Deep Filter Banks for Texture Recognition and Segmentation

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

Subhransu Maji
University of Massachusetts, Amherst
smaji@cs.umass.edu

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

Abstract

Research in texture recognition often concentrates on the problem of material recognition in uncluttered conditions, an assumption rarely met by applications. In this work we conduct a first study of material and describable texture attributes recognition in clutter, using a new dataset derived from the OpenSurface texture repository. Motivated by the challenge posed by this problem, we propose a new texture descriptor, FV-CNN, obtained by Fisher Vector pooling of a Convolutional Neural Network (CNN) filter bank. FV-CNN substantially improves the state-of-the-art in texture, material and scene recognition. Our approach achieves 79.8% accuracy on Flickr material dataset and 81% accuracy on MIT indoor scenes, providing absolute gains of more than 10% over existing approaches. FV-CNN easily transfers across domains without requiring feature adaptation as for methods that build on the fully-connected layers of CNNs. Furthermore, FV-CNN can seamlessly incorporate multi-scale information and describe regions of arbitrary shapes and sizes. Our approach is particularly suited at localizing "stuff" categories and obtains state-of-the-art results on MSRC segmentation dataset, as well as promising results on recognizing materials and surface attributes in clutter on the OpenSurfaces dataset.

1. Introduction

Texture is ubiquitous and provides useful cues of material properties of objects and their identity, especially when shape is not useful. Hence, a significant amount of effort in the computer vision community has gone into recognizing texture via tasks such as texture perception [1, 2, 12, 13] and description [7, 11], material recognition [26, 36, 37], segmentation [20, 29], and even synthesis [9, 43].

Perhaps the most studied task in texture understanding is the one of material recognition, as captured in benchmarks such as CuRET [8], KTH-TIPS [5], and, more recently, FMD [38]. However, while at least the FMD dataset contains images collected from the Internet, vividly dubbed "images in the wild", all these datasets make the simplifying assumption that textures fill images. Thus, they are not necessarily representative of the significantly harder problem of recognising materials in natural images, where textures appear in clutter. Building on a recent dataset collected by the computer graphics community, the **first contribution** of this paper is a *large-scale analysis of material and perceptual texture attribute recognition and segmentation in clutter* (Fig. 1 and Sect. 2).

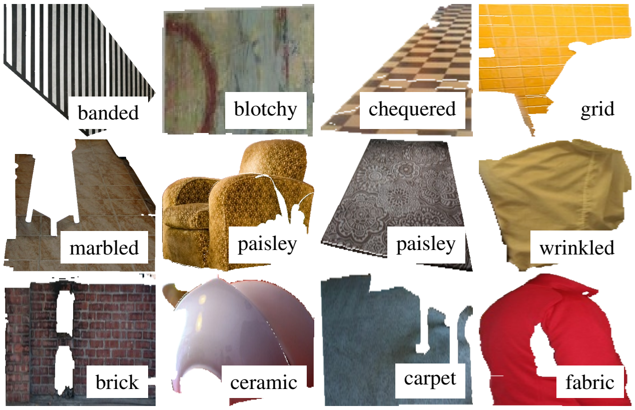

Figure 1. **Texture recognition in clutter**. Example of top retrieved texture segments by attributes (top two rows) and materials (bottom) in the OpenSurfaces dataset.

Motivated by the challenge posed by recognising texture in clutter, we develop a new texture descriptor. In the simplest terms a texture is characterized by the arrangement of local patterns, as captured in early works [26, 41] by the distribution of local "filter bank" responses. These filter banks were designed to capture edges, spots and bars at different scales and orientations. Typical combinations of the filter responses, identified by vector quantisation, were used as the computational basis of the "textons" proposed by Julesz [22]. Texton distributions were the early versions of "bag-of-words" representations, a dominant approach in recognition in the early 2000s, since then improved by new pooling schemes such as soft-assignment [27, 42, 48] and Fisher Vectors (FVs) [32]. Until recently, FV with SIFT features [28] as a local representation was the state-of-the-art method for recognition, not only for textures, but for objects and scenes too.

Later, however, Convolutional Neural Networks (CNNs) have emerged as the new state-of-the-art for recognition, exemplified by remarkable results in image classification [23], detection [14] and segmentation [16] on a number of widely used benchmarks. Key to their success is the ability to leverage large *labelled* datasets to learn increasingly complex transformations of the input to capture invariances. Importantly, CNNs pre-trained on such large datasets have been shown [6, 14, 30] to contain general-purpose feature extractors, transferrable to many other domains.

Domain transfer in CNNs is usually achieved by using as features the output of a deep, fully-connected layer of the network. From the perspective of textures, however, this choice has three drawbacks. The first one (I) is that, while the convolutional layers are akin to non-linear filter banks, the fully connected layers capture their spatial layout. While this may be useful for representing the shape of an object, it may not be as useful for representing texture. A second drawback (II) is that the input to the CNN has to be of fixed size to be compatible with the fully connected layers, which requires an expensive resizing of the input image, particularly when features are computed for many different regions [14, 15]. A third and more subtle drawback (III) is that deeper layers may be more domain-specific and therefore potentially less transferrable than shallower layers.

The **second contribution** of this paper is FV-CNN (Sect. 3), a *pooling method that overcomes these drawbacks*. The idea is to regard the convolutional layers of a CNN as a filter bank and build an orderless representation using FV as a pooling mechanism, as is commonly done in the bag-of-words approaches. Although the suggested change is simple, the approach is remarkably flexible and effective. First, pooling is orderless and multi-scale, hence suitable for textures. Second, any image size can be processed by convolutional layers, avoiding costly resizing operations. Third, convolutional filters, pooled by FV-CNN, are shown to transfer more easily than fully-connected ones even without fine-tuning. While other authors [15, 18] have recently proposed alternative pooling strategies for CNNs, we show that our method is more natural, faster and often significantly more accurate.

