ densely and at multiple scales. It then soft-quantizes the descriptors by using a Gaussian Mixture Model (GMM) with K modes. The Gaussian covariance matrices are assumed to be diagonal, but local descriptors are first decorrelated and optionally dimensionality reduced by PCA. The *improved* version of the descriptor adds signed square-rooting and l^2 normalization. We are not the first to use SIFT or IFV in texture recognition. For example, SIFT was used in [31], and Fisher Vectors were used in [33]. However, neither work tested the standard IFV formulation [29], which we found to give excellent results.

DeCAF. The *DeCAF features* [11] are obtained from an image ℓ as the output of the deep convolutional neural network of [18]. This network, which alternates several layers of linear filtering, rectification, max pooling, normalization, and full linear weighting, is learned to discriminate 1,000 object classes of the ImageNet challenge. It is used as a texture descriptor by removing the softmax and last fully-connected layer of the network, resulting in a $\phi(\mathbf{x}) \in \mathbb{R}^{4096}$ dimensional descriptor vector which is l^2 normalized before use in an SVM classifier. To the best of our knowledge, we are the first to test these features on texture analysis tasks.

3.2. Describable attributes as a representation

The main motivation for recognizing describable attributes is to support human-centric applications, enriching the vocabulary of visual properties that machines can understand. However, once extracted, these attributes may also be used as texture descriptors in their own right. As a simple incarnation of this idea, we propose to collect

Local descr.	Kernel Linear	Hellinger	add-χ^2	exp-χ^2
MR8	15.9±0.8	19.7 ± 0.8	24.1 ± 0.7	30.7 ± 0.7
LM	18.8 ± 0.5	25.8 ± 0.8	31.6 ± 1.1	39.7 ± 1.1
Patch$_{3\times3}$	14.6 ± 0.6	22.3 ± 0.7	26.0 ± 0.8	30.7 ± 0.9
Patch$_{7\times7}$	18.0 ± 0.4	26.8 ± 0.7	31.6 ± 0.8	37.1 ± 1.0
LBP^u	8.2 ± 0.4	9.4 ± 0.4	14.2 ± 0.6	24.8 ± 1.0
LBP-VQ	21.1 ± 0.8	23.1 ± 1.0	28.5 ± 1.0	34.7 ± 1.3
SIFT	**34.7 ± 0.8**	**45.5 ± 0.9**	**49.7 ± 0.8**	**53.8 ± 0.8**

Table 1: Comparison of local descriptors and kernels on the DTD data, averaged over ten splits.

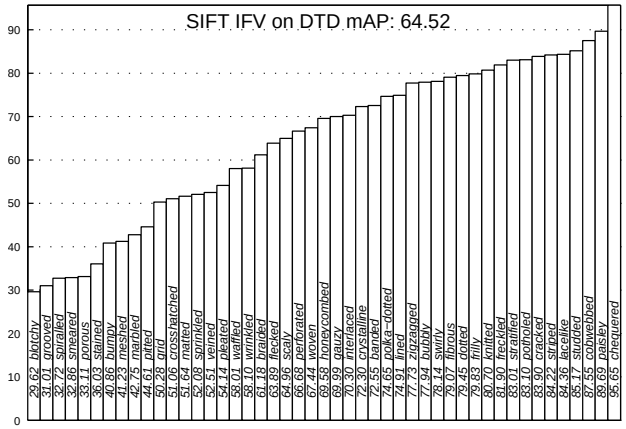

Figure 4: Per-class AP of the 47 describable attribute classifiers on DTD using the IFV$_{\text{SIFT}}$ representation and linear classifiers.

the response of attribute classifiers trained on DTD in a 47-dimensional feature vector $\phi(\ell) = (c_1(\ell), \ldots, c_{47}(\ell))$. Sect. 4 shows that this very compact representation achieves excellent performance in material recognition; in particular, combined with IFV (SIFT and color) and/or DeCAF it sets the new state-of-the-art on KTH-TIPS2-b and FMD. In addition to the contribution to the best results, our proposed attributes generate meaningful descriptions of the materials from KTH-TIPS, *e.g. aluminium foil: wrinkled; bread: porous*.

4. Experiments

4.1. Object descriptors for textures

This section demonstrates the power of IFV and DeCAF (Sect. 3.1) as a texture representation by comparing it to established texture descriptors. Most of these representations can be broken down into two parts: computing local image descriptors $\{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$ and encoding them into a global image statistics $\phi(\ell)$.

