Multipath Sparse Coding Using Hierarchical Matching Pursuit

Liefeng Bo
ISTC-PC Intel Labs
liefeng.bo@intel.com

Xiaofeng Ren
ISTC-PC Intel Labs
xren@cs.washington.edu

Dieter Fox
University of Washington
fox@cs.washington.edu

Abstract

Complex real-world signals, such as images, contain discriminative structures that differ in many aspects including scale, invariance, and data channel. While progress in deep learning shows the importance of learning features through multiple layers, it is equally important to learn features through multiple paths. We propose Multipath Hierarchical Matching Pursuit (M-HMP), a novel feature learning architecture that combines a collection of hierarchical sparse features for image classification to capture multiple aspects of discriminative structures. Our building blocks are MI-KSVD, a codebook learning algorithm that balances the reconstruction error and the mutual incoherence of the codebook, and batch orthogonal matching pursuit (OMP); we apply them recursively at varying layers and scales. The result is a highly discriminative image representation that leads to large improvements to the state-of-the-art on many standard benchmarks, e.g.*, Caltech-101, Caltech-256, MIT-Scenes, Oxford-IIIT Pet and Caltech-UCSD Bird-200.*

1. Introduction

Images are high dimensional signals that change dramatically under varying scales, viewpoints, lighting conditions, and scene layouts. How to extract features that are robust to these changes is an important question in computer vision, and traditionally people rely on designed features such as SIFT. While SIFT can be understood and generalized as a way to go from pixels to patch descriptors [2], designing good features is a challenging task that requires deep domain knowledge, and it is often difficult to adapt to new settings.

Feature learning is attractive as it exploits the availability of data and avoids the need of feature engineering. Learning features has become increasingly popular and effective for visual recognition. A variety of learning and coding techniques have been proposed and evaluated, such as deep belief nets [12], deep autoencoders [17], deep convolutional neural networks [15], and hierarchical sparse coding [32, 3]. Many are deep learning approaches that learn to push pixels through multiple layers of feature transforms. The recent work on Hierarchical Matching Pursuit [3] is interesting as it is efficient (using Batch Orthogonal Matching Pursuit), recursive (the same computational structure going from pixels to patches, and from patches to images), and outperforms many designed features and algorithms on a variety of recognition benchmarks.

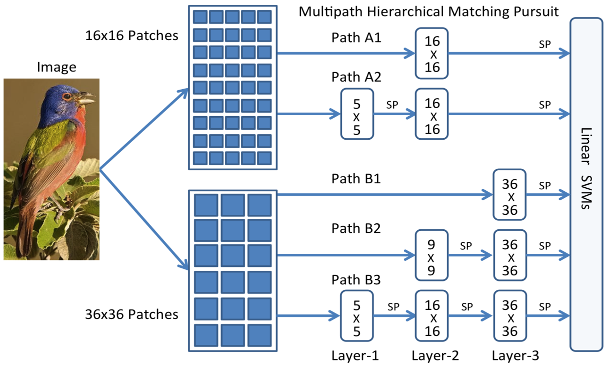

Figure 1: Architecture of multipath sparse coding. Image patches of different sizes (here, 16x16 and 36x36) are encoded via multiple layers of sparse coding. Each path corresponds to a specific patch size and number of layers (numbers inside boxes indicate patch size at the corresponding layer and path). Spatial pooling, indicated by SP, is performed between layers to generate the input features for the next layer. The final layer of each path encodes complete image patches and generates a feature vector for the whole image via another spatial pooling operation. Path features are then concatenated and used by a linear SVM for object recognition.

One crucial problem that is often overlooked in image feature learning is the multi-facet nature of visual structures: discriminative structures, which we want to extract, may appear at varying scales with varying amounts of spatial and appearance invariance. While a generic learning model could capture such heterogeneity, it is much easier to build it into the learning architecture. In this work, we propose Multipath Hierarchical Matching Pursuit (M-HMP), which builds on the single-path Hierarchical Matching Pursuit approach to learn and combine recursive sparse coding through many pathways on multiple bags of patches of varying size, and, most importantly, by encoding each patch through multiple paths with a varying number of layers. See Fig. 1 for an illustration of our system. The multipath architecture is important as it significantly and efficiently expands the richness of image representation and leads to large improvements to the state of the art of image classification, as evaluated on a variety of object and scene recognition benchmarks. Our M-HMP approach is generic and can adapt to new tasks, new sensor data, or new feature learning and coding algorithms.

