A Learned Dictionary Model for Texture Classification

Clara Fannjiang CLARAFJ@STANFORD.EDU

1. Abstract

2. Introduction

Biological visual systems process incessant streams of natural images, and have done so since organisms first developed vision. To capture and interpret visual signals with such high throughput, it makes sense that visual systems have unearthed clever schemes to encode images cheaply. By recognizing statistical patterns that are inherent to natural scenery, visual systems can represent complex images by triggering just a handful of underlying patterns ([2], [8]) from a *dictionary* of possible patterns. Inspired by Nature's *sparse coding* strategy, we develop a novel method for texture classification by learning dictionaries that provide sparse models for each texture class. Unknown textures are then classified based on which dictionary provides the most accurate sparse representation.

More formally, let $\mathbf{D} \in \mathbb{R}^{n \times k}$ be a dictionary whose columns are k prototypical patterns or *atoms*. We can represent a signal $\mathbf{x} \in \mathbb{R}^N$ as a sparse linear combination of atoms, such that $\mathbf{x} = \mathbf{Ds}$ for an exact representation, or $\mathbf{x} \approx \mathbf{Ds}$ in that $||\mathbf{s} - \mathbf{Dx}||_2 \leq \epsilon$ for an approximation. The dictionary can be *overcomplete*, where $k > n$ and there are more explanatory patterns than signal components, or *underdetermined*, where $k < n$. Overcomplete models are more robust to noise and can capture more elaborate structures of a signal; on the other hand, the compactness of underdetermined models may distinguish between different textures more clearly.

For an overcomplete dictionary, note that there are infinitely many representations $\mathbf{s}$. The principle of sparse coding is to constrain $\mathbf{s}$ to be the sparsest representation, or the one with the least number of nonzero components. Such a representation captures the signal most cheaply for us, as we can just store the nonzero components. It is given by

$$\arg\min_{\mathbf{s}} ||\mathbf{s}||_0 \ \ s.t. \ \ \mathbf{x} = \mathbf{Ds} \tag{1}$$

for an exact representation, or

$$\arg\min_{\mathbf{s}} ||\mathbf{s}||_0 \ \ s.t. \ \ ||\mathbf{x} - \mathbf{Ds}||_2 \leq \epsilon \tag{2}$$

for an approximation, where the so-called "ℓ-0 norm" $||\cdot||_0$ gives the number of nonzero components of the argument. A related formulation, where we cap the number of nonzero components in $\mathbf{s}$ to c, is given by

$$\arg\min_{\mathbf{s}} ||\mathbf{x} - \mathbf{Ds}||_2 \ \ s.t. \ \ ||\mathbf{s}||_0 \leq c \tag{3}$$

These are combinatorial optimization problems and NP-hard. Approximation schemes include the greedy *matching pursuit* and *orthogonal matching pursuit* (OMP; [13]) algorithms, as well as convex relaxations that replace the ℓ-0 norm with an ℓ-1 norm.

The other variable in problems (1) and (2), which our texture classification model focuses on, is the choice of the dictionary $\mathbf{D}$. If we know our signals $\mathbf{x}$ belong to a particular class of textures, how can we design $\mathbf{D}$ to adapt to the distinguishing patterns of $\mathbf{x}$? In other words, what set of atoms enables the sparsest representations of a class of textures? Contrary to fixed bases like wavelets, we can indeed adapt dictionaries for particular classes of signals. Algorithms for learning dictionaries include the probabilistic approach in [10], MOD [7], and K-SVD [1]; we focus on implementing K-SVD, as it is the most generalized framework and computationally cheapest. If we have a set of training signals $\{\mathbf{x}_i\}_i^m$ that are the columns of $\mathbf{X} \in \mathbb{R}^{n \times m}$, the learning task is described by the optimization problem

$$\min_{\mathbf{D},\mathbf{S}} ||\mathbf{X} - \mathbf{DS}||_F \ \ s.t. \ \ ||\mathbf{s}_i||_0 \leq c \tag{4}$$

where the sparse representations $\mathbf{s}_i$ are the columns of $\mathbf{S} \in \mathbb{R}^{k \times m}$ and c is the maximum number of atoms we allow in the sparse representation. The optimization problem is both non-smooth and non-convex in $(\mathbf{D}, \mathbf{S})$ due to the sparsity constraint, but [1] is proven to find a stationary point.

