Generative Hierarchical Structure Learning of Sparse FRAME Models

Anonymous CVPR submission

Paper ID 3692

Abstract

This paper proposes a framework for generative learning of hierarchical structures of visual objects, based on training hierarchical random field models. The resulting model, which we call the structured sparse FRAME model, is a straightforward variation on decomposing the original sparse FRAME model into multiple parts that are allowed to shift their locations, orientations and scales, so that the resulting model becomes a reconfigurable template. The model is trained in an unsupervised manner by iterating the following two steps: (1) Inference: Given the current model, we match the model to each image by inferring the location, rotation, and scale of each part template as well as the location of the object template by recursive sum-max maps. (2) Re-learning: Given the inferred geometric configurations of the parts, re-learn the model structure and parameters by maximum likelihood estimation via stochastic gradient algorithm. Experiments show that an improvement is obtained in generalizing the original sparse FRMAE model to the structured version, the proposed model is capable of learning meaningful template and interpretable structure, and the learned templates are useful for object detection 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. Some machine learning models are opaque, non-intuitive, and difficult for people to understand. Explainable model will be highly desired, if users are to understand, interpret, trust, and effectively manage the behavior of the model. Therefore, discovery of explainable and predictive models from visual data becomes a central problem in computer vision and artificial intelligence.

Structured models with hierarchical and compositional representations, such as deformable part-based model [5], stochastic And-Or template [9], have shown to be a powerful basis for achieving impressive model accuracy and explainability. The success lies in that they are capable of learning reconfigurable representation to deal with both structural and appearance variations of objects.

These structured models can be paired with either discriminative learning method or generative learning method. Discriminative structure learning seeks to identify and weigh the most discriminant features and structures for explaining the object categories, while the generative structure learning enable us to learn the parameters and interpretable structures for explaining the image data instead of predicting the object categories. Moreover, generative structure learning is not only important for making the model explainable, it is also of fundamental importance for unsupervised learning, where unlabeled images are provided, since the labeled images are scarce and expensive to acquire.

Recently, Xie et al. proposed a sparse FRAME model [12, 13] as a generative model for representing natural image patterns. It can be considered as a deformable template consisting of a small number of perturbable Gabor wavelets (sketches) at selected locations, scales, and orientations. The model is explainable in a sense that the learned knowledge can be visualized by Markov chain Monte Carlo (MCMC) sampling. However, due to the lack of reconfigurability, the sparse FRAME models can only deal with small deformation (e.g., edge perturbation), and may fail when there exists a large structural change (e.g., part deformation). To address this limitation, we propose to extend the original sparse FRAME model to a structured version, which we call the structured sparse FRAME model, by explicitly modeling part-level structure and deformation for an object.

Method overview. (1) Representation: The structured 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 orientations 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 is also a stochastic reconfigurable template that corresponds to a set of valid configurations or templates. The model inference is to determine a cer-

tain configuration for the object in a testing image, which can be achieved by a bottom-up and top-down dynamic programming procedure. (3) Generative learning: The model is learned in a generative manner in the sense that the learning is preformed by maximum likelihood estimation and also it involves synthesizing image patterns via MCMC sampling. (4) Unsupervised learning: As the model is a fully generative model, it can be learned in an unsupervised manner, where all the locations, scales, and orientations of the object, parts, and edges (Gabor wavelets) are unknown, by an EM-type algorithm that alternates model inference and part re-learning. A mixture of structured sparse FRAME models can also be learned in an unsupervised manner as a And-OR hierarchical structure.

Figure 1 illustrates the basic idea of the structured sparse FRAME model and the mixture models. A two-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 the inference of the structured sparse FRAME model on a testing image, with bounding boxes showing the inferred locations, orientations, and scales of the object (red) and parts (blue). Figure 1(c) illustrates a mixture of structured sparse FRAME model 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 means OR node for selection. The blue empty square 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 in the current And-Or graph.