2. Related Work

In the past few years, a growing amount of research on visual recognition has focused on learning rich features using unsupervised and supervised hierarchical architectures.
Deep Networks: Deep belief nets [12] learn a hierarchy of features, layer by layer, using the unsupervised restricted Boltzmann machine. The learned weights are then further adjusted to the current task using supervised information. To make deep belief nets applicable to full-size images, convolutional deep belief nets [18] use a small receptive field and share the weights between the hidden and visible layers among all locations in an image. Deconvolutional networks [33] convolutionally decompose images in an unsupervised way under a sparsity constraint. By building a hierarchy of such decompositions, robust representations can be built from raw images for image recognition. Deep autoencoders [17] build high-level, class-specific feature detectors from a large collection of unlabeled images, for instance human and cat face detectors. Deep convolutional neural networks [15] won the ImageNet Large Scale Visual Recognition Challenge 2012 and demonstrated their potential for training on large, labeled datasets.
Sparse Coding: For many years, sparse coding [21] has been a popular tool for modeling images. Sparse coding on top of raw patches or SIFT features has achieved state-of-the-art performance on face recognition, texture segmentation [19], and generic object recognition [28, 5, 9]. Very recently, multi-layer sparse coding networks including hierarchical sparse coding [32] and hierarchical matching pursuit [3, 4] have been proposed for building multiple level features from raw sensor data. Such networks learn codebooks at each layer in an unsupervised way such that image patches or pooled features can be represented by a sparse, linear combination of codebook entries. With learned codebooks, feature hierarchies are built from scratch, layer by layer, using sparse codes and spatial pooling [3].

3. Multipath Sparse Coding

This section provides an overview of our Multipath Hierarchical Matching Pursuit (M-HMP) approach. We propose a novel codebook learning algorithm, MI-KSVD, to maintain mutual incoherence of the codebook, and discuss how multi-layer sparse coding hierarchies for images can be built from scratch and how multipath sparse coding helps capture discriminative structures of varying characteristics.

3.1. Codebook Learning with Mutual Incoherence

The key idea of sparse coding is to represent data as sparse linear combinations of codewords selected from a codebook/dictionary [21]. The standard sparse coding approaches learn the codebook $D = [d_1, \cdots, d_m, \cdots, d_M] \in R^{H\times M}$ and the associated sparse codes $X = [x_1, \cdots, x_n, \cdots, x_N] \in R^{M\times N}$ from a matrix $Y = [y_1, \cdots, y_n, \cdots, y_N] \in R^{H\times N}$ of observed data by minimizing the reconstruction error

$$\min_{D,X} \|Y - DX\|_F^2 \tag{1}$$
$$s.t. \ \forall m, \ \|d_m\|_2 = 1 \ \text{and} \ \forall n, \ \|x_n\|_0 \leq K$$

where H, M, and N are the dimensionality of the codewords, the size of the codebook, and the number of training samples, respectively, $\|\cdot\|_F$ denotes the Frobenius norm, the zero-norm $\|\cdot\|_0$ counts non-zero entries in the sparse codes x_n, and K is the sparsity level controlling the number of the non-zero entries.

When sparse coding is applied to object recognition, the data matrix Y consists of raw patches randomly sampled from images. Since the patches frequently observed in images have a higher probability of being included in Y than the ones less frequently observed in images, the learned codebooks may overfit to the frequently observed patches. In order to balance the roles of different types of image patches, it is desirable to maintain large mutual incoherence during the codebook learning phase. On the other hand, theoretical results on sparse coding [7] have also indicated that it is much easier to recover the underlying sparse codes of data when the mutual incoherence of the codebook is large.

This motivates us to balance the reconstruction error and the mutual incoherence of the codebook

$$\min_{D,X} \|Y - DX\|_F^2 + \lambda \sum_{i=1}^{M} \sum_{j=1, j\neq i}^{M} |d_i^\top d_j| \tag{2}$$
$$s.t. \ \forall m, \ \|d_m\|_2 = 1 \ \text{and} \ \forall n, \ \|x_n\|_0 \leq K$$

Here, the mutual coherence $\lambda \sum_{i=1}^{M} \sum_{j=1,j\neq i}^{M} |d_i^\top d_j|$ has been included in the objective function to encourage large mutual incoherence where $\lambda \geq 0$ is a tradeoff parameter.

We propose MI-KSVD to solve the above optimization by adapting the well-known KSVD algorithm [1]. KSVD has led to state-of-the-art results in various image processing and recognition tasks [1, 19, 4]. Like KSVD, MI-KSVD decomposes the above optimization problem (2) into two subproblems, **Encoding** and **Codebook Update**, and solves them in an alternating manner. During each iteration, the current codebook D is used to encode the data Y by computing the sparse code matrix X. Then, the codewords of the

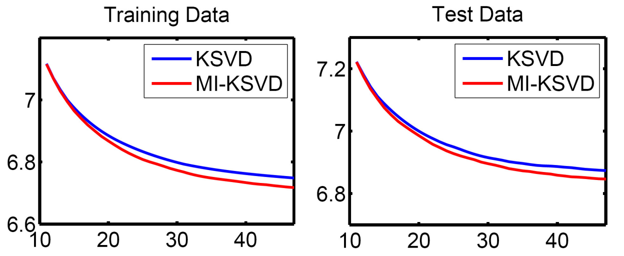

Figure 2: Mean square error as a function of iterations. Training and test data consists of 1,000,000 36x36 image patches and 100,000 36x36 image patches sampled from images, respectively.

codebook are updated one at a time, resulting in a new codebook. This new codebook is then used in the next iteration to recompute the sparse code matrix followed by another round of codebook update. Note that MI-KSVD is quite different from dictionary learning algorithms proposed in [26], which use L_2 norm to measure mutual incoherence and L_1 norm to enforce sparsity.