Learned dictionaries have proven to match or exceed the state of the art in problems such as image compression, de-noising, and in-painting, which have traditionally been the domains of fixed bases such as wavelets, edgelets, noiselets, etc. To demonstrate the power of learned dictionaries for texture classification, we implement K-SVD to learn dictionaries for different texture classes, then run a simple classification method on test sets of 10 and 30 textures from the Normalized Brodatz Texture Dataset ([12]), a widely used benchmark for texture processing algorithms.

3. Texture Classification

3.1. Past Work

The state of the art in texture classification involves a trade-off between complex but much slower neural networks ([4], [6]), which provide higher classification rates for more texture classes, and simpler SVM-based methods ([9]), which provide somewhat lower classification rates but are much faster. dARTEX, the elaborate neural architecture developed in [4], models how the visual cortex discriminates textures and involves layers for surface-based texture classification, contour-based edge grouping, surface filling-in, spatial attention, and object attention. [6], another neural architecture, uses Gabor filtering over multiple scales for its feature extraction and improves on dARTEX's classification rates slightly. On the other end of the spectrum, [9] presents a multi-class SVM that cannot match the performance of [4] and [6], but still offers classification rates in the mid-90s on the same Brodatz test sets and is far simpler and faster.

We will compare our classification results against [4], [5], [6], and [9] on the same sets of 10 and 30 Brodatz textures. Though our learned dictionary model cannot beat these methods, we highlight that its novelty lies in its simplicity and intuitiveness. Simpler than a multi-class SVM, which must tackle the problem of selecting the right kernel and feature representation, we can train our model (i.e. run K-SVD) within about 10 seconds per texture class and run a fast greedy method to classify each new texture within about a second. A learned dictionary is an intuitive extrapolation of k-means clustering, where each observation belongs to multiple clusters (atoms) with varying degrees of attachment (given by the sparse coefficients), rather than each observation being assigned to just one cluster. Along with this aesthetic intuitiveness, our model also reaches classification rates in the mid-90s. Unlike [6] and [9], which use fixed dictionariesmulti-scale Gabor filter banks and discrete wavelet transforms, respectivelyto extract features, our learned dictionary model performs adequately despite its simplicity because it is explicitly designed to adapt to each texture class.

3.2. Learned Dictionary Models

4. Methods

4.1. K-SVD: Dictionary Learning

To learn a dictionary for a texture class, we train K-SVD as described in [1] (reproduced in Fig. 1) on 1000 16×16-pixel patches extracted from an image of the texture. For the "Sparse Coding Stage" of K-SVD, we use OMP as described in [13] (reproduced in Fig. 2).

The two key K-SVD parameters to choose are c in Eq. 4, the number of atoms permitted in the sparse representation, and k, the number of atoms in the dictionary $\mathbf{D}$. When $c = 1$, K-SVD reduces to the special case of k-means clustering; when $c > 1$, a texture patch can be expressed as a combination of multiple underlying patterns. Similarly, the redundancy ratio k/n can tune the descriptive power of the model: the more redundant the dictionary, the more complex patterns that can be captured in a sparse representation. Small k/n, however, creates highly compact models that can discriminate between different textures more easily. We will show classification results across a range of values for both k and c.

