

Describing Textures in the Wild

Mircea Cimpoi¹, Subhransu Maji², Iasonas Kokkinos³, Sammy Mohamed⁴, and Andrea Vedaldi¹

¹Department of Engineering Science, University of Oxford

²Toyota Technological Institute, Chicago (TTIC)

³Center for Visual Computing, Ecole Centrale Paris

⁴Stony Brook University

Abstract

Patterns and textures are defining 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 analytical dimension in image understanding, we address the challenging problem of describing textures with semantic attributes. We identify a rich 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 the basis to seek for the best texture representation for recognizing describable texture attributes in images. We port from object recognition to texture recognition the Improved Fisher Vector (IFV) and show that, surprisingly, it outperforms specialized texture descriptors not only on our problem, but also in established material recognition datasets. We also show that the describable attributes are excellent texture descriptors, transferring between datasets and tasks; in particular, combined with IFV, they significantly outperform the state-of-the-art by more than 8% 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, 11, 17, 25]. 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



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.

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.

The study of perceptual attributes of textures has a long history starting from pre-attentive aspects and grouping [16], to coarse high-level attributes [1, 2, 33], to some recent work aimed at discovering such attributes by automatically mining descriptions of images from the Internet [3, 12]. 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 Bhushan 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

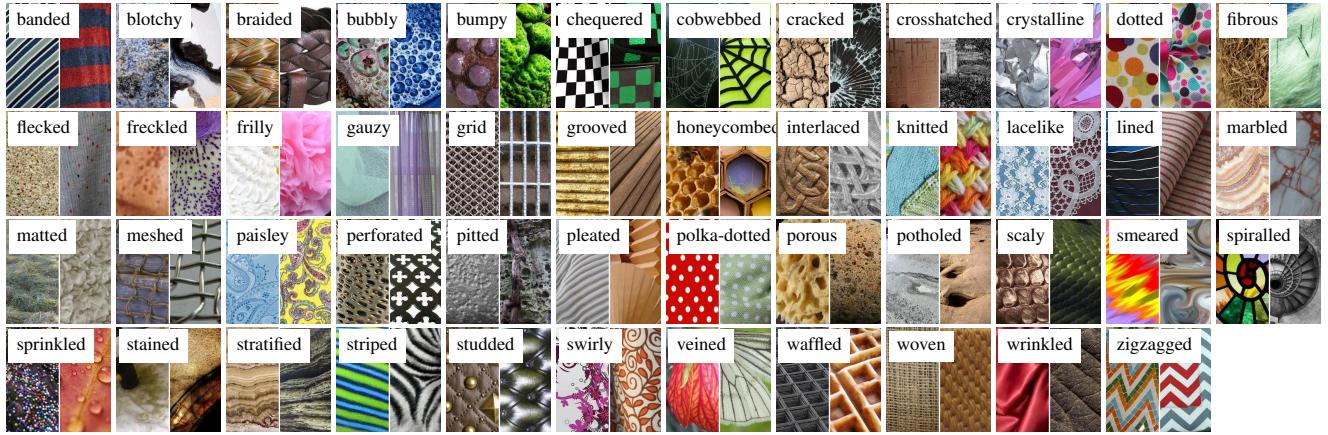


Figure 2: The 47 texture words in the **describable texture dataset** introduced in this paper. Two examples of each attribute are shown to illustrate the significant amount of variability in the data.

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 Bhushan et al., that capture a wide variety of visual properties of textures and to introduce a corresponding *describable texture 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) [30], 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, the appearance of the textures, and the reliability of annotators (Sect. 2.1).

Our **second contribution** is to identify a *gold standard texture representation* that achieves optimal 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* [18], *UMD* [39], *Outex* [23], *Drexel Texture Database* [24], and *KTH-TIPS* [7, 14]. 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 [19, 23, 35, 36]. In contrast, generic object recognition features such as SIFT was 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 [26] computed on SIFT features and certain color features can significantly boost performance. Surprisingly, these descriptors outperform specialized state-of-the-art texture representations not only in recognizing our de-

scribable attributes, but also in a variety of datasets for material recognition, achieving an accuracy of 63.3% on FMD and 67.5% on KTH-TIPS2-b dataset (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 as features can boost recognition rates even more for material classification achieving an accuracy of 53.1% on FMD and 64.6% on KTH when used alone as a 47 dimensional feature, and 65.4% on FMD and 74.6% on KTH when combined with SIFT and simple color descriptors (Sect. 4.2). *These represent more than an absolute gain of 8% in accuracy over previous state of the art. Our 47 dimensional feature contributed with 2.2 to 7% to the gain.* Furthermore, these attribute are easy to describe by design, hence they can serve as intuitive dimensions to explore large collections of texture patterns – for e.g., 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*, *flecked*, *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 judgments, 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 as well (Sect. 3.2 and 4.2).