Encoding: Given a codebook D, the encoding problem is to find the sparse code x of y, leading to the following optimization

$$\min_{x} \|y - Dx\|^2 \qquad s.t.\ \ \|x\|_0 \leq K \tag{3}$$

Computing the optimal solution involves searching over all the $\binom{M}{K}$ possible combinations and thus is NP-hard. Here, orthogonal matching pursuit (OMP) [25] is used to compute the sparse code x due to its efficiency and effectiveness. OMP selects the codeword best correlated with the current residual at each iteration, which is the reconstruction error remaining after the codewords chosen thus far are subtracted. At the first iteration, this residual is exactly the observation y. Once a new codeword is selected, the observation is orthogonally projected onto the span of all the previously selected codewords and the residual is recomputed. The procedure is repeated until the desired sparsity level K is reached.

Codebook Update: Given the sparse code matrix X, the codewords d_m are optimized sequentially. In the m-th step, the m-th codeword and its sparse codes can be computed by minimizing the residual matrix and the mutual coherence corresponding to that codeword

$$\begin{aligned} &\min_{d_m} \|Y - DX\|_F^2 + \lambda \sum_{i=1}^{M} \sum_{j=1, j\neq i}^{M} |d_i^\top d_j| \\ &s.t. \|d_m\|_2 = 1 \end{aligned} \tag{4}$$

Removing the constant terms, the above optimization problem can be simplified to

$$\begin{aligned} &\min_{d_m} \{\bar{x}_m^\top \bar{x}_m d_m^\top d_m - 2R_m\bar{x}_m + \lambda \sum_{j=1, j\neq m}^{M} |d_j^\top d_m|\} \\ &s.t. \|d_m\|_2 = 1 \end{aligned} \tag{5}$$

where $\bar{x}_i^\top$ are the rows of X, and $R_m = Y - \sum_{i\neq m} d_i \bar{x}_i^\top$ is the residual matrix for the m-th codeword. This matrix contains the differences between the observations and their approximations using all other codewords and their sparse codes. To avoid introducing new non-zero entries in the sparse code matrix X, the update process only considers observations that use the m-th codeword. We solve (5) by standard gradient descent with initialization $d_m^0 = \frac{R_m\bar{x}_m}{\|R_m\bar{x}_m\|_2}$, which is optimal when ignoring the mutual incoherence penalty.

It can be observed that the proposed alternative approach decreases the objective function (2) at each iteration. In practice, we find that MI-KSVD converges to good codebooks for a wide range of initializations. Fig. 2 compares the reconstruction error of the proposed MI-KSVD and KSVD on both training data and test data (see Section 4.1). MI-KSVD leads to small reconstruction error on both training and test data, compared with KSVD. This is remarkable since the additional penalty usually increases the reconstruction error. The codebook learned by MI-KSVD has average mutual coherence (AMC) $\frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{j=1, j\neq i}^{M} |d_i^\top d_j| = 0.118$, substantially smaller than 0.153 yielded by KSVD.

3.2. Hierarchial Matching Pursuit

In our hierarchical matching pursuit, MI-KSVD is used to learn codebooks at three layers, where the data matrix Y in the first layer consists of raw patches sampled from images, and Y in the second and third layers are sparse codes pooled from the lower layers. With the learned codebooks D, hierarchical matching pursuit builds a feature hierarchy, layer by layer, using batch orthogonal matching pursuit for computing sparse codes, spatial pooling for aggregating sparse codes, and contrast normalization for normalizing feature vectors, as shown in Fig. 3.

First Layer: The goal of the first layer in HMP is to extract sparse codes for small patches (*e.g.*, 5x5) and generate pooled codes for mid-level patches (*e.g.*, 16x16). Orthogonal matching pursuit is used to compute the sparse codes x of small patches (*e.g.*, 5x5 pixels). Spatial max pooling is then applied to aggregate the sparse codes. In our terminology, an image patch P is divided spatially into smaller cells. The features of each spatial cell C are the max pooled sparse codes, which are simply the component-wise maxima over all sparse codes within a cell:

$$\begin{aligned} F(C) = \max_{j\in C} [&\max(x_{j1}, 0), \cdots, \max(x_{jM}, 0), \cdots, \\ &\max(-x_{j1}, 0), \cdots, \max(-x_{jM}, 0)] \end{aligned} \tag{6}$$

Here, j ranges over all entries in the cell, and x_{jm} is the m-th component of the sparse code vector x_j of entry j. We split the positive and negative components of the sparse codes into separate features to allow higher layers weight positive