Related work. Most existing methods to learn hierarchical structures of object patterns are usually supervised, such as [5, 15, 7] , where labels are given. In this paper, we learn hierarchical structures of objects from unaligned images without annotations. Our work is similar to [6, 16], which also learn structures of hierarchical compositions of Gabor wavelets or edgelets. They learn the structures via a bottom-up layer-by-layer scheme, where once the lower layers are learned, they are fixed in the learning of higher layers. In contrast, our iterative structure learning algorithm re-learns the object template as well as the part-templates, and re-selects the Gabor wavelets and the part templates at each iteration. Our work is also related to And-Or template [9] and hierarchical compositional model [2]. To represent visual parts in the structures, the former uses hybrid image template (HIT) [8], and the latter uses active basis template (ABT) [11]. 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 MCMC computation in learning. In our model, parts are represented by sparse FRAME models, which do not make above simplifying assumption, so that our model is more mathematically rigorous and is capable of visualizing the learned model by synthesizing patterns via MCMC simulation, which makes our model more explainable.

2. Background of sparse FRAME model

This section reviews the background of the sparse FRAME models [12], which serves as the foundation of the structured 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 filer 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 (or natural parameters of the above exponential family distribution), $\langle \mathbf{I}, B_{x,s,\alpha} \rangle$ 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 images $\{\mathbf{I}_m, m = 1, ..., M\}$ from the same object category, where M is the number of training images, we can learn the unknown weight parameters λ by maximizing the log-likelihood $L(\lambda) = \sum_{m=1}^{M} \log p(\mathbf{I}_m; \lambda)/M$, and thus leading to the stochastic gradient ascent algorithm [14]

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

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

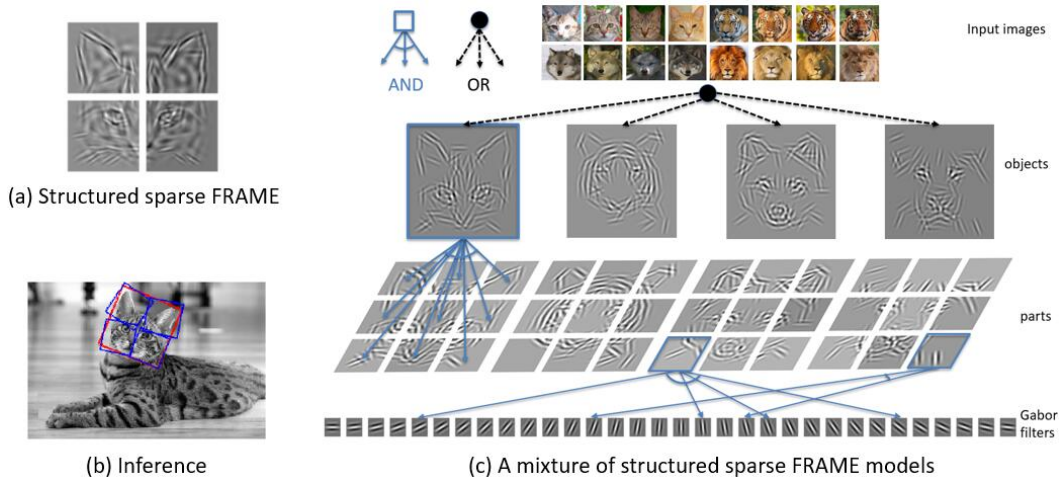


Figure 1: (a) Structured sparse FRAME model: A structured sparse FRAME model with 2×2 part is learned from roughly aligned observed images. The layout of parts is visualized by displaying the synthesized images generated by the 4 part models that composes the object model. (b) Inference: A testing image with bounding boxes showing the inferred locations, orientations, and scales of the object (red) and parts (blue) by the learned model. (c) A mixture of structured sparse FRAME models is learned by an EM-like 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 choices and an AND node (in blue) represent 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.

$\log Z(\lambda^{(0)}) = 0$, 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 wavelets are selected from the given dictionary. We can explicitly write the sparsified model as

$$p(\mathbf{I}; \mathbf{B}, \lambda) = \frac{1}{Z(\lambda)} \exp\left(\sum_{i=1}^{n} \lambda_i |\langle \mathbf{I}, B_{x_i,s_i,\alpha_i} \rangle| \right) q(\mathbf{I}), \quad (4)$$

where $\mathbf{B} = (B_{x_i,s_i,\alpha_i}, i = 1, ..., n)$ are the n wavelets selected from a given dictionary (n is assumed 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 the basis functions and the estimation of the corresponding weight parameters.

A two-stage learning algorithm [12] or a single-stage learning algorithm [13] can be used to train the sparse FRAME model. In this paper, we will use the two-stage learning algorithm.

