Learning Discriminative Hierarchical Features for Object Recognition

Zhen Zuo, and Gang Wang

Abstract

Hierarchical feature learning methods have demonstrated substantial improvements over the conventional hand-designed local features. However, recent approaches mainly perform feature learning in an unsupervised manner, where subtle differences between different classes can hardly be captured. In this letter, we propose a discriminative hierarchical feature learning method, which learns a non-linear transformation to encode discriminative information in the feature space. We apply our features on two general image classification benchmarks: Caltech 101, STL-10, and a new fine-grained image classification dataset: NTU Tree-51. The results show that by employing discriminative constraint, our method consistently improves the performance with 3% to 7% in classification accuracy.

Index Terms

Discriminant analysis, hierarchical feature learning, patch-to-class distance, object recognition.

I. Introduction

Feature representation is a critical component of a modern visual recognition system. Numerous works have been done to develop advanced hand-crafted feature descriptors, famous works include SIFT [1], HOG [2], etc. Although such descriptors can lead to good performance, they might not be able to capture the essential information hidden in the data. In contrast, feature learning has shown great advantages in learning data-adaptive image representation. Especially recently, deep learning techniques, such as

The authors are with the School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Gang Wang is also with Advanced Digital Sciences Center, University of Illinois, Singapore 138632 (e-mail: zzuo1@e.ntu.edu.sg; wanggang@ntu.edu.sg).

This research is supported by the Singapore National Research Foundation under its IDM Futures Funding Initiative and administered by the Interactive & Digital Media Programme Office, Media Development Authority.

Auto-Encoders [3], and Hierarchical Spatial-Temporal Feature [4], have achieved great success on many challenging research problems.

However, most existing feature learning methods process in an unsupervised manner, which might miss discriminative information, and limit the representation capability. In this letter, we propose a discriminative information encoding method to improve the discriminative power of the learned features. Specifically, we assume the local image patches contain the class specific information. Based on this assumption, we assign all the local image patches the same class labels as the images they extracted from. To get more discriminative local image representation, we aim to learn such a feature space, in which, the feature patches (transformed image patches) from the same class are close together, while the feature patches from different classes are separable from each other. However, local patches from the same class can be highly diverse. Simply forcing all the feature patches from the same class to be close will bring too much noise. Instead, a feature patch only needs to be close to a small subset of the patches from the same class. Thus, we introduce the 'Patch-to-Class' distance (P2CD) (inspired by the 'Image-to-Class' distance proposed in Naive Bayes Nearest Neighbour [5]) to directly measure the distance between each feature patch and its nearest neighbour patches from difference classes. As shown in Figure 1, this framework forces the training feature patches to be close to their corresponding classes (positive), while to be far away from other classes (negative), which means shortening P2CD(q, NN_p) while elongating P2CD(q, NN_n).

In this letter, we build a discriminative hierarchical feature learning framework based on the hierarchical Reconstruction Independent Component Analysis (RICA) structure [4], [6], [7] (our method can also be applied in other feature learning frameworks involving learning transformation matrix). As shown in the orange box on the left of Figure 2, the first layer features are learned through small input image patches (yellow boxes), then they are convolved with a larger region (red box) to generate the inputs to the second layer. The final features are the combination of outputs of both layers. Since we focus on learning discriminative multi-layer local features, we simply follow the Bag-of-Words (BoW) to get global image representation, and use linear SVM to do classification. Our overall object recognition pipeline is shown in Figure 2.