In IFV the **local descriptors** $\mathbf{d}_i$ are 128-dimensional *SIFT* features, capturing a spatial histogram of the local gradient orientations; here spatial bins have an extent of 6×6 pixels and descriptors are sampled every two pixels and at scales $2^{i/3}, i = 0, 1, 2, \ldots$. We also evaluate as local descriptors the *Leung and Malik* (LM) [21] (48-D) and *MR8* (8-D) [14, 39] filter banks, the 3×3 and 7×7 raw image patches of [38], and the *local binary patterns* (LBP) of [25].

Encoding maps image descriptors $\{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$ to a statistics $\phi(\ell) \in \mathbb{R}^d$ suitable for classification. Encoding can be as simple as averaging (sum-pooling) descriptors [23], although this is often preceded by a high-dimensional sparse coding step. The most common coding method is to vector quantize the descriptors using an algorithm such as K-means [21], resulting in the so-called *bag-of-visual-words* (BoVW) representation [8]. Variations include soft quantization by a GMM in FV (Sect. 3.1), soft quantization with a kernel in KCB [37], Locality-constrained Linear Coding (LLC) [41], or specialized quantization schemes, such as mapping LBPs to *uniform patterns* [25] (LBP^u; we use the rotation invariant multiple-radii version of [24] for comparison purposes). For LBP, we also experiment with a variant (LBP-VQ) where standard LBP^{u2} is computed in 8×8 pixel neighborhoods, and the resulting local descriptors are further vector quantized using K-means and pooled as this scheme performs significantly better in our experiments.

For each of the selected features, we experimented with several **SVM kernels** $K(\mathbf{x}', \mathbf{x}'')$: linear, Hellinger's, additive-χ^2, and exponential-χ^2 kernels sign-extended as in [40]. The λ parameter of the exponential kernel [40] is selected as one over the mean of the kernel matrix on the training set. The data is normalized so that $K(\mathbf{x}', \mathbf{x}'') = 1$ as this is often found to improve performance. Learning uses a standard non-linear SVM solver and validation to select the parameter C. When multiple features are used, the corresponding kernels are averaged.

Local descriptor comparisons on DTD. This experiment compares local descriptors and kernels on DTD (Tab. 1). All comparison use the bag-of-visual-word pooling/encoding scheme using K-means for vector quantization the descriptors. The DTD data is used as a benchmark averaging the results on the ten train-val-test splits. K was cross-validated, finding an optimal setting of 1024 visual words for SIFT and color patches, 512 for LBP-VQ, 470 for the filter banks. Tab. 1 reports the mean Average Precision (mAP) for 47 SVM attribute classifiers. In these experiments, only the key attribute labels for each image are used; joint annotations are evaluated as DTD-J in Tab. 2, with similar results. As expected, the best kernel is exp-χ^2, followed by additive χ^2 and Hellinger, and then linear. Dense SIFT (53.8% mAP) outperforms the best specialized texture descriptor on the DTD data (39.7% mAP for LM). Fig. 4 shows AP for each attribute: concepts like *chequered* achieve nearly perfect classification, while others such as *blotchy* and *smeared* are far harder.

Encoding comparisons on DTD. Having established the excellent performance of SIFT in texture recognition, this