The **third contribution** of the paper is a *thorough evaluation of these descriptors* on a variety of benchmarks, from textures to objects (Sect. 4). In textures, we evaluate material and describable attributes recognition and segmentation on new datasets derived from OpenSurfaces (Sect. 2). When used with linear SVMs, FV-CNN improves the state of the art on texture recognition by a significant margin. Like textures, scenes are also weakly structured and a bag-of-words representation is effective. FV-CNN obtains 81.1% accuracy on the MIT indoor scenes dataset [34], significantly outperforming the current state-of-the-art of 70.8% [47]. What is remarkable is that, where [47] finds that CNNs trained on scene recognition data perform better than CNNs trained on an object domain (ImageNet), when used in FV-CNN not only is there an overall performance improvement, but the domain-specific advantage is entirely removed (Tab. 3). This indicates that FV-CNN are in fact better at domain transfer. Our method also matches the previous best in PASCAL VOC 2007 classification dataset providing measurable boost over CNNs and closely approaches competitor methods on CUB 2010-2011 datasets when ground-truth object bounding boxes are given.

FV-CNN can be used for describing regions by simply pooling across pixels within the region. Combined with a low-level segmentation algorithm this suffices to localize textures within images. This approach is similar to a recently proposed method called "R-CNN" for localizing objects [14]. However, in contrast to it we do not need repeated evaluations of the CNN since the convolutional features can be computed just once for the entire image and pooled differently. This makes FV-CNN not only faster, but also as experiments suggest, much more accurate at texture localization. We achieve state of the art results on the MSRC segmentation dataset using a simple scheme of classifying "superpixels" obtaining an accuracy of 87.0% (previous best 86.5%). The corresponding R-CNN obtains 57.7% accuracy. Segmentation results are promising in the *OpenSurfaces* dataset [4] as well.

Finally, we analyze the utility of different network layers and architectures as filter banks, concluding that: SIFT is competitive only with the first few layers of a CNN (Fig. 4) and that significant improvement to the underlying CNN architecture, such as the ones achieved by the *very deep* models of Simonyan and Zisserman [39], directly translate into much better filter banks for texture recognition.

2. Texture recognition in clutter

A contribution of this work is the analysis of materials and texture attributes in realistic imaging conditions. Earlier datasets such as KTH-TIPS were collected in controlled conditions, which makes their applicability to natural images unclear. More recent datasets such as FMD and DTD remove this limitation by building on images downloaded from the Internet, dubbed images "in the wild". However, in these datasets texture always fill the field of view of the camera. In this paper we remove this limitation by experimenting for the first time with a large dataset of textures collected in the wild and in cluttered conditions.

In particular, we build on the *Open Surfaces* (OS) dataset that was recently introduced by Bell *et al.* [4] in computer graphics. OS comprises 25,357 images, each containing a number of high-quality texture/material segments. Many of these segments are annotated with additional attributes such as the material name, the viewpoint, the BRDF, and the object class. Not all segments have a complete set of annotations; the experiments in this paper focus on the 58,928 that contain material names. Since material classes are highly unbalanced, only the materials that contain at least 400 examples are considered. This result in 53,915 annotated material segments in 10,422 images spanning 22 different classes.[1] Images are split evenly into training, validation, and test subsets with 3,474 images each. Segment sizes are highly variable, with half of them being relatively small, with an area smaller than 64×64 pixels. While the lack of exhaustive annotations makes it impossible to define a complete background class, several less common materials (including for example segments that annotators could not assign to a material) are merged in an "other" class that acts as pseudo-background.

In order to study perceptual properties as well as materials, we augment the OS dataset with some of the 47 attributes from the DTD [7]. Since the OS segments do not trigger with sufficient frequency all the 47 attributes, the evaluation is restricted to eleven of them for which it was possible to identify at least 100 matching segments.[2] The attributes were manually labelled in the 53,915 segments retained for materials. We refer to this data as OSA. The complete list of images, segments, labels, and splits are available at http://www.robots.ox.ac.uk/~vgg/data/dtd/.

3. Method

This section describes the methodological contributions of this paper: region description and segmentation.

3.1. Region description

This section introduces a number of visual descriptors suitable to model the appearance of image regions. Texture is traditionally described by orderless pooling of filter bank responses as, unlike in objects, the overall shape information is usually unimportant. However, small under-sampled textures may benefit if recognized in the context of an object. Thus, the primacy of orderless pooling may not always hold in the recognition of textures in natural conditions.

In order to explore the interplay between shape and orderless pooling, we evaluate two corresponding region descriptors: FC-CNN for shape and FV-CNN for texture. Both descriptors are based on the same CNN features [23] obtained from an off-the-shelf CNN pre-trained on the ImageNet ILSVRC 2012 data as suggested in [6, 21, 35]. Since the underlying CNN is the same, it is meaningful to compare FC- and FV-CNN directly.

Object descriptor: FC-CNN. The FC-CNN descriptor is obtained by extracting as features the output of the penultimate Fully-Connected (FC) layer of a CNN, including the non-linear gating function, applied to the input image. This can be considered an object descriptor because the fully connected layers allow FC-CNN to *capture the overall shape of the object* contained in the region. FC-CNN is applied to an image region R (which may be the whole image) by warping the bounding box enclosing R (plus a 10% border) to a square of a fixed size matching the default CNN input geometry, obtaining the same R-CNN descriptor introduced by Girshick *et al.* [14] as a state-of-the-art object detector in the PASCAL VOC [10] data.

Texture descriptor: FV-CNN. The FV-CNN descriptor is inspired by the state-of-the-art texture descriptors of [7] based on the Fisher Vector (FV). Differently from FC-CNN, FV pools local features densely within the described regions *removing global spatial information*, and is therefore more apt at describing textures than objects. Here FV is computed on the output of a single (last) convolutional layer of the CNN, but we compared features from other layers as well (Sect 4.4). By avoiding the computation of the fully connected layers, the input image does not need to be rescaled to a specific size; in fact, the dense convolutional features are extracted at multiple scales and pooled into a single FV just like for SIFT. The pooled convolutional features are extracted immediately after the last linear filtering operator and are not otherwise normalised.