DTD contains **textures in the wild**, *i.e.* texture images extracted from the web rather than being 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 where 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 judgments, 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 will be made publicly available on the web at [anonymized], along with standardized evaluation, as well as code reproducing the results in Sect. 4.

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 [22], where they analyzed images in the Outex dataset [23] using a subset of the attributes we consider. Their attributes were demonstrated to perform better than several low-level descriptors, but these were trained and evaluated on the *same* dataset. Hence it is not clear if their learned attributes generalize well to other settings. In contrast, we show that: (i) our texture attributes trained on DTD outperform their semantic attributes on Outex and (ii) they can significantly boost performance on a number of other material and texture benchmarks (Sect. 4.2).

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, collecting a full set of multiple attribute labels for each image in the dataset, and addressing annotation noise.

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 Bhushan, 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 analyzed the remaining words and merged similar ones such as “coiled”, “spiraled” 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-thousands 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 texture 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 was found to be too expensive. To reduce this cost we propose to exploiting the correlation and sparsity of the attribute occurrences (Fig. 3). For each attribute q , twelve key images

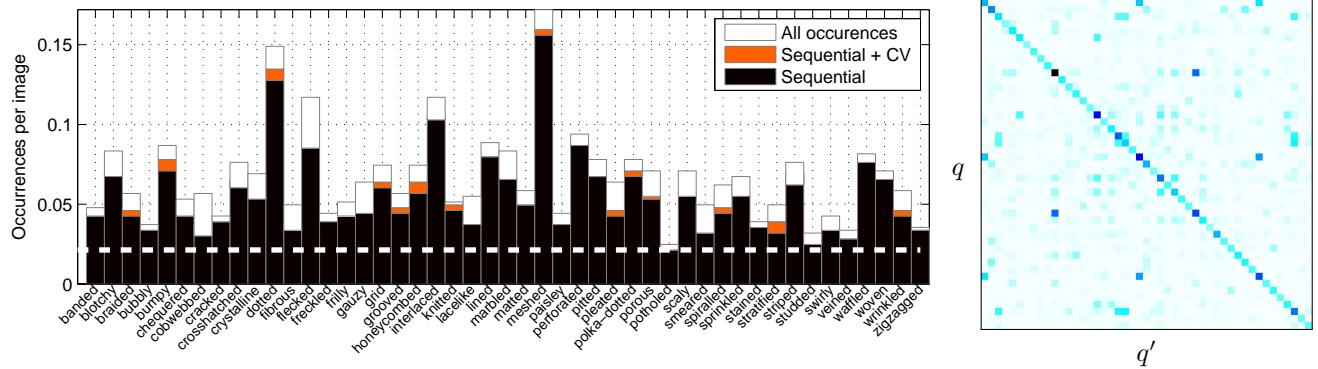


Figure 3: Quality of joint sequential annotations. Each bar shows the average number of occurrences of a given attribute in a DTD image. The horizontal dashed line corresponds to a frequency of 1/47, the minimum given the design of DTD (Sect. 2.1). The black portion of each bar is the amount of attributes 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).

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 individual appearance of an image ℓ_i . This was done by using the attribute classifier learned in Sect. 4; after Platt’s calibration [28] on an held-out test set, the classifier score $c_{q'}(\ell_i) \in \mathbb{R}$ is 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 annotate for each image. As shown in Fig. 3, for a fixed annotation budget 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.

Handling noisy annotations. So far it was assumed that annotators are perfect: deterministic and noise-free. This

is not the case, in part due to the intrinsic subjectivity of describable texture attributes, and in part due to distracted, adversarial, or unqualified annotators. As commonly done, we address this problem by collecting the same annotation multiple times (five) using different annotators, and forming a consensus.