The two-stage learning algorithm includes 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 deformation, 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 coefficients of the selected wavelets. The selection is accomplished by minimizing the least squares reconstruction error for all the training images

$$\sum_{m=1}^{M} \|\mathbf{I}_m - \sum_{i=1}^{n} c_{m,i} B_{x_i+\Delta x_{m,i},s_i,\alpha_i+\Delta\alpha_{m,i}}\|^2, \quad (5)$$

The minimization of (5) can be accomplished 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 log-likelihood of image $\mathbf{I}$, which is computed by

$$L(\mathbf{I}|\mathbf{B}) = \sum_{i=1}^{n} \lambda_i \max_{\Delta x, \Delta\alpha} |\langle \mathbf{I}, B_{x_i+\Delta x, s_i, \alpha_i+\Delta\alpha}\rangle| - \log Z(\lambda), \quad (6)$$

serves as the template matching score in object recognition.

3. Structured Sparse FRAME Models

Hierarchical structure. In this section, we will extend the original sparse FRAME model to a structured version that we call the structured sparse FRAME model, which is a composition of shiftable parts at different locations, scales, and, orientations, while parts themselves are compositions of a number of shiftable basis functions at different locations, scales, and orientations. The structured sparse FRAME model is a hierarchical probability distribution defined on $\mathbf{I}$,

$$p(\mathbf{I}; \mathbf{H}, \lambda) = \frac{1}{Z(\lambda)} \exp\left(\sum_{j=1}^{K} \sum_{i=1}^{n_j} \lambda_i^{(j)} |\langle \mathbf{I}, B_{x_i^{(j)}, s_i^{(j)}, \alpha_i^{(j)}}\rangle| \right) q(\mathbf{I}), \quad (7)$$

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 Gabor wavelets. Each group represents a part template. n_j is the number of Gabor wavelets 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 Gabor from a given dictionary to form a hierarchy and estimating their associated parameters.

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 deformation. Learning model (7) from training images requires inference of the deformations of parts and basis functions.

3.1. Hierarchical shared sparse coding

As the original sparse FRAME model corresponds to the shared sparse coding, the structured sparse FRAME model in (7) corresponds to the hierarchical shared sparse coding, with which we can easily explicitly deal with the model deformation.

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

$$\mathbf{I} = \sum_{i=1}^{n} c_i B_{X+x_i+\Delta x_i, s_i, \alpha_i+\Delta\alpha_i} + \epsilon = C\mathbf{B}_X + \epsilon, \quad (8)$$

where $C = (c_i, i = 1, ..., n)$, $\mathbf{B}_X = (B_{X+x_i+\Delta x_i, s_i, \alpha_i+\Delta\alpha_i}, i = 1, ..., n)$ 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

$$\sum_{i=1}^{n} \lambda_i \max_{\Delta x, \Delta\alpha} |\langle \mathbf{I}, B_{X+x_i+\Delta x, s_i, \alpha_i+\Delta\alpha}\rangle| - \log Z(\lambda) \quad (9)$$

where $\lambda = (\lambda_i, i = 1, ..., n)$ are weight parameters of the template and $\log Z(\lambda)$ is the normalizing constant.

We can generalize the representation $\mathbf{B}_X$ by using $\mathbf{B}_{X,S,A}$ to denote the part template at location X, scale S, and orientation A, in which we take rotation and scale of the template into account. 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 structured 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 orientation of the j-th part template in the object template $\mathbf{H}$. Then we can represent the image $\mathbf{I}_m$ by a template of K parts:

$$\mathbf{I}_m = \sum_{j=1}^{K} C_{m,j} \mathbf{B}^{(j)}_{X_j, S_j, A_j} + \epsilon_m, \quad (10)$$

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 parts are allowed to perturb their locations, scales, and orientations to account for the structural deformation in the image, we can extend (10) to

$$\mathbf{I}_m = \sum_{j=1}^{K} C_{m,j} \mathbf{B}^{(j)}_{X_j+\Delta X_{m,j}, S_j+\Delta S_{m,j}, A_j+\Delta A_{m,j}} + \epsilon_m, \quad (11)$$

where $(\Delta X_{m,j}, \Delta S_{m,j}, \Delta A_{m,j})$ are perturbations of the location, scale, and orientation 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 within a squared region centered at the part, $\Delta S_{m,j} \in \{-1, 0, 1\} \times 0.1$, and $\Delta A_{m,j} \in \{-1, 0, 1\} \times \pi/16$). We use $L(\mathbf{I}_m|\mathbf{B}_{X_j, S_j, A_j})$ to denote the log-likelihood of part $\mathbf{B}_{X_j, S_j, A_j}$. Also, we assume parts do not overlap with each other, i.e., the subspaces spanned by parts are orthogonal to each other, then the log-likelihood score of the image $\mathbf{I}_m$ given the object template $\mathbf{H}$ is