The FV-CNN descriptor is related to the one proposed by Gong *et al.* [15]. There VLAD pooling, which is similar to FV, is applied to FC-CNN-like descriptors extracted from densely sampled image windows. A key difference of FV-CNN is that dense features are extracted from the convolutional rather than fully-connected layers. This is more natural, significantly more efficient (as it does not require recomputing the network for each extracted local descriptor) and, as shown in Sect. 4, more accurate.

3.2. Region segmentation

This section discusses our method to automatically partition an image into a number of recognisable regions. Inspired by Cimpoi *et al.* [7] that successfully ported object description methods to texture descriptors, here we propose a segmentation technique inspired by object detection. An increasingly popular method for object detection, followed for example by R-CNN [14], is to first propose a number of candidate object regions using low-level image cues, and then verifying a shortlist of such regions using a powerful classifier. Applied to textures, this requires a low-level mechanism to generate textured region proposals, followed by a region classifier. A key advantage of this approach is that it allows applying object- (FC-CNN) and texture-like (FV-CNN) descriptors alike. After proposal classifica-

[1] The classes and corresponding number of example segments are: brick (610), cardboard (423), carpet/rug (1,975), ceramic (1,643), concrete (567), fabric/cloth (7,484), food (1,461), glass (4,571), granite/marble (1,596), hair (443), other (2,035), laminate (510), leather (957), metal (4,941), painted (7,870), paper/tissue (1,226), plastic/clear (586), plastic/opaque (1,800), stone (417), tile (3,085), wallpaper (483), wood (9,232).

[2] These are: banded, blotchy, chequered, flecked, gauzy, grid, marbled, paisley, pleated, stratified, wrinkled.

tion, each pixel can be assigned more than one label; this is solved with simple voting schemes, also inspired by object detections methods.

The paper explores two such region generation methods: the crisp regions of [19] and the multi-scale combinatorial grouping of [3]. In both cases, region proposals are generated using low-level image cues, such as colour or texture consistency, as specified by the original methods. It would of course be possible to incorporate FC-CNN and FV-CNN among these energy terms to potentially strengthen the region generation mechanism itself. However, this contradicts partially the logic of the scheme, which breaks down the problem into cheaply generating tentative segmentations and then verifying them using a more powerful (and likely expensive) model. Furthermore, and more importantly, these cues focus on separating texture *instances*, as presented in each particular image, whereas FC-CNN and FV-CNN are meant to identify a texture class. It is reasonable to expect instance-specific cues (say the colour of a painted wall) to be better for segmentation.

4. Results

This section evaluates the proposed region recognition methods for classifying and segmenting materials, describable texture properties, and higher-level object categories. Sect. 4.1 evaluates the *classification task* by assessing how well regions can be classified given that their true extent is known and Sect. 4.3 evaluates both *classification and segmentation*. The rest of the section introduces the evaluation benchmarks and technical details of the representations.

Datasets. The evaluation considers three texture recognition benchmarks other than OS (Sect. 2). The first one is the *Flickr Material Dataset* (FMD) [37], a recent benchmark containing 10 material classes. The second one is the *Describable Texture Datasets* (DTD) [7], which contains texture images *jointly* annotated with 47 describable attributes drawn from the psychological literature. Both FMD and DTD contain images "in the wild", *i.e*. collected in uncontrolled conditions. However, differently from OS, these images are uncluttered. The third texture dataset is *KTH-TIPS-2b* [5, 17], containing a number of example images for each of four samples of 11 material categories. For each material, images of one sample are used for training and the remaining for testing.

Object categorisation is evaluated in the *PASCAL VOC 2007* [10] dataset, containing 20 object categories, any combination of which may be associated to any of the benchmark images. Scene categorisation uses the *MIT Indoor* [34] dataset, containing 67 indoor scene classes. Fine-grained categorisation uses the *Caltech/UCSD Bird* dataset (CUB) [45], containing images of 200 bird species.

Note that some of these datasets come with ground truth region/object localisation. The +R suffix will be appended to a dataset to indicate that this information is used both at training and testing time. For example, OS means that segmentation is performed automatically at test time, whereas OS+R means that ground-truth segmentations are used.

Evaluation measures. For each dataset the corresponding standard evaluator protocols and accuracy measures are used. In particular, for FMD, DTD, MIT Indoor, CUB, and OS+R, evaluation uses average classification accuracy, per-segment/image and normalized for each class. When evaluating the quality of a segmentation algorithm, however, one must account for the fact that in most datasets, and in OS and MSRC in particular, not all pixels are labelled. In this case, accuracy is measured per-pixel rather than per-segment, ignoring all pixels that are unlabelled in the ground truth. For MSRC, furthermore, accuracy is normalised across all pixels regardless of their category. For OSA, since some segments may have more than one label, we are reporting mAP, following the standard procedure for multi-label datasets. Finally, PASCAL VOC 2007 classification uses mean average precision (mAP), computed using the TRECVID 11-point interpolation [10].[3]

Descriptor details. FC-CNN and FV-CNN build on the pre-trained VGG-M [6] model as this performs better than other popular models such as [21] while having a similar computational cost. This network results in 4096-dimensional FC-CNN features and 512-dimensional local features for FV-CNN computation. The latter are pooled into a FV representation with 64 Gaussian components, resulting in 65K-dimensional descriptors. While the FV-CNN dimensionality is much higher than the 4K dimensions of FC-CNN, the FV is known to be highly redundant and can be typically compressed by one order of magnitude without appreciable reduction in the classification performance [31], so the effective dimensionality of FC- and FV-CNN is likely comparable. We verified that by PCA-reducing FV to 4096 dimensions and observing only a marginal reduction in classification performance in the PASCAL VOC object recognition task described below. In addition to VGG-M, the recent state-of-the art VGG-VD (very deep with 19 layers) model of Simonyan and Zisserman [39] is also evaluated.

