Learning Hybrid Image Templates (HIT) by Information Projection

Zhangzhang Si and Song-Chun Zhu

Abstract—This paper presents a novel framework for learning a generative image representation – the *hybrid image template* (HIT) from a small number (i.e, $3 \sim 20$) of image examples. Each learned template is composed of, typically, $50 \sim 500$ image patches whose geometric attributes (location, scale, orientation) may adapt in a local neighborhood for deformation, and whose appearances are characterized respectively by four types of descriptors: local sketch (edge or bar), texture gradients with orientations, flatness regions, and colors. These heterogeneous patches are automatically ranked and selected from a large pool according to their information gains using an information projection framework. Intuitively, a patch has a higher information gain if (i) its feature statistics is consistent within the training examples and is distinctive from the statistics of negative examples (i.e. generic images or examples from other categories); and (ii) its feature statistics has less intra-class variations. The learning process pursues the most informative (for either generative or discriminative purpose) patches one at a time and stops when the information gain is within statistical fluctuation. The template is associated with a well-normalized probability model that integrates the heterogeneous feature statistics. This automated feature selection procedure allows our algorithm to scale up to a wide range of image categories, from those with regular shapes to those with stochastic texture. The learned representation captures the intrinsic characteristics of the object or scene categories. We evaluate the hybrid image templates on several public benchmarks, and demonstrate classification performances on par with state-of-art methods like HoG+SVM, and when small training sample sizes are used the proposed system shows a clear advantage.

Index Terms—Image Representation, Deformable Templates, Information Projection, Visual Learning, Statistical Modeling

1 Introduction

1.1 Motivation and objective

If asked what a tomato looks like, one may describe it as an object with "round shape, red color, smooth surface, ...". This description represents different visual features that are common to tomato and distinct from other objects. In this paper, we present a novel framework for learning a fully generative image representation – the *hybrid image template* (HIT) from a small number (i.e. $3 \sim 20$) of image examples. Figure 1 shows two hybrid image templates learned from a few tomato and pear examples respectively. Each template is composed of a number of image patches (typically $50 \sim 100$) whose geometric attributes (location, scale, orientation) may adapt in a local neighborhood to account for deformations and variations, and whose appearances are characterized respectively by four types of descriptors: local sketch (edge or bar), texture gradients (with orientation field), flatness regions (smooth surface and lighting), and colors. Naturally, there are large variations in the representations of different classes, for example, teapots may have common shape outline, but do not have common texture or color, the hedgehog in Figure 2 has distinct texture and shape, but its color is often less distinguishable from its background. So the essence of our learning framework is to automatically select, in a principled way, informative patches from a large pool and compose them into a template with a well-normalized probability model. It is fast to learn a HIT. For 100 training images, it takes about one minute on a standard PC.

The authors are with the Department of Statistics, University of California, Los Angeles. Email $\{zzsi, sczhu\}$*@stat.ucla.edu*

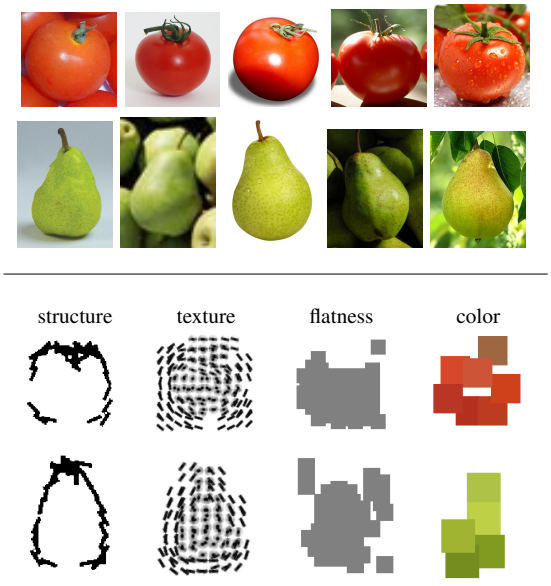


Fig. 1: Learning Hybrid Image Templates (HIT) from a few examples for tomato and pear respectively. Each template consists of a number of image patches: sketches (shape elements), texture / gradients, flat area (smooth surface and lighting), and colors.

In the following, we outline the four major issues in learning the hybrid image templates.

1), The space of atomic image patches and a hybrid dictionary. The HIT quantizes the space of small image

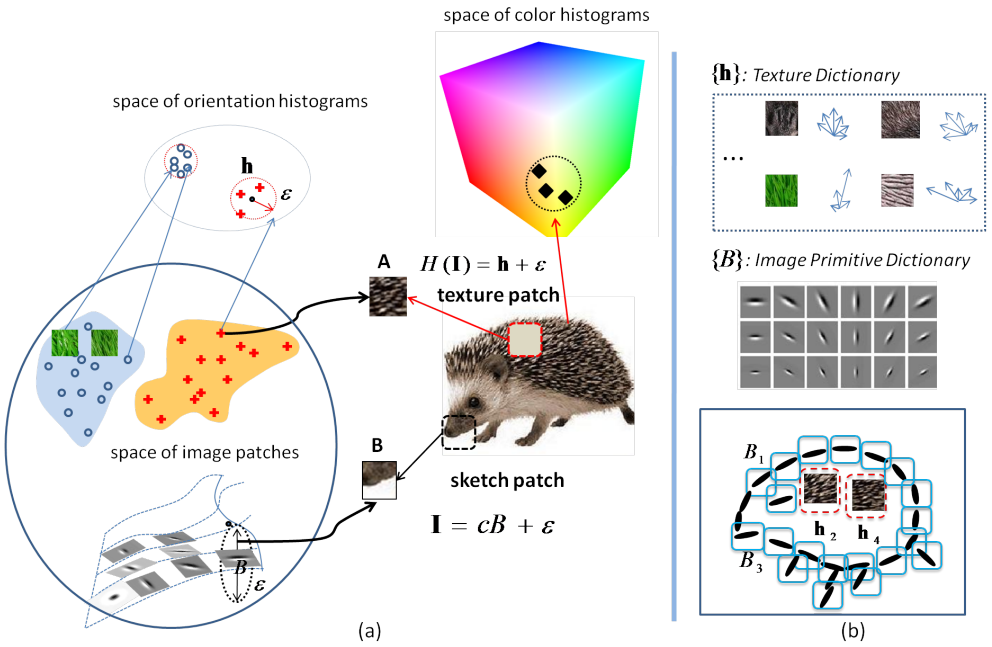


Fig. 2: Analyzing the image space and its composition. A hedgehog image may be seen as a collection of local image patches which are from different subspaces of varying dimensions and complexities.

patches (say $11^2 \sim 19^2$ pixels) into small subspaces that can be explained by heterogeneous feature prototypes (sketch, texture, flatness and color). The total number of prototypes can be very large, and a rough computation estimates around 10^6 of them. It learns an image template (of a larger size, *e.g.* 128 by 128) composed of a small number (*e.g.* 50 $\sim$ 500) of feature prototypes explaining small patches at different locations. This is a very sparse representation, given that the number of overlapping patches together with the candidate prototypes explaining them easily form a huge over-complete dictionary of more than 10^8 in size.

It is illustrative to look at the hedgehog example in Figure 2. Patch A in the body of the hedgehog is a texture pattern and belongs to a very high dimensional subspace. Patch B at the nose is an edge primitive and is from a low dimensional subspace. Besides textures and primitives, there are also flat patches which do not have structures, such as surfaces, walls, and the sky with smooth shading, and chromatic patches which are decomposed from the intensity image. The template of the hedgehog is then composed of selected patches from the hybrid dictionaries of four types of patches.

2), The criterion in selecting and ranking the atomic patches. To compose the template, we seek image patches that are informative in the following sense. i) It should be consistently shared by images from a certain category with little statistical fluctuation; and ii) it should be distinguishable from other images, i.e. negative examples. We consider two cases in learning the model: a) The negative examples are generic natural images and thus the learned templates are generic and generative; and b) The negative examples are from a competing object class and thus the learned templates are discriminative. For example, the templates in Figure 1 are generic. We may also learn a tomato template against a pear, then the selected features and their weights are adjusted. The selection and ranking of these patches is guided by an information projection principle.

3), The probability model on the templates. To compose the image patches from the heterogeneous subspaces (manifolds), we need a well-normalized probability model that integrates these patches under a common information theoretic principle.

Starting from an initial reference model, we pursue a sequence of probability model so as to minimize a Kullback-Leibler divergence. At each step, we choose a patch and its feature descriptor which leads to the maximum reduction of the KL-divergence. The pursuit process stops when the information gain is within the statistical fluctuation. This information projection allows us to learn the probabilistic image model with a relatively small number of examples. For categories with structural variations, we learn multiple hybrid templates through an EM-like procedure with unknown object subcategories as missing data.