$$L(\mathbf{I}_m|\mathbf{H}) = \sum_{j=1}^{K} \max_{\Delta X, \Delta S, \Delta A} L(\mathbf{I}_m|\mathbf{B}^{(j)}_{X_j+\Delta X, S_j+\Delta S, A_j+\Delta A}). \quad (12)$$

3.2. Learning and inference

Objective function. The learning of the model is to learn the K part templates $\{\mathbf{B}^{(j)}, j = 1, ..., K\}$ and their layout $\{(X_j, S_j, A_j), j = 1, ..., K\}$ in $\mathbf{H}$ from the training images $\{\mathbf{I}_m\}$, while inferring the part perturbations $(\Delta X_{m,j}, \Delta S_{m,j}, \Delta A_{m,j})$, by maximizing the objective function which is defined as the sum of the log-likelihood given $\mathbf{H}$ over all the training images,

$$\sum_{m=1}^{M} \sum_{j=1}^{K} L(\mathbf{I}_m | \mathbf{B}^{(j)}_{X_j+\Delta X_{m,j}, S_j+\Delta S_{m,j}, A_j+\Delta A_{m,j}}), \quad (13)$$

subject to the constraint that there are no overlapping parts in each $\mathbf{I}_m$. This can be achieved by iterating the model inference step and the model re-learning step.

Step 1: Inference of structured sparse FRAME model. Given the structured sparse FRAME model $\mathbf{H} = \{\mathbf{B}^{(j)}_{X_j, S_j, A_j}, j = 1, .., K\}$ and a testing image $\mathbf{I}$, the inference of the model is to infer the location $\widehat{\mathcal{X}}$ of the object template in the testing image $\mathbf{I}$ by

$$\widehat{\mathcal{X}} = \arg\max_{\mathcal{X}} \sum_{j=1}^{K} \max_{\Delta X, \Delta S, \Delta A} L(\mathbf{I} | \mathbf{B}_{\mathcal{X}+X_j+\Delta X, S_j+\Delta S, A_j+\Delta A}), \quad (14)$$

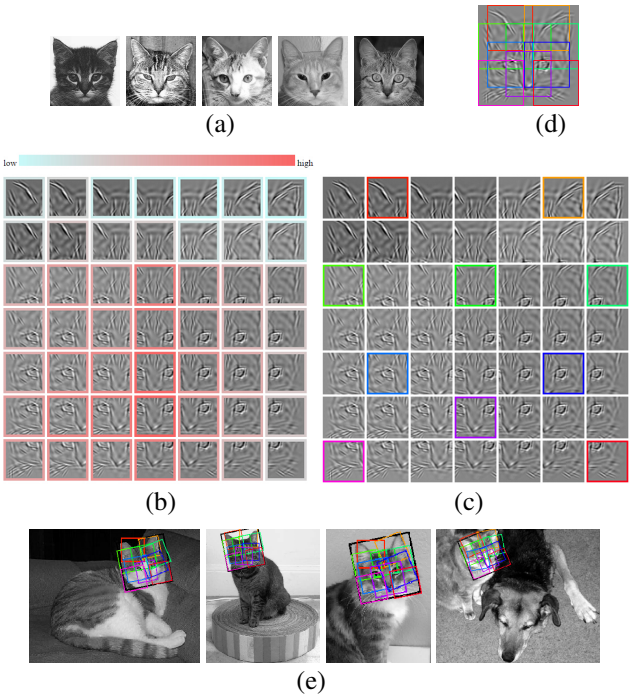
and the perturbations in locations, scale, and orientations of K parts by

$$(\Delta X_j, \Delta S_j, \Delta A_j) = \arg\max_{\Delta X, \Delta S, \Delta A} L(\mathbf{I} | \mathbf{B}_{\widehat{\mathcal{X}}+X_j+\Delta X, S_j+\Delta S, A_j+\Delta A}), \quad (15)$$

as well as the perturbations of all basis functions in each part. The inference can be efficiently accomplished by recursive sum-max maps shown in Algorithm 1, which involves a bottom-up/top-down inference procedure. For notation simplicity, we do not take scales and orientations of both the object template and its part templates into account in Algorithm 1.