Dataset			SIFT					DeCAF	IFV + DeCAF	Previous Best
Source	Splits	Metric	IFV	BoVW	VLAD	LLC	KCB			
CUReT	20	acc.	**99.5±0.4**	98.1±0.9	98.8±0.6	97.1±0.4	97.7±0.6	97.9±0.4	**99.8±0.1**	99.4±n/a [36]
UMD	20	acc.	99.2±0.4	98.1±0.8	99.3±0.4	98.4±0.7	98.0±0.9	96.4±0.7	**99.5±0.3**	**99.7±0.3** [34]
UIUC	20	acc.	97.0±0.9	96.1±2.4	96.5±1.8	96.3±0.1	91.4±1.4	94.2±1.1	**99.0±0.5**	**99.4±0.4** [34]
KT	20	acc.	**99.7±0.1**	98.6±1.0	99.2±0.8	98.1±0.8	98.5±0.8	96.9±0.9	**99.8±0.2**	**99.4±0.4** [34]
KT-2a$^{\alpha}$	4	acc.	**82.2±4.6**	**74.8±5.4**	**76.5±5.2**	**75.7±5.6**	**72.3±4.5**	**78.4±2.0**	**84.7±1.5**	73.0±4.7 [33]
KT-2b$^{\beta}$	4	acc.	**69.3±1.0**	58.4±2.2	63.1±2.1	57.6±2.3	58.3±2.2	**70.7±1.6**	**76.2±3.1**	66.3 [36]
FMD	14	acc.	**58.2±1.7**	49.5±1.9	52.6±1.5	50.4±1.6	45.1±1.9	**60.7±2.0**	**65.5±1.3**	57.1 $^{\gamma}$[31]
DTD	10	acc.	**61.2±1.0**	55.5±1.1	59.7±1.1	54.7±1.1	53.2±1.6	54.8±0.9	**66.7±0.9**	–
DTD	10	mAP	**63.5±1.0**	54.9±0.9	61.3±0.8	54.3±1.0	52.5±1.3	55.0±1.1	**69.4±1.2**	–
DTD-J$^{\delta}$	10	mAP	**63.5±0.9**	56.1±0.8	61.1±0.7	54.8±1.0	53.2±0.8	48.9±1.1	**68.9±0.9**	–

Table 2: Comparison of encodings and state-of-the-art texture recognition methods on DTD as well as standard material recognition benchmarks (in boldface results on par or better than the previous state-of-the-art). All experiments use a linear SVM. α : three samples for training, one for evaluation; β : one sample for training, three for evaluation. γ : with ground truth masks ([31] Sect. 6.5); our results do not use them. δ : DTD considers only the key attribute label of each texture occurrence and DTD-J includes the joint attribute annotations too (Sect. 2.1), reporting mAP.

experiment compares three encodings: BoVW, VLAD [16], LLC, KCB, and IFV (first five columns of Tab. 2). VLAD is similar to IFV, but uses K-means for quantization and stores only first-order statistics of the descriptors. Dense SIFT is used as a baseline descriptor and performance is evaluated on ten splits of DTD in Tab. 2. IFV (256 Gaussian modes) and VLAD (512 K-means centers) performs similarly (61-63% mAP) and significantly better than BoVW (54.9% mAP). For BoVW we considered a vocabulary size of 4096 words, while for LLC and KCB we used vocabularies of size 10k. As we will see next, however, IFV significantly outperforms VLAD in other texture datasets. We also experimented with the state-of-the-art descriptor of [34] which we did not find to be competitive with IFV on FMD (41.4% acc.) and DTD (40.3% acc.).

State-of-the-art material classification. This experiments evaluates the encodings on several texture recognition benchmarks: CUReT [9], UMD [42], UIUC [20], KTH-TIPS [15], KTH-TIPS2 (a and b) [7], and material – FMD [32]. Tab. 2 compares with the existing state-of-the-art [33, 34, 36] on each of them. For saturated datasets such as CUReT, UMD, UIUC, KTH-TIPS the performance of most methods is above to 99% mean accuracy and there is little difference between them. IFV performs as well or nearly as well as the state-of-the-art, but DeCAF is not as good. However, in harder datasets the advantage of IFV and DeCAF becomes evident: KTH-TIPS-2a (+5%/5% resp.), KTH-TIPS-2b (+3%/4.3%), and FMD (+1%/+3.6%). Remarkably, DeCAF and IFV appear to capture complementary information as their combination results in significant improvements over each descriptor individually, *substantially outperforming any other descriptor* in KTH (+11.7% on the former state-of-the-art), FMD (+9.9%), and DTD (+8%). In particular, while FMD includes manual segmentations of the textures, these are not used when reporting our results. Furthermore, IFV and DeCAF are conceptually simpler than the multiple specialized features used in [33] for material recognition.

4.2. Describable attributes as a representation

This section evaluates the 47 describable attributes as a texture descriptor for material recognition (Tab. 3). The attribute classifiers are trained on DTD using the various representations such as IFV_{SIFT}, DeCAF, or combinations and linear classifiers as in the previous section. As explained in Sect. 3.2, these are then used to form 47-dimensional descriptors of each texture image in FMD and KTH-TIPS2-b. We call this as $\text{DTD}^{\text{feat}}_{\text{method}}$, denoting the choice of the final classifier (method) and underlying features (feat) used for DTD attribute estimation.