4), The latent variables for deformation. To robustly model visual objects, we allow each patch to perturb locally to account for the deformation and occlusion as in the active basis model [2]. These local perturbations are denoted by latent (nuisance) variables. Illustrated in Figure 3 are hybrid image templates of pigeon, hedgehog and pig head matched to image instances. For each of the three figures, on the left is the learned hybrid template. Black bars denote sketch features and red dots denote texture features. The red dots illustrate the

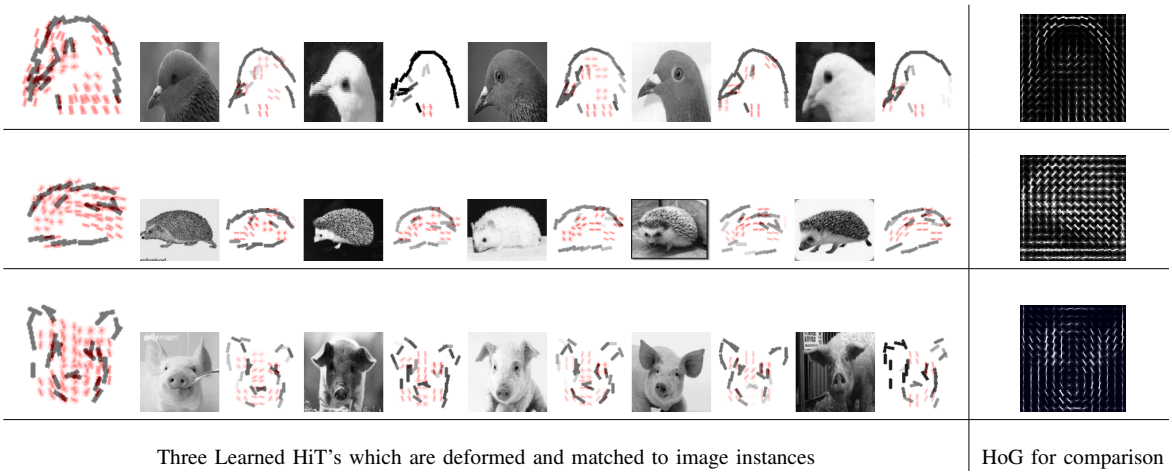


Fig. 3: Three automatically learned Hybrid Image Templates (HIT) together with their matching result on example images. The black bars illustrate sketch features and the red dots visualize texture (orientation field) features. For visual comparison, the weights of linear classifier on the HoG [1] feature map are shown to the right. HIT presents a much sparser image representation.

local orientation field, which can be strongly oriented or isotropic, depending on different object categories and locations. On the right are matched templates on image instances. In these instances, the strength of the bar reflects the feature response. The red dots in the right figure denote the texture features fired on these images. The deformation is captured by the local translation and rotation of sketches, and by the perturbation in the local orientation field (orientation histogram). The occlusion is captured by missed correspondences. On the right hand side of Figure 3, for each category we show the weights of linear classifier on the histogram of gradient (HoG [1]) map. The HoG map also captures important information of the object category but with a much denser representation. A sparser representation is needed for understanding the intrinsic structures that underlie the object categories.

1.2 Related work and comparison

1.2.1 Local feature descriptors

There has been a large amount of work on designing image features in the literature, and they can be roughly divided into two types. i) Geometric features, e.g. Haar-like features, Canny edge detector [3], Gabor-like primitives [4], and shape context descriptor [5]), explicitly record locations of edges/bars and are good for image patches with noticeable structures. We generally call them sketch features in this paper. ii) Texture features, in contrast, tend to be better described by histogram statistics, for example, GIST [6], SIFT [7] and HoG [1]. In object recognition, sketch features are shown to work well on objects with regular shapes, while texture features are more suitable for complex objects with cluttered appearance.

These two types of features are often studied separately for structures at different resolutions, but in real images, they are connected continuously through image scaling [8]. That is, viewed in low resolution, geometric structures become blurred and merge into texture appearance, and can become flat area (white noise) at extremely low resolution. Thus a good representation must be a hybrid one adaptive to different image scales.

1.2.2 Global image representation

Image templates, especially deformable ones, have been extensively studied for detection, recognition and tracking, for example, deformable templates [9], active appearance models [10], pictorial structures [11], constellation model [12], part-based latent SVM model [13], recursive compositional model [14], region-based taxonomy [15], hierarchical parts dictionary[16] and Active Basis model [2]. Our HIT model is closely related to the Active Basis model which only uses sketches to represent object shapes. In contrast, the HIT model integrates texture, flatness and color features and is more expressive. Our model is also related to the primal sketch representation [17] which combines sketchable (primitives) and non-sketchable (textures and flatness) image components. The difference is that the primal sketch is a low-middle level visual representation for generic images while the HIT model is a high level vision representation for objects that have similar configurations in a category.

1.2.3 Model learning and pursuit

In the literature, feature selection and model pursuit has been studied in three families of statistical models.

i) In Markov random (Gibbs) fields models, automated feature selection and model pursuit has been studied in text modeling [18], texture modeling [19], and other tasks [20]. These learning algorithms iteratively find the most informative feature statistics, and match them in the learned probabilistic model.

ii) In sparse coding and generative modeling, the active basis model [2] learns a sparse shape template composed of Gabor wavelet elements at different locations by information projection [21], [19]. Another line

of work learns hierarchical dirichlet processes [22] on interest points extracted from images.

iii) In discriminative modeling, boosting algorithms *e.g.* adaboost [23] and support vector machines [24] learn hyperplanes in the feature space that optimize the prediction of labels. In recent years, by combining different types of features in a discriminative framework [25], [26], [27], [28], [29], [30], [31], [32], much progress has been made towards improving the accuracy of object categorization.

The hybrid image templates include both texture features used in Gibbs model and image primitives used in active basis models. Thus the model is pursued in a product space composed of both low-dimensional subspaces for structures and high dimensional subspaces for textures. We also incorporate latent variables for each patch to perturb locally so that the deformable template can be registered to the object instances.

1.3 Relation to HoG

The proposed HIT can be interpreted in context of the popular HoG descriptor [1], but is one step beyond it. It has far shorter (at least $1/10$) feature dimension, and thus lead to more robust classification performance especially using small training sizes (e.g. 20$\sim$100 training positives). In HoG the image lattice is partitioned into equal sized cells (*e.g.* 8 by 8 pixels). Within each cell a gradient histogram (a 30$\sim$40 dimensional vector, depending on implementation) is computed by pooling over gradients, which is robust to local deformation. However, the detailed deformation is not recorded in the histogram. In HIT, the image lattice is divided into overlapping patches, and each patch can be described by one of the four feature types. One patch is similar to one cell in HoG. More precisely, HIT has the following advantages:

1) HIT is more sparse because it i) makes local decision by inhibition; ii) eliminates patches with high intra-class variations under information projection; iii) allows for local deformation of constituent elements and records them explicitly by local maximization and iv) records high order statistics by prototypes $\{\mathbf{h}\}$. Table 1 gives an example of feature dimensions comparing HIT with two other closely related systems for the VOC horse category. And HIT is customizable for different image categories; while HoG is a generic image descriptor densely populated over pixels.
2) HIT can be trained either using generative criterion towards a hierarchical model, or with discriminative criterion tuned towards classification.
3) HIT performs on par with the fine-tuned HoG feature on public benchmarks, though its feature dimension is only $1/10$ of HoG. When using fewer training examples, HIT outperforms HoG with a clear margin.

	HIT	HoG [1]	part-based latent SVM[13]
feature length	8×10^2	6.3×10^3	4.9×10^4

TABLE 1: Comparison of feature length.

HIT is also related to the part-based latent SVM model [13]. In [13] the template includes a coarse-level root template and several fine-level part templates, all of which are discriminatively trained using SVM. It is shown that the part-based latent SVM model performs better than the baseline HoG+SVM in many public benchmarks. HIT is not yet a part-based model, because its components are atomic. It is expected that composing HIT into part-based hierarchical template will lead to capability to model larger deformation as well as better classification performance. Hierarchical HIT is our ongoing work and is beyond the scope of this paper.

2 Representation

2.1 Hybrid image template

Let Λ be the image lattice for the object template which is typically of 150×150 pixels. This template will undergo a similarity transform to align with object instance in images. The lattice is decomposed into a set of K patches $\{\Lambda_k, k = 1, 2..., K\}$ selected from a large pool in the learning process through feature pursuit. As it was illustrated in Figure 1, these patches belong to four bands: sketch, texture/gradient field, flatness, and color respectively, and do not form a partition of the lattice Λ for two reasons:

- Certain pixels on Λ are left unexplained due to inconsistent image appearances at these positions.
- Two selected patches from different bands may overlap each other in position. For example, a sketch patch and a color patch can occupy the same region, but we make sure the sketch feature descriptor and color descriptor extracted from them would represent largely uncorrelated information.