II. Discriminative hierarchical feature learning

We aim to learn a transformation matrix to transform the local image patches from the original image space to the discriminative feature space. Recently, RICA has shown its power in several challenging image and video recognition tasks. Thus, we build our discriminative hierarchical feature learning scheme

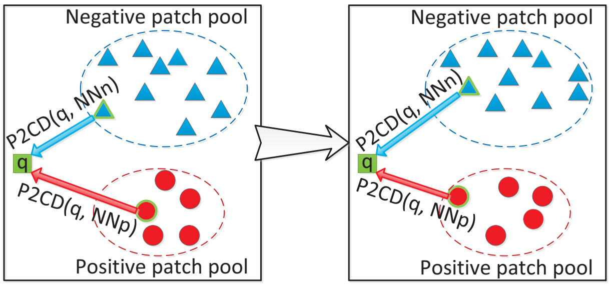


Fig. 1. Schematic diagram of our discriminative information encoding scheme. The square in green denotes the training patch; the circles in red denote patches from the same category (positive); the triangles in blue denote patches from other categories (negative). This framework aims to reduce P2CD(q, NN_p), while elongate P2CD(q, NN_n).

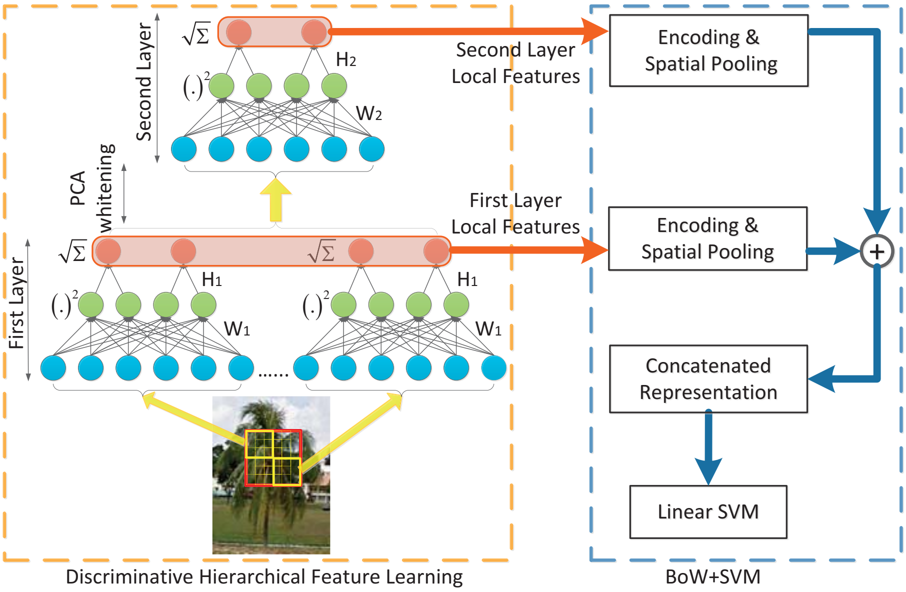


Fig. 2. Illustration of our object recognition pipeline. (Best viewed in color.)

based on the RICA structure, but our method can also be applied in other feature learning frameworks, such as Convolutional Neural Networks [8].

A. Basic Single Layer RICA Learning Module

The basic single layer RICA learning module [7] consists of a linear auto-encoder term and a non-linear term, as shown in Figure 2. We set the non-linear function in an ‘energy pooling’ manner. Given $x \in \mathbb{R}^d$ as a raw-pixel value training patch with dimensionality d, the RICA 1) uses the matrix $W \in \mathbb{R}^{d\times d}$ to linearly transform the input data into Wx; 2) applies energy pooling [9] to represent the subspace structure of Wx, and get the non-linear transformed feature vector $q \in \mathbb{R}^{d/2}$:

$$q = \sqrt{H(Wx)^2} \tag{1}$$

where the square of Wx and the square root of $H(Wx)^2$ are processed element-wisely. $H \in \mathbb{R}^{d/2 \times d}$ is the subspace pooling matrix used to reduce feature dimension, each row of H selects and sums two adjacent feature dimensions without overlapping. While W can be learned by minimizing the following equation:

$$\sum\nolimits_{j=1}^{N} \left(\left\| x^j - W^T W x^j \right\|_2^2 + \gamma \sum\nolimits_{r=1}^{d/2} q_r^j \right) \tag{2}$$

in which, x^j and q^j denote the jth training patch in the image space and feature space respectively, and N denotes the number of training patches. The first 'auto-encoder' term is used to prevent the bases of W from degenerating. The second 'sparse' term is used to ensure the sparsity of the learned feature descriptors.