Task: Find the best dictionary to represent the data samples $\{\mathbf{y}_i\}_{i=1}^N$ as sparse compositions, by solving

$$\min_{\mathbf{D},\mathbf{X}} \{\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2\} \quad \text{subject to} \quad \forall i,\ \|\mathbf{x}_i\|_0 \le T_0.$$

Initialization : Set the dictionary matrix $\mathbf{D}^{(0)} \in \mathbf{R}^{n \times K}$ with ℓ^2 normalized columns. Set $J = 1$.
Repeat until convergence (stopping rule):

- *Sparse Coding Stage*: Use any pursuit algorithm to compute the representation vectors $\mathbf{x}_i$ for each example $\mathbf{y}_i$, by approximating the solution of

$$i = 1, 2, \ldots, N, \quad \min_{\mathbf{x}_i} \{\|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2^2\} \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \le T_0.$$

- *Codebook Update Stage*: For each column $k = 1, 2, \ldots, K$ in $\mathbf{D}^{(J-1)}$, update it by
  - Define the group of examples that use this atom, $\omega_k = \{i | 1 \le i \le N,\ \mathbf{x}_T^k(i) \ne 0\}$.
  - Compute the overall representation error matrix, $\mathbf{E}_k$, by

$$\mathbf{E}_k = \mathbf{Y} - \sum_{j \ne k} \mathbf{d}_j \mathbf{x}_T^j.$$

  - Restrict $\mathbf{E}_k$ by choosing only the columns corresponding to ω_k, and obtain $\mathbf{E}_k^R$.
  - Apply SVD decomposition $\mathbf{E}_k^R = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T$. Choose the updated dictionary column $\tilde{\mathbf{d}}_k$ to be the first column of $\mathbf{U}$. Update the coefficient vector $\mathbf{x}_R^k$ to be the first column of $\mathbf{V}$ multiplied by $\mathbf{\Delta}(1, 1)$.
- Set $J = J + 1$.

Figure 1. The K-SVD algorithm (Fig. 2 in [1]).

INPUT:
- An $N \times d$ measurement matrix $\boldsymbol{\Phi}$
- An N-dimensional data vector $\boldsymbol{v}$
- The sparsity level m of the ideal signal

OUTPUT:
- An estimate $\widehat{\boldsymbol{s}}$ in $\mathbb{R}^d$ for the ideal signal
- A set Λ_m containing m elements from $\{1, \ldots, d\}$
- An N-dimensional approximation $\boldsymbol{a}_m$ of the data $\boldsymbol{v}$
- An N-dimensional residual $\boldsymbol{r}_m = \boldsymbol{v} - \boldsymbol{a}_m$

PROCEDURE:
1) Initialize the residual $\boldsymbol{r}_0 = \boldsymbol{v}$, the index set $\Lambda_0 = \emptyset$, and the iteration counter $t = 1$.
2) Find the index λ_t that solves the easy optimization problem

$$\lambda_t = \arg\max_{j=1,\ldots,d} |\langle \boldsymbol{r}_{t-1}, \boldsymbol{\varphi}_j \rangle| .$$

If the maximum occurs for multiple indices, break the tie deterministically.
3) Augment the index set and the matrix of chosen atoms: $\Lambda_t = \Lambda_{t-1} \cup \{\lambda_t\}$ and $\boldsymbol{\Phi}_t = [\, \boldsymbol{\Phi}_{t-1} \quad \boldsymbol{\varphi}_{\lambda_t} \,]$. We use the convention that $\boldsymbol{\Phi}_0$ is an empty matrix.
4) Solve a least squares problem to obtain a new signal estimate:

$$\boldsymbol{x}_t = \arg\min_{\boldsymbol{x}} \|\boldsymbol{v} - \boldsymbol{\Phi}_t \boldsymbol{x}\|_2 .$$

5) Calculate the new approximation of the data and the new residual

$$\boldsymbol{a}_t = \boldsymbol{\Phi}_t \boldsymbol{x}_t$$
$$\boldsymbol{r}_t = \boldsymbol{v} - \boldsymbol{a}_t .$$

6) Increment t, and return to Step 2 if $t < m$.
7) The estimate $\widehat{\boldsymbol{s}}$ for the ideal signal has nonzero indices at the components listed in Λ_m. The value of the estimate $\widehat{\boldsymbol{s}}$ in component λ_j equals the jth component of $\boldsymbol{x}_t$.