Step 2: Structure learning and parameter estimation. Given the inferred structural deformations (i.e., object bounding box and part bounding boxes), we can firstly align the objects and parts by morphing the corresponding image patches. We learn an original sparse FRAME model on the well aligned training images, and then dividing the template into a collection of highly overlapping part templates, then we greedily pursue a small number of part templates according to their log-likelihood scores to enforce that the selected part templates only have limited overlap. For simplicity, we can divide the object template into $d \times d$ non-overlapping parts and include them all into the model without selection.

Figure 2 illustrates the inference step and the structure learning step. Figure 2(a) displays 5 aligned training images where we assume there are no deformations in parts and objects. Figure 2(b) displays a collection of 49 part templates



(a) (b) (c) (d) (e)

Figure 2: Model inference and structure learning. (a) Well aligned training images. (b) A collection of learned part templates with 80% overlapping. (c) Selected part templates. (d) Layout of selected parts. (e) Inference of the objects by the learned model, where bounding boxes indicate the detected objects and parts.

with 80% overlapping learned from the 5 aligned training images. We visualize each part template by displaying its synthesized image. The log-likelihood score of each part template in the collection is indexed by color, where redder color indicates higher score. Figure 2(c) shows 10 selected parts with color bounding boxes in the collection. Each selected part is associated with a unique color. Figure 2(d) shows the layout of the selected parts in the whole template by putting all together according to their selected locations. Figure 2(e) displays the results of the inference of objects by the learned model. The black bounding boxes locate the detected objects, while the others locate parts.

Unsupervised learning by EM-type algorithm. In the setting of unsupervised learning, the training images are non-aligned in the sense that the location, scale, and orientation of the object in each image are unknown. As to the layout of parts in the whole template, for simplicity, we will not precisely pursuit parts from a collection of highly overlapping part templates, instead, we divide the whole template into $d \times d$ non-overlapping parts regularly and include them all into the template. The unsupervised learning of the structured sparse FRAME models can be achieved by iterating the following two steps: (1) Model inference: Given

the current structured sparse FRAME model, we match the model to each image by inferring the location, rotation, and scale of each part template as well as the overall location of the object template by recursive sum-max maps shown in Algorithm 1. (2) Part re-learning: Given the inferred geometric configurations of the parts, re-learn sparse FRAME model for each part, which includes basis function selection and parameter estimation.

Figure 3 displays one example of unsupervised learning of structured sparse FRAME model. The object template is divided into 2×2 parts. The size of each part template is 50×50. The number of non-aligned training images is 26. The total number of selected wavelets is 300. The number of iteration is 6. (a) displays 2×2 parts of synthesized images generated by the learned model. (b) displays 2×2 parts of sketch templates illustrate the selected wavelets by shared matching pursuit in each part. (c) illustrates 12 examples of 26 non-aligned training images from cat category, with bounding boxes showing the inferred location, rotation, and scaling of the object (black) and parts (different colors) after the model is learned. (d) shows the inference results of the leaned model on 2 testing images, with bounding box indicating the geometric configuration of the detected object (black) and parts (different colors).

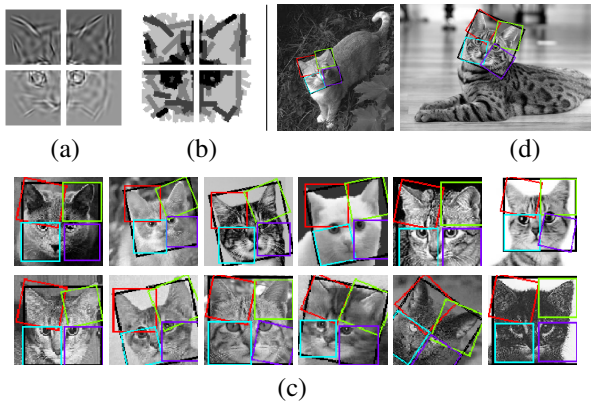


Figure 3: Unsupervised learning of structured sparse FRAME model. (a) 2×2 parts of synthesized images generated by the learned model. (b) 2×2 parts of sketch templates. (c) 12 examples of 26 non-aligned training images from cat category, with bounding boxes showing the inferred location, rotation, and scaling of the object (black) and parts (different colors) after the model is learned. (d) Inference results of the leaned model on 2 testing images.

