Learning Hierarchical Image Representation with Sparsity, Saliency and Locality

Jimei Yang
jyang44@ucmerced.edu
Ming-Hsuan Yang
mhyang@ucmerced.edu
University of California, Merced
California, USA

We present a deep learning model for hierarchical image representation in which we build the hierarchy by stacking up the base models layer by layer. In each layer, the base model receives the features of the lower layer as input and produces a more invariant and complex representation. The bottom layer receives raw images as input and the top layer produces an image representation that can be used for high-level vision tasks.

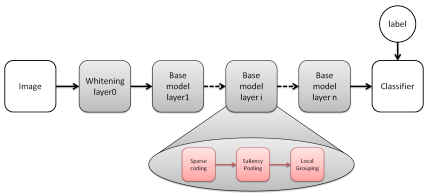

Figure 1: Illustration of the HSSL model. The shadowed components denote the learning process. At layer 0, a standard preprocessing is applied to reduce noise (e.g., whitening and normalization). From layer 1 and onward, the base model at layer i is repeatedly built upon layer $i-1$. The output of the layer n is fed into classifiers for vision tasks.

The base model defines a nonlinear mapping from the lower layer to the next latent layer within a restricted domain. We denote Ω^ℓ as the working domain of layer ℓ. Let $\mathbf{x}_i^{\ell-1}$ be a feature vector of layer $\ell-1$ and $\mathbf{X}^{\ell-1} = \{\mathbf{x}_i^{\ell-1}\}_{i\in\Omega^\ell}$ be the feature set in the working domain Ω^ℓ of layer ℓ. The nonlinear mapping function is defined as

$$\mathbf{x}^\ell = f(\mathbf{X}^{\ell-1}), \|\mathbf{x}^\ell\| \leq 1. \quad (1)$$

The function f maps a set of feature vectors from a lower layer to a single feature vector in the current layer. As a result, the first layer extracts features from small local patches and the last layer extracts a single feature of the entire image domain. Formally, we have $\Omega^1 \subset \Omega^2 \subset \ldots \subset \Omega^n = E$, where E denotes the entire image domain.

To construct a hierarchical representation, we need to learn the base model layer by layer. By removing the domain notations for presentation clarity, learning hierarchical image representation is defined by a recursive function

$$\mathbf{x}^n = \underbrace{f \circ f \circ \cdots \circ f}_{n}(\mathbf{x}^0). \quad (2)$$

In each layer, the base model function f is bounded so as the recursive function. Thus, the variations of hierarchical representations can be well controlled. Similar to other hierarchical models, it is not clear how the optimal number of layers can be learned easily. In this paper, we only use two layers in our HSSL model for category level object recognition.

The base model consists of three component functions: sparse coding s, salient pooling p and local grouping g. Thus, the nonlinear function f is composed of a chain: $f = g \circ p \circ s$.

Sparse coding We use sparse coding in our base model to learn a set of atom or basis signals from the lower layer so that raw features fed to the current layer can be well quantified. At each layer, we need to encode a large set of raw features in the image domain, and thus the sparse coding is the main computational bottleneck of our model. We develop a parallel implementation of ℓ_1 norm sparse coding by a coordinate descent algorithm. This implementation allows us to encode raw features of the entire image domain simultaneously and significantly improve the computational performance of our model.

Saliency pooling As a result of using the over-complete dictionary, the sparse codes are more sensitive to variation (e.g., slight translation or rotation) and noise. To alleviate this, pooling functions are often used to characterize the statistics of sparse codes within certain local image region. For image representation, we observe that irrelevant parts of the image (background, non-target objects) may have large sparse coefficients. Consequently, such sparse representations may encode more non-essential visual information. We propose a saliency-weighted max pooling function to address this problem. By using the bottom-up saliency to guide pooling, in general better sparse representations focusing on the foreground objects can be obtained.

Local grouping By grouping the pooled sparse codes in local neighborhood, the base model can produce increasingly complex representation for the use of the upper layer in the sense of bridging the semantic gap between successive layers. We first concatenate the pooled sparse codes in the square grids (*e.g.* $3\times3, 4\times4$) to form a vector. As the dimension of this vector is high, we use PCA (Principle Component Analysis) to reduce the dimension while preserving the representation ability for local grouping.

We refer the proposed model as HSSL (Hierarchical model with Sparsity, Saliency and Locality) model and Figure 1 illustrates the architecture and processes.

Instead of using hand-crafted descriptors (SIFT, HoG, Gabor, LBP), the proposed HSSL model learns effective representation directly from images in a unsupervised data-driven manner. The proposed model requires minimum expert knowledge for specific tasks or laborious labeling process. Therefore, it can be easily applied to vision tasks with different sensor data such as infrared and depth images where descriptive features cannot be easily crafted.

Table 1: Experimental results with Caltech 101 dataset.

	Method	15 samples	30 samples
Bio-inspired	Pinto [2]		67%
Deep Learning	Zeiler [5]	58.6 ± 0.7%	66.9 ± 1.1%
	HSSL	**68.7± 0.4%**	**76.1± 1.3%**
SIFT-based	Yang [4]	67.0 ± 0.45%	73.2 ± 0.54%

Table 2: Experimental results with Caltech 256 dataset.

Methods	15 samples	30 samples	45 samples	60 samples
Yang [4]	27.8 ± 0.51%	34.0 ± 0.35%	37.5 ± 0.55%	40.1 ± 0.91%
HSSL	**29.8± 0.4%**	**35.4± 0.4%**	**38.7±0.3%**	**41.6± 0.3%**

Table 3: Experimental results with Oxford Flowers dataset.

Methods	40 trainings / 20 tests	60 trainings / 20 tests
Varma [3]	68.9 ± 2.0%	
Nilsback [1]		71.8 ± %
HSSL	**69.7±2.7%**	**76.2±3.8%**

We validate our HSSL model with object categorization experiments using the Caltech 101 database. We mainly compare our method with the state-of-the-art biologically-inspired [2], deep learning [5] and SIFT-based methods [4]. The results are summarized in Table 1. More experimental results on Caltech 256 and Oxford Flowers are listed in Tab 2 and Tab 3.

[1] Maria-Elena Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *CVPR*, 2006.

[2] Nicolas Pinto, David D. Cox, and James J. Dicarlo. Why is real-world visual object recognition hard? *PLoS Computational Biology*, 4(1): e27+, 2008.

[3] Manik Varma and Debajyoti Ray. Learning the discriminative power-invariance trade-off. In *ICCV*, 2007.

[4] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[5] Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and Rob Fergus. Deconvolutional networks. In *CVPR*, 2010.