The best results are obtained when IFV_{SIFT} + DeCAF features are used as the underlying representation for predicting DTD attributes. When combined with a linear SVM classifier $\text{DTD}^{\text{IFV + DeCAF}}_{\text{LIN}}$[1], results are promising: on KTH-TIPS2-b, the describable attributes yield 71.2% mean accuracy and 55.9% on FMD outperforming the aLDA model of [31] combining color, SIFT and edge-slice (44.6%). While results are not as good as the IFV_{SIFT} + DeCAF representation directly, the dimensionality of this descriptor is *three orders of magnitude smaller*. For this reason, using an RBF classifier with the DTD features is relatively cheap. Doing so improves the performance by 1–2% ($\text{DTD}^{\text{IFV + DeCAF}}_{\text{RBF}}$). DTD descriptors constructed out of IFV alone are also quite competitive achieving 62.9% and 49.8% on KTH-2b and FMD respectively. They also show a 2–3% improvement when combined with RBF kernels. Combining the DTD RBF kernels obtained from IFV_{SIFT} and IFV_{SIFT} + DeCAF improves performance further.

We also investigated combining multiple IFV features

[1] Note: we drop SIFT in IFV_{SIFT} for brevity

with DTD descriptors: $\text{DTD}^{\text{IFV}}_{\text{RBF}}$ with IFV_{SIFT} and IFV_{RGB}. IFV_{RGB} computes the IFV representation on top of all the 3×3 RGB patches in the image in the spirit of [38]. The performance of IFV_{RGB} is notable given the simplicity of the local descriptors; however, it is not as good as $\text{DTD}^{\text{IFV}}_{\text{RBF}}$ which is also 26 times smaller. The combination of IFV_{SIFT} and IFV_{RGB} is already notably better than the previous state-of-the-art results and the addition of $\text{DTD}^{\text{IFV}}_{\text{RBF}}$ improves by another significant margin. Similarly the $\text{DTD}^{\text{IFV}}_{\text{RBF}}$ descriptors also provide a significant improvement over DeCAF features alone.

Overall, our best result on KTH-TIPS-2b is **77.3%** acc. (vs. the previous best of 66.3) and on FMD of **67.1%** acc. (vs. 57.1) on FMD, an improvement of more than **10%** in both cases over the previous state of the art.

Finally, we compared the semantic attributes of [24] with $\text{DTD}^{\text{IFV}}_{\text{LIN}}$ on the Outex data. Using IFV_{SIFT} as an underlying representation for our attributes, we obtain 49.82% mAP on the retrieval experiment of [24], which is is not as good as their result with LBP^{u} (63.3%). However, LBP^{u} was developed on the Outex data, and it is therefore not surprising that it works so well. To verify this, we retrained our DTD attributes with IFV using LBP^{u} as local descriptor, obtaining a score of 64.5% mAP. This is remarkable considering that their retrieval experiment contains the data used to *train* their own attributes (target set), while our attributes are trained on a completely different data source. Tab. 1 shows that LBP^{u} is not competitive on DTD.

4.3. Search and visualization

Fig. 5 shows an excellent semantic correlation between the ten categories in KTH-TIPS-2b and the attributes in DTD. For example, aluminium foil is found to be *wrinkled*, while bread is found be *bumpy*, *pitted*, *porous* and *flecked*.

As an additional application of our describable texture attributes we compute them on a large dataset of 10,000 wallpapers and bedding sets from houzz.com. The 47 attribute classifiers are learned as explained in Sect. 4.1 using the IFV_{SIFT} representation and them apply them to the 10,000 images to predict the strength of association of each attribute and image. Classifiers scores are re-calibrated on the target data and converted to probabilities by examining the extremal statistics of the scores. Fig. 6 shows some example attribute predictions, selecting for a number of attribute an image that would score perfectly (excluding images used for calibrating the scores), and then including additional top two attribute matches. The top two matches tend to be very good description of each texture or pattern, while the third is a good match in about half of the cases.

