Generative Hierarchical Learning of Sparse FRAME Models

Jianwen Xie[1], Yifei Xu[2], Erik Nijkamp[1], Ying Nian Wu[1], and Song-Chun Zhu[1]
[1]University of California, Los Angeles (UCLA), USA
[2]Shanghai Jiao Tong University, Shanghai, China
jianwen@ucla.edu, fei960922@sjtu.edu.cn, enijkamp@ucla.edu, {ywu,sczhu}@stat.ucla.edu

Abstract

This paper proposes a method for generative learning of hierarchical random field models. The resulting model, which we call the hierarchical sparse FRAME (Filters, Random field, And Maximum Entropy) model, is a generalization of the original sparse FRAME model by decomposing it into multiple parts that are allowed to shift their locations, scales and rotations, so that the resulting model becomes a hierarchical deformable template. The model can be trained by an EM-type algorithm that alternates the following two steps: (1) Inference: Given the current model, we match it to each training image by inferring the unknown locations, scales, and rotations of the object and its parts by recursive sum-max maps, and (2) Re-learning: Given the inferred geometric configurations of the objects and their parts, we re-learn the model parameters by maximum likelihood estimation via stochastic gradient algorithm. Experiments show that the proposed method is capable of learning meaningful and interpretable templates that can be used for object detection, classification and clustering.

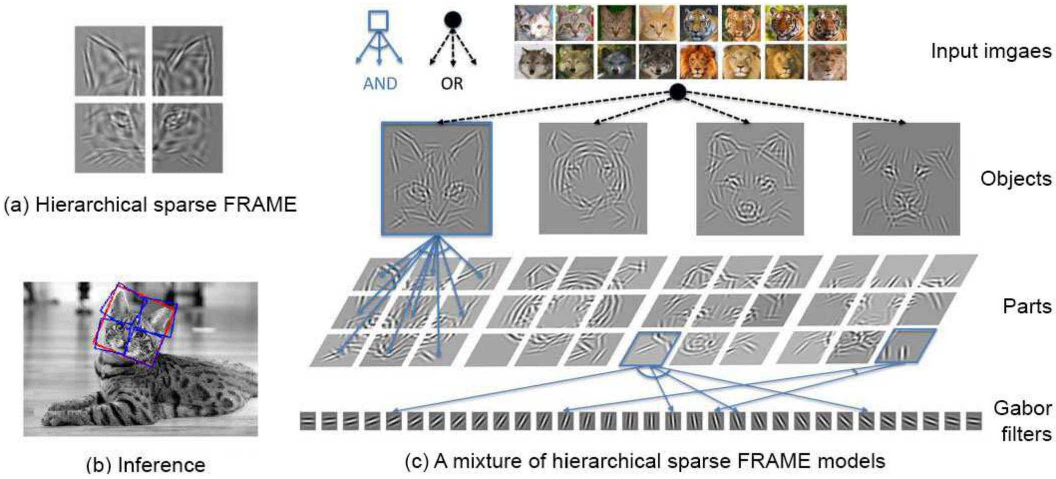
1. Introduction

Motivation and objective. We are entering a new age of computer vision applications, where machine learning technology plays a critical role in achieving a high level of prediction performance, e.g., [7, 8, 12, 13]. However, some machine learning models are opaque and difficult for people to understand. Explainable models are highly desirable, if users are to understand, interpret and effectively manage the behaviors of the models. Therefore, discovering explainable models for visual data is an important problem in computer vision and artificial intelligence.

Models with hierarchical and compositional representations, such as deformable part-based models [5] and stochastic And-Or templates [11], have been shown to be a powerful basis for achieving both prediction accuracy and explainability. They are capable of learning reconfigurable representations to deal with both structural and appearance variations of objects. These models can be paired with either discriminative learning method or generative learning method. Discriminative learning seeks to identify and weigh the most discriminant features and structures for explaining the object categories, while generative learning enables us to learn the parameters and interpretable patterns for explaining the image data instead of predicting the image categories. Moreover, generative learning is not only important for making the model explainable, it can also be used for unsupervised learning from unlabeled images.

Recently, Xie et al. proposed a sparse FRAME (Filters, Random field, And Maximum Entropy) model [16, 17] as a generative model for image patterns. It can be considered a template consisting of a small number of perturbable Gabor wavelets (sketches) at selected locations, scales and orientations. The learned knowledge in the model can be visualized by sampling from the model. However, the sparse FRAME models can only deal with small deformations (e.g., edge perturbations), and may fail when there exist large geometric changes (e.g., part deformations). To address this limitation, we propose to extend the original sparse FRAME model to a hierarchical version, which we call the hierarchical sparse FRAME model, by explicitly involving part-level representations and deformations.

Method overview. (1) Representation: The hierarchical sparse FRAME model is a hierarchical compositional deformable template, which is composed of a group of part templates that are allowed to shift their locations and rotations relative to each other. Each part template is in turn composed of a group of Gabor wavelets that are allowed to shift their locations and orientations relative to each other. (2) Inference: The model inference is to determine a certain geometric configuration of the template for a given object such that the log-likelihood is maximized. This can be efficiently achieved by a bottom-up/top-down dynamic programming, which is implemented by recursive sum-max maps. (3) Generative learning: The model is learned in a generative manner in the sense that the learning is carried out by maximum likelihood estimation and also it involves synthesizing image patterns via Markov chain Monte Carlo


(a) Hierarchical sparse FRAME

(b) Inference

(c) A mixture of hierarchical sparse FRAME models

Figure 1: (a) Hierarchical sparse FRAME model: A hierarchical sparse FRAME model with 2 × 2 parts is learned from roughly aligned observed images. The parts are visualized by displaying the synthesized images generated by the 4 part models that are composed into the object model. (b) Inference: A testing image with bounding boxes showing the inferred locations, rotations and scales of the object (red) and parts (blue). (c) A mixture of hierarchical sparse FRAME models is learned by an EM-type algorithm from animal face images of four categories without manual labeling. The learned mixture model is visualized as an And-Or graph, where an OR node (in black) represents a selection between difference alternatives and an AND node (in blue) represents a composition of terminal nodes or children nodes. The object and part templates shown in the And-Or graph are synthesized image patterns generated by the learned model via MCMC.