4. Experiments

4.1. Object, part, and keypoint localization

The inference step plays an important role in the unsupervised learning of the structure sparse FRAME models.

Algorithm 1 Inference algorithm for structured sparse FRAME model

Input: A structured sparse FRAME model $\mathbf{H} = \{\mathbf{B}^{(j)}_{X_j}, j = 1, .., K\}$, where $\mathbf{B}^{(j)} = \{B_{x_i^{(j)}, s_i^{(j)}, \alpha_i^{(j)}}, i = 1, ..., n_j\}$, and a testing image $\mathbf{I}$

Output: Location $\widehat{\mathcal{X}}$ of the whole 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 $\{(\Delta x_i^{(j)}, \Delta\alpha_i^{(j)}), i = 1, ..., n_j, j = 1, ..., K\}$.

1: **Up-1** compute feature map SUM1 of Gabor B on $\mathbf{I}$ for all locations x, scales s, and orientations α:

$$\text{SUM1}(x, s, \alpha) = |\langle \mathbf{I}, B_{x,s,\alpha} \rangle|, \forall x, s, \alpha$$

2: **Up-2** compute MAX1 by local max-pooling to account for the shifts of Gabor wavelets:

$$\text{MAX1}(x, s, \alpha) = \max_{\Delta x, \Delta\alpha} \text{SUM1}(x + \Delta x, s, \alpha + \Delta\alpha), \forall x, s, \alpha$$

3: **Up-3** compute part matching score SUM2 of $\mathbf{B}^{(j)}$ on the image $\mathbf{I}$ for all locations X:

$$\text{SUM2}^{(j)}(X) = \sum_{i=1}^{n_j} \lambda_i^{(j)} \text{MAX1}(X + x_i^{(j)}, s_i^{(j)}, \alpha_i^{(j)}) - \log Z(\lambda^{(j)}), \forall X, j$$

4: **Up-4** compute the MAX2 by local max-pooling to account for the shifts of parts:

$$\text{MAX2}^{(j)}(X) = \max_{\Delta X} \text{SUM2}^{(j)}(X + \Delta X), \forall X, j$$

5: **Up-5** compute the matching score SUM3 of template $\mathbf{H}$ on the image $\mathbf{I}$ for all locations $\mathcal{X}$:

$$\text{SUM3}(\mathcal{X}) = \sum_{j=1}^{K} \text{MAX2}^{(j)}(\mathcal{X} + X_j), \forall \mathcal{X}$$

6: **Up-6** compute the optimum matching score of $\mathbf{H}$:

$$\text{MAX4} = \max_{\mathcal{X}} \text{SUM3}(\mathcal{X})$$

7: **Down-1** compute the location of the template $\mathbf{H}$ on the image $\mathbf{I}$:

$$\widehat{\mathcal{X}} = \arg\max_{\mathcal{X}} \text{SUM3}(\mathcal{X})$$

8: **Down-2** compute the perturbations of all parts on the image $\mathbf{I}$:

$$\Delta X_j = \arg\max_{\Delta X} \text{SUM2}^{(j)}(\widehat{\mathcal{X}} + X_j + \Delta X), \forall j$$

9: **Down-3** compute the perturbations of Gabor wavelets in all parts on the image $\mathbf{I}$:

$$(\Delta x_i^{(j)}, \Delta\alpha_i^{(j)}) = \arg\max_{\Delta x, \Delta\alpha} \text{SUM1}(\widehat{\mathcal{X}} + X_j + \Delta X_j + x_i^{(j)} + \Delta x, s_i^{(j)}, \alpha_i^{(j)} + \Delta\alpha), \forall i, j$$

We evaluate the accuracy of the inference of our model on detection tasks, in comparison to AND-OR template model (AOT) [9] and part-based latent SVMs model (LSVM) [5].

We measure the performance of detection by evaluating the accuracy of localizing keypoints, 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 identifi-

Table 1: 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

able keypoints are selected manually as pixel-level ground truths by a human labeler. The keypoints are also manually grouped into different semantic parts as ground truths for parts. All these key points are not used in training models, and only for evaluating detection performance. Once AND-OR template 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 our model since the bottom level of representational units in our model are also Gabor wavelets. For latent SVM models, we associate each key point with the most likely nearest part and record the most likely location of the key point relative to the center of that part. With these association, we can then predict the pixel-level key points via the templates of these models for each testing image. We trained structured sparse FRAME models with 3×3 non-overlapping moving parts. 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.

