# Learning Multi-Scale Representations for Material Classification

Wenbin Li*

Max Planck Institute for Informatics, Saarbrücken, Germany

**Abstract.** The recent progress in sparse coding and deep learning has made unsupervised feature learning methods a strong competitor to hand-crafted descriptors. In computer vision, success stories of learned features have been predominantly reported for object recognition tasks. In this paper, we investigate if and how feature learning can be used for material recognition. We propose two strategies to incorporate scale information into the learning procedure resulting in a novel multi-scale coding procedure. Our results show that our learned features for material recognition outperform hand-crafted descriptors on the FMD and the KTH-TIPS2 material classification benchmarks.

## 1 Introduction

Perceiving and recognizing material is a fundamental aspect of visual perception. In contrast to texture recognition it requires generalization over large variations between material instances and discriminance between visually similar materials. Studies have shown that material recognition in real-world scenarios is far from solved [6, 18]. While well established manually designed features [21, 20, 23] have been shown to be most powerful on material recognition tasks. It is non-trivial to come up with a good design of visual features and efforts are clearly needed to explore the question how we can automatically learn features for this challenging and relevant problem. Further, it is known that multi-scale representations are key for competitive performance on this task [6, 16, 20]. However, current feature learning techniques do not include multi-scale representations. Therefore, we present the first study of applying unsupervised feature discovery algorithms for material recognition and show improved performance over hand-crafted feature descriptors. Further, we investigate different ways to incorporate multi-scale information in the feature learning process and propose the first multi-scale coding procedure that results in a joint representation of multi-scale patches (see Figure 1 for examples of multi-scale codes).

## 2 Related Work

*Material Recognition* Curet database [9] was first proposed to address the recognition of single material instance, which motivated progress on texture recognition [27, 25]. Later research [12, 6] shifted the focus towards whole material

* Recommended for submission to YRF2014 by Dr. Mario Fritz

(a) Filters learned on the KTH-TIPS2. (b) Filters learned on the FMD.

**Fig. 1.** Examples of multi-scale filters learned on KTH-TIPS2a (a) and FMD (b) from the proposed MS4C model.

class, emphasizing challenges like scale and intra-class variations. Liu et al [18] presented the Flickr material dataset using images captured under unknown real-world conditions and has been evaluated in [14, 17, 23].

*Feature Learning* In machine learning literature, a rich set of models have been proposed for feature learning aimed to find a better representation for data. Examples include sparse coding [24], restricted Boltzmann machines [13, 8] and various autoencoder-based models [4]. The Spike-and-Slab Sparse coding (S3C) [11] has recently been proposed to combine the advantages of SC and RBMs and has shown superior performance. Our representation is based on the S3C model and extended to multi-scale representation.

*Multi-Scale Representation* Early texton work included multi-scale filters to enrich the representation. Although the clustering step can be seen as a form of feature learning, the filters are hand-crafted. Also the LBP work has multi-scale extension [20] that has substantially improved the performance. Recently, a multi-scale convolutional neural network (CNN) [10] has been proposed. It differs from our multi-scale feature learning approach as we learn a representation jointly across scales. The image codes derived from our representation directly encode the multi-scaled information. Figure 1 illustrates some examples of multi-scale codes learned by our model.

## 3 Feature Learning

In a typical patch-based feature learning setting, random patches $\{v_n\}$ are firstly extracted from training images and a feature mapping $f$ is learned (dictionary learning). Then one can encode the patches covering the input image and pool the codes together in order to form the final feature representation (feature extraction). By altering the model used for feature mapping, we can get different feature representations.

*Sparse Coding (SC)* Sparse coding for visual feature coding was originally proposed [22] as an unsupervised learning model of low-level sensory processing in humans. The dictionary $W$ is obtained by optimizing $\underset{W,s_i}{\text{minimize}} \sum_i \|v_i - W s_i\|_2^2 + \beta\|s_i\|_1, w.r.t. \|W_j\|_2 \leq 1, \forall j$, then feature $s_i$ for each input $v_i$ is obtained by solving the same form of optimization problem but with the learned dictionary.

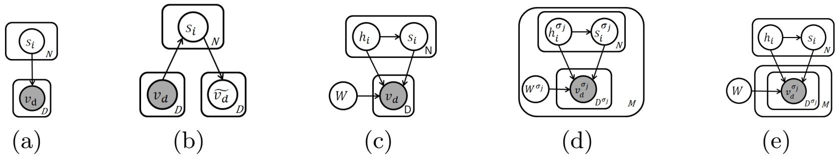

**Fig. 2.** Graphical model for: (a) SC, (b) AE, (c) S3C, (d) S4C, (e) MS4C.

*Auto-Encoder (AE)* Auto-encoder is another popular model for learning feature representation. $v$ is firstly mapped into a latent representation $s$ (encoding) with a nonlinear function $f$ such as the sigmoid function:$s = f(Wv + b)$. Then it is mapped back into a reconstruction $\widetilde{v}$ through a similar transformation $\widetilde{v} = f(\widetilde{W}s + \widetilde{b})$, and the dictionary (or weights) $W$ is obtained by optimizing the reconstruction error $W = \underset{W}{\operatorname{argmin}} \sum_i L(v_i, \widetilde{v_i})$, where $L(v_i, \widetilde{v})$ is a loss function such as the squared error $L(v, \widetilde{v}) = \|v - \widetilde{v}\|_2^2$. During encoding phase, the features are computed by applying the forward-pass only in order to obtain $s$.