Beyond simple voting, we found that the method of [38] can effectively remove or down-weigh bad annotators improving agreement. This method models each annotator α_j as a classifier with a given bias and error rate. Then, given a collection $\hat{a}_{qij} \in \{0, 1\}$ of binary annotations for attribute q , image i , and annotator j , it tries to estimate simultaneously the ground truth labels a_{qi} and the quality α_j of the individual annotators. The method is appealing as several quantities are easily interpretable. For example, the prior $p(\alpha_j)$ on annotators encodes how frequently we expect to find good and bad annotators (*e.g.* we found that 0.5% of them labeled images randomly). A major difference compared to the scenario considered in [38] is that, in our case, the key attribute of each image is already known. By incorporating this as additional prior, the method can use the key attributes to implicitly benchmark and calibrate annotators. The final set of annotations $\{a_{qi}\}$ is obtained by thresholding the (approximated) posterior marginal $p(a_{qi}|\{\hat{a}_{qij}\})$ to 60%, similar to choosing three out of five votes in the basic voting scheme, computed using variational inference. In general, we found most probabilities to be very close to 100% or 0%, suggesting that there is little residual noise in the process. We also inspected the top 30 images of each attribute based on simple voting and this posterior marginals and found the ranking to be significantly improved.

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 ℓ matches a given perceptual category. We propose two such representations: a gold-standard low-level texture descriptor based on the improved Fisher Vector (Sect. 3.1) and a mid-level texture descriptor consisting of the describable attributes themselves (Sect. 3.2). The details of the classifiers are discussed in Sect. 4.

3.1. Improved Fisher vectors

This section introduces our gold-standard low-level texture representation, the *Improved Fisher Vector* (IFV) of and relates it to existing texture descriptors. We port IFV from the object recognition literature [27] and we show that it substantially outperforms specialized texture representations (Sect. 4).

Given an image ℓ , the *Fisher Vector* (FV) formulation of [26] starts by extracting local SIFT [20] descriptors $\{\mathbf{d}_1, \dots, \mathbf{d}_n\}$ densely and at multiple scales. It then soft-quantizes the descriptors by using a Gaussian Mixture Model (GMM) with K modes, prior probabilities π_k , mode means μ_k and mode covariances Σ_k . Covariance matrices are assumed to be diagonal, but local descriptors are first decorrelated and optionally dimensionality reduced by PCA. Then first and second order statistics are computed as

$$u_{jk} = \frac{1}{n\sqrt{\pi_k}} \sum_{i=1}^n q_{ik} \frac{d_{ji} - \mu_{jk}}{\sigma_{jk}},$$

$$v_{jk} = \frac{1}{n\sqrt{2\pi_k}} \sum_{i=1}^n q_{ik} \left[\left(\frac{d_{ji} - \mu_{jk}}{\sigma_{jk}} \right)^2 - 1 \right],$$

where j spans descriptor dimensions and q_{ik} is the posterior probability of mode k given descriptor \mathbf{d}_i , i.e. $q_{ik} \propto \exp\left[-\frac{1}{2}(\mathbf{d}_i - \mu_k)^T \Sigma_k^{-1} (\mathbf{d}_i - \mu_k)\right]$. These statistics are then stacked into a vector $(\mathbf{u}_1, \mathbf{v}_1, \dots, \mathbf{u}_K, \mathbf{v}_K)$. In order to obtain the *improved* version of the representation, the signed square root $\sqrt{|z|} \text{ sign } z$ is applied to its components and the vector is l^2 normalized.

At least two key ideas in IFV were pioneered in texture analysis: the idea of sum-pooling local descriptors was introduced by [21], and the idea of quantizing local descriptors to construct histogram of features was pioneered by [19] with their computational model of textons. However, three key aspects of the IFV representation were developed in the context of object recognition. The first one is the use of the SIFT descriptors, originally developed for object matching [20], that are more distinctive

that local descriptors popular in texture analysis such as filter banks [13, 19, 36], local intensity patterns [23], and patches [35]. The second one is replacing histogramming with the more expressive FV pooling method [26]. And the third one is the use of the square-root kernel map [27] in the improved version of the Fisher Vector.

We are not the first to use SIFT or IFV in texture recognition. For example, SIFT was used in [29], and Fisher Vectors were used in [31]. However, neither work tested the standard IFV formulation [27], which is well tuned for object recognition, developing instead variations specialized for texture analysis. We were therefore somewhat surprised to discover that the off-the-shelf method surpasses these approaches (Sect. 4.1).

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 the response of attribute classifiers trained on DTD in a 47-dimensional feature vector $\phi(\ell) = (c_1(\ell), \dots, c_{47}(\ell))$. Sect. 4 shows that this very compact representation achieves excellent performance in material recognition; in particular, combined with IFV (SIFT and color) 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-TIPS2-b (aluminium foil: wrinkled; bread: porous).