We draw imprecision-recall curves and use area under curve (AUC) to measure the performance of the localization of key points. A higher curve indicates larger AUC and better performance. The horizontal axis of the curve is the tolerance for the normalized key point deviation (dividing 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 associated with the part. The deviation of the object is computed by averaging the deviations of all key points. Table 1 shows the comparisons of accuracies of localization of key points, parts, and object. Our approach outperforms the other methods in terms of localization accuracies on the detection tasks.

Figure 4 shows the comparison of the templates of structured sparse FRAME models, LSVM models, and And-Or Templates learned from cat, lion, tiger, and wolf categories. (a) displays the templates of structured sparse FRAME models. They are synthesized pixel-level image patterns sampled from the part models via MCMC. (b) displays the HoG feature template for LSVM models. (c) shows the sketch templates, where each symbolic bar represents each selected Gabor wavelet. Figure 5 shows the detection results on several challenging testing images with the learned models. From these examples we can see that the structured sparse FRAME models can locate the object and internal parts with higher precision, so that they can obtain higher localization accuracies.

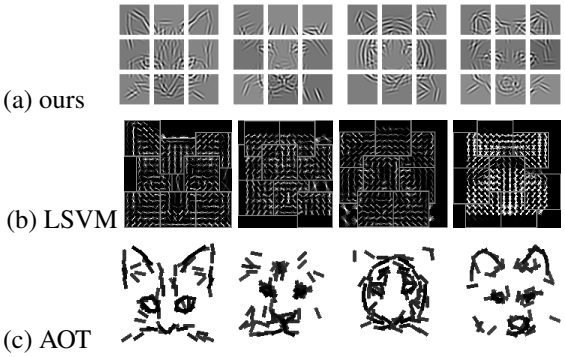


Figure 4: Comparison of template learned by different models. (a) shows the templates of the structured sparse FRAME models, which are generated by sampling from the learned model via HMC. (b) shows the HoG templates for LSVM. (c) shows the symbolic sketch templates for And-Or templates, where each bar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

4.2. Evaluating mixture models by clustering tasks

We evaluate the accuracy of the unsupervisd learning of the mixture models on clustering tasks. We test it on a benchmark clustering dataset [13] that consists of 12 clustering tasks, where each task has 2 or 5 clusters and each

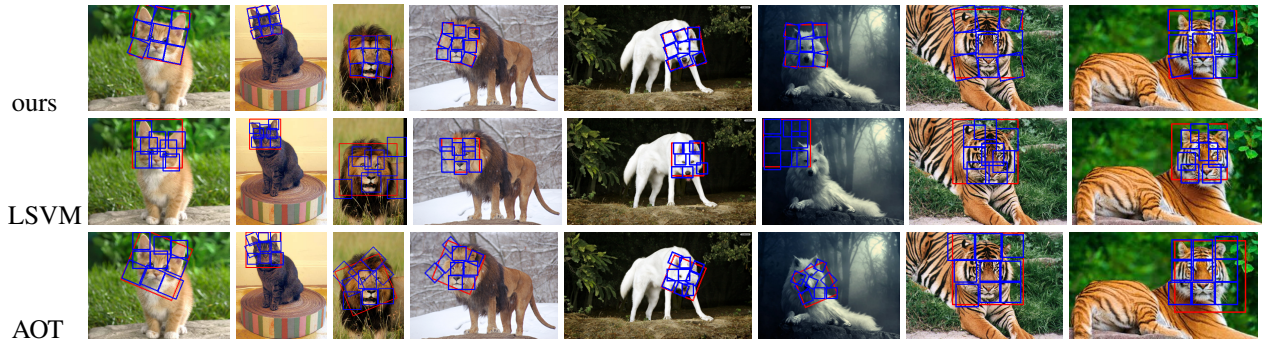


Figure 5: Comparison of localizing objects, parts, and keypoints. From top to bottom, we display the results of structured sparse FRAME models, LSVM models, and AOT templates. For each testing image, the detected bounding boxes for objects (red) and parts (blue) are shown. Best viewed in color.