The hybrid image template consists of the following components,

$$\text{HIT} = (\{\Lambda_k, \ell_k, \{B_k \text{ or } \mathbf{h}_k\}, \delta_k :, k = 1, 2, ..., K\}, \Theta) \quad (1)$$

1) $\Lambda_k \subset \Lambda$ is the k-th patch lattice described above.
2) $\ell_k \in \{'\text{skt}', '\text{txt}', '\text{flt}', '\text{clr}'\}$ is the type of the patch.
3) B_k or $\mathbf{h}_k$ is the feature prototype for the k-th patch. If $\ell_k = '\text{skt}'$, then the patch is described by a basis function B_k for the image primitive, otherwise it is described by a histogram $\mathbf{h}_k$ for texture gradients, flatness or color respectively.
4) $\delta_k = (\delta_k x, \delta_k y, \delta_k \theta)$: the latent variables for the local variabilities of the k-th patch, *i.e.* the local translations and rotations of selected patches.
5) $\Theta = \{\lambda_k, z_k : k = 1, 2, ..., K\}$ are the parameters of the probabilistic model p (to be discussed in the subsection). λ_k, z_k are the linear coefficient and normalizing constant for the k-th patch.

2.2 Prototypes, ϵ - balls, and saturation function

Let $\mathbf{I}_{\Lambda_k}$ be the image defined on the patch $\Lambda_k \subset \Lambda$. For $\ell_k = '\text{skt}'$, the prototype B_k defines a subspace through an explicit function for $\mathbf{I}_{\Lambda_k}$ (a sparse coding model),

$$\Omega(B_k) = \{\mathbf{I}_{\Lambda_k} : \mathbf{I}_{\Lambda_k} = c_k B_k + \epsilon\}. \quad (2)$$

For $\ell_k \in \{'\text{txt}', '\text{flt}', '\text{clr}'\}$, the prototype defines a subspace through an implicit function for $\mathbf{I}_{\Lambda_k}$ which constrains the histogram (a Markov random field model),

$$\Omega(\mathbf{h}_k) = \{\mathbf{I}_{\Lambda_k} : H(\mathbf{I}_{\Lambda_k}) = \mathbf{h}_k + \epsilon\}. \quad (3)$$

$H(\mathbf{I}_{\Lambda_k})$ extracts the histogram (texture gradient, flatness, or color) from $\mathbf{I}_{\Lambda_k}$.

In $\Omega(B_k)$, the distance is measured in the image space,

$$\rho^{\text{ex}}(\mathbf{I}_{\Lambda_k}) = \|\mathbf{I}_{\Lambda_k} - cB_k\|^2 \quad (4)$$

while in $\Omega(h_k)$, the distance is measured in the projected histogram space with L1 or L2 norm.

$$\rho^{\text{im}}(\mathbf{I}_{\Lambda_k}) = \|H(\mathbf{I}_{\Lambda_k}) - \mathbf{h}_k\|^2 \quad (5)$$

Intuitively, we may view the $\Omega(B_k)$ and $\Omega(\mathbf{h}_k)$ as ϵ-balls centered at the prototypes B_k and $\mathbf{h}_k$ respectively, with different metrics. Each ϵ-ball is a set of image patches which are perceptually equivalent. Thus the image space of HIT is the product space of these heterogeneous subspaces: $\Omega(HIT) = \prod_{k=1}^{K} \Omega_k$, on which a probability model is concentrated. Due to statistical fluctuations in small patches, these ϵ-balls have soft boundaries. Thus we use a sigmoid function to indicate whether a patch $\mathbf{I}_{\Lambda_k}$ belongs to a ball $\Omega(B_k)$ or $\Omega(\mathbf{h}_k)$.

$$r(\mathbf{I}_{\Lambda_k}) = \text{S}(\rho(\mathbf{I}_{\Lambda_k})), \quad (6)$$

where ρ can be either ρ^{ex} or ρ^{im}. $S(x)$ is a saturation function with maximum at $x = 0$:

$$\text{S}(x) = \tau\left(\frac{2}{1 + e^{-2(\eta - x)/\tau}} - 1\right), \quad (7)$$

with shape parameters τ and η. Following [2] we set $\tau = 6$ and η is locally adaptive: $\eta = \|\mathbf{I}_{\Lambda_k}\|^2$ where $\mathbf{I}_{\Lambda_k}$ denotes the local image patch. We call $r(\mathbf{I}_{\Lambda_k})$ the *response* of the feature (prototype B_k or $\mathbf{h}_k$) on patch $\mathbf{I}_{\Lambda_k}$.

2.3 Projecting image patches to 1D responses

Though the image patches are from heterogeneous subspaces of varying dimensions with different metrics, we project them into the one-dimensional feature response $r(\mathbf{I}_{\Lambda_k})$, on which we can calculate the statistics (expectation) of $r(\mathbf{I}_{\Lambda_k})$ over the training set regardless of the types of patches. This way it is easy to integrate them in a probabilistic model.

In the following we discuss the details of computing the responses for the four different image subspaces.

Given an input color image $\mathbf{I}$ on lattice Λ, we first transform it into a HSV-space with HS being the chromatic information and V the gray level image. We apply a common set of filters Δ to the gray level image. The dictionary Δ includes Gabor filters (sine and cosine) at 3 scales and 16 orientations. The Gabor filter of the canonical scale and orientation is of the form: $F(x, y) \propto \exp\{-(x/\sigma_1)^2 - (y/\sigma_2)^2\}e^{ix}$ with $\sigma_1 = 5$, $\sigma_2 = 10$.

1). Calculating responses on primitives. When a patch $\mathbf{I}_{\Lambda_k}$ contains a prominent primitive, such as an edge or bar, it is dominated by a filter which inhibits all the other filters. Thus the whole patch is represented by a single filter, which is called a Basis function $B_k \in \Delta$. The response is calculated as the local maximum over the activity δ_k,

$$r^{\text{skt}}(\mathbf{I}_{\Lambda_k}) = \max_{\delta x, \delta y, \delta\theta} \text{S}(\|\mathbf{I} - cB_{x+\delta x, y+\delta y, o+\delta o}\|^2). \quad (8)$$

The local maximum pooling is proposed by [33] as a possible function of complex cells in V1.

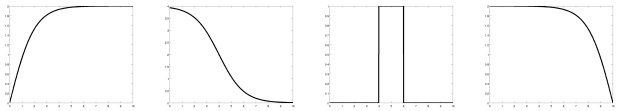
2). Calculating responses on texture. In contrast to the primitives, a texture patch usually contains many small elements, such as the patch on the hedgehog body in Figure 1. As a result, many filters have medium responses on the image patch. Thus we pool a histogram of these filters collectively over the local patch to form a histogram descriptor $H(\mathbf{I})$.

The texture response is calculated by

$$r^{\text{txt}}(\mathbf{I}_{\Lambda_k}) = \text{S}(\|H(\mathbf{I}_{\Lambda_k}) - \mathbf{h}\|^2), \quad (9)$$

where $\mathbf{h}$ is a pre-computed histogram prototype (one may consider it as a cluster center of similar texture patches). More specifically, $\mathbf{h}$ is obtained by averaging the histograms at the same position of roughly aligned positive example images. For texture, we are only interested in the medium to strong strength along certain directions. So we replace the indicator function, which is often used in histogram binning , by a continuous function $a(x) = \frac{12}{1+e^{-x/3}} - 6$. The histogram is then weighted into one bin for each filter,

$$H_o(\mathbf{I}_{\Lambda_k}) = \frac{1}{|\Lambda_k|} \sum_{(x,y)\in\Lambda_k} a(|F_o * \mathbf{I}_{\Lambda_k}|^2). \quad (10)$$



(a) $a(x)$ (b) $b(x)$ (c) $\mathbf{1}(x)$ (d) $S(x)$

Fig. 4: Plotting the four functions: $a(x)$, $b(x)$, $\mathbf{1}(x)$, $S(x)$.

Thus we obtain the oriented histogram for all filters as a $|\mathcal{O}|$-vector,

$$H(\mathbf{I}_{\Lambda_k}) = (H_1, ..., H_{|\mathcal{O}|}). \quad (11)$$

It measures the strengths in all orientations.

3). Calculating responses on flat patch. By flat patch we mean image area that are void of structures, especially edges. Thus filters have near-zero responses. They are helpful for suppressing false alarms in cluttered areas. As a texture-less measure, we choose a few small filters

$\Delta^{\text{flt}} = \{\nabla_x, \nabla_y, LoG\}$ and further compress the texture histogram into a single scalar,

$$H(\mathbf{I}_{\Lambda_k}) = \sum_{F \in \Delta^{\text{flt}}} \sum_{(x,y) \in \Lambda_k} b(|F_o * \mathbf{I}_{\Lambda_k}|^2). \quad (12)$$

$b()$ is a function that measures the featureless responses. It takes the form of a sigmoid function like $S()$ but with different shape parameters. In Figure 4 we plot the four functions $a()$, $b()$, $\mathbf{1}()$ and $S()$ for comparison.