*Spike-and-Slab Sparse Coding (S3C)* The Spike-and-Slab Sparse Coding (S3C) [11] has been recently proposed to combine the merits of both sparse coding and RBMs: the first layer is a real-valued $D$-dimensional visible vector $v \in R^D$, where $v_d$ corresponding to the pixel value at position d; the second layer consists of two different kinds of latent variables, the binary *spike* variables $h \in \{0, 1\}^N$ and the real-valued *slab* variables $s \in R^N$. The spike variable $h_i$ gates the slab variable $s_i$, and those two jointly define the $i^{th}$ hidden unit as $h_i s_i$. The process can be more formally described as follows:

$$\begin{aligned} \forall i \in \{1, ..., N\}&, d \in \{1, ..., D\} \\ p(h_i = 1) &= g(b_i) \\ p(s_i | h_i) &= N(s_i | h_i \mu_i, \alpha_{ii}^{-1}) \\ p(v_d | s, h) &= N(v_d | W_{d:}(h \circ s), \beta_{dd}^{-1}) \end{aligned}$$

where $g$ is the logistic sigmoid function, $b$ is the biases on the spike variables, $\mu$ and $W$ govern the linear dependence of $s$ on $h$ and $v$ on $s$ respectively, $\alpha$ and $\beta$ are diagonal precision matrices of their respective conditionals, and $h \circ s$ denotes the element-wise product of $h$ and $s$. Columns of $W$ are constrained to have unit norm, $\alpha$ is restricted to be a diagonal matrix and $\beta$ to be a diagonal matrix or a scalar. The model has shown to outperform previous feature learning techniques [11] and is the best performer on a recent transfer learning challenge [1].

## 4 Multi-Scale Feature Learning

As shown in [6, 17, 20, 19], encoding scale information is important for material recognition task. Hence we propose two different strategies to include multi-scale information in feature learning:

*Stacked Spike-and-Slab Sparse Coding (S4C)* We perform the encoding at multiple scales and stack the obtained codes, then use this code for classification. We convolve the patch with different sized Gaussians before encoding in order to represent scale information. While there is a common dictionary, the representation already encodes how the patch evolves in scale-space and therefore multi-scale information is captured:

$$\begin{aligned}
&\forall i \in \{1, ..., N\}, j \in \{1, ..., M\}, d \in \{1, ..., D\}\\
&\quad p(h_i^{\sigma_j} = 1) = g(b_i^{\sigma_j})\\
&\quad p(s_i^{\sigma_j} | h_i^{\sigma_j}) = N(s_i^{\sigma_j} | hi^{\sigma_j} \mu_i^{\sigma_j}, (\alpha_{ii}^{\sigma_j})^{-1})\\
&p(v_d^{\sigma_j} | s^{\sigma_j}, h^{\sigma_j}) = N(v_d^{\sigma_j} | W_{d:}^{\sigma_j} (h^{\sigma_j} \circ s^{\sigma_j}), (\beta_{dd}^{\sigma_j})^{-1})
\end{aligned}$$

where $M$ denotes the number of scales and $\sigma_j$ indexes units and parameters at specific scale.

*Multi-Scale Spike-and-Slab Sparse Coding (MS4C)* We construct a multi-scale pyramid for each image, apply the feature learning directly on the pyramid and then use the obtained codes for classification. In contrast to the S4C approach, the MS4C approach yields codes that model each patch jointly across scales:

$$\begin{aligned}
&\forall i \in \{1, ..., N\}, j \in \{1, ..., M\}, d \in \{1, ..., D\}\\
&\quad p(h_i = 1) = g(b_i)\\
&\quad p(s_i | h_i) = N(s_i | hi\mu_i, \alpha_{ii}^{-1})\\
&\quad p(v_d^{\sigma_j} | s, h) = N(v_d^{\sigma_j} | W_{d:}(h \circ s), \beta_{dd}^{-1})
\end{aligned}$$

where $v_d^{\sigma_j}$ denotes the joint representation of visible units at specific scale $\sigma_j$. Inference is carried out as in the S3C model as the different scales can be seen as a decomposition of a larger multi-scale patch that includes all the scales. Figure 2 shows the graphical models for both single-scale and multi-scale models.