Due to the similarity between FV-CNN and the dense SIFT FV descriptors used for texture recognition in [7], the latter is evaluated as well. Since SIFT descriptors are smaller (128-dimensional) than the convolutional ones (512-dimensional), a larger number of Gaussian components (256) are used to obtain FV descriptors with a comparable dimensionality. The SIFT descriptors support is 32×32 pixels at the base scale.

In order to make results comparable to [7], we use the same settings whenever possible. FV-CNN and D-SIFT compute features after rescaling the image by factors

[3] The definition of AP was changed in later versions of the benchmark.

	dataset	meas.	IFV	VGG-M			VGG-VD			FV-SIFT	SoA
		(%)		FC	FV	FC+FV	FC	FV	FC+FV	FC+FV-VD	
(a)	KTH-T2b	acc	70.8$_{\pm 2.7}$	71$_{\pm 2.3}$	73.3$_{\pm 4.7}$	73.9$_{\pm 4.9}$	75.4$_{\pm 1.5}$	**81.8**$_{\pm 2.5}$	**81.1**$_{\pm 2.4}$	**81.5**$_{\pm 2.0}$	76.0$_{\pm 2.9}$ [40]
	FMD	acc	59.8$_{\pm 1.6}$	70.3$_{\pm 1.8}$	73.5$_{\pm 2.0}$	76.6$_{\pm 1.9}$	77.4$_{\pm 1.8}$	79.8$_{\pm 1.8}$	**82.4**$_{\pm 1.5}$	**82.2**$_{\pm 1.4}$	57.7$_{\pm 1.7}$ [33, 37]
	OS+R	acc	39.8	41.3	52.5	54.9	43.4	59.5	**60.9**	58.7	–
(b)	DTD	acc	58.6$_{\pm 1.2}$	58.8$_{\pm 0.8}$	66.8$_{\pm 1.6}$	69.8$_{\pm 1.1}$	62.9$_{\pm 0.8}$	72.3$_{\pm 1.0}$	**74.7**$_{\pm 1.0}$	**75.5**$_{\pm 0.8}$	–
	OSA+R	mAP	56.5	54.3	65.2	**67.9**	49.7	67.2	**67.9**	**68.2**	–
(d)	MSRC+R	acc	85.7	85.0	95.4	96.9	79.4	97.7	**98.8**	**99.1**	–
	MSRC+R	msrc-acc	92.0	84.0	97.6	98.1	82.0	**99.2**	**99.6**	**99.5**	–
	VOC07	mAP11	59.9	76.8	76.4	79.5	81.7	**84.9**	**85.1**	**84.5**	85.2 [44]
	MIT Indoor	acc	54.9	62.5	74.2	74.4	67.6	**81.0**	**80.3**	80.0	70.8 [47]
	CUB	acc	17.5	46.1	49.9	54.9	54.6	**66.7**	**67.3**	65.4	73.9 (62.8*) [46]
	CUB+R	acc	27.7	56.5	65.5	68.1	62.8	73.0	**74.9**	73.6	76.37 [46]

Table 1. Evaluation of texture descriptors. The table compares FC-CNN, FV-CNN on two networks trained on ImageNet – VGG-M and VGG-VD, and IFV on dense SIFT. We evaluated these descriptors on (a) material datasets (FMD, KTH-T2b, OS+R), (b) texture attributes (DTD, OSA+R) and (c) general categorisation datasets (MSRC+R,VOC07,MIT Indoor) and fine grained categorisation (CUB, CUB+R). For this experiment the region support is assumed to be known (and equal to the entire image for all the datasets except OS+R and MSRC+R and for CUB+R, where it is set to the bounding box of a bird). (*) using a model without parts. Best results are marked in bold.

		VGG-M			VGG-VD			
dataset	measure (%)	FC-CNN	FV-CNN	FV+FC-CNN	FC-CNN	FV-CNN	FC+FV-CNN	SoA
OS	pp-acc	36.3	48.7 (46.9)	50.5	38.8	**55.4 (55.7)**	**55.2**	–
MSRC	msrc-acc	56.1	82.3	75.5	57.7	**87.0**	80.4	86.5 [24]

Table 2. Segmentation and recognition using crisp region proposals of materials (OS) and things & stuff (MSRC). Per-pixel accuracies are reported, using the MSRC variant (see text) for the MSRC dataset. Results using MCG proposals [3] are reported in brackets for FV-CNN.

$2^s, s = -3, -2.5, \dots 1.5$ (but, for efficiency, discarding scales that would make the image larger than 1024^2 pixels). Before pooling descriptors with a FV, these are usually de-correlated by using PCA. Here PCA is applied to SIFT, additionally reducing its dimension to 80, as this was empirically shown to improve the overall recognition performance. However, PCA is not applied to the convolutional features in FV-CNN as in this case results were worse.

Learning details. The region descriptors (FC-CNN, FV-CNN, and D-SIFT) are classified using 1-vs-rest Support Vector Machine (SVM) classifiers. Prior to learning, descriptors are L^2 normalised and the learning constant set to $C = 1$. This is motivated by the fact that, after data normalisation, the exact choice of C has a negligible effect on performance. Furthermore, the accuracy of the 1-vs-rest classification scheme is improved by recalibrating the SVM scores after training, by scaling the SVM weight vector and bias such that the median scores of the negative and positive training samples for each class are mapped respectively to the values -1 and 1.

4.1. Region recognition: textures

This and the following section evaluate region recognition assuming that the ground-truth region R is known (Table 1), for example because it fills the entire image. This section focuses on textures (materials and perceptual attributes), while the next on objects and scenes.