Then the flatness response is defined as,

$$r^{\text{flt}}(\mathbf{I}_{\Lambda_k}) = \text{S}(H(\mathbf{I}_{\Lambda_k}) - h). \quad (13)$$

In the above $h = 0$ is a scalar for flatness prototype.

4). Calculating responses on color. The chromatic descriptors are informative for certain object categories. Similar to orientation histogram, we calculate a histogram $H^{\text{clr}}(\mathbf{I}_{\Lambda_k})$ on the color space (we use the 2D HS-space in the HSV format). Then the color patch response is defined as the saturated distance between the color histogram of the observed image and the prototype histogram $\mathbf{h}$,

$$r^{\text{clr}}(\mathbf{I}_{\Lambda_k}) = \text{S}(\|H^{\text{clr}}(\mathbf{I}_{\Lambda_k}) - \mathbf{h}\|^2). \quad (14)$$

In summary, a HIT template consists of K prototypes $\{B_k \text{ or } \mathbf{h}_k, k = 1, \cdots, K\}$ for sketch, texture/gradient, flatness, and color patches respectively which define K-subspaces (or ϵ-balls) $\Omega(B_K)$ or $\Omega(\mathbf{h}_k)$ of varying dimensions. These ϵ-balls quantize the image space with different metrics. An input image $\mathbf{I}$ on lattice Λ is then projected to the HIT and is represented by a vector of responses:

$$\mathbf{I} \to (r_1, r_2, ..., r_K)$$

where r_k is a soft measure for whether the image patch $\mathbf{I}_{\Lambda_k}$ belongs to the subspace defined by the corresponding prototype. In the next section we will define a probability model on image $\mathbf{I}$ based on these responses.

3 Learning the Hybrid Image Templates

We present an algorithm for learning the hybrid image templates automatically from a set of image examples. It pursues the image patches, calculates their prototypes, and derive a probability model sequentially until the information gain is within the statistical fluctuation – a model complexity criterion similar to AIC [34].

3.1 Template pursuit by information projection

Let $f(\mathbf{I})$ be the underlying probability distribution for an image category, and our objective is to learn a series of models that approach f from an initial or reference model q,

$$q = p_0 \to p_1 \to p_2 \to \cdots \to p_K \approx f. \quad (15)$$

These models sequentially match the observed marginal statistics collected from the samples of f. With more marginal statistics matched between the model p and f, p will approach f in terms of reducing the Kullback-Leibler divergence $KL(f||p)$ monotonically.

The main input to the learning algorithm is a set of positive examples

$$D^+ = \{\mathbf{I}_1, ..., \mathbf{I}_n\} \sim f,$$

where f is the underlying target image distribution and $\sim$ means sampled from. For simplicity, we may assume these images contain roughly aligned objects that can be explained by a common HIT template. When this alignment assumption is not satisfied, we can adopt an EM-like iterative procedure with the unknown object localization as missing data. See [2] and Sec. 5.5 for examples of learning from non-aligned objects. We are also given a set of negative examples

$$D^- = \{\mathbf{J}_1, ..., \mathbf{J}_N\} \sim \text{ reference distribution } q.$$

The negative examples are only used for pooling marginal histograms of one-dimensional feature responses in a pre-computation step.

The image lattice Λ is divided into overlapping patches for multiple scales by a scanning window with a step size about 10% of the window size. Then we calculate their corresponding prototypes and responses for all images in D^+. The sketch prototypes B_i are specified by the Gabor dictionary Δ, and the histogram prototypes $\mathbf{h}_k$ are obtained by computing the histograms for positive examples in the same region of template lattice and then taking the average. As a result, we obtain an excessive number of candidate patches.

$$\Omega_{\text{cand}} = \{\Lambda_j, \ell_j, \{B_j \text{ or } \mathbf{h}_j\} : j = 1, 2, ..., M\}. \quad (16)$$

From Ω_{cand}, we will select the most informative patches and their corresponding prototypes for HIT.

By induction, at the k-th step, we have a HIT with $k-1$ patches and a model $p = p_{k-1}$:

$$\text{HIT}_{k-1} = (\{\Lambda_j, \ell_j, B_j \text{ or } \mathbf{h}_j, \delta_j, j = 1, ..., k-1\}, \Theta_{k-1}).$$

Consider a new candidate patch Λ_k in Ω_{cand} and its responses on n positive examples and N negative examples:

$$\{r^+_{k,i}, i = 1, ..., n\} \quad \{r^-_{k,i}, i = 1, ..., N\}. \quad (17)$$

And let $\bar{r}^+_k$ and $\bar{r}^-_k$ be the sample means on the two sets.

The gain of adding this patch to the template is measured by the KL divergence between the target marginal distribution $f(r_k)$ and the current model $p_{k-1}(r_k)$, as this represents the new information in the training data that is not yet captured in the model. Among all the candidate patches, the one with the largest gain is selected.

To estimate this gain, we use Monte-Carlo methods with samples from $f(r_k)$ and $p_{k-1}(r_k)$. Obviously $\{r^+_{k,i}\}$ is a fair sample from $f(r_k)$. While to sample from $p_{k-1}(r_k)$, one may use importance sampling on $\{r^-_{k,i}\}$, *i.e.* re-weighting the examples by $\frac{p_{k-1}(r_k)}{q(r_k)}$. Here we simplify the problem by a conditional independence assumption as stated in previous section. A feature response $r_1(\mathbf{I}_{\Lambda_1})$

is roughly uncorrelated with $r_2(\mathbf{I}_{\Lambda_2})$ if one of the following holds: i) the two patches Λ_1 and Λ_2 have little overlap; ii) Λ_1 and Λ_2 are from different scales. If at the k-th step we have removed from Ω_{cand} all the candidate patches that overlap with selected patches, then r_k is roughly uncorrelated with all the previously selected responses $r_1, ..., r_{k-1}$. As a result, $p_{k-1}(r_k) = q(r_k)$ and $\{r^-_{k,i}\}$ can be used as a sample of $p_{k-1}(r_k)$. The exact formula for estimating the gain (*i.e.* KL divergence between $f(r_k)$ and $p_{k-1}(r_k)$) is given in Sec. 3.2 once we have derived the parametric form of p in the following.

For a selected patch Λ_k, the new model $p = p_k$ is required to match certain observed statistics (*e.g.* first moment) while it should be also close to the learned model p_{k-1} to preserve the previous constraints. This is commonly expressed as a constrained optimization problem [18], [19])

$$p^*_k = \arg\min \quad KL(p_k|p_{k-1}) \tag{18}$$

$$s.t. \qquad E_{p_k}[r_k] = E_f[r_k] \tag{19}$$

By solving the Euler-Lagrange equation with Lagrange multipliers $\{\lambda_j\}$ and γ,

$$\frac{\partial}{\partial p_k}\Big\{ \sum_{\mathbf{I}} p_k(\mathbf{I}) \log \frac{p_k(\mathbf{I})}{p_{k-1}(\mathbf{I})} + \lambda_k(E_{p_k}[r_j] - E_f[r_j]) + \gamma(\sum_{\mathbf{I}} p_k(\mathbf{I}) - 1)\Big\} = 0.$$

Thus we have,

$$p_k(\mathbf{I}) = p_{k-1}(\mathbf{I}) \frac{1}{z_k} \exp\{-\lambda_k r_k(\mathbf{I})\}. \tag{20}$$

$z_k = E_q\left[\exp\{\lambda_k r_k(\mathbf{I}_{\Lambda_k})\}\right]$ is a normalizing constant. This can be estimated by the negative samples,

$$z_k \approx \frac{1}{N} \sum_{i=1}^{N} e^{\lambda_k r(\mathbf{J}_{i,\Lambda_k})}. \tag{21}$$

λ_k is the parameter (Lagrange multiplier) to satisfy constraint in eqn (19),

$$E_p[r_k] \approx \frac{1}{N} \sum_{i=1}^{N} \left[r(\mathbf{J}_{i,\Lambda_k}) e^{\lambda_k r(\mathbf{J}_{i,\Lambda_k})} \right] \frac{1}{z_k} = \bar{r}^+_k. \tag{22}$$

In computation, we can look up $\bar{r}^+_k$ in the table to find the best λ_k. The importance sampling is a good estimation in calculating λ_k and z_k because in our model r is one dimensional.