4. Experiments

4.1. Improved Fisher Vectors for textures

This section demonstrates the power of IFV 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 $\{\mathbf{d}_1, \dots, \mathbf{d}_n\}$ 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 $2^{i/3}, i = 0, 1, 2, \dots$. We also evaluate as local descriptors the *Leung and Malik* (LM) [19] (48-D) and *MR8* (8-D) [13, 36] filter banks, the 3×3 and 7×7 raw image patches of [35], and the *local binary patterns* (LBP) of [23].

Encoding maps image descriptors $\{\mathbf{d}_1, \dots, \mathbf{d}_n\}$ to a statistics $\phi(\ell) \in \mathbb{R}^d$ suitable for classification. Encoding can be as simple as averaging (sum-pooling) descriptors [21], although this is often preceded by a high-

Local d.	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 _{3×3}	14.6 ± 0.6	22.3 ± 0.7	26.0 ± 0.8	30.7 ± 0.9
Patch _{7×7}	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.

dimensional sparse coding step. The most common coding method is to vector quantize the descriptors using an algorithm such as K -means [19], resulting in the so-called *bag-of-visual-words* (BoVW) representation [8]. Variations include soft quantization by a GMM in FV (Sect. 3.1) or specialized quantization schemes, such as mapping LBPs to *uniform patterns* [23] (LBP^u; we use the rotation invariant multiple-radii version of [22] 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**: linear $K(\mathbf{x}', \mathbf{x}'') = \langle \mathbf{x}', \mathbf{x}'' \rangle$, Hellinger's $\sum_{i=1}^d \sqrt{\mathbf{x}'_i \mathbf{x}''_i}$, additive- χ^2 $\sum_{i=1}^d \mathbf{x}'_i \mathbf{x}''_i / (\mathbf{x}'_i + \mathbf{x}''_i)$, and exponential- χ^2 $\exp \left[-\lambda \sum_{i=1}^d (\mathbf{x}'_i - \mathbf{x}''_i)^2 / (\mathbf{x}'_i + \mathbf{x}''_i) \right]$ kernels sign-extended as in [37]. In the latter case, λ 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 in order to select the parameter C in the range $\{0.1, 1, 10, 100\}$ (the choice of C was found to have little impact on the result).

Local descriptor comparisons on DTD. This experiments compares local descriptors and kernels on DTD. 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. As expected, the best kernel is exp- χ^2 , followed by additive χ^2 and Hellinger, and then linear. Dense SIFT (53.82% mAP) outperforms the best specialized texture descriptor on the DTD data (39.67% mAP for LM). Fig. 4 shows AP for each attribute: concepts like *chequered* achieve nearly perfect classification, while oth-

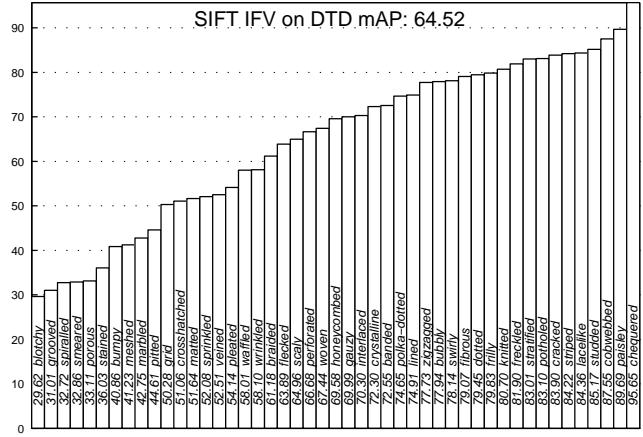


Figure 4: Per-class AP of the 47 describable attribute classifiers on DTD using the IFV_{SIFT} representation and linear classifiers.

ers such as *blotchy* and *smeared* are far harder.

Encoding comparisons on DTD. This experiment compares three encodings: BoVW, VLAD [15] and IFV. 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 (about 60% mAP) and significantly better than BoVW (53.82% mAP). 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 [32] which we did not find to be competitive with IFV on FMD and DTD (Tab. 2); unfortunately could not obtain an implementation of [29] to try on our data – however IFV_{SIFT} outperforms it on material recognition.

State-of-the-art material classification. This experiments evaluates the encodings on several material recognition datasets: CUReT [9], UMD [39], UIUC [18], KTH-TIPS [14], KTH-TIPS2(a and b) [7], and FMD [30]. Tab. 2 compares with the existing state-of-the-art [31, 32, 34] 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. In harder datasets the advantage of IFV is evident: KTH-TIPS-2a (+5%), KTH-TIPS-2b (+3%), and FMD (+1%). In particular, while FMD includes manual segmentations of the textures, these are not used here here. Furthermore, IFV is conceptually simpler than the multiple specialized features used in [31] for material recognition.