Texture recognition without clutter. This experiment evaluates the performance of FC-CNN, FV-CNN, D-SIFT, and their combinations in standard texture recognition benchmarks such as FMD, KTH-TIPS-2, and DTD. FC-CNN is roughly equivalent to the DeCAF method used in [7] for this data as regions fill images; however, while the performance of our FC-CNN is similar in KTH ($\sim$ 70%), it is substantially better in FMD (60.7% $\rightarrow$ 70.4% accuracy) and DTD (54.8% $\rightarrow$ 58.7%). This likely is caused by the improved underlying CNN, an advantage which is more obvious in FMD and DTD that are closer to object recognition than KTH. FV-CNN performs within $\pm 2\%$ in FMD and KTH but substantially better for DTD (58.7% $\rightarrow$ 66.6%). D-SIFT is comparable in performance to FC-CNN in DTD and KTH, but substantially worse (70.4% $\rightarrow$ 59.2%) in FMD. Our conclusion is that, even when textures fill the input image as in these benchmarks, orderless pooling in FV-CNN and D-SIFT can be either the same or substantially better than the pooling operated in the fully-connected layers by FC-CNN.

Combining FC- and FV-CNN improves performance in all datasets by $1 - 3\%$. While this combination is already significantly above the state-of-the-art in DTD and FMD (+2.6%/11.2%), the method of [7] still outperforms these descriptors in KTH. However, replacing VGG-M with VGG-VD significantly improves the performance in all cases – a testament to the power of deep features. In particular, the best method FC+FV-CNN-VD, improves the state of the art by at least 6% in all datasets. Interestingly, this is obtained by using a *single* low-level feature type as FC- and

FV-CNN build on the same convolutional features. Adding D-SIFT results in at most $\sim 1\%$ improvement, and in some cases it slightly degrades performance.

Texture recognition in clutter. The advantage of FV-CNN over FC-CNN is much larger when textures do not fill the image but are extracted from clutter. In OS+R (Sect. 2), material recognition accuracy starts at about 46% for both FC-CNN and D-SIFT; however, FV-CNN improves this by more than 11% ($46.5\% \rightarrow 58.1\%$). The combination of FC- and FV-CNN improves results further by $\sim 2\%$, but adding SIFT deteriorates performance. With the very deep CNN conclusions are similar; however, switching to VGG-VD barely affects the FC-CNN performance ($46.5 \rightarrow 48.0\%$), but strongly affects the one of FV-CNN ($58.1\% \rightarrow 65.1\%$). This confirms that FC-CNN, while excellent in object detection, is not a very good descriptor for classifying textured regions. Results in OSA+R for texture attribute recognition (Sect. 2) and in MSRC+R for semantic segmentation are analogous; it is worth noting that, when ground-truth segments are used in this experiment, the best model achieves a nearly perfect 99.7% classification rate in MSRC.

4.2. Region recognition: objects and scenes

This section shows that the FV-CNN descriptor, despite its orderless nature that make it an excellent texture descriptor, excels at object and scene recognition as well. In the remainder of the section, and unless otherwise noted, region descriptors are applied to images as a whole by considering these single regions.

FV-CNN vs FC-CNN. As seen in Table 1, in PASCAL VOC and MIT Indoor the FC-CNN descriptor performs very well but in line with previous results for this class of methods [6]. FV-CNN performs similarly to FC-CNN in PASCAL VOC, but substantially better ($+5\%$ for VGG-M and $+13\%$ for VGG-VD) in MIT Indoor. As further discussed below, this is an example of the ability of FV-CNN to transfer between domains better than FC-CNN. The gap between FC-CNN and FV-CNN is the highest for the very deep VGG-VD models ($68.1\% \rightarrow 81.1\%$), a trend also exhibited by other datasets as seen in Tab. 1. In the CUB dataset, FV-CNN significantly outperforms FC-CNN both whether the descriptor is computed from the whole image (CUB) or from the bird bounding box (CUB+R). In the latter case, the difference is very large ($+10 - 14\%$).

Comparison with alternative pooling methods. FV-CNN is related to the method of [15], which uses a similar underlying CNN and VLAD instead of FV for pooling. Notably, however, FV-CNN results on MIT Indoor are markedly better than theirs for both VGG-M and VGG-VD ($68.8\% \rightarrow 73.5\%/81.1\%$ resp.) and marginally better ($68.8\% \rightarrow 69.1\%$) when the same CAFFE CNN is used (Tab. 3). The key difference is that FV-CNN pools convolutional features, whereas [15] pools fully-connected descriptors extracted from square image patches. Thus, even without spatial information as used by [15], FV-CNN is not only substantially faster, but at least as accurate. Using the same settings, that is, the same net and the same three scales, our approach results in an 8.5× speedup.

	Accuracy (%)		
CNN	FC-CNN	FV-CNN	FC+FV-CNN
PLACES	65.0	67.6	73.1
CAFFE	58.6	69.7	71.6
VGG-M	62.5	74.2	74.4
VGG-VD	67.6	**81.0**	80.3

Table 3. **Accuracy of various CNNs on the MIT indoor dataset.**

Comparison with the state-of-the-art. The best result obtained in PASCAL VOC is comparable to the current state-of-the-art set by the deep learning method of [44] ($85.2\% \rightarrow 85.0\%$), but using a much more straightforward pipeline. In MIT Places our best performance is also substantially superior ($+10\%$) to the current state-of-the-art using deep convolutional networks learned on the MIT Place dataset [47] (see also below). In the CUB dataset, our best performance is a little short ($\sim 3\%$) of the state-of-the-art results of [46]. However, [46] uses a category-specific part detector and corresponding part descriptor as well as a CNN fine-tuned on the CUB data; by contrast, FV-CNN and FC-CNN are used here as *global image descriptors* which, furthermore, *are the same for all the datasets considered*. Compared to the results of [46] without part-based descriptors (but still using a part-based object detector), our global image descriptors perform substantially better ($62.1\% \rightarrow 69.1\%$).