(MCMC) sampling. (4) Unsupervised learning: As the model is a fully generative model, it can be learned in an unsupervised manner, where the locations, scales and orientations of the object, parts, and edges (Gabor wavelets) are unknown, by an EM-type algorithm that alternates inference and re-learning steps. A mixture of hierarchical sparse FRAME models can also be learned unsupervisedly as an And-Or graph [22].

Figure 1 illustrates the basic idea of the hierarchical sparse FRAME model and the mixture model. A three-layer hierarchical model with 2 × 2 parts is visualized in Figure 1(a) by displaying the synthesized images generated from its part models by MCMC. Figure 1(b) displays an example of inference of the hierarchical sparse FRAME model on a testing image, with bounding boxes showing the inferred locations, scales and rotations of the object (red) and parts (blue). Figure 1(c) illustrates a mixture of hierarchical sparse FRAME models as an And-Or graph, which is learned from 50 animal face images of four categories, where the category labels are unknown. The black solid dot represents an OR node for selection. The blue empty squares denote AND nodes, which are compositions of terminal nodes (Gabor wavelets) or children AND nodes (parts). Each AND node (object or part) or each terminal node (Gabor wavelet) is also associated with a geometric OR node which accounts for its deformation. For clarity, the geometric OR nodes are not visualized.

Related work. Most existing methods for learning hierarchical representations of object patterns are based on discriminative learning [5, 20, 9]. In this paper, we learn a generative model for hierarchical representation of objects. Our work is similar to [6, 21], which also learn hierarchical compositions of Gabor wavelets or edgelets. They learn the models via bottom-up layer-by-layer schemes. Once the lower layers are learned, they are fixed in the learning of higher layers. In contrast, our iterative learning algorithm re-learns the object and part templates, and re-selects the Gabor wavelets in each iteration. Our work is also related to And-Or template (AOT) [11] and hierarchical compositional model [2]. To represent visual parts in the models, the former uses hybrid image template (HIT) [10], and the latter uses active basis template (ABT) [15]. Both HIT and ABT are templates of Gabor wavelets and make the simplifying assumptions that the selected Gabor wavelets are orthogonal and independent in order to avoid time-consuming MCMC computation in learning. In our model, parts are represented by sparse FRAME models, which do not make the above simplifying assumptions, so that our model is mathematically rigorous and is capable of visualizing the learned model by synthesizing patterns via MCMC, which makes our model more explainable.

2. Background of sparse FRAME model

This section reviews the background of the sparse FRAME model [16], which serves as the foundation of the hierarchical sparse FRAME model.

2.1. Inhomogeneous FRAME model

Let $\mathbf{I}$ be an image defined on a square or rectangular domain $\mathcal{D}$. Let $B_{x,s,\alpha}$ denote a basis function such as Gabor wavelet (or difference of Gaussian (DoG) filter) centered at pixel x (a two-dimensional vector) and tuned to scale s and orientation α. Given a dictionary of basis functions or filter bank $\{B_{x,s,\alpha}, \forall x, s, \alpha\}$, the dense version of the inhomogeneous FRAME model is a spatially non-stationary random field that reproduces statistical properties of filter responses at all the locations x, scales s and orientations α. The model is of the following form

$$p(\mathbf{I}; \lambda) = \frac{1}{Z(\lambda)} \exp\left(\sum_{x,s,\alpha} \lambda_{x,s,\alpha} |\langle \mathbf{I}, B_{x,s,\alpha} \rangle| \right) q(\mathbf{I}), \quad (1)$$

where $\lambda = (\lambda_{x,s,\alpha}, \forall x, s, \alpha)$ are the weight parameters, $\langle \mathbf{I}, B_{x,s,\alpha} \rangle$ is the inner product between $\mathbf{I}$ and $B_{x,s,\alpha}$, $Z(\lambda)$ is the normalizing constant, and $q(\mathbf{I})$ is a known Gaussian white noise reference distribution.

Given a set of roughly aligned training images $\{\mathbf{I}_m, m = 1, ..., M\}$ from the same object category, where M is the number of training images, we can learn the weight parameters λ by maximizing the log-likelihood $L(\lambda) = \sum_{m=1}^{M} \log p(\mathbf{I}_m; \lambda)/M$, using the stochastic gradient ascent algorithm [19]

$$\lambda^{(t+1)}_{x,s,\alpha} = \lambda^{(t)}_{x,s,\alpha} + \gamma_t \left(\frac{1}{M} \sum_{m=1}^{M} |\langle \mathbf{I}_m, B_{x,s,\alpha} \rangle| - \frac{1}{\tilde{M}} \sum_{m=1}^{\tilde{M}} |\langle \tilde{\mathbf{I}}_m, B_{x,s,\alpha} \rangle| \right), \quad (2)$$

where γ_t is the step size, $\{\tilde{\mathbf{I}}_m, m = 1, ..., \tilde{M}\}$ are the synthesized images sampled from $p(\mathbf{I}; \lambda^{(t)})$ using Hamiltonian Monte Carlo (HMC) algorithm [4]. $\tilde{M}$ is the number of independent parallel Markov chains that sample from $p(\mathbf{I}; \lambda^{(t)})$. The difference $\sum_{m=1}^{M} |\langle \mathbf{I}_m, B_{x,s,\alpha} \rangle|/M - \sum_{m=1}^{\tilde{M}} |\langle \tilde{\mathbf{I}}_m, B_{x,s,\alpha} \rangle|/\tilde{M}$ is the Monte Carlo estimate of the gradient of the log-likelihood $L(\lambda)$ at $\lambda^{(t)}$.