Dataset	SIFT			Published		Feature	KTH-TIPS-2b	FMD
	IFV	BoWV	VLAD	Best	[32]			
CUReT	99.6 ± 0.3	98.1 ± 0.9	98.8 ± 0.6	→	99.4	DTD _{LIN}	61.1 ± 2.8	48.9 ± 1.9
UMD	99.2 ± 0.4	98.1 ± 0.8	99.3 ± 0.4	→	99.7 ± 0.3	DTD _{RBF}	64.6 ± 1.5	53.1 ± 2.0
UIUC	97.0 ± 0.9	96.1 ± 2.4	96.5 ± 1.0	→	99.4 ± 0.4	IFV _{SIFT}	69.3 ± 1.0	58.2 ± 1.7
KTH-TIPS	99.7 ± 0.1	98.6 ± 1.0	99.2 ± 0.8	→	99.4 ± 0.4	IFV _{RGB}	58.8 ± 2.5	47.0 ± 2.7
KTH-TIPS-2a ^α	82.5 ± 5.2	74.8 ± 5.4	76.5 ± 5.2	73.0 ± 4.7 [31]	—	IFV _{SIFT} + IFV _{RGB}	67.5 ± 3.3	63.3 ± 1.9
KTH-TIPS-2b ^β	69.3 ± 1.0	58.4 ± 2.2	63.1 ± 2.1	66.3 [34]	—	DTD _{RBF} + IFV _{SIFT}	68.4 ± 1.4	60.1 ± 1.6
FMD	58.2 ± 1.7	49.5 ± 1.9	52.6 ± 1.5	57.1 / 55.6 [29] ^γ	41.4 ± 1.3	DTD _{RBF} + IFV _{RGB}	70.9 ± 3.5	61.3 ± 2.0
DTD	61.5 ± 1.4	55.6 ± 1.3	59.8 ± 1.0	—	40.2 ± 0.5	All three	74.6 ± 3.0	65.4 ± 2.0
						Prev. state of the art	66.3 [34]	57.1 [29]

Table 2: **Left:** Comparison of encodings and state-of-the-art texture recognition methods on DTD as well as standard material recognition benchmarks. α : three samples for training, one for evaluation; β : one sample for training, three for evaluation. γ : with/without ground truth masks ([29] Sect. 6.5); our results do not use them. **Right:** Combined with IFV_{SIFT} and IFV_{RGB}, the DTD_{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.

4.2. Describable attributes as a representation

This section evaluates using the 47 describable attributes as a texture descriptor applying it to the task of material recognition. The attribute classifiers are trained on DTD using the IFV+SIFT representation and linear classifiers as in the previous section (DTD_{LIN}). 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.

When combined with a linear SVM classifier, results are promising (Tab. 2): on KTH-TIPS2-b, the describable attributes yield 61.1% mean accuracy and 49.0% on FMD outperforming the aLDA model of [29] combining color, SIFT and edge-slice (44.6%). While results are not as good as the IFV_{SIFT} representation, the dimensionality of this descriptor is *three orders of magnitude smaller* than IFV. For this reason, using an RBF classifier with the DTD features is relatively cheap. Doing so improves the performance by 3.5–4% (DTD_{RBF}).

We also investigated combining multiple features: DTD_{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 [35]. The performance of IFV_{RGB} is notable given the simplicity of the local descriptors; however, it is not as good as DTD_{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 DTD_{RBF} improves by another significant margin. Overall, our best result on KTH-TIPS-2b is **74.6%** (vs. the previous best of 66.3) and on FMD of **65.4%** (vs. 57.1) on FMD, with an improvement of more than **8%** accuracy in both cases.

Finally, we compared the semantic attributes of [22] with DTD_{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 [22], which 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 64.52% 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 that there is 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 as: *bumpy*, *pitted*, *porous* and *flecked*.

In what follows, we experimented with describing images from a challenging material dataset, FMD and encouraged by the good results, we applied the same technique to images from the wild, from some online catalog.

4.3.1 Subcategorizing FMD materials using describable texture attributes

The results shown in Fig. 6 extends the results in Table 2 and Sect. 4.2 and illustrate the classification performance of the 47-dimensional DTD descriptors on the FMD materials – note the excellent performance obtained for foliage, wood, and water, which are above 70%.