B. Discriminative Hierarchical Feature Learning

1) Single Layer Discriminative Feature Learning: The basic learning method described in SectionII-A can hardly capture discriminative information hidden in different classes. Ideally, for classification, we expect the learned features to be close to the features from the same class, while to be far away from the features from other classes. Thus, we propose the following framework to encode discriminative information.

In our discriminative learning scheme, we aim to maximize the following function for a training feature patch q:

$$\frac{P\left(c|q\right)}{P\left(\bar{c}|q\right)} = \frac{P\left(q|c\right) \cdot P\left(c\right)}{P\left(q|\bar{c}\right) \cdot P\left(\bar{c}\right)} \tag{3}$$

where $\bar{c}$ denotes all the classes except class c, and q is from c.

Assuming the class priors are equal, then the posteriors are equal to the likelihoods, which can be approximated by applying the Parzen window estimator as described in [5]:

$$\hat{P}\left(q|c\right) = \exp\left(-\frac{1}{2\sigma^2} \left\| q - NN_c\left(q\right) \right\|^2 \right) \tag{4}$$

in which, $NN_c\left(q\right)$ is the nearest neighbour belonging to class c of the training patch q in the feature space. If we further take the log probability and ignore the constant, we can rewrite the right-hand side of Equation 4 as $-\left\| q - NN_c\left(q\right) \right\|^2$, which can be considered as the negative 'Patch-to-Class' distance

(P2CD). Then Equation 3 can be written in a simplified form:

$$\log \frac{P\left(q|c\right)}{P\left(q|\bar{c}\right)} = -\left\|q{-}NN_c\left(q\right)\right\|^2 + \left\|q{-}NN_{\bar{c}}\left(q\right)\right\|^2 \tag{5}$$

where $NN_{\bar{c}}\left(q\right)$ is the nearest neighbour of q in the feature space, and it is from classes other than c. Based on our discriminative learning method, we get the representation of the single layer learning module:

$$\begin{gathered}\min_W E_u + \eta E_s \\ \text{where, } E_u{=}\sum\nolimits_{j=1}^{N}\left(\left\|x^j - W^T W x^j\right\|_2^2 + \gamma \sum\nolimits_{r=1}^{d/2} q_r^j\right) \\ E_s{=}\sum\nolimits_{j=1}^{N}\left(\left\|q^j - NN_c\left(q^j\right)\right\|_2^2 - \left\|q^j - NN_{\bar{c}}\left(q^j\right)\right\|_2^2\right)\end{gathered} \tag{6}$$

where E_u represents the unsupervised term used to enforce low reconstruction error and sparsity, E_s represents the supervised discriminative constraint. γ and η are the tradeoff parameters used to control the level of sparsity and discriminative power.

We adopt the gradient descent to optimize the object function 6, and the gradients can be computed as follows:

$$\begin{aligned}\frac{\partial E_u}{\partial W_{mn}} &{=}\sum_{j=1}^{N}\frac{\partial\left\|x^j{-}W^T W x^j\right\|_2^2}{\partial W_{mn}} + \gamma\sum_{j=1}^{N}\sum_{r=1}^{d/2}\frac{\partial q_r^j}{\partial W_{mn}} \\ &{=}\sum\nolimits_{j=1}^{N}\frac{\partial E_{ae}}{\partial W_{mn}} + \gamma\sum\nolimits_{j=1}^{N}\sum\nolimits_{r=1}^{d/2}\frac{\partial E_{sparse}}{\partial W_{mn}} \\ \frac{\partial E_s}{\partial W_{mn}} &{=}\sum_{j=1}^{N}\frac{\partial\left\|q^j{-}NN_c\left(q^j\right)\right\|_2^2}{\partial W_{mn}} {-}\sum_{j=1}^{N}\frac{\partial\left\|q^j{-}NN_{\bar{c}}\left(q^j\right)\right\|_2^2}{\partial W_{mn}} \\ &{=}\sum\nolimits_{j=1}^{N}\frac{\partial E_{pos}}{\partial W_{mn}} - \sum\nolimits_{j=1}^{N}\frac{\partial E_{neg}}{\partial W_{mn}} \\ \text{where } \frac{\partial E_{sparse}}{\partial W_{mn}} &{=}H_{rm}\left(W_m x^j\right)x_n^j/\sqrt{H_r(Wx^j)^2} \\ \frac{\partial E_{ae}}{\partial W_{mn}} &{=}{-}4W_m x^j x_n^j{+}Tr\left[\left[2W^T W\left(x^j\left(x^j\right)^T\right)\right]^T\left(W^T J^{mn}{+}J^{mn}W\right)\right] \\ \frac{\partial E_{pos}}{\partial W_{mn}} &{=}2\left(q^j - NN_c\left(q^j\right)\right)^T\left(\frac{\partial q^j}{\partial W_{mn}} - \frac{\partial NN_c\left(q^j\right)}{\partial W_{mn}}\right) \\ \frac{\partial q^j}{\partial W_{mn}} &{=}H_m\left(W_m x^j\right)x_n^j/\sqrt{H(Wx^j)^2}, \quad \left(J^{mn}\right)_{kl}{=}\delta_{mk}\delta_{nl}\end{aligned} \tag{7}$$

where $\partial E_{neg}/\partial W_{mn}$ has the same form as $\partial E_{pos}/\partial W_{mn}$, $\partial NN_c\left(q^j\right)/\partial W_{mn}$ and $\partial NN_{\bar{c}}\left(q^j\right)/\partial W_{mn}$ have the same form as $\partial q^j/\partial W_{mn}$. The transformation matrix W can be updated with step size α until convergence: $W = W - \alpha \times (\partial E_u/\partial W + \eta \partial E_s/\partial W)$.

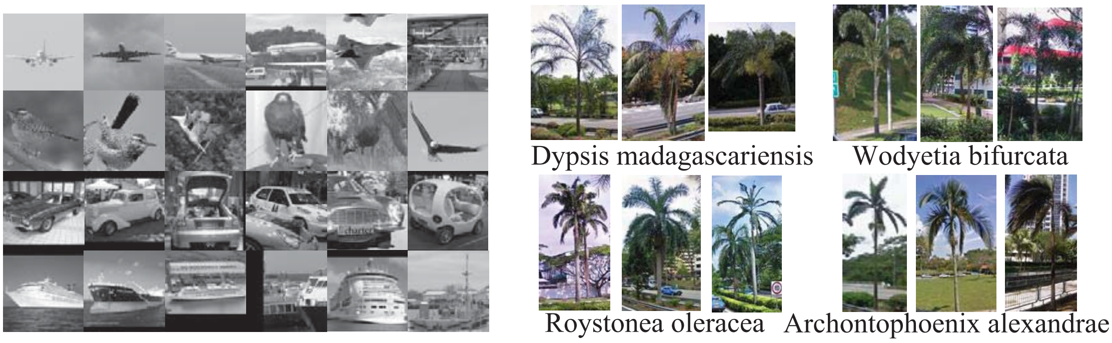
2) Hierarchical Learning Structure: Though the single layer feature learning module can achieve good performance, it still has some limitations. For example, it is not able to share statistical information among different local features, and it cannot extract information from multiple visual levels. To get higher level visual representations that can not only tolerate non-trivial transformations in small local areas, but also capture contextual information of the first layer features, we leverage a multi-layer scheme to learn hierarchical features. In this letter, we adopt a two-layer framework: in the first layer, feature learning is performed on small image areas (16x16 image patches) to extract the first layer discriminative features; while in the second layer, higher level image representation is learned from bigger image areas (32x32 image patches) to get the second layer discriminative features.