Figure 2. The OMP algorithm (Alg. 3 in [13]).

4.2. Classification on the Normalized Brodatz Texture Dataset

We train and test our model on the same set of 10 Brodatz textures used in [4] and [6] (listed on the left-hand side of Fig. 11), and the same set of 30 Brodatz textures used in [5], [6], and [9] (listed on the left-hand side of Fig. 13).

To create disjoint training and testing instances per texture, we follow a similar protocol to [6]: each texture is represented in the Brodatz database as a single 640×640-pixel image, which we split into a 3×3-grid of 9 non-overlapping 213×213-pixel sub-images. For each texture, we train K-SVD on 1000 randomly selected 16×16-pixel patches from the central sub-image. We then use the remaining 8 sub-images as test images, so that there are 8 test images per texture. The training and testing pipeline proceeds as follows:

- **Inputs**: training patches for each texture class, K-SVD parameters k and c, a test image
- **Training**: For each texture class, train K-SVD on representative patches of the texture to obtain a learned dictionary $\mathbf{D}$.
- **Testing**: Extract patches from the test image. For each patch $\mathbf{x}$, run OMP to obtain a c-sparse representation $\mathbf{s}$ from each learned dictionary. Classify the test image as the texture whose dictionary provides the smallest average re-projection error $||\mathbf{x} - \mathbf{Ds}||_2$ across all patches.

Fig. 3 gives an example of the random dictionary used to initialize K-SVD. As a demonstration, the algorithm can iteratively adapt these random atoms to learn the distinctive patterns of the Brodatz textures D74 and D106 in Figs. 4 and 7, as seen in Figs. 5, 6, 8, and 9. Note howwithout any prior knowledgethe smooth, curved edges of the coffee bean texture emerge from noise-like atoms in Figs. 5 and 6, and the distinctive stripes of the mesh texture develop in Figs. 8 and 9.

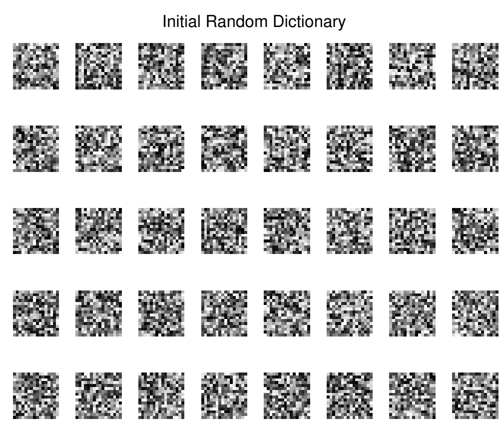

Figure 3. Random dictionary we use to initialize K-SVD.

5. Classification Results

5.1. 10 Brodatz Textures

Fig. 10 demonstrates how varying the K-SVD parameters k, giving the redundancy of the dictionary, and c, the number of atoms chosen in the sparse representations, affect classification rate on the 10 Brodatz

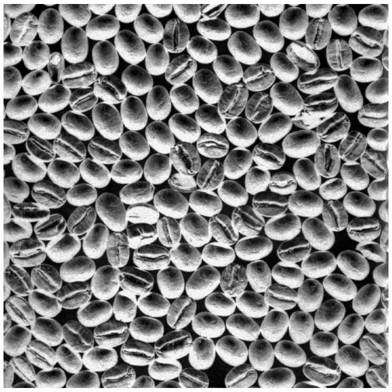

Figure 4. Original Brodatz texture D74 (coffee beans).

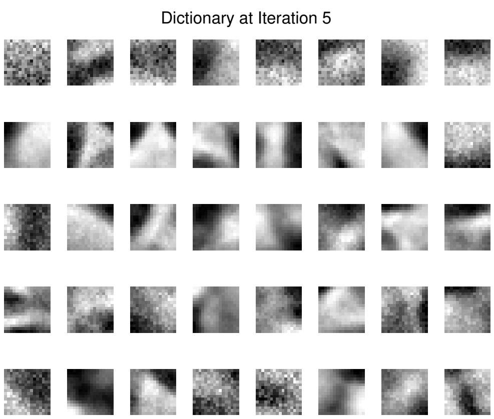

Figure 5. Dictionary for sparse representation of 16×16-pixel patches of Brodatz texture D74 ($k = 40, c = 4$) after 5 iterations of K-SVD.