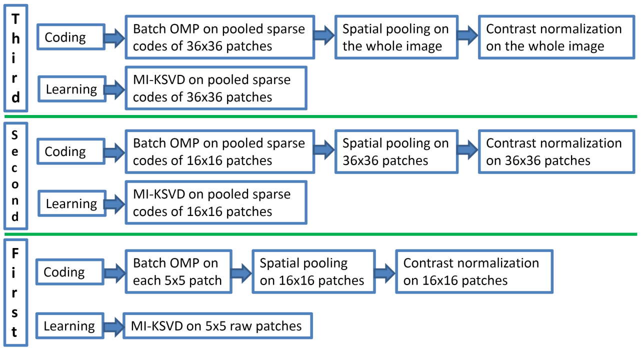
T h i r d
Coding
Batch OMP on pooled sparse codes of 36x36 patches
Spatial pooling on the whole image
Contrast normalization on the whole image
Learning
MI-KSVD on pooled sparse codes of 36x36 patches
S e c o n d
Coding
Batch OMP on pooled sparse codes of 16x16 patches
Spatial pooling on 36x36 patches
Contrast normalization on 36x36 patches
Learning
MI-KSVD on pooled sparse codes of 16x16 patches
F i r s t
Coding
Batch OMP on each 5x5 patch
Spatial pooling on 16x16 patches
Contrast normalization on 16x16 patches
Learning
MI-KSVD on 5x5 raw patches

Figure 3: A three-layer architecture of Hierarchical Matching Pursuit.

and negative responses differently. The feature F_P describing an image patch P is the concatenation of aggregated sparse codes in each spatial cell

$$F_P = \left[F(C_1^P), \cdots, F(C_s^P), \cdots, F(C_S^P)\right] \quad (7)$$

where $C_s^P \subseteq P$ is a spatial cell generated by spatial partitions, and S is the total number of spatial cells. We additionally normalize the feature vectors F_P by L_2 norm $\sqrt{\|F_P\|^2 + \varepsilon}$, where ε is a small positive number. Since the magnitude of sparse codes varies over a wide range due to local variations in illumination and occlusion, this operation makes the appearance features robust to such variations, as commonly done in SIFT features. We find that $\varepsilon = 0.1$ works well for all the recognition problems we consider.

Second Layer: The goal of the second layer in HMP is to gather and code mid-level sparse codes and generate pooled codes for large patches (e.g. 36x36). To do so, HMP applies batch OMP and spatial max pooling to features F_P generated in the first layer. The codebook for this level is learned by sampling features F_P over images. The process to extract the feature describing a large image patch is identical to that for the first layer: sparse codes of each image patch are computed using batch orthogonal matching pursuit, followed by spatial max pooling on the large patch. The feature vector is then normalized by its L2 norm.

Third Layer: The goal of the third layer in HMP is to generate pooled sparse codes for the whole image/object. Similar to the second layer, the codebook for this level is learned by sampling these pooled sparse codes in the second layer. With the learned codebook, just as in the second layer, sparse codes of each image patch (for instance, 36x36) are computed using batch OMP, followed by spatial max pooling on the whole images. The features of the whole image/object are the concatenation of the aggregated sparse codes of the spatial cells. The feature vector is then normalized by dividing with its L_2 norm.

A one-layer HMP has the same architecture as the final layer of a three-layer HMP, except that MI-KSVD and batch OMP are performed on 36x36 raw image patches instead of pooled sparse codes. A two-layer HMP has the same architecture as the second and third layers of a three-layer HMP, except that MI-KSVD and batch OMP in the second layer are performed on raw image patches.

3.3. Architecture of Multipath Sparse Coding

In visual recognition, images are frequently modeled as unordered collections of local patches, i.e. a bag of patches. Such models are flexible, and the image itself can be considered as a special case (bag with one large patch). Traditional bag-of-patches models introduce invariance by completely ignoring spatial positions of and relationships between patches, generally useful for visual recognition. The spatial pyramid bag-of-patches model [16] overcomes this problem by organizing patches into spatial cells at multiple levels and then concatenating features from spatial cells into one feature vector. Such models effectively balance the importance of invariance and discriminative power, leading to much better performance than simple bags. Spatial pyramid bags are a compelling strategy for unsupervised feature learning because of a number of advantages: (1) bag-of-patches virtually generates a large number of training samples for learning algorithms and decreases the chance of overfitting; (2) the local invariance and stability of the learned features are increased by pooling features in spatial cells; (3) by varying patch sizes, feature learning can capture structures at multiple levels of scale and invariance.

A single-path HMP already has the first two advantages. To exploit the advantage of multiple patch sizes as well as the strength of multi-layer architectures, our Multipath Hierarchical Matching Pursuit (M-HMP) configures matching pursuit encoders in multiple pathways, varying patch sizes and the number of layers (see Fig. 1). Note that in the final layer, sparse coding is always applied to the full image patches (16x16 on the top and 36x36 on the bottom). M-HMP encodes patches of different sizes, such as 16x16 and 36x36, which contain structures of different scales. More importantly, we argue that multiple paths, by varying the number of layers in HMP, is important for a single patch size. For instance, we could learn features for 36x36 image patches using a one-layer HMP or a two-layer HMP or a three-layer HMP. These HMP networks with different layers capture different aspects of 36x36 image patches. Intuitively, features of 36x36 image patches learned by a one-layer HMP capture basic structures of patches and are sensitive to spatial displacement. Features of 36x36 image patches learned by a two-layer HMP introduce robustness to local deformations due to the usage of spatial max pooling in the first layer. Features of 36x36 image patches learned by a three-layer HMP are highly abstract and robust due to their ability to recursively eliminate unimportant structures and increase invariance to local deformations by spatial max pooling in the previous layers.