The estimation of the normalizing constant is required in unsupervised learning. Starting from $\lambda^{(0)} = 0$ and $\log Z(\lambda^{(0)}) = 0$, we can estimate $\log Z(\lambda^{(t)})$ along the learning process by $\log Z(\lambda^{(t+1)}) = \log Z(\lambda^{(t)}) + \log \frac{Z(\lambda^{(t+1)})}{Z(\lambda^{(t)})}$, where the ratio of the normalizing constants at two consecutive steps can be approximated by

$$\frac{Z(\lambda^{(t+1)})}{Z(\lambda^{(t)})} \approx \frac{1}{\tilde{M}} \sum_{m=1}^{\tilde{M}} \left[\exp\Big(\sum_{x,s,\alpha} (\lambda^{(t+1)}_{x,s,\alpha} - \lambda^{(t)}_{x,s,\alpha}) \times |\langle \tilde{\mathbf{I}}_m, B_{x,s,\alpha} \rangle| \Big) \right]. \quad (3)$$

2.2. Sparse FRAME model

The sparse FRAME model is a sparsified version of the dense model in (1), where only a small number of basis functions are selected from the given dictionary. We can explicitly write the sparsified model as

$$p(\mathbf{I}; \mathbf{B}, \lambda) = \frac{1}{Z(\lambda)} \exp\left(\sum_{i=1}^{n} \lambda_i |\langle \mathbf{I}, B_{x_i,s_i,\alpha_i} \rangle| \right) q(\mathbf{I}),$$

where $\mathbf{B} = (B_{x_i,s_i,\alpha_i}, i = 1, ..., n)$ are the n basis functions selected from a given dictionary (n is assumed to be given, e.g., $n = 200$), and $\lambda = (\lambda_i, i = 1, ..., n)$ are the corresponding weight parameters. The learning of the sparse model involves the selection of basis functions and the estimation of the corresponding weight parameters.

A two-stage learning algorithm [16] or a single-stage learning algorithm [17] can be used to train the sparse FRAME model. In this paper, we will use the two-stage learning algorithm that consists of the following two stages: (1) In the first stage, a shared sparse coding scheme is used to select $\mathbf{B} = (B_{x_i,s_i,\alpha_i}, i = 1, ..., n)$ by simultaneously reconstructing all the observed images $\{\mathbf{I}_m, m = 1, ..., M\}$. To account for shape deformations, B_{x_i,s_i,α_i} are allowed to locally perturb their locations and orientations on each observed image during reconstruction. Therefore, we have $\mathbf{I}_m = \sum_{i=1}^{n} c_{m,i} B_{x_i+\Delta x_{m,i},s_i,\alpha_i+\Delta\alpha_{m,i}}$, where $(\Delta x_{m,i}, \Delta\alpha_{m,i})$ are the local perturbations of the location and orientation of the i-th basis function B_{x_i,s_i,α_i} in the m-th training image, and $c_{m,i}$ are the reconstruction coefficients of the selected wavelets. The selection is accomplished by minimizing the overall least squares reconstruction error

$$\sum_{m=1}^{M} \|\mathbf{I}_m - \sum_{i=1}^{n} c_{m,i} B_{x_i+\Delta x_{m,i},s_i,\alpha_i+\Delta\alpha_{m,i}}\|^2.$$

This can be achieved by a shared matching pursuit algorithm. (2) After selecting $\mathbf{B} = (B_{x_i,s_i,\alpha_i}, i = 1, ..., n)$, the second stage estimates the corresponding weight parameters $\lambda = (\lambda_i, i = 1, ..., n)$ by maximum likelihood using the stochastic gradient ascent algorithm as in equation (2) and estimates $\log Z(\lambda)$ by equation (3).

The image log-likelihood $L(\mathbf{I}|\mathbf{B})$, which is computed by

$$\sum_{i=1}^{n} \lambda_i \max_{\Delta x, \Delta\alpha} |\langle \mathbf{I}, B_{x_i+\Delta x,s_i,\alpha_i+\Delta\alpha} \rangle| - \log Z(\lambda),$$

serves as the template matching score for object recognition.

3. Hierarchical Sparse FRAME Model

3.1. Representation

Hierarchical random field model. In this section, we will extend the original sparse FRAME model to a hierarchical version which we call the hierarchical sparse FRAME model. It is a composition of shiftable parts, while the parts themselves are compositions of a number of shiftable basis functions. The model is a probability distribution defined on $\mathbf{I}$,

$$p(\mathbf{I};\mathbf{H},\Lambda)=\frac{1}{Z(\Lambda)}\exp\left[f(\mathbf{I};\mathbf{H},\Lambda)\right]q(\mathbf{I}), \tag{4}$$

where the scoring function $f(\mathbf{I};\mathbf{H},\Lambda)$ is

$$f(\mathbf{I};\mathbf{H},\Lambda)=\sum_{j=1}^{K}\sum_{i=1}^{n_j}\lambda_i^{(j)}|\langle\mathbf{I},B_{x_i^{(j)},s_i^{(j)},\alpha_i^{(j)}}\rangle|,$$

where $\mathbf{H}=\{(B_{x_i^{(j)},s_i^{(j)},\alpha_i^{(j)}},i=1,...,n_j),j=1,...,K\}$ represents a template of K groups of selected basis functions. Each group represents a part template. n_j is the number of basis functions in group j. $\Lambda=\{(\lambda_i^{(j)},i=1,...,n_j),j=1,...,K\}$ collects the parameters. Learning such a hierarchical random field model requires selecting basis functions from a given dictionary to form a hierarchy and estimating their associated parameters. In our current implementation, we simply divide the image domain into $K=d\times d$ non-overlapping parts, so that the basis functions within each part form a group, and the parts are allowed to shift.

Hierarchical deformation. We may treat $\mathbf{H}$ as a hierarchical deformable template, so that when it is fitted to each training image $\mathbf{I}_m$, the part templates and the basis functions are allowed to perturb their locations and orientations to account for shape deformations. Learning model (4) from roughly aligned training images requires inference of the deformations of both parts and basis functions.