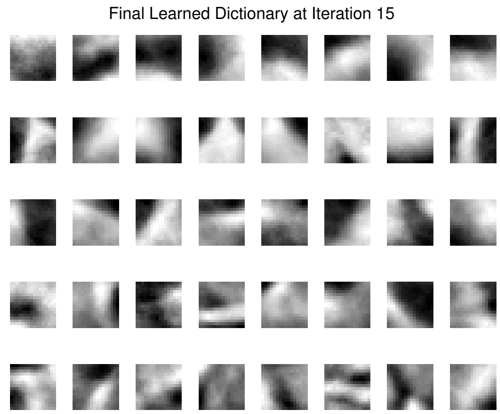

Figure 6. Final learned dictionary for sparse representation of 16×16-pixel patches of Brodatz texture D74 ($k = 40, c = 4$) after 15 iterations of K-SVD.

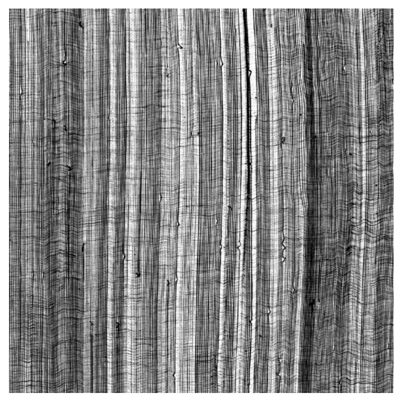

Figure 7. Original Brodatz texture D106 (mesh).

textures. The highest classification rate of 96.25% is obtained with $k = 64, c = 4$. We observe that k needs to be large enough that a dictionary can capture sufficiently elaborate patterns of the texture, but if k is too large, a dictionary starts incorporating patterns induced by noise rather than the texture itself. These spurious patterns provide no discriminative power between different textures, as shown in the “128-Atom Dictionary” curve, which is outperformed by the 64-atom dictionary.

Similarly, we observe that c needs to be large enough that complex textures can be expressed as combinations of multiple patterns. However, if c is too large, OMP starts selecting atoms that are less representative of a texture patch. Again, these irrelevant atoms weaken the discriminatory power of the model: almost all curves decline in performance once c grows too large.

Fig. 11 shows the confusion matrix of the best learned dictionary model on the 10 Brodatz textures, where row label gives the true texture and the column label gives the predicted texture. Our model only misclassifies 3 test images of herringbone, canvas, and jeans as raffia, and perfectly classifies the rest.

Tab. 1 compares the learned dictionary model's classification rate to those of other methods. Though it cannot match their performance, it performs adequately enough to reach the mid-90s, which is satisfying given the model's simplicity.

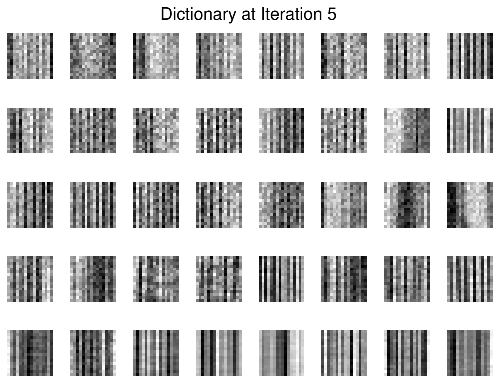

Figure 8. Dictionary for sparse representation of 16×16-pixel patches of Brodatz texture D106 ($k = 40, c = 4$) after 5 iterations of K-SVD.