By recursion, we have a *factorized* log-linear form,

$$p_K(\mathbf{I}) = q(\mathbf{I}) \prod_{j=1}^{K} \left[\frac{1}{z_j} \exp\{\lambda_j r_j(\mathbf{I}_{\Lambda_j})\} \right] \tag{23}$$

The above pursuit algorithm is related to projection pursuit [21]. But instead of using product of marginal histograms, our model is a product of parametric likelihood ratio functions, which has much fewer parameters and more robust than the classic projection pursuit, especially when the training sample size is small (*e.g.* 10 50). Besides, each sketch feature is associated with a latent variable describing its deformation or perturbation that varies different training examples.

3.2 Interpretation of the learning procedure

Each learning step in the previous subsection observes the following Pythagorean theorem which is known in information projection [18], [19].

Proposition 1: The model p_{k-1}, p_k and the underlying probability f satisfy the following equation,

$$KL(f\,||\,p_{k-1}) - KL(f\,||\,p_k) = KL(p_k\,||\,p_{k-1}) > 0. \tag{24}$$

This ensures the convergence of the learning process, given that we can find informative feature responses r_k that can tell the difference between $E_f[r_k] = E_{p_k}[r_k]$ and $E_q[r_k] = E_{p_{k-1}}[r_k]$.

Figure 5 shows the geometric interpretation.

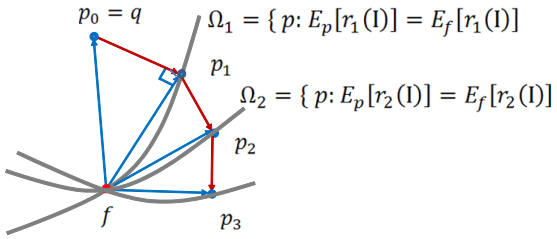


Fig. 5: Learning by information projection. The series of models $p_0, p_1, ..., p_K$ converge to the target probability f monotonically by sequentially matching the constraints.

In Figure 5, we consider the space of all possible probabilities where each point is a model. Our underlying probability f and the initial probability q are two points in the space with a large divergence $KL(f\,||\,q)$. The learning process iterates the following two steps.

1), Min-step. Suppose we have chosen a patch Λ_k and its prototype, and calculated its statistics $E_f[r_k] \approx \bar{r}^+_k$. We denote the set of all models p that has the same statistics by a set,

$$\Omega_k = \{p: \; E_p[r_k(\mathbf{I}_{\Lambda_k})] = E_f[r_k(\mathbf{I}_{\Lambda_k})]\}.$$

This is illustrated by a curve passing point f. Thus we find a model p^*_k on this curve through a perpendicular projection from p_{k-1} to Ω_k. In other words, p^*_k is the model that is closest to p_{k-1} on Ω_k to preserve the previously learned statistics,

$$p^*_k \text{ or } \lambda^*_k = \arg\min_{\Omega_k} KL(p_k\,||\,p_{k-1}). \tag{25}$$

This step solves for λ_k and z_k in equations (21) and (22).

2). Max-step. Among all the candidate patches and their prototypes in Ω_{cand}, we need to choose a patch/prototype which has the largest difference between $E_{p_{k-1}}[r_k]$ and $E_{p_k}[r_k]$.

$$p^*_k \text{ or } (\lambda_k, B_k/\mathbf{h}_k)^* = \arg\max_{\Omega_{\text{cand}}} KL(p_k\,||\,p_{k-1}).$$

Intuitively, this is to choose a curve in Figure 5 which is the farthest away from the current model p_{k-1}. By

equation (24), this is to choose the patch that maximizing the reduction of KL-divergence,

$$(\lambda_k, B_k/\mathbf{h}_k)^* = \arg\max_{\Omega_{\text{cand}}} [KL(f \,||\, p_{k-1}) - KL(f \,||\, p_k)].$$

We define the information gain at the k-th step by,

$$\begin{aligned} \textbf{gain:} \quad Ig_k &= KL(p_k \,||\, p_{k-1}) \\ &= \lambda_k E_f[r_k] - \log z_k \approx \lambda_k \bar{r}_k^+ - \log z_k \end{aligned} \tag{26}$$

After k steps, the total information gain is:

$$KL(p_k \,||\, q) \approx \sum_{j=1}^{k} (\lambda_j \bar{r}_j^+ - \log z_j). \tag{27}$$

3.3 Correcting the information gain.

Due to limited training examples, the estimated information gain is subject to fluctuation error. We propose to correct it considering the bias and variance of the estimated expectation on positive examples. Recall that λ_k is the parameter learned according to $E_f[r_k]$, and empirically we estimate $\hat{\lambda}_k$ from $\bar{r}_k^+$. While $p_k(\mathbf{I}; \lambda_k)$ is the desired model, we can only get $\hat{p}_k(\mathbf{I}; \hat{\lambda}_k)$ in practice. Consequently, the estimated information gain is,

$$\hat{Ig}_k \triangleq KL(\hat{p}_k \| p_{k-1}) = \hat{\lambda}_k E_{p_k}[r_k(\mathbf{I})] - \log z_k.$$

The true information gain Ig_k is discounted with an AIC type ([34]) of penalty

$$Ig_k \approx \hat{Ig}_k - \frac{1}{n} \frac{\text{Var}_{\hat{f}}(r_k)}{\text{Var}_{\hat{p}_k}(r_k)}, \tag{28}$$

where $\text{Var}_{\hat{f}}(r_k)$ is estimated on n positive examples. That is, the information gain is discounted by the relative variance of the marginal feature statistic. When $\hat{p}_k(r_k)$ is a good fit for $\hat{f}(r_k)$, we may assume the discount factor only depends on the training sample size n.

3.4 Discriminative adjustments of HIT

The template matching score of an HIT on a testing image is defined as the log likelihood ratio computed from Eq. (23):

$$\text{Score}(\mathbf{I}; \text{HIT}) = \log \frac{p_K(\mathbf{I}; \text{HIT})}{q(\mathbf{I})} = \sum_{j=1}^{K} \lambda_j r_j - \log z_j \tag{29}$$

This template matching score is linear in the feature responses $\{r_j\}$ and can be interpreted discriminatively if we treat the two image distributions $f(\mathbf{I})$ and $q(\mathbf{I})$ as the generating models for positive and negative examples:

$$\log \frac{p(+|\mathbf{I})}{p(-|\mathbf{I})} = \sum_{j=1}^{K} w_j r_j + w_0 \tag{30}$$

where $+, -$ denote binary labels of images, $w_j = \lambda_j$ and $w_0 = -\sum_{j=1}^{K} \log z_j$.

The two likelihood ratios in Eq. (29) and (30) are closely connected by the following Bayesian formula,

$$\frac{p(+|\mathbf{I})}{p(-|\mathbf{I})} = \frac{p(\mathbf{I}|+)}{p(\mathbf{I}|-)} \cdot \frac{p(+)}{p(-)} = \frac{p_K(\mathbf{I}; \text{HIT})}{q(\mathbf{I})} \cdot \frac{p(+)}{p(-)}$$

and their forms are different only by a constant $\frac{p(+)}{p(-)}$, the ratio between amount of positive and negative examples.

To obtain good classification performances on large data sets, it is often desirable to adjust the parameters in Eq. (29) or (30) using a discriminative criterion. In this paper we use logistic regression to adjust the parameters.

Algorithm 1: Learning a hybrid image template.

```
Let template T = empty ;
Divide the template lattice into overlapping candidate
  patches at multiple scales;
Prepare candidate sketch, texture, flatness and color
  features for the candidate patches: the feature responses
  are computed according to Section 2.3;
foreach candidate feature response r_k do
    Compute its information gain by Eq. (26);
    Adjust the information gain by Eq. (28);
end
repeat
    Select r_{k*} that maximizes gain;
    Estimate the model parameter λ_k (feature weight) that
    best satisfies Eq. (22);
    Perform local inhibition such that neighboring
    features of the same type will not be selected;
until gain is smaller than a threshold τ ;
Output:
The template T with selected features, each with a weight
  λ and a normalizing constant z;
```

4 Algorithms for Learning & Detection

4.1 Learning

The stepwise learning algorithm for hybrid image templates is described in Algorithm 1, with the stopping criterion τ being a global parameter. To accelerate the feature selection, we may separate candidate features of different types and scales into several groups that are not correlated with one another. Within each small group of candidate features, the cost of feature selection is greatly reduced. For fast computation, we utilize a rank preserving (or monotonic) function of the information gain. Let $\bar{r}^+$ be the sample average of feature response on positive examples. Let $q(r_k)$ be the frequency histogram pooled from the feature responses on negative examples. The maximum likelihood estimator $\lambda^* \triangleq \arg\max_\lambda Ig_k$ is determined by $\bar{r}^+$ and $q(r)$. It can be shown that, if we can assume the reference distribution $q(r_k)$ stays the same for different k, then Ig_k computed using λ^* is a monotonic increasing function of $\bar{r}^+$. This assumption holds for sketch and flatness features, and is a good approximation for texture and color histogram features.