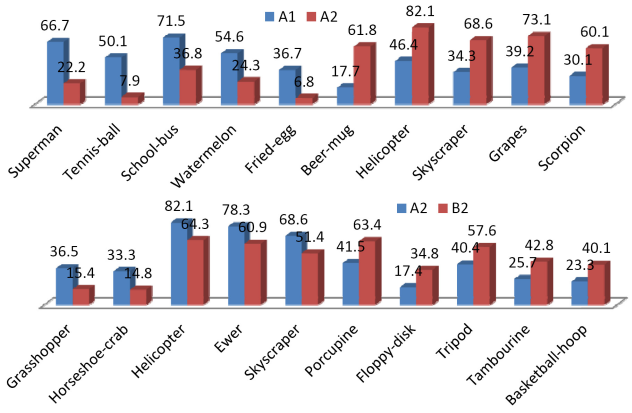

Figure 4: Top five categories on which the HMP pairs lead to the largest differences On Caltech-256 with 60 training images per category. *Top:* One-layer and two-layer HMP networks on image patches of size 16x16. *Bottom:* Two-layer HMP networks on image patches of size 16x16 and on images patches of size 36x36.

Next, we outline the detailed architecture of the five HMP networks used in the experiments. On image patches of size 16x16 (A), we learn a one-layer HMP on RGB images (A1) and a two-layer HMP on grayscale images (A2). For the one-layer HMP, we learn codebooks of size 1000 with sparsity level 5 on a collection of 16x16 raw patches sampled from RGB images. For the two-layer HMP, we first learn first-layer codebooks of size 75 with sparsity level 5 on a collection of 5x5 raw patches sampled from grayscale images. We then generate the pooled sparse codes on 4x4 spatial cells of 16x16 image patches with a pooling size of 4x4 pixels. Finally, we learn second-layer codebooks of size 1000 with sparsity level 10 on the resulting pooled sparse codes on 16x16 image patches.

On image patches of size 36x36 (B), we learn one-layer HMP on RGB images, and two-layer and three-layer HMP on grayscale images. For the one-layer HMP (B1), we learn codebooks of size 1000 with sparsity level 5 on a collection of 36x36 raw patches sampled from RGB images. For the two-layer HMP (B2), we first learn codebooks of size 300 with sparsity level 5 on a collection of 10x10 raw patches sampled from grayscale images. We then generate the pooled sparse codes on 4x4 spatial cells of 36x36 image patches with a pooling size of 9x9 pixels. Finally, we learn codebooks of size 1000 with sparsity level 10 on the resulting pooled sparse codes on 36x36 image patches. For the three-layer HMP (B3), the first two layers are the same as A2. For the third layer, we first generate the pooled sparse codes on 3x3 spatial cells of the pooled sparse codes in the second layer with a pooling size of 3x3. Finally, we learn codebooks of size 1000 with sparsity level 10 on the resulting pooled sparse codes based 36x36 image patches (36=4x3x3).

In the final layer of A1, A2, B1, B2 and B3, we generate image-level features by computing sparse codes of 36x36 image patches and performing max pooling followed by contrast normalization on spatial pyramids 1x1, 2x2 and 4x4 on the whole images. Note that the above architecture of multi-layer HMP networks leads to fast computation of pooled sparse codes.

To investigate how different HMP architectures help each other, we report the top five categories on which the HMP pairs A1 and A2, and the HMP pairs A2 and B2 yield the largest accuracy gaps. As can been seen in Fig. 4, the HMP networks with different architectures lead to significantly different accuracies on some categories. Generally speaking, the architecture A1 works well for the categories which have consistent appearances (colors, textures and so on) while the architecture A2 does well for categories which have consistent shapes. Architecture B2 outperforms architecture A2 for categories on which large scale shapes are more discriminative than small scale shapes, and vice versa. Note that the accuracy of M-HMP is robust with respect to patch size choice and other reasonable patch sizes such as 20x20 and 40x40 give similar results.

4. Experiments

We evaluate the proposed M-HMP models on five standard vision datasets on object, scene, and fine-grained recognition, extensively comparing to state-of-the-art algorithms using designed and learned features. All images are resized to 300 pixels on the longest side. We remove the zero frequency component from raw patches by subtracting their mean in the first layer of the HMP networks. The set of hyperparameters for all five HMP networks are optimized on a small subset of the ImageNet database. We keep this set of hyperparameters in all the experiments, even though per-dataset tweaking through cross validation may further improve accuracy. With the learned M-HMP features, we train linear SVMs for recognition.

4.1. MI-KSVD

We compare KSVD and MI-KSVD for a one-layer HMP network with image patches of size 36x36 on the Caltech-101 dataset. We choose a one-layer HMP due to the convenience of showing the learned codebooks. We learn codebooks of size 1000 with sparsity level 5 on 1,000,000 sampled 36x36 raw patches. The tradeoff parameter λ is chosen by performing five-fold cross validation on the training set.