3.2. Hierarchical deformable template

Part template. Each part in the model can be considered a sparse FRAME model, so we can simply generalize the notation for the original sparse FRAME templates to obtain the one for the part templates. Given a sparse FRAME template $\mathbf{B}=(B_{x_i,s_i,\alpha_i},i=1,..,n)$, for simplicity, we shall temporarily assume $\mathbf{B}$ is only allowed spatial translation in encoding images. Suppose $\mathbf{B}$ appears at location X in image $\mathbf{I}$, then we can write the representation as

$$\mathbf{I}=\sum_{i=1}^{n}c_iB_{X+x_i+\Delta x_i,s_i,\alpha_i+\Delta\alpha_i}+\epsilon=C\mathbf{B}_X+\epsilon,$$

where $C=(c_i,i=1,...,n)$ collects all coefficients, $\mathbf{B}_X=(B_{X+x_i+\Delta x_i,s_i,\alpha_i+\Delta\alpha_i},i=1,...,n)$ is the deformed template spatially translated to X. $\mathbf{B}_X$ explains the part of $\mathbf{I}$ that is covered by $\mathbf{B}_X$. For image $\mathbf{I}$ and location X, the log-likelihood $L(\mathbf{I}|\mathbf{B}_X)$ is computed by

$$\sum_{i=1}^{n}\lambda_i\max_{\Delta x,\Delta\alpha}|\langle\mathbf{I},B_{X+x_i+\Delta x,s_i,\alpha_i+\Delta\alpha}\rangle|-\log Z(\lambda).$$

We can generalize $\mathbf{B}_X$ by using $\mathbf{B}_{X,S,A}$ to denote the part template at location X, scale S and rotation A. We will use $L(\mathbf{I}|\mathbf{B}_{X,S,A})$ to denote the log-likelihood of the part template $\mathbf{B}_{X,S,A}$.

Object template. With the notation of part template, we can denote a hierarchical sparse FRAME model, which is a template of K part templates, by $\mathbf{H}=\{\mathbf{B}^{(j)}_{X_j,S_j,A_j},j=1,..,K\}$, where (X_j,S_j,A_j) are the location, scale and rotation of the j-th part template in the object template $\mathbf{H}$. Then we can represent image $\mathbf{I}_m$ by a template of K parts:

$$\mathbf{I}_m=\sum_{j=1}^{K}C_{m,j}\mathbf{B}^{(j)}_{X_j,S_j,A_j}+\epsilon_m, \tag{5}$$

where each $\mathbf{B}^{(j)}_{X_j,S_j,A_j}$ is assumed to deform its basis functions by local max pooling when it encodes the image.

Since the object template $\mathbf{H}$ is deformable in the sense that all the parts are allowed to perturb their locations, scales and rotations to account for the deformation in the image, we can extend (5) to

$$\mathbf{I}_m=\sum_{j=1}^{K}C_{m,j}\mathbf{B}^{(j)}_{X_j+\Delta X_{m,j},S_j+\Delta S_{m,j},A_j+\Delta A_{m,j}}+\epsilon_m,$$

where $(\Delta X_{m,j},\Delta S_{m,j},\Delta A_{m,j})$ are perturbations of the location, scale and rotation of the j-th part template $\mathbf{B}^{(j)}_{X_j,S_j,A_j}$, and assumed to take values within limited and properly discretized ranges (default setting: $\Delta X_{m,j}\in[-1,1]\times[-1,1]$ pixels, $\Delta S_{m,j}\in\{-1,0,1\}\times 0.1$, and $\Delta A_{m,j}\in\{-1,0,1\}\times\pi/16$). We use $L(\mathbf{I}_m|\mathbf{B}^{(j)}_{X_j,S_j,A_j})$ to denote the log-likelihood of part $\mathbf{B}^{(j)}_{X_j,S_j,A_j}$. Further, we assume the parts do not overlap with each other, i.e., the subspaces spanned by the basis functions in different parts are orthogonal to each other, then the log-likelihood score of the image $\mathbf{I}_m$ given the object template $\mathbf{H}$ is

$$L(\mathbf{I}_m|\mathbf{H})=\sum_{j=1}^{K}\max_{\Delta X,\Delta S,\Delta A}L(\mathbf{I}_m|\mathbf{B}^{(j)}_{X_j+\Delta X,S_j+\Delta S,A_j+\Delta A}). \tag{6}$$

3.3. EM-type learning algorithm

Objective function. Equation (6) assumes all objects are aligned with only part-level deformations. In unsupervised learning, objects in the training images can be non-aligned in the sense that they might appear at different locations, even with different scales and rotations. For notation simplicity, we temporarily assume $\mathbf{H}$ is only allowed spatial translation in matching objects. We will use $\mathbf{H}_{\mathcal{X}}$ to denote the object template at location $\mathcal{X}$, and let $L(\mathbf{I}_m|\mathbf{H}_{\mathcal{X}})$ be the log-likelihood score of $\mathbf{H}_{\mathcal{X}}$. The learning of the hierarchical sparse FRAME model is to learn the K part templates $\{\mathbf{B}^{(j)}, j = 1, ..., K\}$ from non-aligned training images $\{\mathbf{I}_m, m = 1, ..., M\}$, while inferring the object locations $\mathcal{X}_m$, the part perturbations $(\Delta X_{m,j}, \Delta S_{m,j}, \Delta A_{m,j})$, and the perturbations of basis functions, by maximizing the objective function defined as the sum of the log-likelihood given $\mathbf{H}$ over all the training images, $\sum_{m=1}^{M} L(\mathbf{I}_m|\mathbf{H}_{\mathcal{X}_m})$, which is

$$\sum_{m=1}^{M}\left[\sum_{j=1}^{K} L(\mathbf{I}_m|\mathbf{B}^{(j)}_{\mathcal{X}_m+X_j+\Delta X_{m,j},S_j+\Delta S_{m,j},A_j+\Delta A_{m,j}})\right], \tag{7}$$

subject to the constraint that there are no overlapping parts in each $\mathbf{I}_m$. The learning can be done by an EM-type algorithm that iterates the inference step and the re-learning step in order to maximize the objective function (7):