We conclude that FV-CNN is a very strong descriptor. Results are usually as good, and often significantly better, than FC-CNN. In most applications, furthermore, FV-CNN is many times faster as it does not require evaluating the CNN for each target image region. Finally, FC- and FV-CNN can be combined outperforming the state-of-the-art in many benchmarks.

Domain transfer. So far, the same underlying CNN features, trained on ImageNet's ILSVCR, were used for all datasets. Here we investigate the effect of using domain-specific features. To do so, we consider the PLACES [47], trained to recognize places on a dataset of about 2.5 million labeled images. [47] showed that, applied to the task of scene recognition in MIT Indoor, these features outperform similar ones trained on ILSVCR (denoted CAFFE [21] below) – a fact explained by the similarity of domains. Below, we repeat this experiment using FC- and FV-CNN descriptors on top of VGG-M, VGG-VD, PLACES, and CAFFE.

Results are shown in Table 3. The FC-CNN results are in line with those reported in [47] – in scene recognition with FC-CNN the same CNN architecture performs better if trained on the Places dataset instead of the ImageNet data

(58.6% → 65.0% accuracy[4]). Nevertheless, stronger CNN architectures such as VGG-M and VGG-VD can approach and outperform PLACES even if trained on ImageNet data (65.0% → 63.0%/68.1%).

However, when it comes to using the filter banks with FV-CNN, conclusions are very different. First, FV-CNN outperforms FC-CNN in all cases, with substantial gains up to 20% in correspondence of a domain transfer from ImageNet to MIT Indoor. Second, *the advantage of using domain-specific CNNs disappears*. In fact, the same CAFFE model that is 6.4% worse than PLACES with FC-CNN, is actually 1.5% *better* when used in FV-CNN. The conclusion is that FV-CNN appears to be immune, or at least substantially less sensitive, to domain shifts.

Our tentative explanation of this surprising phenomenon is that the convolutional layers are less committed to a specific dataset than the fully ones. Hence, by using those, FV-CNN tends to be a more general than FC-CNN.

4.3. Texture segmentation

The previous section considered the problem of region recognition when the region support is known at test time. This section studies the problem of recognising regions when their extent R is *not* known and also be estimated.

The first experiment (Tab. 2) investigates the simplest possible scheme: combining the region descriptors of Sect. 4.1 with a general-purpose image segmentation method, namely the *crisp regions* of [19]. Two datasets are evaluated: OS for material recognition and MSRC for things & stuff. Compared to OS+R, classifying crisp regions results in a drop of about 5% points for all descriptors. As this dataset is fairly challenging with best achievable performance is 55.4%, this is a satisfactory result. But it also illustrates that there is ample space for future improvements. In MSRC, the best accuracy is 87.0%, just a hair above the best published result 86.5% [25]. Remarkably, these algorithms not use any dataset-specific training, nor CRF-regularised semantic inference: they simply greedily classify regions as obtained from a general-purpose segmentation algorithms. Qualitative segmentation results (sampled at random) are given in Fig. 2 and 3.

Unlike crisp regions, the proposals of [3] are overlapping and a typical image contains thousands of them. We propose a simple scheme to combine prediction from multiple proposals. For each proposal we set its *label* to the highest scoring class, and *score* to the highest score. We then sort the proposals in the increasing order of their score divided by their *area* and paste them one by one. This has the effect of considering larger regions before smaller ones and more confident regions after less ones for regions of the same area. Results using FV-CNN shown in Tab. 2 in brackets (FC-CNN was too slow for our experiments). The results are comparable to those using crisp regions, and we obtain 55.7% accuracy on the OS dataset. Our initial attempts at schemes such as non-maximum suppression of overlapping regions that are quite successful for object segmentation [16] performed rather poorly. We believe this is because unlike objects, material information is fairly localized and highly irregularly shaped in an image. However, there is room for improvement by combining evidence from multiple segmentations.

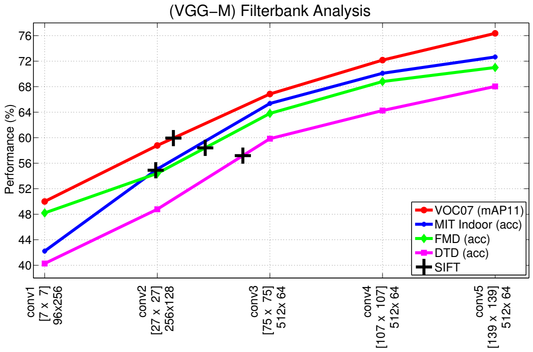

Figure 4. **CNN filterbank analysis for VGG-M.** Performance of filter banks extracted from various layers network are shown on various datasets. For each layer $conv\{1,...,5\}$ we show, the size of the receptive field $[N \times N]$, and $FB \times D$, where FB is the size of filter bank and D is the dictionary size in the FV representation. The performance using SIFT is shown in black plus (+) marks.

4.4. Convolutional layer analysis

We study the performance of filter banks extracted from different layers of a CNN in the FV-CNN framework. We use the VGG-M network which has five convolutional layers. Results on various datasets, obtained as in Sect. 4.1 and 4.2, are shown in Fig. 4. In addition we also show the performance using FVs constructed from dense SIFT using a number of words such that the resulting FV is roughly the same size of FV-CNN. The CNN filter banks from layer 3 and beyond significantly outperform SIFT. The performance monotonically improves from layer one to five.

5. Conclusions

We have conducted a range of experiments on material and texture attribute recognition in a large dataset of textures in clutter. This benchmark was derived from OpenSurfaces, an earlier contribution of the computer graphics community, highlights the potential for collaboration between computer graphics and vision communities. We have also evaluated a number of state-of-the-art texture descriptors on these and many other benchmarks. Our main finding is that orderless pooling of convolutional neural network features is a remarkably good texture descriptor, versatile enough to dubbed as a scene and object descriptor, resulting in new state-of-the-art performance in several benchmarks.