We visualize the codebooks learned by KSVD and MI-KSVD in Fig. 5. First of all, the learned dictionaries have very rich appearances and include uniform colors of red, green and blue, transition codewords between different colors, gray and color edges, double gray and color edges, center-surround (dot) codewords, and so on. This suggests that a large variety of discriminative structures is captured. Moreover, the codebook learned by MI-KSVD is more balanced and diverse than that learned by KSVD: there are

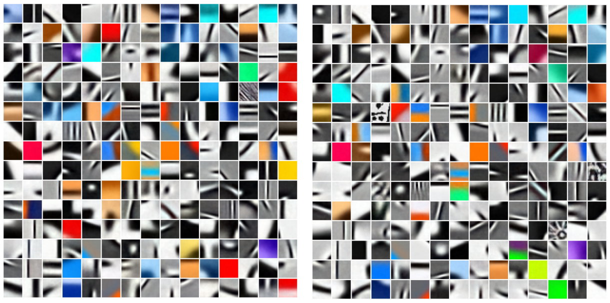

Figure 5: Learned codebooks by KSVD (*left*) and MI-KSVD (*right*) on Caltech-101. 225 codewords randomly selected from 1000 codewords are shown.

	Test Accuracy	AMC	Training Time
KSVD	71.9	0.153	6126.1s
MI-KSVD	73.1	0.118	6713.8s

Table 1: MI-KSVD and KSVD on Caltech-101. Both of them are stopped after 100 iterations that are sufficient for convergence.

less color codewords and more codewords with some complex structures in MI-KSVD. We compare KSVD and MI-KSVD in Table 1 in terms of test accuracy, training time and average mutual coherence (AMC). As can been seen, MI-KSVD leads to higher test accuracy and lower average mutual coherence than KSVD, with comparable training time.

4.2. Object Recognition

We investigate the behavior of M-HMP for object category recognition on Caltech-101. The dataset contains 9,144 images from 101 object categories and one background category. We use Caltech-101 because a large number of algorithms have been evaluated on this dataset, despite its known limitations. In the following sections, we will demonstrate multipath sparse coding on more standard vision datasets. Following the standard experimental setting, we train models on 30 images and test on no more than 50 images per category [16].

We show the results of M-HMP in Fig. 6 (A1, A2, B1, B2 and B3 are defined in Section 3.3). As can bee seen, the one-layer HMP networks (A1 and B1) work surprisingly well and already outperform many existing computer vision approaches, showing the benefits of learning from pixels. The two-layer HMP networks (A2 and B2) achieve the best results among five single pathways. The three-layer HMP network (B3) is superior to the one-layer networks (A1 and B1), but inferior to the two layer HMP networks (A2 and B2). The combination of HMP networks of different depths (A1+A2 and B1+B2+B3) leads to significant improvements over the best single HMP network (A2 and B2). Multipath coding combining all five HMP networks achieves the best result of 82.5%, suggesting that different paths complement one another and capture different image structures.

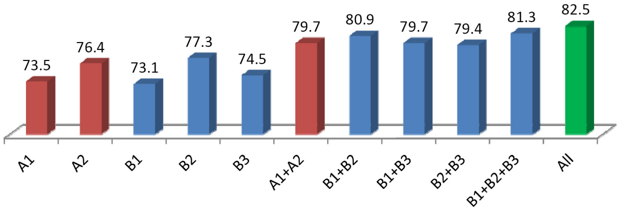

Figure 6: Test accuracy of single-path and multipath HMP. A1,A2,B1,B2 and B3 denotes the HMP networks of different architectures (See Section 3.3). A1+A2 indicates the combination of A1 and A2. "All" means the combination of all five paths: A1,A2,B1,B2 and B3.

SIFT+T [9]	67.7	HSC [32]	74.0
Local NBNN [20]	71.9	Asklocals [5]	77.1
LC-KSVD [13]	73.6	LP-β [11]	77.7
LLC [28]	73.4	FK [8]	77.8
HMP [3]	76.8	M-HMP	**82.5±0.5**

Table 2: Test accuracy on Caltech-101.

Training Images	15	30	45	60
Local NBNN [20]	33.5	40.1	/	/
LLC [28]	34.4	41.2	45.3	47.7
CRBM [27]	35.1	42.1	45.7	47.9
LP-β [11]	/	45.8	/	/
M-HMP	**42.7**	**50.7**	**54.8**	**58.0**

Table 3: Test accuracy on Caltech-256.

We compare M-HMP with recently published state-of-the-art recognition algorithms in Table 2. LLC [28], LC-KSVD [13], and Asklocals [5] are one-layer sparse coding approaches. SIFT+T [9] is soft threshold coding and CRBM [27] is a convolutional variant of Restricted Boltzmann Machines (RBM). FK [8] is a Fisher kernel based coding approach. All of them are based on SIFT. HSC [32] is a two layer sparse coding network using L1-norm regularization. Local NBNN [20] is an extension of Naive Bayesian Nearest Neighbor (NBNN). LP-β [11] is a boosting approach to combine multiple types of designed features. M-HMP achieves test accuracy superior to all of them; by a large margin.

To further evaluate the scalability of the proposed approach with respect to the number of categories and the number of images in each category, we perform experiments on Caltech-256. The dataset consists of 30,607 images from 256 object categories and background, where each category contains at least 80 images. Caltech-256 is much more challenging than Caltech-101 due to the larger number of classes and more diverse lighting conditions, poses, backgrounds, object sizes, etc. Following the standard setup [29], we gradually increase the training set size from 15 to 60 images per category with a step of 15 and test trained models on the rest of the images.