To get the second layer inputs, convolution is applied on the first layer outputs at multiple grid locations to get a highly over-complete set of first layer features in the 32x32 image areas. Concatenating these first layer features will generate a high-dimensional representation, which cannot be efficiently processed. Thus, PCA is applied for dimension reduction and data whitening, the output of which is the input to the second layer. We get the final feature descriptors by concatenating the output features of both layers as shown in Figure 2.

3) Approximation: In each layer, it's time consuming to search nearest neighbour from a large collection of image patches. Especially in our case, the membership of nearest neighbours $NN(q)$ of each training patch x change when the feature transformation matrix W updates. For simplification, firstly, we fix $NN(q)$ and update W until W converges to a suboptimal value. Secondly, we search nearest neighbours in the learned suboptimal feature space, and renew the membership of nearest neighbours $NN(q)$. We iterate these two steps for several times. In our experiments, when the number of iterations increases, the performance slightly increases. For efficiency consideration, we merely apply one iteration. We initialize W as the matrix learned by the basic learning module without discriminative term, and then search for $NN(q)$ based on this W, and update W afterwards. Furthermore, to speed up the procedure of nearest neighbour search, we use FLANN [10], which is a library making use of multiple randomized k-d trees to achieve fast NN approximation.

III. Experiments and analysis

We test our discriminative hierarchical feature learning algorithm on two general image classification benchmarks: Caltech-101, STL-10, and one new fine-grained image classification dataset we collected: NTU Tree-51. To make a better comparison with other methods, we only use gray-scale images. In layer



(a) Examples of STL-10

(b) Examples of NTU Tree-51

Fig. 3. (a) Examples from STL-10 dataset. (b) Examples from four different tree species of NTU Tree-51 dataset.

1, we learn on 16x16 patches, and get 128 dimensional first layer features. In layer 2, based on the learned first layer features, we convolve them within the larger 32x32 patches with a stride of 2. We concatenate the responses and get 10,368 dimensional data, then PCA is applied to produce 300 dimensional input data for the second layer. After processing layer 2, we get 150 dimensional second layer features. Since this letter focuses on local feature learning, we simply employ the most general settings in the BoW framework. We use Spatial Pyramid Matching [11] to get the global image representation, and linear SVM as classifier.

To ensure the accuracy of nearest neighbour search, we densely extract patches from the training images to build the positive and negative patch pool. Additionally, we discard patches extracted from the low contrast areas for denoising. Specifically, for each class, we densely extract 1000-2000 patches per training image for the first layer, and 200-400 patches for the second layer to build the positive patch pool. Meanwhile, we randomly select patches from images belonging to negative categories, and build the negative patch pool with 10 times the size of the positive pool. Finally, we randomly select 10% of the patches from each class-specific positive patch pool as the training patches, and learn our features based on 100,000-500,000 training patches in the first layer, and 30,000-100,000 training patches in the second layer.

A. Results

Caltech-101 [12] has images from 101 different object classes with high intra-class variance. There are 31 to 800 images in each class. Following the most general settings, we randomly select 30 images per class for training, and 50 images for testing. We resize all the images to 150x150 pixels. The numerical results of our method and other algorithms are reported in Table I. We test the hierarchical RICA with training patches randomly extracted from training images, and get 66.7% in accuracy as the baseline. With our discriminative learning method, the accuracy can be improved to 73.0%, which is 6.3% higher than the baseline.