E-step: Inference. Given the current hierarchical sparse FRAME model $\mathbf{H} = \{\mathbf{B}^{(j)}_{X_j,S_j,A_j}, j = 1, .., K\}$, we match it to each image $\mathbf{I}_m$ by inferring the location $\widehat{\mathcal{X}}_m$ of the object template in $\mathbf{I}_m$ via $\widehat{\mathcal{X}}_m =$

$$\arg\max_{\mathcal{X}} \sum_{j=1}^{K} \max_{\Delta X,\Delta S,\Delta A} L(\mathbf{I}_m|\mathbf{B}^{(j)}_{\mathcal{X}+X_j+\Delta X,S_j+\Delta S,A_j+\Delta A}),$$

and the perturbations in locations, scales and rotations of K parts via

$$\begin{aligned}&(\Delta X_{m,j}, \Delta S_{m,j}, \Delta A_{m,j})\\ &= \arg\max_{\Delta X,\Delta S,\Delta A} L(\mathbf{I}_m|\mathbf{B}^{(j)}_{\widehat{\mathcal{X}}_m+X_j+\Delta X,S_j+\Delta S,A_j+\Delta A}),\end{aligned}$$

as well as the perturbations of all basis functions in each part. The inference can be efficiently accomplished by recursive sum-max maps described in Algorithm 1, which is a bottom-up/top-down procedure. For notation simplicity, we omit the scales and rotations of both the object template and its part templates in the description of Algorithm 1.

M-step: Re-learning. Given the inferred deformations (i.e., object and part bounding boxes), we can first align the objects and parts by morphing the corresponding image patches. We then learn an original sparse FRAME model on the aligned training images, which involves the selection of basis functions and the parameters estimation, and then divide the object template into $K = d \times d$ non-overlapping part templates.

The EM-type algorithm is initialized by randomly assigning an initial bounding box of object to each training image. It is run for a few of iterations until convergence.

If the scales and rotations of the objects are also inferred by the $\arg\max$ operation in E-step, the learning algorithm can deal with learning from non-aligned objects with unknown locations, scales, and rotations. Figure 2 displays one example of learning from non-aligned images. The object template consists of 2×2 part templates. Each part template is of size 50×50. The number of non-aligned training images is 26. The total number of the selected basis functions (Gabor wavelets) is 300. The number of iterations is 6. Figure 2 (a) displays 2×2 parts of synthesized images generated by the learned model. Figure 2(b) displays 2×2 parts of sketch templates which illustrate the selected Gabor wavelets by shared matching pursuit. Figure 2(c) illustrates 12 examples of 26 non-aligned training images from cat category, with bounding boxes showing the inferred locations, scales, and rotations of the objects (black) and their parts (colored) after the model is learned. Figure 2(d) shows the inference results of the learned model on 2 testing images, with bounding boxes indicating the geometric configurations of the detected objects (black) and their parts (colored).

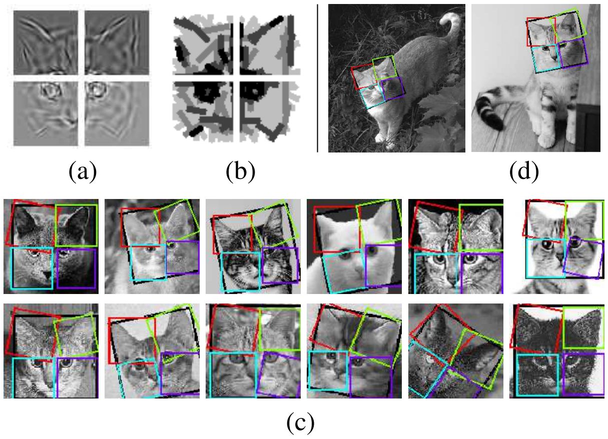


Figure 2: Learning a hierarchical sparse FRAME model from non-aligned images. (a) 2×2 parts of synthesized images generated by the learned model. (b) 2×2 parts of sketch templates where each Gabor wavelet is illustrated by a bar (smaller scales of bars appear darker). (c) 12 examples of 26 non-aligned training images from cat category, with bounding boxes showing the inferred locations, scales, and rotations of the objects (black) and parts (colored) by the learned model in E-step. (d) Inference results of the learned model on 2 testing images.

4. Experiments

4.1. Evaluating mixture models by clustering tasks

A mixture of hierarchical sparse FRAME models can be trained in an unsupervised manner by an EM-type algorithm that iterates the following two steps: (1) classifying images into different clusters based on the current mixture model, (2) re-learning the model of each cluster from images. Mixture models can be evaluated by clustering tasks, and we use a benchmark clustering dataset [17] that consists of 12 clustering tasks, where the number of clusters of each task varies from 2 to 5, and each cluster has 15 images.