We report the average accuracy over 5 random trials in Table 3. We keep the same architecture as that for Caltech-

DPM [10]	30.4	DPM+Gist+SPM [22]	43.1
SPM [22]	34.4	HMP [4]	47.6
RBoW [23]	37.9	M-HMP	**51.2**

Table 4: Test accuracy on MIT-Scenes

101 (Section 4.2), with the only exception that the number of codewords in the final layer of HMP is increased to 2000 to accommodate for more categories and more images. As can be seen, our M-HMP approach makes exciting progress on this benchmark and is significantly better than all previously published results. More importantly, the performance gap grows larger with an increasing number of training images. For instance, the gap between M-HMP and CRBM is about 7.6% for 15 training images per category, but it increases to about 10.1% for 60 training images. This suggests that (1) rich features are important for large-scale recognition with a large number of categories and a large number of images; (2) M-HMP is well suited for extracting rich features from images, particularly important as we move toward high-resolution images.

4.3. Scene Recognition

We evaluate our M-HMP approach on the popular MIT Scene-67 dataset. This dataset contains 15,620 images from 67 indoor scene categories. We use the same M-HMP architecture and hyperparameters as for Caltech-101. Following standard experimental setting [22], we train models on 80 images and test on 20 images per category.

We report the accuracy of M-HMP on the training/test split provided on the authors' website in Table 4. Again, M-HMP achieves much higher accuracy than state-of-the-art recognition algorithms: spatial pyramid matching (SPM) [22], deformable parts models (DPM) [10], Reconfigurable Models (RBoW) [23], Hierarchical Matching Pursuit [4], and even the combination of SPM, DPM, and color GIST [22]. The best single-path M-HMP is an architecture of two layers on 16x16 image patches that obtains 44.4% accuracy. Multipath HMP dramatically increases the accuracy to 51.2%, suggesting that rich but complementary features are essential for achieving good performance on scene recognition.

4.4. Fine-grained Object Recognition

In the last decade, most work has been focused on basic-level recognition tasks: distinguishing different categories of objects, such as table, computer and human. Recently, there is increasing interest and attention on fine-grained (subordinate-level) recognition that classifies similar object categories, such as different species of birds, cats and dogs [6]. We evaluate our M-HMP approach on the Oxford-IIIT Pet [24] and the Caltech-UCSD Bird-200 [6] datasets. We use the same architecture as for Caltech-101, with the exception that the number of codewords is increased to 3000 in the final layer of HMP.

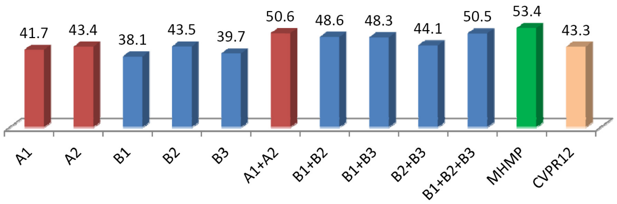

Figure 7: Test Accuracy on the Oxford-IIIT Pet Dataset.

MKL [6]	19.0	Pose Pooling [34]	28.2
LLC [31]	18.0	UTL [30]	28.2
RF [31]	22.4	M-HMP	**30.3**

Table 5: Test Accuracy on Caltech-UCSD Bird-200.

The Oxford-IIIT Pet dataset is a collection of 7,349 images of cats and dogs of 37 different breeds, of which 25 are dogs and 12 are cats. The dataset contains about 200 images for each breed, which have been split randomly into 50 for training, 50 for validation, and 100 for testing. Following the standard experimental setting [6], we train M-HMP on the training+validation set and compute recognition accuracy on the test set. We report test accuracy of M-HMP in Fig. 7. Our results are obtained on image layout only and should be compared with those on the same setting. We are not able to evaluate M-HMP on image+head and image+head+body layouts due to incomplete annotations in the publicly available dataset. As can been seen, M-HMP outperforms the Shape+Appearance approach [24] on image layout by a large margin, 53.4% vs. 43.3%. M-HMP achieves about 10 percent higher accuracy than all single HMP networks. The best single network is a two-layer HMP (B2) over grayscale images that achieves 43.5% accuracy.

The Caltech-UCSD Bird-200 dataset contains 6,033 images from 200 bird species in North America. In each image, the bounding box of a bird is given. Following the standard setting [6], 15 images from each species are used for training and the rest for testing. In Table 5, we compare our M-HMP with recently published algorithms such as multiple kernel learning [6], LLC [31], random forest [31], multi-cue [14], pose pooling [34] and unsupervised template learning [30]. Multipath HMP outperforms the state of the art by a large margin and sets a new record for fine-grained object recognition. Note that the previous approaches all use multiple types of features such as SIFT, color SIFT, color histograms, *etc.* to boost the classification accuracy; pose pooling [34] exploits additional labeled parts to train and test models.