5. Summary

We introduced a large dataset of 5,640 images collected "in the wild" jointly labelled with 47 describable texture attributes and used it to study the problem of extracting semantic properties of textures and patterns, addressing real-world human-centric applications. Looking for the best representation to recognize such describable attributes in natural images, we have ported IFV and DeCAF, object recognition representations, to the texture domain. Not only they work best in recognizing describable attributes, but they also outperform specialized texture representations on a number of challenging material recognition benchmarks. We have shown that the describable attributes, while not being designed to do so, are good predictors of materials as well, and that, when combined with IFV, significantly outperform the state-of-the-art on FMD and KTH-TIPS2-b.

Feature	KTH-2b	FMD
$\text{DTD}^{\text{IFV}}_{\text{LIN}}$	62.9±3.8	49.8±1.3
$\text{DTD}^{\text{IFV}}_{\text{RBF}}$	66.0±4.3	52.4±1.3
$\text{DTD}^{\text{IFV + DeCAF}}_{\text{LIN}}$	71.2±0.6	55.9±2.3
$\text{DTD}^{\text{IFV + DeCAF}}_{\text{RBF}}$	72.0±0.5	58.0±1.8
$\text{DTD}^{\text{IFV}}_{\text{RBF}}$ + $\text{DTD}^{\text{IFV + DeCAF}}_{\text{RBF}}$	73.8±1.3	61.1±1.4
DeCAF	70.7±1.6	60.7±2.1
IFV_{RGB}	58.8±2.5	47.0±2.7
IFV_{SIFT} + IFV_{RGB}	67.5±3.3	63.3±1.9
$\text{DTD}^{\text{IFV}}_{\text{RBF}}$ + IFV_{SIFT}	70.2±2.4	60.1±1.6
$\text{DTD}^{\text{IFV}}_{\text{RBF}}$ + IFV_{RGB}	70.9±3.5	61.3±2.0
Combined	74.6±3.0	65.4±2.0
IFV_{SIFT} + $\text{DTD}^{\text{IFV}}_{\text{RBF}}$	70.2±2.4	60.0±1.9
IFV_{SIFT} + $\text{DTD}^{\text{IFV + DeCAF}}_{\text{RBF}}$	75.6±1.8	65.5±1.2
DeCAF + $\text{DTD}^{\text{IFV}}_{\text{RBF}}$	75.4±1.8	64.6±1.6
DeCAF + $\text{DTD}^{\text{IFV + DeCAF}}_{\text{RBF}}$	73.7±1.8	64.1±1.5
IFV_{SIFT} +DeCAF + $\text{DTD}^{\text{IFV}}_{\text{RBF}}$	**77.3±2.3**	66.7±1.7
IFV_{SIFT} +DeCAF + $\text{DTD}^{\text{IFV + DeCAF}}_{\text{RBF}}$	76.4±2.8	66.9±1.6
Combined	**77.1±2.4**	**67.1±1.5**
Prev. best	66.3 [36]	57.1 [31]

Table 3: **DTD for material recognition.** Combined with IFV_{SIFT} and IFV_{RGB}, the $\text{DTD}^{\text{IFV}}_{\text{RBF}}$ features achieve a significant improvement in classification performance on the challenging KTH-TIPS-2b and FMD compared to published state of the art results. See the text for the details on the notation and the methods.

Acknowledgements. This research is based on work done at the 2012 CLSP Summer Workshop, and was partially supported by NSF Grant #1005411, ODNI via the JHU HLTCOE and Google Research. Mircea Cimpoi was supported by the ERC grant VisRec no. 228180 and Iasonas Kokkinos by ANR-10-JCJC-0205.

References

[1] M. Amadasun and R. King. Textural features corresponding to textural properties. *Systems, Man, and Cybernetics*, 19(5), 1989. 1

[2] R. Bajcsy. Computer description of textured surfaces. In *IJCAI*, IJCAI. Morgan Kaufmann Publishers Inc., 1973. 1

[3] T. Berg, A. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. *ECCV*, 2010. 1

[4] B. Berlin and P. Kay. *Basic color terms: Their universality and evolution*. Univ of California Press, 1991. 3

[5] N. Bhushan, A. Rao, and G. Lohse. The texture lexicon: Understand-

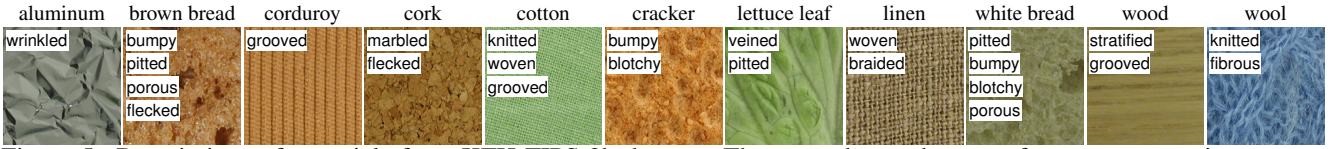

Figure 5: Descriptions of materials from KTH-TIPS-2b dataset. These words are the most frequent top scoring texture attributes (from the list of 47 we proposed), when classifying the images from the KTH-TIPS-2b dataset.