cluster has 15 images. The numbers of clusters are assumed known in these task. 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 [10] are used to measure the clustering performance. Let x be the ground-true 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 $\sum_y p(y) \sum_x p(x|y) \log(1/p(x|y))$, where 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. We fit a mixture of structured sparse FRAME model with 3×3 non-overlapping moving parts in an unsupervised setting, where the locations, orientations, and scales of objects and parts are unknown and need to be inferred in the learning process. $\tilde{M} = 100$ chains of sampled images are generated to estimate the parameters and normalizing constants. The other parameter setting is the same as the above experiment. We compare our model with (a) the sparse FRAME model via generative boosting [13], (b) the active basis model [11], (c) k-means with HoG features [3], and (d) two-step EM [1]. Table 2 summarize the comparisons by showing the average clustering accuracies based on 5 repetitions for 12 tasks. The results show that our method performs better than other representations. Especially, an improvement is obtained when we generalize the original sparse FRMAE model into the structured version by explicitly modeling the part-level deformation.

Table 2: Comparison of conditional purity and conditional entropy on clustering

(a) Conditional purity

Tasks	ours	sparse FRAME	active basis	two-way EM	HoG k-means
1	**0.9800**	0.8867	0.6667	0.8733	0.7600
2	**0.9933**	0.9067	0.7867	0.8200	0.6400
3	**0.9867**	0.9733	0.9600	0.7133	0.7933
4	**0.9956**	0.9200	0.7289	0.7202	0.8000
5	**0.9867**	0.9822	0.6578	0.8578	0.8400
6	**1.0000**	1.0000	0.8355	0.8000	0.9333
7	**0.9267**	0.8500	0.8300	0.7734	0.8067
8	**0.9300**	0.9200	0.9033	0.7300	0.7800
9	**0.9700**	0.9533	0.9233	0.8500	0.8400
10	**0.8901**	0.8827	0.7973	0.8693	0.7147
11	**0.9321**	0.9227	0.8880	0.7573	0.7840
12	**0.8867**	0.8800	0.8053	0.8133	0.7680
Avg.	**0.9565**	0.9231	0.8152	0.7981	0.7883

(b) Conditional entropy

Tasks	ours	sparse FRAME	active basis	two-way EM	HoG k-means
1	**0.0660**	0.2130	0.5846	0.3451	0.4786
2	**0.0249**	0.2457	0.4534	0.4040	0.6355
3	**0.0499**	0.0821	0.1390	0.5296	0.4337
4	**0.0166**	0.1773	0.5941	0.5935	0.4914
5	**0.1240**	0.0665	0.6581	0.3022	0.3327
6	**0.0000**	0.0000	0.2595	0.3548	0.0924
7	**0.1432**	0.2080	0.3211	0.4208	0.2724
8	**0.1525**	0.1625	0.1758	0.5521	0.5189
9	**0.0612**	0.0669	0.1693	0.2804	0.2649
10	**0.2259**	0.2859	0.4472	0.3013	0.5155
11	**0.1091**	0.1121	0.2249	0.4861	0.3866
12	**0.2790**	0.2902	0.3537	0.4586	0.4772
Avg.	**0.1044**	0.1592	0.3651	0.4190	0.4083

5. Conclusion

This paper proposes a novel generative structure learning framework applied to hierarchical representation of object patterns. Our model is defined as a structured learning extension to the sparse FRAME model. The model is capable of capturing structural deformation and is learned in an unsupervised manner. It can be visualized by MCMC sampling, which make it much more explainable. Compared to existing structure leaning method, our method performs better in terms of accuracies of localization of object, parts, and keypoints in detection and object clustering.

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. 8

[2] J. Dai, Y. Hong, W. Hu, S.-C. Zhu, and Y. Nian Wu. Unsupervised learning of dictionaries of hierarchical compositional models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2505–2512, 2014. 2

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893. IEEE, 2005. 8

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

[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 *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 2

[7] P. Schnitzspan, M. Fritz, S. Roth, and B. Schiele. Discriminative structure learning of hierarchical representations for object detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2238–2245. IEEE, 2009. 2

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

[9] 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

[10] 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. 8

[11] 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, 8

[12] 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

[13] 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, 7, 8

[14] 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. 2

[15] L. Zhu, Y. Chen, A. Yuille, and W. Freeman. Latent hierarchical structural learning for object detection. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1062–1069. IEEE, 2010. 2

[16] L. L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille. Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. In *European Conference on Computer Vision*, pages 759–773. Springer, 2008. 2