[4] [47] report 68.3% for PLACES applied to MIT Indoor, a small difference explained by implementation details such as the fact that, for all the methods, we do not perform data augmentation by jittering.

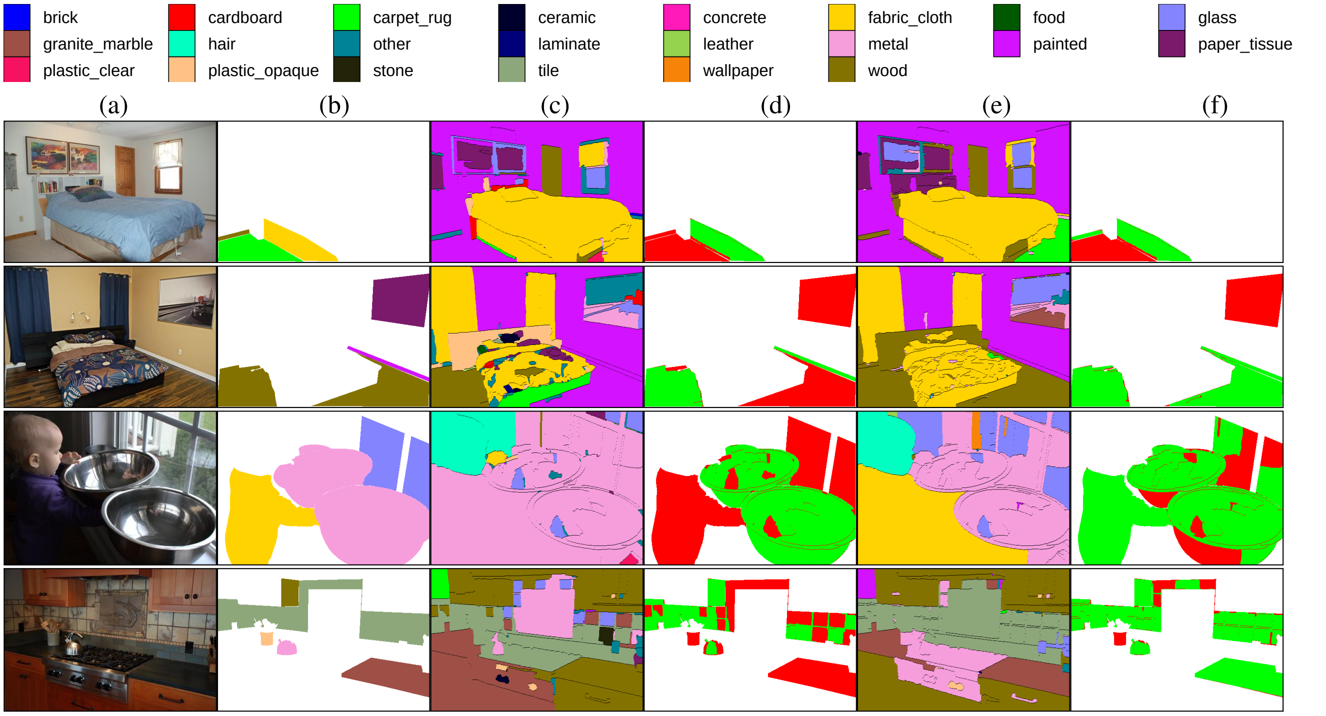

Figure 2. **OS material recognition results.** Example test image with material recognition and segmentation on the OS dataset. (a) original image. (b) ground truth segmentations from the OpenSurfaces repository (note that not all pixels are annotated). (c) FC-CNN and crisp-region proposals segmentation results. (d) incorrectly predicted pixels (restricted to the ones annotated). (e-f) the same, but for FV-CNN.