Algorithm 1 Inference algorithm for hierarchical sparse FRAME model

```
Input: A hierarchical sparse FRAME model $\mathbf{H} = \{\mathbf{B}^{(j)}_{X_j}, j = 1, .., K\}$,
    where $\mathbf{B}^{(j)} = (B_{x_i^{(j)}, s_i^{(j)}, \alpha_i^{(j)}}, i = 1, ..., n_j)$, model parameters
    $\Lambda = \{(\lambda_i^{(j)}, i = 1, ..., n_j), j = 1, ..., K\}$, and a testing image $\mathbf{I}$.
Output: Location $\widehat{\mathcal{X}}$ of the object template $\mathbf{H}$ on image $\mathbf{I}$, perturbations
    $\{\Delta X_j, j = 1, ..., K\}$ of the parts, and perturbations of the basis functions in all parts
    $\{(\Delta x_i^{(j)}, \Delta\alpha_i^{(j)}), i = 1, ..., n_j, j = 1, ..., K\}$.
Up-1 compute feature map SUM1 of Gabor $B$ on $\mathbf{I}$ for all locations $x$, scales $s$ and orientations $\alpha$:
    $\mathrm{SUM1}(x, s, \alpha) = |\langle \mathbf{I}, B_{x,s,\alpha} \rangle|, \forall x, s, \alpha$
Up-2 compute MAX1 by local max-pooling to account for the shifts of Gabor wavelets:
    $\mathrm{MAX1}(x, s, \alpha) = \max_{\Delta x, \Delta\alpha} \mathrm{SUM1}(x + \Delta x, s, \alpha + \Delta\alpha), \forall x, s, \alpha$
Up-3 compute the matching score SUM2 of part template $\mathbf{B}^{(j)}$ on the image $\mathbf{I}$ for all locations $X$:
    $\mathrm{SUM2}^{(j)}(X) = \sum_{i=1}^{n_j} \lambda_i^{(j)} \mathrm{MAX1}(X + x_i^{(j)}, s_i^{(j)}, \alpha_i^{(j)}) - \log Z(\lambda^{(j)}), \forall X, j$
Up-4 compute the MAX2 by local max-pooling to account for the shifts of parts:
    $\mathrm{MAX2}^{(j)}(X) = \max_{\Delta X} \mathrm{SUM2}^{(j)}(X + \Delta X), \forall X, j$
Up-5 compute the matching score SUM3 of object template $\mathbf{H}$ on the image $\mathbf{I}$ for all locations $\mathcal{X}$:
    $\mathrm{SUM3}(\mathcal{X}) = \sum_{j=1}^{K} \mathrm{MAX2}^{(j)}(\mathcal{X} + X_j), \forall \mathcal{X}$
Up-6 compute the optimum matching score of $\mathbf{H}$:
    $\mathrm{MAX4} = \max_{\mathcal{X}} \mathrm{SUM3}(\mathcal{X})$
Down-1 compute the location of the object on the image $\mathbf{I}$:
    $\widehat{\mathcal{X}} = \arg\max_{\mathcal{X}} \mathrm{SUM3}(\mathcal{X})$
Down-2 compute the perturbations of all parts on the image $\mathbf{I}$:
    $\Delta X_j = \arg\max_{\Delta X} \mathrm{SUM2}^{(j)}(\widehat{\mathcal{X}} + X_j + \Delta X), \forall j$
Down-3 compute the perturbations of Gabor wavelets in all parts on the image $\mathbf{I}$:
    $(\Delta x_i^{(j)}, \Delta\alpha_i^{(j)}) = \arg\max_{\Delta x, \Delta\alpha} \mathrm{SUM1}(\widehat{\mathcal{X}} + X_j + \Delta X_j + x_i^{(j)} + \Delta x, s_i^{(j)}, \alpha_i^{(j)} + \Delta\alpha), \forall i, j$
```

The numbers of clusters are assumed known in these tasks. The image ground-truth category labels are provided for the sake of computing the clustering accuracies but assumed unknown to the learning algorithm. Conditional purity and conditional entropy [14] are used to measure the clustering performance. Let x be the ground-truth category label and y be the inferred category label of an image. The conditional purity is defined as $\sum_y p(y) \max_x p(x|y)$, and the conditional entropy is $\sum_y p(y) \sum_x p(x|y) \log(1/p(x|y))$. Both $p(y)$ and $p(x|y)$ can be estimated from the training images. Higher purity and lower entropy are expected for a better clustering algorithm.

For each task, a model-based clustering is performed by fitting a mixture of hierarchical sparse FRAME models with 2×2 parts in an unsupervised setting. We infer the unknown locations, scales, and rotations of objects and parts, as well as category labels in the learning process. $\tilde{M} = 100$ chains of sampled images are generated to estimate the parameters and normalizing constants. The ranges of perturbations for both Gabor wavelets and part templates are 1 pixel in locations and $\pi/8$ in orientations. Typical template sizes are 100×100. Typical number of Gabor wavelets for object template is 300. The range of rotations for object templates is $\pi/8$. In classification, we search over 4 different resolutions of the images to account for different scales of objects.

We compare our model with (a) the original sparse FRAME model without shiftable parts [16], (b) the sparse FRAME model learned via generative boosting [17], (c) the active basis model [15], (d) two-step EM method [1], (e) k-means with HoG features [3], and (f) And-Or template (AOT) [11]. Table 1 summarizes the comparisons by showing the average clustering accuracies based on 5 repetitions for 12 tasks. The results show that our method performs better than the other models. An improvement is obtained when we generalize the original sparse FRAME model to the hierarchical version by explicitly modeling the part-level deformations.

4.2. Object, part, and key point localization

The inference step (E-step) plays an important role in the unsupervised learning of our model. We evaluate the accuracy of the inference of the hierarchical sparse FRAME model on detection tasks, in comparison to two baseline methods, which are And-Or template (AOT) [11] and part-based latent SVMs (LSVM) [5].

The performance of detection is measured by evaluating the accuracy of localizing key points, parts, and objects. We collect an animal face detection dataset with 8 categories, where each category includes 10 training images and 30 testing images. For each image, roughly twenty identifiable key points are selected manually as pixel-level ground truths by a human labeler. The key points are manually grouped into different semantic parts as ground truths for parts. These key points are not used in training the models. They are used only for evaluating detection performance. Once our model is trained from the training images, we associate each key point with the most likely nearest Gabor wavelet in the template. Similar strategy is used for And-Or template, since its bottom level representational units are

Table 1: Comparison of conditional purity and conditional entropy on clustering tasks

(a) Conditional purity

Task	Ours	Sparse FRAME	Generative Boosting	Active Basis	Two-step EM	k-means +HoG	AOT
1	**0.993**	0.967	0.887	0.667	0.873	0.760	0.813
2	**0.993**	0.980	0.907	0.787	0.820	0.640	0.773
3	**0.993**	0.960	0.973	0.960	0.713	0.793	0.907
4	**0.920**	0.907	**0.920**	0.729	0.720	0.800	0.876
5	**0.996**	0.987	0.982	0.658	0.858	0.840	0.849
6	**1.000**	**1.000**	**1.000**	0.836	0.800	0.933	**1.000**
7	**0.920**	0.917	0.850	0.830	0.773	0.807	0.830
8	**0.993**	0.953	0.920	0.903	0.730	0.780	0.770
9	**0.960**	0.893	0.953	0.923	0.850	0.840	0.880
10	**0.907**	0.797	0.883	0.797	0.869	0.715	0.824
11	**0.960**	0.872	0.923	0.888	0.757	0.784	**0.960**
12	**0.909**	0.907	0.880	0.805	0.813	0.768	0.712
Avg.	**0.962**	0.928	0.923	0.815	0.798	0.788	0.849