Our experiments illustrate how the DTD attributes can be used to find “semantic structures” in a dataset such as FMD, for example by distinguishing between “knitted vs pleated fabric”, “gauzy vs crystalline glass”, “veined vs frilly foliage” etc. To do so, FMD images for each material were clustered based on the 47 attribute vectors using K -means into 3–5 clusters each. Examples of the most meaningful clusters are shown in Fig. 8 along with the dominant attributes in each.

Notable fine-grained material distinctions include *knitted* vs *pleated* fabrics and *frilly* vs *pleated & veined* foliage

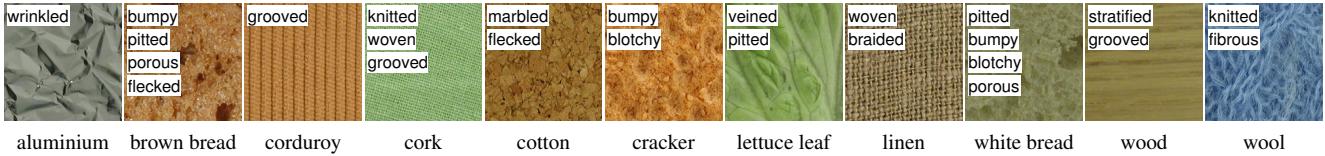


Figure 5: Descriptions of materials from KTH-TIPS-2b dataset. These words are the most frequent top scoring texture attributes (from the list of 47 we proposed), when classifying the images from the KTH-TIPS-2b dataset.

which contain linear structures. In the latter case, veins often have a radial pattern which is captured by the dominant *spiralled* attribute. The method distinguishes *bumpy* stones such as pebbles from *porous* or *pitted* stones for zoomed / detailed views of stone blocks. Water is divided into *swirly* & *spiralled* images, which show the orientation of the waves, and *bubbly*, *sprinkled* images, which show splashing drops. Glass is more challenging but some images are correctly identified as *crystalline*. Fig. 7 shows other challenging examples illustrating the variety of materials and patterns that can be described by the DTD attributes. Metal is one of the hardest class to identify (Fig. 6), but attributes such as “interlaced” and “braided” are still correctly recognized in the third (jewelry) and last (metal wires) image.

4.3.2 Examples in the wild

As an additional application of our describable texture attributes we compute them on a large dataset of 10,000 wall-papers and bedding sets (about 5,000 for each of the two categories) 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 recalibrated on a subset of the target data and converted to probabilities using Platt’s method [28], for each individual attribute. Fig. 11 and

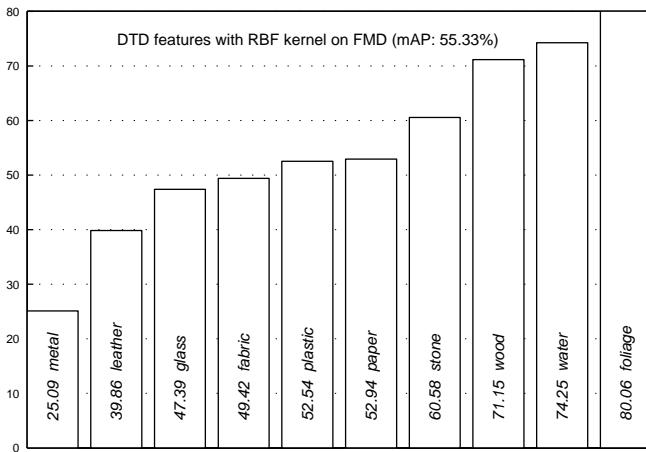


Figure 6: Per class AP results on FMD dataset using DTD classification scores as features.

Fig. 12 shows some example attribute predictions (excluding images used for calibrating the scores), for the best scoring 3-4 images for each category – by top attribute. We show for each images the top three attributes – the top two being very accurate, while the third is correct in about half of the cases. Please note that each score is calibrated on a per attribute basis, to the scores do not add up to 1.

5. Summary

We introduced a large dataset of 5,640 images collected “in the wild” jointly labeled 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, an object recognition representation, to the texture domain. Not only IFV works best in recognizing describable attributes, but it also outperforms specialized texture representation 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 the FMD and KTH-TIPS recognition tasks.

References

- [1] M. Amadasun and R. King. Textural features corresponding to textural properties. *Systems, Man, and Cybernetics*, 19(5), 1989.
- [2] R. Bajcsy. Computer description of textured surfaces. In *IJCAI*, IJCAI. Morgan Kaufmann Publishers Inc., 1973.
- [3] T. Berg, A. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. *ECCV*, 2010.
- [4] B. Berlin and P. Kay. *Basic color terms: Their universality and evolution*. Univ of California Press, 1991.
- [5] N. Bhushan, A. Rao, and G. Lohse. The texture lexicon: Understanding the categorization of visual texture terms and their relationship to texture images. *Cognitive Science*, 21(2):219–246, 1997.
- [6] L. Bourdev, S. Maji, and J. Malik. Describing people: A poselet-based approach to attribute classification. In *ICCV*, 2011.
- [7] B. Caputo, E. Hayman, and P. Mallikarjuna. Class-specific material categorisation. In *ICCV*, 2005.
- [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.
- [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.
- [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.