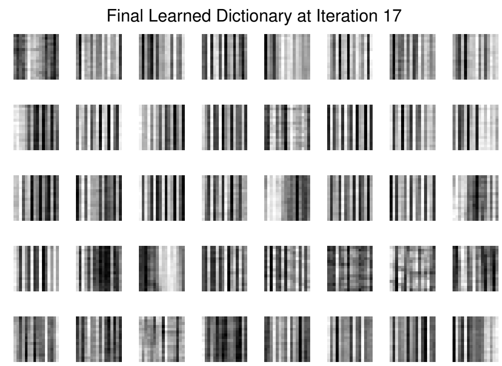

Figure 9. Final learned dictionary for sparse representation of 16×16-pixel patches of Brodatz texture D106 ($k = 40, c = 4$) after 15 iterations of K-SVD.

Classification Method	Classification Rate
Dictionary Learning Model	**0.9625**
Bhatt et al. [4]	**0.9810**
Diaz-Pernas et al. [6]	**0.9956**

Table 1. Comparison of the texture classification methods in [4] ("1 Texture/Image with attention"), [6], and our learned dictionary model ($k = 64, c = 4$) on 10 Brodatz textures.

Classification Method	Classification Rate
Dictionary Learning Model	**0.9250**
Chen et al. [5]	**0.9572**
Li et al. [9]	**0.9634**
Diaz-Pernas et al. [6]	**0.9890**

Table 2. Comparison of the texture classification methods in [5], [6], [9], and our learned dictionary model ($k = 32, c = 2$) on 30 Brodatz textures.

5.2. 30 Brodatz Textures

Fig. 12, Fig. 13, and Tab. 2 provide analogous information to Fig. 10, Fig. 11, and Tab. 1 on the larger set of 30 Brodatz textures.

We observe similar trends: Fig. 12 reveals it is important that k and c are tuned to be not too large, yet not too small, and Tab. 2 shows that though we cannot compete with more complex methods, we can still reach classification rates in the 90s. Fig. 13 reveals that much of our mis-classification occurs between 3 different sand textures, which is reasonable given the inherent similarity between sand textures.

6. Conclusion

We have developed a simple, novel model for texture classification, based on learned dictionaries for sparse representations of textures. By tuning the parameters k and c of the dictionary learning algorithm K-SVD, which control the redundancy of the dictionary and the sparsity of the representations, we can control the discriminative power of our model. We have evaluated the model on sets of 10 and 30 textures from the Brodatz database, a widely used benchmark for texture processing algorithms. Though it cannot match the performance of much more complex methods, it can still reach classification rates in the mid-90s. Given the simplicity and intuitiveness of the model and classification method, we consider this a success.

There are many extensions that can be explored for the learned dictionary model. Learning dictionaries at different scales of the textures, for example, could simulate the multi-scale feature extraction highlighted in [6]. We also intend to replace K-SVD with a hierarchical learning algorithm such as [3], which will provide dictionary models invariant to rotation and scaling.

References

[1] Aharon, M., Elad, M., & Bruckstein, A. (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Sig. Proc.*, 54:4311-4322.

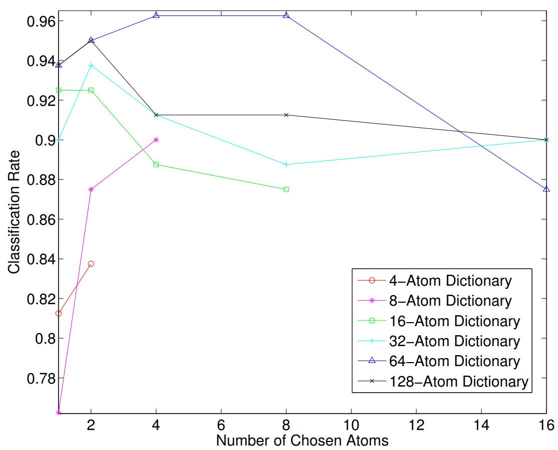

Figure 10. Classification rate of the learned dictionary model on 10 Brodatz textures for $k \in \{4, 8, 16, 32, 64, 128\}$ and $c \in \{1, 2, 4, 8, 16\}$.