Algorithm	Acc.
SPM [11]	64.6%
Hierarchical RICA [4], [7]	66.7%
NBNN [5]	70.4%
local NBNN [13]	71.9%
LLC [14]	73.4%
Hierarchical SC [15]	74.0%
CRBM [16]	77.8%
Ours (first layer)	66.3%
Ours (two layers)	**73.0**%

TABLE I

RESULTS ON CALTECH-101

Algorithm	Acc.
K-means (Triangle) [17]	51.5%
RICA [7]	52.9%
Hierarchical RICA [4], [7]	54.4%
Sum-Product networks[18]	62.3%
Ours (first layer)	53.3%
Ours (two layers)	**56.7**%

TABLE II

RESULTS ON STL-10

Algorithm	Acc.
SPM [11]	69.6%
Hierarchical RICA [4]	70.9%
LLC [14]	75.3%
Ours (first layer)	74.9%
Ours (two layers)	**78.1**%

TABLE III

RESULTS ON NTU TREE-51

STL-10 [17] is a newly proposed challenging dataset for deep learning networks, which contains 96x96 pixel images from 10 classes as shown in Figure 3(a). We only use the provided labelled data: 5,000 images for training and 8,000 images for testing. The training sets are predefined in 10 folders, where each folder contains 1,000 training images. According to the testing protocol, we train our method on the pre-defined folders, and use the average results as the final testing accuracy. The results are shown in Table II. As this dataset is very challenging, the accuracy is relatively low, but we can still get 2.3% improvement compared with the baseline, and achieve 56.7% in accuracy.

NTU Tree-51 is a fine-grained image dataset we collected, it aims to recognize trees at a distance. All the images were cropped from Google Street View images, which were captured continuously from a distance on a moving vehicle. This dataset contains 2613 street view tree images in total, which is composed of images of 51 common tree species in Singapore, and each species contains 30-70 samples.

This dataset is challenging because of its large intra class variance, and relatively small inter class variance. Image samples of the dataset are shown in Figure 3(b). We resize all the images to 150x150 pixels, for each species, we use 20 images for training, and the rest for testing. The numerical results are shown in Table III. With our discriminative term, we can improve the baseline with 7.2% in accuracy.

Furthermore, according to the comparison results of using the first layer only and two layers shown in Table I, II, and III, with the stacked second layer, the performance can be significantly improved, thus, the hierarchical structure is crucial.

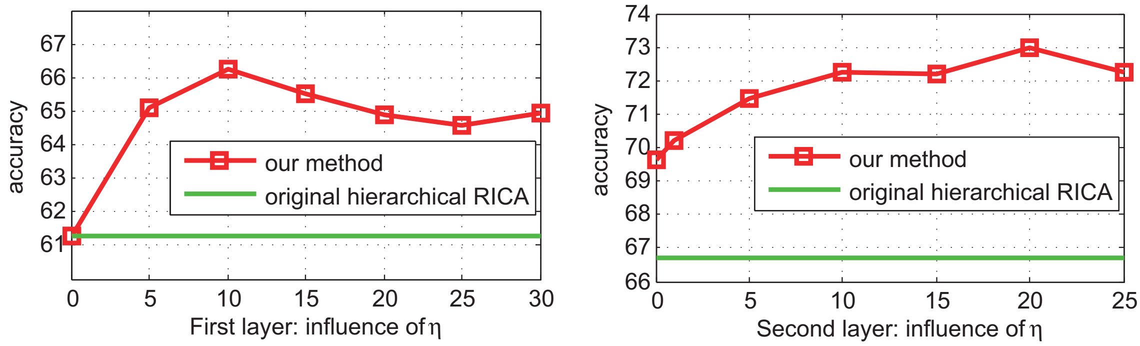
B. Parametric Analysis

We observe the sparse term γ does not bring much influence to the performance, hence we fix it as a constant for all the datasets (50 in the first layer, and 1 in the second layer), and focus on η: weight of the discriminative term in layer 1 and layer 2. In this section, we experimentally investigate how they may affect the performance.