Algorithm 2: Detecting a hybrid image template.

foreach *x,y,t* **do**
  Compute the feature response map (SUM1 maps) $r^{(t)}(x,y)$ where t indexes feature type ;
end
foreach *x,y,t* **do**
  $r^{\mathrm{LMAX}^{(t)}}(x,y) \leftarrow LocalMaxPooling(r^{(t)}(x,y))$;
  where the maximization is over local translation and rotation; we call them MAX1 maps. We also record the local perturbations (ARGMAX1 maps) corresponding to local maxima.
end
foreach *x,y,o,s* **do**
  Compute $Score(x,y,o,s)$ (SUM2 maps) by scanning the template over the MAX1 maps:

$$\mathrm{Score}(x,y,o,s) = \sum_{j=1}^{K} \left(\lambda_j r_j^{(x,y,o,s)} - \log z_j \right)$$

  where $r_j^{(x,y,o,s)}$ denotes the response of the j-th selected feature after transforming the template by translation (x,y), rotation o and scaling s.
end
$(x*, y*, o*, s*) \leftarrow \arg\max Score(x,y,o,s)$;
The object is detected by HIT at (x*,y*, o*, s*).
Given the localization of the object, find the locations and rotations of its components according to ARGMAX1 maps, and detailed deformation of HIT (a parse tree) is found.

4.2 Detection

Detecting a HIT in a testing image is straight-forward. If we discard the hidden variables controlling the template deformation, then the detection is the same as one would run a face detector (as a linear classifier) with a sliding window on an image. Now to detect a HIT, we not only find the global translation and rotation of the matched template, but also infer all the perturbations of the small patches inside the template. Inspired by the cortex-like systems [33], [35], [36], we have adopted a recursive SUM-MAX procedure similar to [2], which is decribed in Algorithm 2.

5 Experiment and Evaluation

We present six experiments studying the properties of HIT learning and evaluate their performance for classification on commonly used benchmarks.

5.1 E1: Learning HIT for image categories

In the first experiment we are interested in whether the learning algorithm can identify and select meaningful descriptors for different image categories. We apply the learning algorithm to 14 object or scene image categories. Figure 6 shows the learned hybrid image templates. The number of training images for each category varies. Most categories have around 30 training examples, and some categories have as few as 6. Sketch and texture patches are selected for most categories. Flatness patches are selected for tomato and the sky areas. Many color patches are selected for tomato, pizza, highway and forest, and some are chosen for panda, palm tree and sunset. These learned HIT's capture human intuition better in comparison to other popular representations, such as HoG [1] as is shown in Figure 3, and have richer features than the active basis model [2] and the classical AAM models [10] and deformable templates [9].

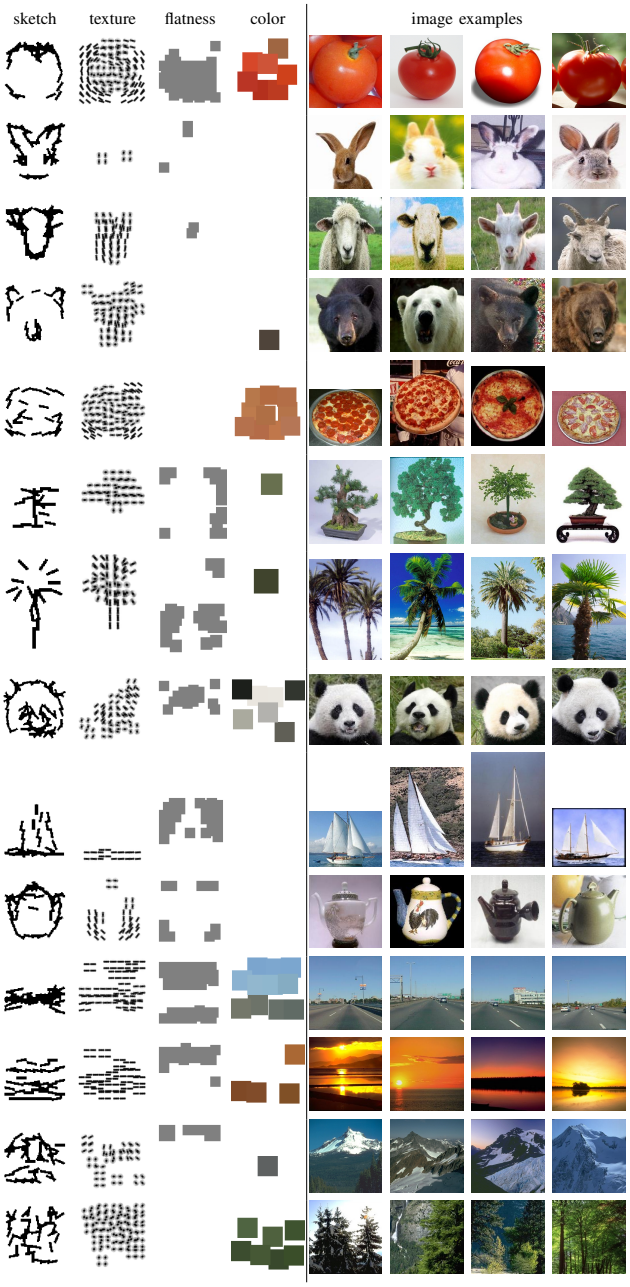


Fig. 6: Learned hybrid image templates with sketch, texture, flatness and color and features. For each row in the figure, in the first column is the image template with selected sketch, texture, flatness and color features. Then four training examples are shown in the final column. Best viewed in color.

5.2 E2: Sketch-texture contributions to classification

In the second experiment, we study how the sketch and texture patches are ordered by their information gains

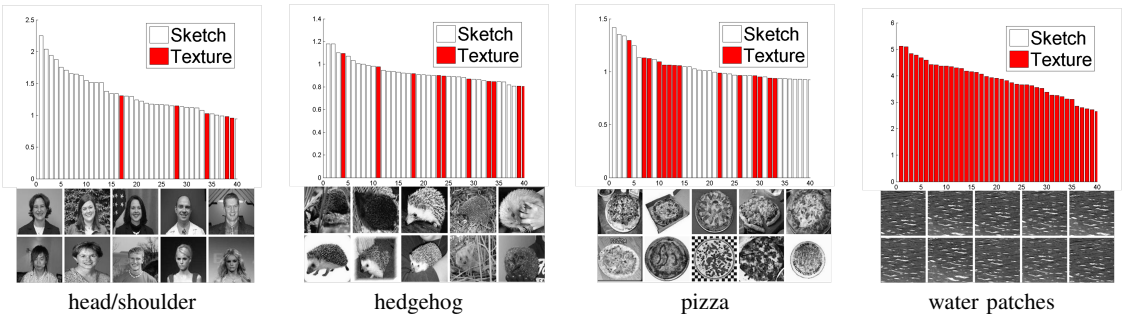
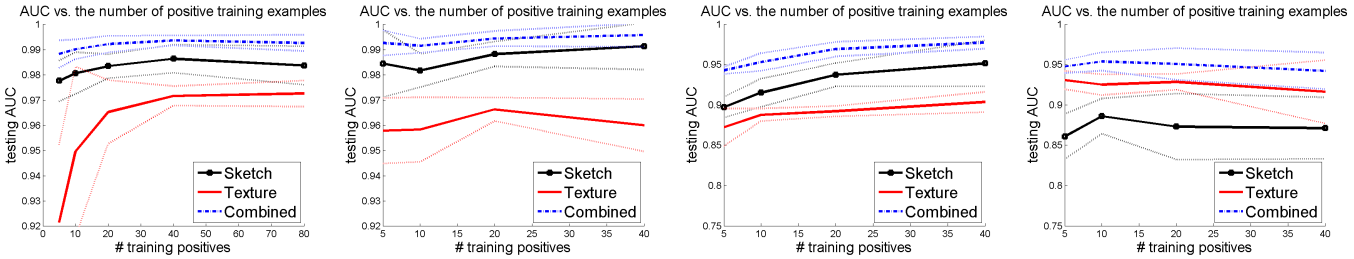


Fig. 7: Competition of sketch and texture patches. Top 40 selected patches are ordered by their information gains in decreasing order in each category. Hollow bars are for sketch patches, and solid red bars are for texture patches.



(a) human head/shoulder (b) cat head (c) pig head (d) hedgehog

Fig. 8: Improvement on classification due to the combination of sketch and texture features. In each plot, the area under ROC curve (AUC) is averaged over cross validation runs and plotted against the number of positive training examples. The dotted lines indicate 95% confidence bounds.

in different categories, and how much they contribute to classification.