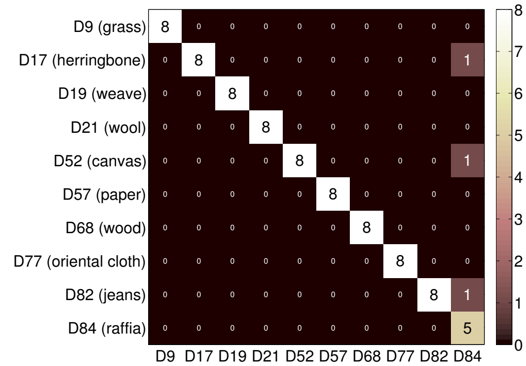

Figure 11. Confusion matrix of the best learned dictionary model ($k = 64, c = 4$) on 10 Brodatz textures, where there are 8 test images per texture class.

[2] Atick, J. J. & Redlich, A. N. (1992). What does the retina know about natural scenes? *Neural Computation*, 2:308-320.

[3] Bar, L. & Sapiro, G. (2010). Hierarchical dictionary learning for invariant classification. *ICASSP 2010*.

[4] Bhatt, R., Carpenter, G., & Grossberg, S. (2007). Texture segregation by visual cortex: Perceptual grouping, attention, and learning. *Vision Research* 47(25): 3173-3211.

[5] Chen, X.W., Zeng, X., & van Alphen, D. (2006). Multi-class feature selection for texture classification. *Pattern Recognition Letters*, 27(14): 1685-1691.

[6] Diaz-Pernas, F. J. et al. (2009). Texture classification of the entire Brodatz database through an orientational-invariant neural architecture. *IWINAC 2009*, 294-303.

[7] Engan, K., Rao, B. D., & Kreutz-Delgado, K. (1999). Method of optimal directions for frame design. *IEEE Int. Conf. Acoust., Speech, Signal Process.* 5:2443-2446.

[8] Field, D.J. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Opt. Soc. of Am. A*, 4:2379:2394.

[9] Li, S., Kwok, J.T., Zhu, H., & Wang, Y. (2003). Texture classification using the support vector machines. *Pattern Recognition* 36(12): 2883-2893.

[10] Olshausen, B.A. & Field., D.J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Res.*, 37:3311-3325

[11] Peyré, G.(2009). Sparse modeling of textures. *Journal of Mathematical Imaging and Vision*, 34:17-31.

[12] Safia, A. & He, D. (2013). New Brodatz-based Image Databases for Grayscale Color and Multiband Texture Analysis. *ISRN Machine Vision*, 2013: Article ID 876386.

[13] Tropp, J.A., Gilbert, A.C. (2007). Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inf. Theory*, 53:4655-4666.

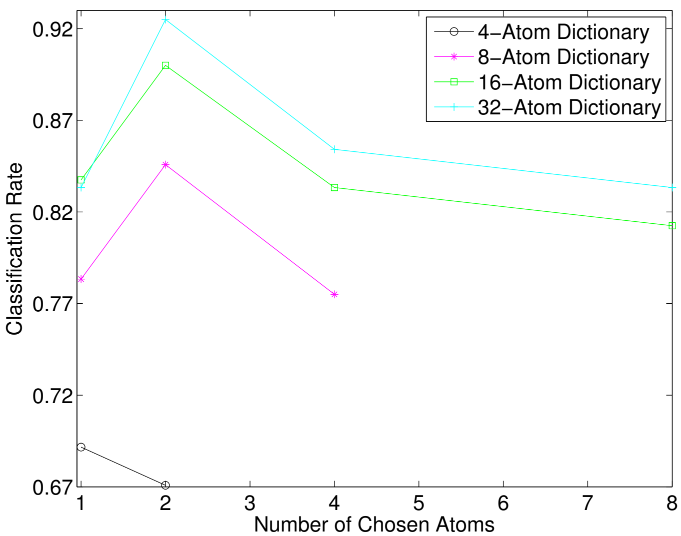

Figure 12. Classification rate of the learned dictionary model on 30 Brodatz textures for $k \in \{4, 8, 16, 32\}$ and $c \in \{1, 2, 4, 8\}$.