Figure 7: Challenging or difficult images which were correctly characterized by our DTD classifier.

- [11] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, pages 1778–1785. IEEE, 2009.
- [12] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2007.
- [13] 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.
- [14] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh. On the significance of real-world conditions for material classification. *ECCV*, 2004.
- [15] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, 2010.
- [16] B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, march 1981.
- [17] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Describable visual attributes for face verification and image search. *PAMI*, 33(10):1962–1977, 2011.
- [18] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *PAMI*, 28(8):2169–2178, 2005.
- [19] 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.
- [20] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999.
- [21] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *JOSA A*, 7(5), 1990.
- [22] T. Matthews, M. S. Nixon, and M. Niranjan. Enriching texture analysis with semantic data. In *CVPR*, June 2013.
- [23] 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.
- [24] 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.
- [25] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012.
- [26] F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007.
- [27] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *Proc. ECCV*, 2010.
- [28] 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.
- [29] 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.
- [30] L. Sharan, R. Rosenholtz, and E. H. Adelson. Material perception: What can you see in a brief glance? *Journal of Vision*, 9:784(8), 2009.
- [31] G. Sharma, S. ul Hussain, and F. Jurie. Local higher-order statistics (lhs) for texture categorization and facial analysis. In *Proc. ECCV*, 2012.
- [32] L. Sifre and S. Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *CVPR*, June 2013.
- [33] 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.
- [34] R. Timofte and L. Van Gool. A training-free classification framework for textures, writers, and materials. In *BMVC*, Sept. 2012.
- [35] M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? In *CVPR*, volume 2, pages II–691. IEEE, 2003.
- [36] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *IJCV*, 62(1):61–81, 2005.
- [37] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *CVPR*, 2010.
- [38] P. Welinder and P. Perona. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *CVPR*, 2010.
- [39] Y. Xu, H. Ji, and C. Fermüller. Viewpoint invariant texture description using fractal analysis. *IJCV*, 83(1):85–100, jun 2009.



fabric (knitted)



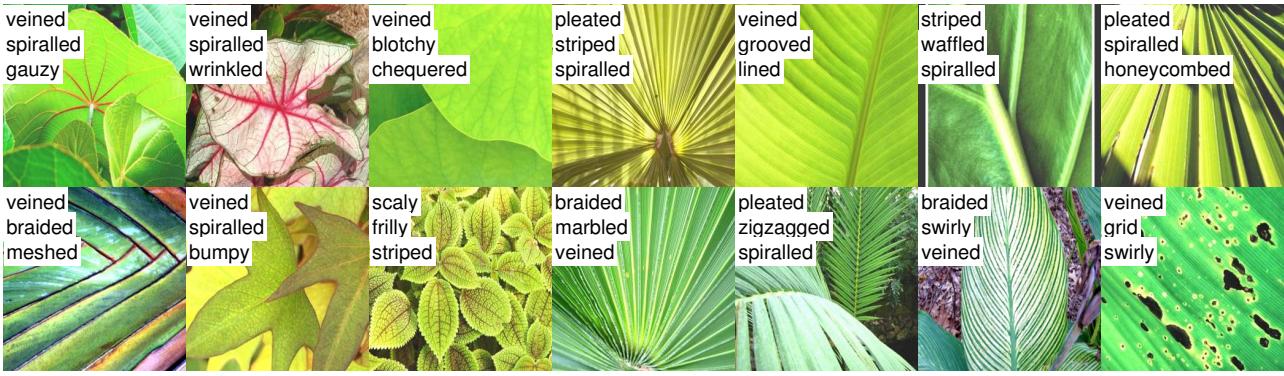
fabric (pleated)



glass (bubbly, gauzy)

glass (crystalline, bubbly)

Figure 8: Example meaningful clusters of FMD categories, obtained using K-means on DTD classification scores. Showing results for fabric and glass – overlayed, we list the most frequently identified attributes. On each image, we show the top 3 scoring texture words.



foliage (pleated, spiralled, veined)



foliage (frilly, sprinkled)



paper (wrinkled, pleated)