(b) Conditional entropy

Task	Ours	Sparse FRAME	Generative Boosting	Active Basis	Two-step EM	k-means +HoG	AOT
1	**0.025**	0.123	0.213	0.585	0.345	0.479	0.371
2	**0.025**	0.066	0.246	0.453	0.404	0.636	0.425
3	**0.025**	0.100	0.082	0.139	0.530	0.434	0.192
4	**0.170**	0.202	0.177	0.594	0.594	0.491	0.305
5	**0.017**	0.050	0.067	0.658	0.302	0.333	0.365
6	**0.000**	**0.000**	**0.000**	0.260	0.355	0.092	**0.000**
7	**0.140**	0.150	0.208	0.321	0.421	0.272	0.313
8	**0.025**	0.118	0.163	0.176	0.552	0.519	0.346
9	0.106	0.191	**0.067**	0.169	0.280	0.265	0.216
10	**0.220**	0.425	0.286	0.447	0.301	0.516	0.359
11	**0.055**	0.191	0.112	0.225	0.486	0.387	0.064
12	**0.189**	0.222	0.290	0.354	0.459	0.477	0.543
Avg.	**0.083**	0.153	0.159	0.365	0.419	0.408	0.291

also Gabor wavelets. For LSVM, each key point is associated with the most likely nearest part, and then we record the most likely location of the key point in that part. With these associations, we can predict the key points via the templates of these models for each testing image. We train the hierarchical sparse FRAME models with 3 × 3 non-overlapping moving parts from aligned images. The ranges of perturbations for both Gabor wavelets and part templates are 2 pixels in locations and $\pi/8$ in orientations. Typical template sizes are 100 × 100. Typical number of wavelets for object template is 370.

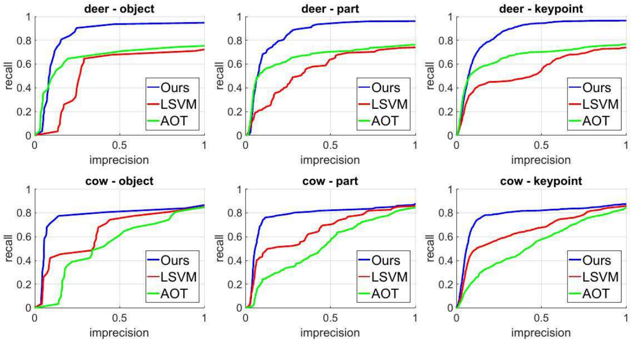
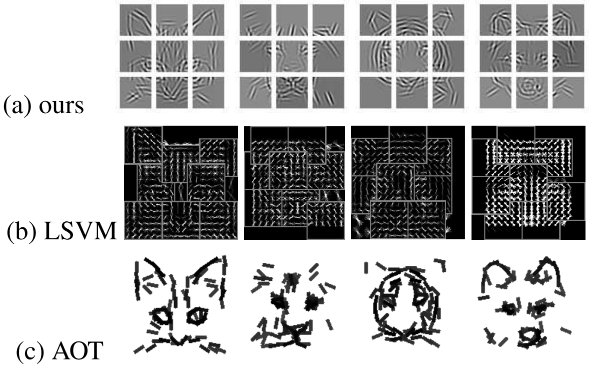


Figure 3: Comparison of imprecision-recall curves of different models for key points, parts, and object in the categories of deer and cow.

We plot imprecision-recall curves and use area under curve (AUC) to measure the performance of the localization of key points. Figure 3 shows the imprecision-recall curves for key points, parts, and object in deer and cow categories. A higher curve indicates larger AUC and better performance. The horizontal axis of the curve is the tolerance for the normalized key point deviation (divided by the size of the template), which is the distance between the predicted location and the ground-truth location of key point. The vertical axis is the recall rate, which is the percentage of the predicted key points within a certain tolerance. For curves of parts and object, the deviation of the part is computed by averaging the deviations of key points inside the part. The deviation of the object is computed by averaging the deviations of all the key points. Table 2 shows the comparisons of accuracies of localization of key points, parts, and object. Our approach outperforms the other methods in terms of average AUC on the detection tasks.

Figure 4 shows a comparison of the templates of hierarchical sparse FRAME models, LSVM models, and And-Or templates learned from cat, lion, tiger, and wolf categories. Figure 5 displays some detection results with the learned models. We can see that our model can locate the objects and internal parts with higher precision.


(a) ours

(b) LSVM

(c) AOT

Figure 4: Comparison of templates learned by different hierarchical models. (a) shows the templates of the hierarchical sparse FRAME models, which are generated by MCMC sampling from the learned models. (b) shows the HoG feature templates for LSVM. (c) displays the symbolic sketch templates for And-Or templates (AOT), where each bar represents the selected Gabor wavelet. (From left to right column: cat, lion, tiger, and wolf.)

4.3. Evaluating unsupervisedly learned models via classification

The model can be used for unsupervised hierarchical feature learning. Supervised classifiers learned on top of these features can be used for classification. We use the LHI-

Table 2: Comparison of AUCs for localization of object, parts and key points

Tasks	object			part			key point		
	ours	AOT	LSVM	ours	AOT	LSVM	ours	AOT	LSVM
cat	**0.954**	0.949	0.700	**0.955**	0.950	0.718	**0.954**	0.949	0.700
lion	**0.879**	0.842	0.834	**0.908**	0.856	0.830	**0.907**	0.857	0.834
tiger	**0.954**	0.948	0.744	**0.956**	0.950	0.744	**0.954**	0.948	0.744
wolf	**0.857**	0.774	0.741	**0.888**	0.826	0.750	**0.887**	0.825	0.741
deer	**0.738**	0.675	0.559	**0.736**	0.673	0.570	**0.738**	0.676	0.565
cougar	**0.960**	0.936	0.831	**0.961**	0.939	0.825	**0.960**	0.938	0.831
cow	**0.757**	0.549	0.663	**0.762**	0.546	0.670	**0.763**	0.556	0.673
bear	**0.769**	0.607	0.744	**0.776**	0.605	0.745	**0.773**	0.611	0.751
Avg.	**0.859**	0.785	0.727	**0.868**	0.793	0.732	**0.867**	0.795	0.730