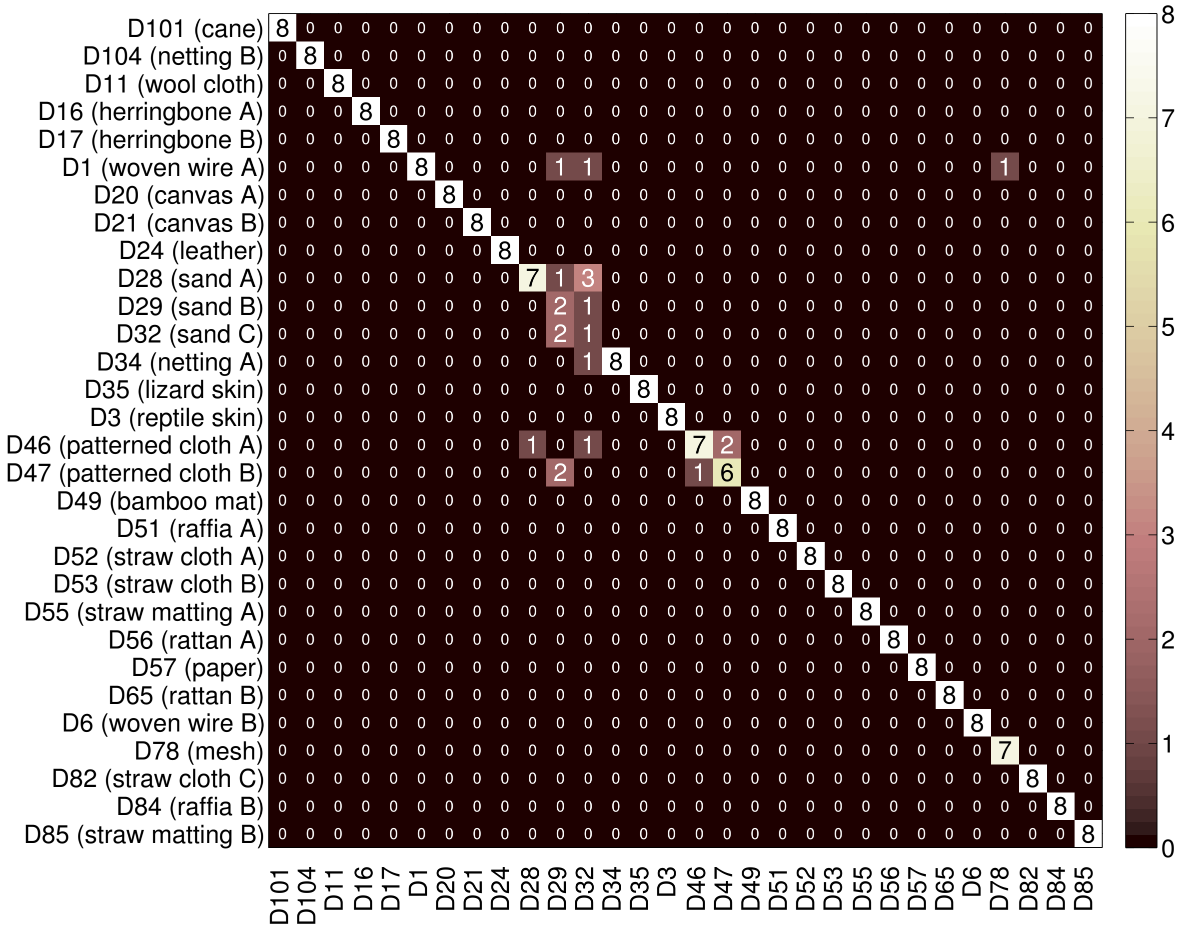

Figure 13. Confusion matrix of the best learned dictionary model ($k = 32, c = 2$) on 30 Brodatz textures, where there are 8 test images per texture class.