## 5 Experiments

In our experiments, we investigate how the learning framework can be used for feature discovery on material recognition task and compare our approach to the state-of-the-art on the FMD and the KTH-TIPS2 databases.

### 5.1 Experimental setup

We use KTH-TIPS2 [6] and FMD [18] in our experiments. For KTH-TIPS2 database, we use two instances for training and the other two for test per category. For FMD, we randomly split half for training and the other half for testing as suggested in [18]. We compare the learned features with hand-crafted features on the two databases with standard SVM classifier [7]. For single scale, we compare to the LBP [21] and its variants. For multi-scale approaches, we consider: Texton [16], MLBP [20]. For the learned representation, we compare to vector quantization, sparse coding, auto encoders and the spike-and-slab approach. In all our experiments we fix the size of dictionary at 1600 for consistency.

### 5.2 Single-scale

For learned features, we apply the K-means, AE, SC and the S3C on the patch data where we vary the patch size; for hand-crafted features, we examine the original LBP and its variants as described in [20]. We use Theano [5] and Pylearn2 [2] for the auto-encoder and the S3C, the SPAMS [3] package for SC.

Results are shown in Figure 3 and Figure 4. On both datasets, the S3C model in combination with the linear kernel outperforms all other hand-crafted and learned features. With a performance of 71.3% and 48.4% for the KTH-TIPS2a and the FMD respectively it improves by 4.1% (over $LBP_u$ with the $exp - \chi^2$ kernel) and 9% (over $LBP$ with the linear kernel) respectively. The best performance is achieved for a patch size of 12. We verified that this parameter can be found via cross-validation on the training set. We attribute the decrease in the performance for the patch size of 24 to a lack of data to learn the required number of parameters. Best performance for feature learning technique is typically obtained in combination with linear kernel, while the hand-crafted features have to rely on the non-linear $exp - \chi^2$ kernel. This is another appealing property of the learned features from a computational point of view.

Overall, we found that the S3C feature performs better than other learning approaches and the hand-crafted features for the single-scale setting, and hence we further developed the S3C model to multi-scale approaches in the following experiments.

### 5.3 Multi-scale

Here we examine multi-scale feature representation for the task. Also we investigate the combination of color information for the learned representation. For hand-crafted features, we include the MLBP and also the texton with the MR8 filter [26]. As shown in Figure 3 (a), (c) and Figure 4 (a), (c). MLBP shows better performance than textons. While the S4C produces slightly worse performance than the MLBP on KTH-TIPS2, we see an improvement of 1.4% for the MS4C. Further including color information improves the performance to 70.5% which is an overall improvement of 3.8% over the best hand-crafted descriptor. From the numbers on the FMD database, we see S4C and MS4C beat the best hand-crafted feature (MLBP) by 7.2% and 8% respectively. On this database, inclusion of color information does not yield additional improvements. The new joint multi-scale coding of of the MS4C consistently improves over the stacked approach of S4C.

Further we reproduced two additional settings in order to provide more points of comparison to the state-of-the-art. We follow the protocol in [15] on the KTH-TIPS2-a data, and then report results via a 3-NN classifier, feature learned by single scale S3C at patch size of 12x12 achieved 70.2%, which is significantly better than the reported results of 64.2% for LQP. Also we did additional experiments on the FMD database, following the settings in [14], with multi-scale collaborated representation, we got average recognition rate of 48.3% and standard deviation of 1.8%, which is comparable to the best single kernel descriptor with 49%.

Figure 1 visualizes the MS4C model where we see how each filter has a multi-scale response. Together with the strong performance in our experiments, we conclude that a multi-scale code indeed captures additional information about how edge structures propagate through scales.

| ClassificationRate(%) | | | | | |
|---|---|---|---|---|---|
| Single-Scale | | | | Multi-Scale | |
| LBP | $LBP_u$ | $LBP_{ri}$ | $LBP_{ri,u}$ | Texton | MLBP |
| 58.7/64.8 | 60.3/67.2 | 55.0/53.6 | 50.9/51.4 | 54.0/58.9 | 66.7/66.1 |

(a) Hand-crafted Feature.

| PatchSize | ClassificationRate(%) | | | |
|---|---|---|---|---|
| | KM | AE | SC | S3C |
| 6 | 60.6/64.8 | 54.3/48.6 | 60.8/64.8 | 63.8/57.5 |
| 12 | 58.4/65.5 | 49.6/44.2 | 66.0/64.8 | 71.3/66.0 |
| 24 | 58.3/65.0 | 48.9/39.1 | * | 55.9/60.8 |

(b) Standard Feature Learning.

| ClassificationRate(%) | | |
|---|---|---|
| S4C | MS4C | MS4C+Color |
| 65.6/58.6 | 68.1/66.6 | 70.5/69.3 |

(c) Multi-scale Feature Learning.