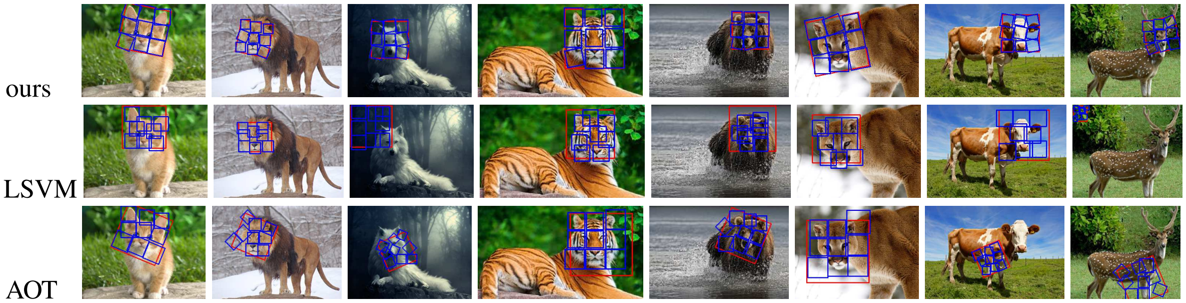


Figure 5: Comparison of localizing objects, parts, and keypoints. From top to bottom, we display the results of hierarchical sparse FRAME models, LSVM models, and AOT templates. For each testing image, the detected bounding boxes for the object (red) and parts (blue) are shown. One example is displayed for each task. Best viewed in color.

Animal-Faces dataset [10], which has around 2200 images of 20 categories of animal faces. Each category exhibits rich appearance variations and shape deformations, e.g., (a) flip and rotation transformations and (b) sub-categories. We randomly select half of the images per category for training and the rest for testing. For each category, we learn a mixture model of 5 or 11 hierarchical sparse FRAME models with 2×2 moving parts in an unsupervised manner. We then combine the object templates from all the learned mixture models into a codebook of $20 \times 5 = 100$ or $20 \times 11 = 220$ codewords. (Each object templte is a codeword.) The maps of template matching scores from all the codewords in the codebook are computed for each image, and then they are fed into spatial pyramid matching (SPM) [18], which equally divides an image into 1, 4, 16 areas, and the maximum scores at different image areas are concatenated into a feature vector. SVM classifiers with ℓ_2 loss are trained on these feature vectors, and are evaluated on the testing data in terms of classification accuracies using the one-versus-all rule.

Table 3 lists a comparison of our models with some baseline methods. For each method, we use the same training/testing data split, the same approach of classifier training, and the same number of clusters in the mixture model learned from each category. The results show that our models outperform the original sparse FRAME models without moving parts and the And-Or templates (AOT) in terms of classification accuracy on this dataset.

Table 3: SVM (with ℓ_2 loss) classification accuracy on features that are unsupervisedly learned from LHI-Animal-Faces dataset with 20 categories

# clusters	AOT	ours w/o parts	ours
5	65.80%	70.62%	**74.33%**
11	62.54%	72.56%	**75.83%**

5. Conclusion

This paper proposes a generative learning framework applied to hierarchical representations of object patterns. Our model is defined as a hierarchical extension of the original sparse FRAME model. The model is capable of capturing geometric deformations and can be learned in an unsupervised manner. It can be visualized by MCMC sampling. Compared to previous generative hierarchical leaning methods, our method performs better in terms of accuracies of localization of object, parts, and key points in detection, object classification, and clustering.

Project page: The data, code, and more results and details can be found at http://www.stat.ucla.edu/~jxie/hsFRAME.html

Acknowledgments

The work is supported by NSF DMS 1310391, DARPA SIMPLEX N66001-15-C-4035, ONR MURI N00014-16-1-2007, and DARPA ARO W911NF-16-1-0579.

References

[1] A. Barbu, T. Wu, and Y. N. Wu. Learning mixtures of bernoulli templates by two-round EM with performance guarantee. *Electronic Journal of Statistics*, 8(2):3004–3030, 2014. 6

[2] J. Dai, Y. Hong, W. Hu, S.-C. Zhu, and Y. Nian Wu. Unsupervised learning of dictionaries of hierarchical compositional models. In *CVPR*, pages 2505–2512. 2

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893, 2005. 6

[4] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987. 3

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 1, 2, 6

[6] S. Fidler and A. Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *CVPR*, pages 1–8, 2007. 2

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, June 2016. 1

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1

[9] P. Schnitzspan, M. Fritz, S. Roth, and B. Schiele. Discriminative structure learning of hierarchical representations for object detection. In *CVPR*, pages 2238–2245, 2009. 2

[10] Z. Si and S.-C. Zhu. Learning hybrid image templates (hit) by information projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1354–1367, 2012. 2, 8

[11] Z. Si and S.-C. Zhu. Learning and-or templates for object recognition and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2189–2205, 2013. 1, 2, 6

[12] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 1

[14] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine. Unsupervised object discovery: A comparison. *International Journal of Computer Vision*, 88(2):284–302, 2010. 6

[15] Y. N. Wu, Z. Si, H. Gong, and S.-C. Zhu. Learning active basis model for object detection and recognitio. *International Journal of Computer Vision*, 90:198–235, 2010. 2, 6

[16] J. Xie, W. Hu, S.-C. Zhu, and Y. N. Wu. Learning sparse frame models for natural image patterns. *International Journal of Computer Vision*, pages 1–22, 2014. 1, 2, 3, 6

[17] J. Xie, Y. Lu, S.-C. Zhu, and Y. N. Wu. Inducing wavelets into random fields via generative boosting. *Journal of Applied and Computational Harmonic Analysis*, 2015. 1, 3, 5, 6

[18] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, pages 1794–1801, 2009. 8

[19] L. Younes. On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates. *Stochastics: An International Journal of Probability and Stochastic Processes*, 65(3-4):177–228, 1999. 3

[20] L. Zhu, Y. Chen, A. Yuille, and W. Freeman. Latent hierarchical structural learning for object detection. In *CVPR*, pages 1062–1069. IEEE, 2010. 2

[21] L. L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille. Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. In *ECCV*, pages 759–773, 2008. 2

[22] S.-C. Zhu, D. Mumford, et al. A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, 2(4):259–362, 2007. 2