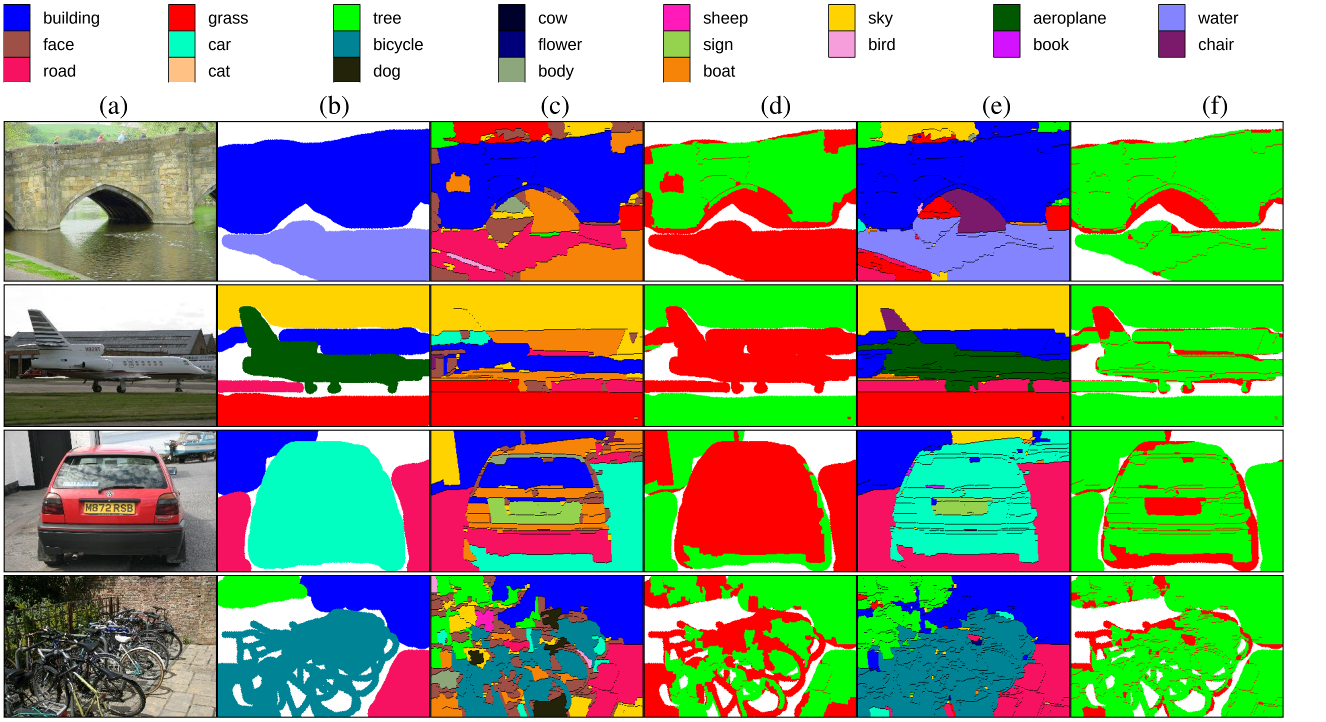

Figure 3. **MSRC object segmentation results.** (a) image, (b) ground-truth, (c-d) FC-CNN, (d-e) FV-CNN segmentation and errors.

Acknowledgements M. Cimpoi was supported by ERC grant VisRec no. 228180 and the XRCE UAC grant. S. Maji acknowledges a faculty startup grant from UMass, Amherst.

References

[1] E. H. Adelson. On seeing stuff: The perception of materials by humans and machines. *SPIE*, 4299, 2001. 1

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

[3] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. CVPR, 2014. 4, 5, 7

[4] S. Bell, P. Upchurch, N. Snavely, and K. Bala. Opensurfaces: A richly annotated catalog of surface appearance. In *Proc. SIGGRAPH*, 2013. 2

[5] B. Caputo, E. Hayman, and P. Mallikarjuna. Class-specific material categorisation. In *ICCV*, 2005. 1, 4

[6] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *Proc. BMVC*, 2014. 2, 3, 4, 6

[7] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proc. CVPR*, 2014. 1, 3, 4, 5

[8] 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. 1

[9] A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *CVPR*, volume 2, 1999. 1

[10] M. Everingham, A. Zisserman, C. Williams, and L. V. Gool. The PASCAL visual obiect classes challenge 2007 (VOC2007) results. Technical report, Pascal Challenge, 2007. 3, 4

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

[12] D. Forsyth. Shape from texture and integrability. In *ICCV*, volume 2, pages 447–452. IEEE, 2001. 1

[13] J. Gårding. Shape from texture for smooth curved surfaces in perspective projection. *Journal of Mathematical Imaging and Vision*, 2(4):327–350, 1992. 1

[14] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. CVPR*, 2014. 2, 3

[15] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *Proc. ECCV*, 2014. 2, 3, 6

[16] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *Computer Vision–ECCV 2014*, pages 297–312. Springer, 2014. 2, 7

[17] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh. On the significance of real-world conditions for material classification. *ECCV*, 2004. 4

[18] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Proc. ECCV*, 2014. 2

[19] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. Crisp boundary detection using pointwise mutual information. In *Proc. ECCV*, 2014. 4, 7

[20] A. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern recognition*, 24(12):1167–1186, 1991. 1

[21] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/, 2013. 3, 4, 6

[22] B. Julesz and J. R. Bergen. Textons, the fundamental elements in preattentive vision and perception of textures. *Bell System Technical Journal*, 62(6, Pt 3):1619–1645, Jul-Aug 1983. 1

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012. 2, 3

[24] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Graph cut based inference with co-occurrence statistics. In *Proc. ECCV*, pages 239–253, 2010. 5

[25] L. Ladicky, P. Sturgess, K. Alahari, C. Russell, and P. Torr. What, where and how many? combining object detectors and crfs. In *Proc. ECCV*, 2010. 7

[26] 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. 1

[27] L. Liu, L. Wang, and X. Liu. In defense of soft-assignment coding. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2486–2493. IEEE, 2011. 1

[28] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999. 1

[29] B. Manjunath and R. Chellappa. Unsupervised texture segmentation using markov random field models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):478–482, 1991. 1

[30] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. In *Proc. CVPR*, 2014. 2

[31] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman. A compact and discriminative face track descriptor. In *Proc. CVPR*, 2014. 4

[32] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Computer Vision–ECCV 2010*, pages 143–156. Springer, 2010. 1

[33] X. Qi, R. Xiao, C. G. Li, Y. Qiao, J. Guo, and X. Tang. Pairwise rotation invariant co-occurrence local binary pattern. *PAMI*, 36(11):2199–2213, Nov 2014. 5

[34] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *Proc. CVPR*, 2009. 2, 4

[35] A. S. Razavin, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *DeepVision workshop*, 2014. 3

[36] G. Schwartz and K. Nishino. Visual material traits: Recognizing per-pixel material context. In *Proc. CVCP*, 2013. 1

[37] 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. 1, 4, 5

[38] 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. 1

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 2, 4

[40] M. Sulc and J. Matas. Fast features invariant to rotation and scale of texture. Technical report, 2014. 5

[41] M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? In *CVPR*, volume 2, pages II–691. IEEE, 2003. 1

[42] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE, 2010. 1

[43] L. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH*, pages 479–488. ACM Press/Addison-Wesley Publishing Co., 2000. 1

[44] Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan. Cnn: Single-label to multi-label. 2014. 5, 6

[45] P. Welinder, S. Branson, T. Mita, C. Wah, and F. Schroff. Caltech-ucsd birds 200. Technical report, Caltech-UCSD, 2010. 4

[46] N. Zhang, J. Donahue, R. Girshickr, and T. Darrell. Part-based R-CNNs for fine-grained category detection. In *Proc. ECCV*, 2014. 5, 6

[47] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Proc. NIPS*, 2014. 2, 5, 6, 7

[48] X. Zhou, K. Yu, T. Zhang, and T. S. Huang. Image classification using super-vector coding of local image descriptors. In *Computer Vision–ECCV 2010*, pages 141–154. Springer, 2010. 1