We vary the value of the regularization parameter η in layer 1 and layer 2 separately. By applying cross validation, we get the results as shown in Figure 4. The performance numbers do not change very dramatically. With layer 1 only, as shown in Figure 4(a), when $\eta = 10$, our method can bring 5% improvement compared to the single layer RICA. Figure 4(b) shows the comparison result of our hierarchical discriminative method versus the hierarchical RICA. Here we fix the η in layer 1 as 10. In layer 2, when $\eta = 0$, it corresponds to the result of only applying discriminative learning on layer 1, and using the basic RICA in layer 2. As the accuracy steadily increase when the value of η increase, we can get 3% improvement when $\eta = 20$. This indicates that our discriminative learning method not only improves the performance of the first layer, but also further improves the performance of the second layer. Generally, setting η to 10 for both layers will lead to good performance on all the datasets.

IV. CONCLUSION

In this paper, we proposed a discriminative hierarchical feature learning algorithm, which aims to force the features from the same class to be close, while features from different classes to be separated. We propose P2CD to measure the distance between a feature descriptor and a class. By applying a two-layer discriminative learning method, we obtain a hierarchical feature representation that can not only represent local discriminative features, but also express multiple visual level features with larger receptive fields by applying convolution and stacking. On two general object recognition benchmarks and a new fine-grained image classification dataset, we experimentally show that learning discriminative features significantly



(a) Influence of η in layer 1

(b) Influence of η in layer 2

Fig. 4. Accuracy of our method on Caltech-101 versus the weight of the discriminative term. Green lines represent the accuracy results of applying the hierarchical RICA, while the red lines indicate the accuracy of our discriminative hierarchical method.

improve the performance. In the future, we will explore information in higher visual levels, and build hierarchical feature learning framework with more layers.

References

[1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *CVPR*, vol. 1. IEEE, 2005, pp. 886–893.

[3] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[4] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, “Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis,” in *CVPR*. IEEE, 2011, pp. 3361–3368.

[5] O. Boiman, E. Shechtman, and M. Irani, “In defense of nearest-neighbor based image classification,” in *CVPR*. IEEE, 2008, pp. 1–8.

[6] W. Y. Zou, S. Y. Zhu, A. Y. Ng, and K. Yu, “Deep learning of invariant features via simulated fixations in video,” in *NIPS*, 2012, pp. 3212–3220.

[7] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng, “Ica with reconstruction cost for efficient overcomplete feature learning,” in *NIPS*, 2011, pp. 1017–1025.

[8] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” in *ICML*. ACM, 2009, pp. 609–616.

[9] A. Hyvärinen, J. Hurri, and P. Hoyer, *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision*. Springer, 2009.

[10] M. Muja and D. G. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration,” in *VISSAPP*, 2009, pp. 331–340.

[11] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *CVPR*, vol. 2. IEEE, 2006, pp. 2169–2178.

[12] L. Fei-Fei, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories,” *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[13] S. McCann and D. G. Lowe, “Local naive bayes nearest neighbor for image classification,” in *CVPR*. IEEE, 2012, pp. 3650–3656.

[14] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, “Locality-constrained linear coding for image classification,” in *CVPR*. IEEE, 2010, pp. 3360–3367.

[15] K. Yu, Y. Lin, and J. Lafferty, “Learning image representations from the pixel level via hierarchical sparse coding,” in *CVPR*. IEEE, 2011, pp. 1713–1720.

[16] K. Sohn, D. Y. Jung, H. Lee, and A. O. Hero, “Efficient learning of sparse, distributed, convolutional feature representations for object recognition,” in *ICCV*. IEEE, 2011, pp. 2643–2650.

[17] A. Coates, H. Lee, and A. Y. Ng, “An analysis of single-layer networks in unsupervised feature learning,” in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 215–223.

[18] R. Gens and P. Domingos, “Discriminative learning of sum-product networks,” in *NIPS*, 2012, pp. 3248–3256.