We choose four categories ranging from structured to textured: head-shoulder, hedgehog, pizza, and wavy water. Each category has 15 image examples, some of which are shown in Figure 7. We plot the information gains of the selected patches in decreasing order: the hollow bars are for sketch patches and the solid (red) bars are for texture patches. For image categories with regular shape, *e.g.* head/shoulder, sketches dominate the information gain. For hedgehog, pizza and wavy water, as there are cluttered structures inside objects, texture patches make bigger contributions.

We test the contributions of sketch and texture for classification on other four categories: human head/shoulder, cat head, pig head and hedgehog. For binary classification, each category has 100 positive examples which are classified against a common set of 600 random negative images. For each category, we compare the AUC (area under ROC curve) of the HIT against performances using only sketch or texture patches respectively. Figure 8 plots the three curves with their confidence intervals with 5 cross validation runs.

5.3 E3: Hybrid image templates over scales

As studied in [8], there is a continuous transition from cartoon sketches (low complexity or low entropy), to object (mixing sketches and textures), to stochastic texture (no sketch), and finally to flatness (pure Gaussian noise with small variance) when we scale down the images. Therefore the HIT must also change over scales. In Figure 9, we show the learned HIT's of cat at four distinct scales. Consider an image patch of cat's whisker. At a very fine scale, individual whiskers are recognizable and many sketches are used to describe the image patch. At a coarse scale, the whisker becomes texture. In other words, each patch exists only for a range of scales.

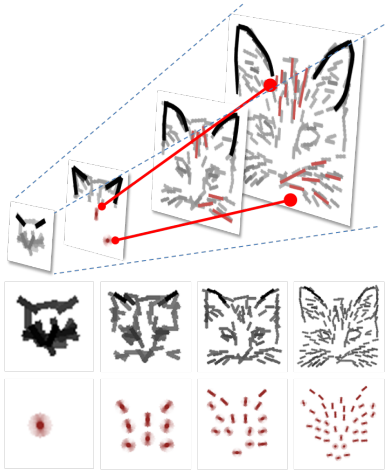


Fig. 9: Top: studying the transition of sketch, texture and flatness patterns by finding correspondences of features across scales. **Bottom**: learned sketch-only and texture-only templates from 20 cat images at multiple scales.

HoG+SVM[1]	Baseline HIT	Mixture of HIT	Part-based LSVM[13]
70.8%	71.6%	75.6%	77.6%

TABLE 2: Recognition accuracies on the animal faces dataset.

5.4 E4: Learning pairwise contrast templates

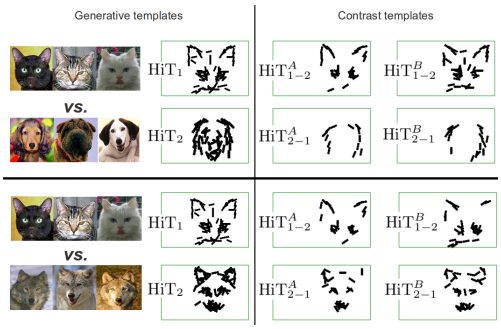


Fig. 10: Contrast templates for cat vs. wolf and cat vs. dog.

What is in common and what is different between cats and wolves? It is interesting to study the *pairwise contrast template* which can be used to discriminate between two categories. Here we provide some intuition into why HIT can be used for classification.

Suppose we are given examples for two categories C_1 and C_2. Let HIT_1 and HIT_2 be the learned HIT templates against a common generic reference model q. Consider two methods for learning contrast templates HIT_{1-2} and HIT_{2-1}. In method A, we replace the generic negative examples by the category that we are discriminating from. During learning, the information gain for any common features between the two categories is reduced, and the resulting contrast templates emphasize the differences.

$$\text{method A:} \quad \text{HIT}^A_{1-2} = \log \frac{p(\mathbf{I}|\text{HIT}_1)}{p(\mathbf{I}|\text{HIT}_2)}$$
$$= \log \frac{p(\mathbf{I}|\text{HIT}_1)}{q(\mathbf{I})} - \log \frac{p(\mathbf{I}|\text{HIT}_2)}{q(\mathbf{I})} = \text{HIT}_1 - \text{HIT}_2.$$

In method B, we take the union of selected features from HIT_1 and HIT_2 and re-weight them by a discriminative learning process such as SVM.

$$\text{method B:} \quad \text{HIT}^B_{1-2} = \arg\max \text{Margin}(\text{HIT})$$

Figure 10 shows two contrast templates: cat vs. wolf, and cat vs. dog. We can see that, by contrasting generative models, we already obtain reasonable one-vs-one classifiers similar to the ones discriminatively trained. This explains why the HIT trained generatively can be adapted for discriminative tasks.

5.5 E5: Weakly supervised clustering of HIT's

In this experiment we are interested in the learning and classification of hybrid image templates in the context of weakly supervised learning and clustering. We also introduce a new dataset: **LHI-Animal-Faces**. Figure 11 provides an overview of the dataset. It contains around 2200 images for 20 categories animal or human faces.Compared to other benchmarks, LHI-Animal-Faces has several good properties: (1) the animal face categories are similar to each other due to evolutional relationship and shared parts, and it is a challenging task to discern them; (2) the animal face categories exhibit interesting within-class variation, which includes (i) rotation and flip transforms, e.g. rotated panda faces and left-or-right oriented pigeon heads; (ii) posture variation, e.g. rabbits with standing ears and relaxed ears; and (iii) sub-types, e.g. male and female lions.

We compare four systems on this dataset: (a) HoG feature trained with SVM [1], (b) HIT, (c) multiple transformation invariant HITs (Mixture of HIT) and (d) part-based HoG feature trained with latent SVM [13]. For system (c), we learn five HITs for each categories (in total 20×5 =100 templates). During learning, we use an iterative EM procedure to infer the unknown rotation, reflection and translation of each template. For system (d), we learn two reflection invariant templates per category. We find using more templates does not help in system (d). Table 2 shows the multi-class recognition accuracies for four systems. By adding rotation/reflection invariance and clustering during learning process, we are able to improve the accuracy from 0.72 to 0.76, outperforming the similar system [1] by a clear margin. The performance is close to the part-based latent SVM model [13] which has much more parameters with a compositional hierarchy. From the confusion matrices, we find the top two confusions are caused by sheep head vs. cow head, and pigeon head vs. eagle head.

Learned templates. Figure 13 shows several distinct clusters of animal face images automatically obtained by the algorithm. Each cluster is modeled by one HIT template, which is invariant to translation, rotation and reflection. For example, the ducks in Figure 13 facing left and right are identified as the same object category and described by one HIT. For illustration purpose only sketch features of the template are shown. Particularly note that the two types of rabbit head images with standing ears vs. with relaxed ears are automatically discovered by the learning algorithm.

5.6 E6: Experiments on commonly used benchmarks

In this section, we evaluate the HIT in terms of its classification performance on commonly used benchmarks and put it in context with other state-or-art methods. The benchmarks include INRIA person [1], VOC2007 [37], Caltech-101 [38] and a new dataset LHI-Animal-Faces to be introduced in Sec. 5.5. We compare the classification performance of HIT with HoG feature [1] trained with SVM, which is equivalent to the root template of part-based latent SVM model [13]. For HoG we use the implementation by [13]. Although HIT is originally designed for generating the image data rather than classification, its performance is on par with state-or-art.

Parameters of HIT. Parameters important for classification performance include: (1) The saturation upper-

Fig. 11: The LHI-Animal-Faces dataset. Three images are shown for each of the 20 categories.

Fig. 12: Learned HIT's from the animal face dataset. One HIT is learned per category. Only sketches are shown for clarity.

Fig. 13: HIT templates clustering results on animal faces.

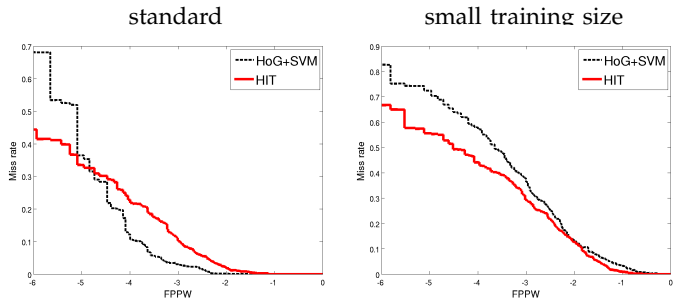


Fig. 14: Comparison on INRIA person dataset using FPPW (horizontal axis) and miss rate (vertical axis).

Fig. 15: False positives (top row) and miss detections (bottom row) of HIT on INRIA person testing examples. The matched template is overlayed on each image.