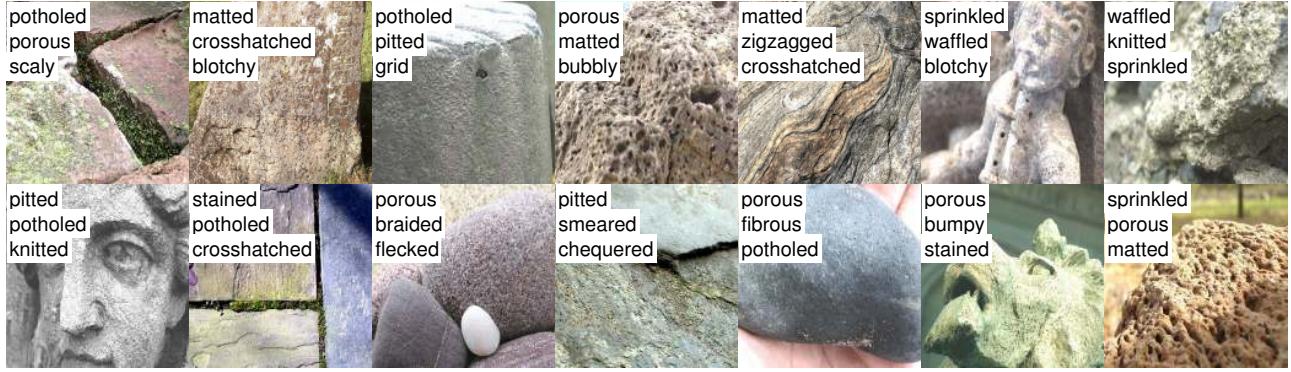


wood (cracked, veined, interlaced)

Figure 9: *Continued from Fig. 8.* Displaying results on foliage, paper and wood.



stone (bumpy)



stone (porous, pitted, flecked)



water (bubbly, smeared)



water (swirly, spiralled)

Figure 10: *Continued from Fig. 9* Subcategories for stone and water images.



Figure 11: Example wallpaper images from an online catalog (houzz.com).

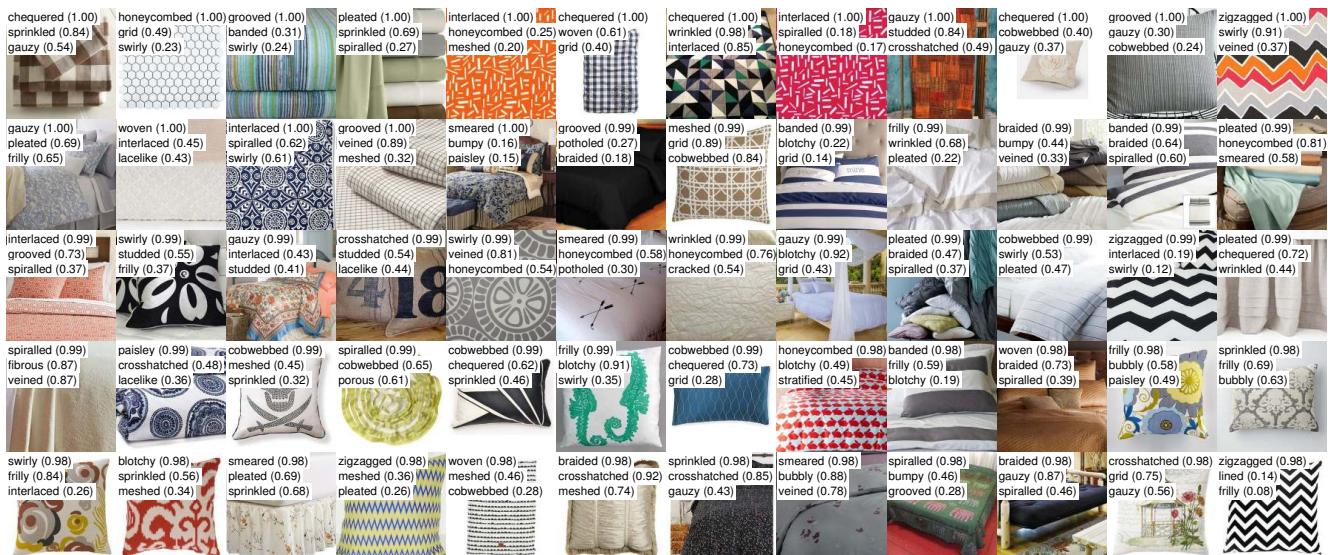


Figure 12: Example bedding sets from an online catalog (houzz.com).