**Fig. 3.** Results on KTH-TIPS2a with linear kernel and the $exp - \chi^2$ kernel.

| ClassificationRate(%) | | | | | |
|---|---|---|---|---|---|
| Single-Scale | | | | Multi-Scale | |
| LBP | $LBP_u$ | $LBP_{ri}$ | $LBP_{ri,u}$ | Texton | MLBP |
| 39.4/36 | 38.2/36.2 | 34.2/35.6 | 27.8/31.8 | 29.4/35.6 | 41.4/42.0 |

(a) Hand-crafted Feature.

| PatchSize | ClassificationRate(%) | | | |
|---|---|---|---|---|
| | KM | AE | SC | S3C |
| 6 | 29.2/38.0 | 37.6/25.0 | 34.8/30.8 | 42.6/39.2 |
| 12 | 26.0/39.6 | 32.4/25.0 | 39.4/26.4 | 48.4/41.8 |
| 24 | 26.8/37.2 | 29.2/22.0 | * | 40.8/44.0 |

(b) Standard Feature Learning.

| ClassificationRate(%) | | |
|---|---|---|
| S4C | MS4C | MS4C+Color |
| 49.2/42.2 | 50.0/41.0 | 48.8/43.2 |

(c) Multi-scale Feature Learning.

**Fig. 4.** Results on FMD with linear kernel and $exp - \chi^2$ kernel.

## 6 Conclusions

We have investigated different feature learning strategies for the task of material classification. Our results match and even surpass standard hand-crafted descriptors. Furthermore, we extend feature learning techniques to incorporate scale information. We propose the first coding procedure that learns and encodes features with a joint multi-scale representation. The comparison of our learned features with state-of-the-art descriptors shows improved performance on standard material recognition benchmarks.

## References

1. Challenges in learning hierarchical models: Transfer learning and optimization. https://sites.google.com/site/nips2011workshop/transfer-learning-challenge

2. Pylearn2 vision, a python library for machine learning. http://deeplearning.net/software/pylearn2/
3. Sparse modeling software, an optimization toolbox for solving various sparse estimation problems. http://spams-devel.gforge.inria.fr/
4. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: NIPS (2007)
5. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., Bengio, Y.: Theano: a CPU and GPU math expression compiler. In: Proceedings of the Python for Scientific Computing Conference (SciPy) (2010)
6. Caputo, B., Hayman, E., Mallikarjuna, P.: Class-specific material categorisation. In: ICCV (2005)
7. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology (2011)
8. Courville, A., Bergstra, J., Bengio, Y.: A spike and slab restricted boltzmann machine. JMLR (2011)
9. Dana, K.J., van Ginneken, B., Nayar, S.K., Koenderink, J.J.: Reflectance and texture of real-world surfaces. ACM Trans. Graph. (1999)
10. Farabet, C., Couprie, C., Najman, L., LeCun., Y.: Scene Parsing with Multiscale Feature Learning, Purity Trees, and Optimal Covers. In: ICML (2012)
11. Goodfellow, I., Couville, A., Bengio, Y.: Large-scale feature learning with spike-and-slab sparse coding. In: ICML (2012)
12. Hayman, E., Caputo, B., Fritz, M., Eklundh, J.O.: On the significance of real-world conditions for material classification. In: ECCV (2004)
13. Hinton, G., Osindero, S., Teh, Y.: A fast learning algorithm for deep belief nets. Neural computation (2006)
14. Hu, D., Bo, L., Ren, X.: Toward robust material recognition for everyday objects. In: BMVC (2011)
15. Hussain, S.U., Triggs, B.: Visual recognition using local quantized patterns. In: ECCV (2012)
16. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. IJCV (2001)
17. Li, W., Fritz, M.: Recognizing materials from virtual examples. In: ECCV (2012)
18. Liu, C., Sharan, L., Adelson, E.H., Rosenholtz, R.: Exploring features in a bayesian framework for material recognition. In: CVPR (2010)
19. Mäenpää, T., Pietikäinen, M.: Multi-scale binary patterns for texture analysis. Image Analysis (2003)
20. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. TPAMI (2002)
21. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recognition (1996)
22. Olshausen, B.A., et al.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature (1996)
23. Qi, X., Xiao, R., Guo, J., Zhang, L.: Pairwise rotation invariant co-occurrence local binary pattern. In: ECCV (2012)
24. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.: Self-taught learning: transfer learning from unlabeled data. In: ICML (2007)
25. Varma, M., Zisserman, A.: A statistical approach to material classification using image patch exemplars. TPAMI (2009)
26. Varma, M., Zisserman, A.: Classifying images of materials: Achieving viewpoint and illumination independence. In: Computer VisionECCV 2002 (2002)

27. Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. IJCV (2005)