Figure 6: Bedding sets (top) and wallpapers (bottom) with the top 3 attributes predicted by our classifier and normalized classification score in brackets.

ing the categorization of visual texture terms and their relationship to texture images. *Cognitive Science*, 21(2):219–246, 1997. 1, 3

[6] L. Bourdev, S. Maji, and J. Malik. Describing people: A poselet-based approach to attribute classification. In *ICCV*, 2011. 1

[7] B. Caputo, E. Hayman, and P. Mallikarjuna. Class-specific material categorisation. In *ICCV*, 2005. 2, 6

[8] G. Csurka, C. R. Dance, L. Dan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Proc. ECCV Workshop on Stat. Learn. in Comp. Vision*, 2004. 5

[9] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink. Reflectance and texture of real world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, 1999. 2, 6

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 2

[11] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proc. ICML*, 2013. 2, 4

[12] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, pages 1778–1785. IEEE, 2009. 1

[13] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2007. 1

[14] J. M. Geusebroek, A. W. M. Smeulders, and J. van de Weijer. Fast anisotropic gauss filtering. *IEEE Transactions on Image Processing*, 12(8):938–943, 2003. 5

[15] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh. On the significance of real-world conditions for material classification. *ECCV*, 2004. 2, 6

[16] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, 2010. 6

[17] B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, march 1981. 1

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012. 4

[19] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Describable visual attributes for face verification and image search. *PAMI*, 33(10):1962–1977, 2011. 1

[20] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *PAMI*, 28(8):2169–2178, 2005. 2, 6

[21] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001. 2, 5

[22] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999. 4

[23] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *JOSA A*, 7(5), 1990. 5

[24] T. Matthews, M. S. Nixon, and M. Niranjan. Enriching texture analysis with semantic data. In *CVPR*, June 2013. 3, 5, 7

[25] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 24(7):971–987, 2002. 2, 3, 5

[26] G. Oxholm, P. Bariya, and K. Nishino. The scale of geometric texture. In *European Conference on Computer Vision*, pages 58–71. Springer Berlin/Heidelberg, 2012. 2

[27] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012. 1

[28] F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007. 2, 4

[29] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *Proc. ECCV*, 2010. 4

[30] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*. Cambridge, 2000. 3

[31] L. Sharan, C. Liu, R. Rosenholtz, and E. H. Adelson. Recognizing materials using perceptually inspired features. *International Journal of Computer Vision*, 103(3):348–371, 2013. 4, 6, 7

[32] L. Sharan, R. Rosenholtz, and E. H. Adelson. Material perceprion: What can you see in a brief glance? *Journal of Vision*, 9:784(8), 2009. 2, 6

[33] G. Sharma, S. ul Hussain, and F. Jurie. Local higher-order statistics (lhs) for texture categorization and facial analysis. In *Proc. ECCV*. 2012. 4, 6

[34] L. Sifre and S. Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *CVPR*, June 2013. 6

[35] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *Systems, Man and Cybernetics, IEEE Transactions on*, 8(6):460 –473, june 1978. 1

[36] R. Timofte and L. Van Gool. A training-free classification framework for textures, writers, and materials. In *BMVC*, Sept. 2012. 6, 7

[37] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders. Kernel codebooks for scene categorization. In *Proc. ECCV*, 2008. 5

[38] M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? In *CVPR*, volume 2, pages II–691. IEEE, 2003. 2, 5, 7

[39] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *IJCV*, 62(1):61–81, 2005. 2, 5

[40] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *CVPR*, 2010. 5

[41] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. *Proc. CVPR*, 2010. 5

[42] Y. Xu, H. Ji, and C. Fermuller. Viewpoint invariant texture description using fractal analysis. *IJCV*, 83(1):85–100, jun 2009. 2, 6