5. Conclusions

We have proposed Multipath Hierarchical Matching Pursuit for learning expressive features from images. Our approach combines sparse coding through several pathways, using multiple patches of varying size, and encoding each patch through multiple paths with a varying number of layers. We have performed extensive comparisons on three types of visual recognition tasks: object recognition, scene recognition, and fine-grained object recognition. Our experiments have confirmed that the proposed approach outperforms the state-of-the-art on various popular vision benchmarks. These results are extremely encouraging, indicating that visual recognition systems can be significantly improved by learning features from raw images.

Acknowledgments

This work was funded in part by the Intel Science and Technology Center for Pervasive Computing and by ONR MURI grant N00014-07-1-0749.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006. 2

[2] L. Bo, X. Ren, and D. Fox. Kernel Descriptors for Visual Recognition. In *NIPS*, 2010. 1

[3] L. Bo, X. Ren, and D. Fox. Hierarchical Matching Pursuit for Image Classification: Architecture and Fast Algorithms. In *NIPS*, 2011. 1, 2, 6

[4] L. Bo, X. Ren, and D. Fox. Unsupervised Feature Learning for RGB-D Based Object Recognition. In *ISER*, 2012. 2, 7

[5] Y. Boureau, N. Roux, F. Bach, J. Ponce, and Y. LeCun. Ask the Locals: Multi-Way Local Pooling for Image Recognition. In *ICCV*, 2011. 2, 6

[6] S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and S. Belongie. Visual Recognition with Humans in the Loop. In *ECCV*, 2010. 7

[7] E. Candes and J. Romberg. Sparsity and Incoherence in Compressive Sampling. *Inverse problems*, 23:969, 2007. 2

[8] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The Devil is in the Details: an Evaluation of Recent Feature Encoding Methods. In *BMVC*, 2011. 6

[9] A. Coates and A. Ng. The Importance of Encoding versus Training with Sparse Coding and Vector Quantization. In *ICML*, 2011. 2, 6

[10] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IEEE PAMI*, 32:1627–1645, 2010. 7

[11] P. Gehler and S. Nowozin. On Feature Combination for Multiclass Object Classification. In *ICCV*, 2009. 6

[12] G. Hinton, S. Osindero, and Y. Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554, 2006. 1, 2

[13] Z. Jiang, Z. Lin, and L. Davis. Learning a Discriminative Dictionary for Sparse Coding via Label Consistent K-SVD. In *CVPR*, 2011. 6

[14] F. Khan, J. van de Weijer, A. Bagdanov, and M. Vanrell. Portmanteau Vocabularies for Multi-cue Image Representations. *NIPS*, 2011. 7

[15] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012. 1, 2

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *CVPR*, 2006. 4, 6

[17] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng. Building High-Level Features Using Large Scale Unsupervised Learning. In *ICML*, 2012. 1, 2

[18] H. Lee, R. Grosse, R. Ranganath, and A. Ng. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. In *ICML*, 2009. 2

[19] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative Learned Dictionaries for Local Image Analysis. In *CVPR*, 2008. 2

[20] S. McCann and D. Lowe. Local Naive Bayes Nearest Neighbor for image classification. In *CVPR*, 2012. 6

[21] B. Olshausen and D. Field. Emergence of Simple-cell Receptive Field Properties by Learning a Sparse Code for Natural Images. *Nature*, 381:607–609, 1996. 2

[22] M. Pandey and S. Lazebnik. Scene Recognition and Weakly Supervised Object Localization with Deformable Part-Based Models. In *ICCV*, 2011. 7

[23] S. N. Parizi, J. Oberlin, and P. Felzenszwalb. Reconfigurable Models for Scene Recognition. In *CVPR*, 2012. 7

[24] O. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Cats and Dogs. In *CVPR*, 2012. 7

[25] Y. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition. In *The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, pages 40–44, 1993. 3

[26] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and Clustering via Dictionary Learning with Structured Incoherence and Shared Features. In *CVPR*, 2010. 3

[27] K. Sohn, D. Jung, H. Lee, and A. Hero III. Efficient Learning of Sparse, Distributed, Convolutional Feature Representations for Object Recognition. In *ICCV*, 2011. 6

[28] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Guo. Locality-constrained Linear Coding for Image Classification. In *CVPR*, 2010. 2, 6

[29] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear Spatial Pyramid Matching using Sparse Coding for Image Classification. In *CVPR*, 2009. 6

[30] S. Yang, L. Bo, J. Wang, and L. Shapiro. Unsupervised template learning for fine-grained object recognition. In *NIPS*, 2012. 7

[31] B. Yao, A. Khosla, and L. Fei-Fei. Combining Randomization and Discrimination for Fine-grained Image Categorization. *CVPR*, 2011. 7

[32] K. Yu, Y. Lin, and J. Lafferty. Learning Image Representations from the Pixel Level via Hierarchical Sparse Coding. In *CVPR*, 2011. 1, 2, 6

[33] M. Zeiler, G. Taylor, and R. Fergus. Adaptive Deconvolutional Networks for Mid and High Level Feature Learning. In *ICCV*, 2011. 2

[34] N. Zhang, R. Farrell, and T. Darrell. Pose Pooling Kernels for Sub-category Recognition. *CVPR*, 2012. 7