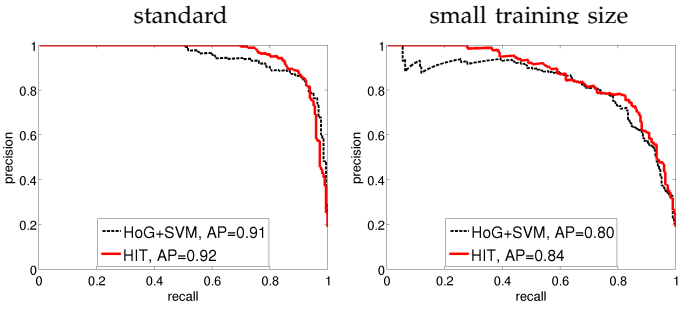


Fig. 16: Comparison on VOC2007 horse using precision-recall.

bound τ in Eq.(7); (2) the neighborhood size for local maximum pooling; (3) the neighborhood size for pooling texture (orientation histogram) and flatness features. We find the best performance is achieved when the upper-bound τ is 5, the local maximum pooling is performed in a 11 by 11 pixels neighborhood, and texture/flatness features are pooled within a 9 by 9 pixels neighborhood. These parameter settings are chosen by cross validation. We also find that performing a simple local normalization on sketch response maps $r^{sk}(x, y, o)$ improves classification performance. It is done by dividing each response $r^{sk}(x, y, o)$ by the local mean response averaged over all orientations pooled in a neighborhood with the same size as the Gabor filter. We only perform local normalization for sketch features.

For simplicity, we use a fixed scale of Gabor filters with size 17 by 17 pixels when computing sketch, texture and flatness features. In total 16 orientations of Gabor sine/cosine filters are used. The classification performance of HIT can be further improved by utilizing multiple scales of Gabor filters.

INRIA person. In Figure 14 we compare HIT with HoG [1] on INRIA person dataset. The left sub-figure of Figure 14 corresponds to training using all 2416 training positives, and the right sub-figure corresponds to training with only the first 100 positive examples. The negative patches are sampled according to the description at the project page of [1]. Both positive and negative image patches are of size 134 by 70 pixels, and they are cropped from original images before feature maps are computed from them. In this way, we make sure that boundary effect is not used unfairly in favor of any training algorithm. Following [1], we use the metric of miss rate plotted against the logarithm of FPPW (false

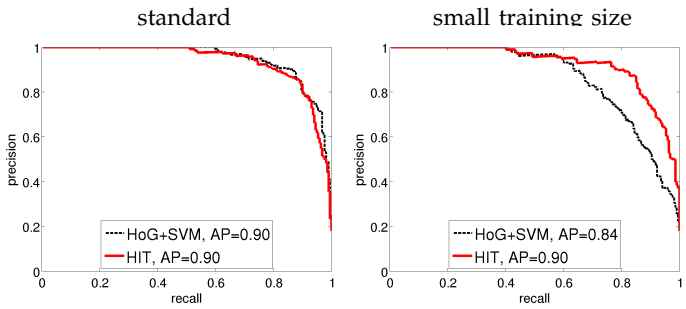


Fig. 17: Comparison on VOC2007 bike using precision-recall.

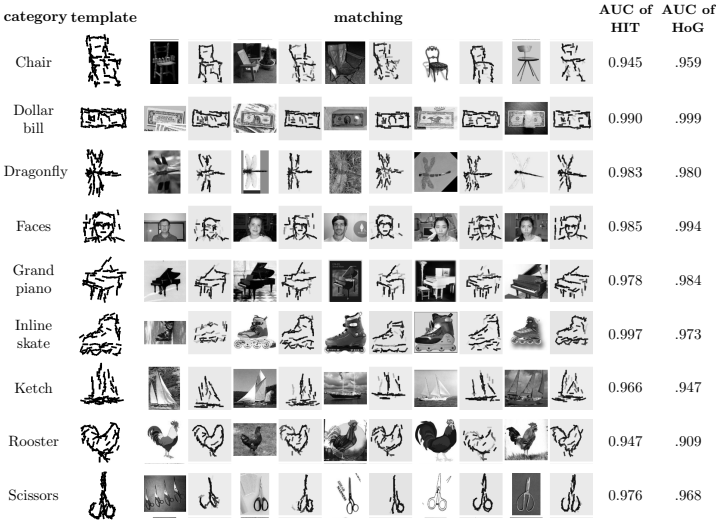


Fig. 18: Learned templates on selected categories in Caltech-101 and the one-vs-all classification accuracy measured by AUC (area under ROC curve) on test images.

positive per window). The lower curve indicates smaller miss rate and better performance. In the standard setting of training size, HIT is on par with HoG and performs much better than HoG at very low false positive rate (*e.g.* 10^{-6}). While using a small training size, HIT has a lower miss rate than HoG for the whole range of FPPW. Figure 15 shows top false positives and miss detections when using HIT to detect persons. Most of false positives have person-like contour in the clutter; while most miss detections correspond to unfamiliar pose or large occlusion.

PASCAL VOC 2007. In Figure 16 and 17, we compare with HoG using precision-recall curves on two Pascal VOC2007 categories: bike and horse. Similar to the INRIA person experiment, we use two settings of training sample sizes. For standard training size (the left sub-figures in Figure 16 and 17), we use all the 241 horse and 139 bike images in TRAINVAL. For small training size (the right sub-figures), we only use the first 20 positive examples. The images in TEST (232 horse and 221 bike images) are used as testing positives. We collect the positive examples by cropping a square patch around the bounding box annotation with a 10% margin and resizing them to 150x150 pixels. For negative examples, we use 150x150 patches cropped from background images as in our INRIA person experiment. It is observed that the performance of HIT is on par with HoG using all training positives. While using fewer training examples, HIT wins over HoG with a big margin.

Caltech-101. For this dataset, the HITs are learned in a translational invariant fashion: during template learning, a hidden variable that accounts for unknown object location needs to be inferred for each image. For every category, we perform 10 EM iterations with a simple initialization that all objects are located in the center. To deal with different aspect ratios of the images, we "inscribe" all images inside a square of 150 by 150 pixels with coinciding centers. Figure 18 lists a subset of learned templates, with 15 training images per category. For illustration purpose we only show the sketch features, and their detections on example images. AUC is shown on the right with comparison to HoG feature. The translational invariant template is able to detect itself and find its detailed correspondences in images, despite object deformation and uncertain location (e.g. faces).

6 Discussion

In this paper, we present a framework for learning a fully generative representation, hybrid image templates, which integrate sketch, texture, flatness and color patches for image modeling. A key advantage of this model, in comparison to previous Gibbs models in language modeling [18] and texture modeling [19], is that it has a much lower computational complexity due to two properties: i) The four types of patches are prototyped into various subspaces (ϵ-balls) and then projected into 1D response r; and ii) the selected patches are mostly independent of each other and thus are factorized in the model so that the parameters and normalizing constants can be computed from a small number of examples. As the comparison of heterogeneous multi-scale features is performed on the relative frequencies of features (*i.e.* the likelihood ratio, p/q) between positive examples and background examples, we make sure all features are compared on the same physical unit (number of bits) in the information projection framework. For classification, the proposed HIT model has a sparser representation than many state-of-art methods (*e.g.* [1], [13]) and demonstrates good performances on par with state-of-art methods on commonly used benchmarks, especially when the training sample size is small.

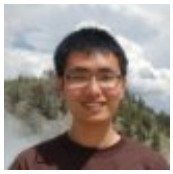
The success of HIT relies on the assumption that objects have a stable configuration with limited structural variation, so that the object can be roughly aligned in the training examples. In our on-going work, we are learning part-based hierarchical models on the HIT configurations to account for explicit structural variabilities within each category.

Acknowledgement: We would like to thank Dr. Ying Nian Wu for his valuable suggestions in experiments and writing. We also thank three anonymous reviewers for their insightful comments. The work is supported by NSF IIS1018751, DMS 1007889 and ONR N000141010933.

Reproducibility page:
www.stat.ucla.edu/∼zzsi/hit.html.

REFERENCES

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.
[2] Y. N. Wu, Z. Si, H. Gong, and S.-C. Zhu, "Learning active basis model for object detection and recognition," *International Journal of Computer Vision*, 2009.
[3] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–714, 1986.
[4] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607 – 609, June 1996.
[5] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.
[6] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 300–312, 2007.
[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.
[8] Y. N. Wu, C.-E. Guo, and S.-C. Zhu, "From information scaling to regimes of statistical models," *Quarterly of Applied Mathematics*, vol. 66, pp. 81–122, 2008.
[9] A. L. Yuille, P. Hallinan, and D. Cohen, "Feature extraction from faces using deformable templates," *International Journal of Computer Vision*, vol. 8, pp. 99–111, 1992.
[10] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.
[11] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.
[12] R. Fergus, P. Perona, and A. Zisserman, "Weakly supervised scale-invariant learning of models for visual recognition," *International Journal of Computer Vision*, vol. 71, no. 3, pp. 273–303, March 2007.
[13] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.
[14] L. Zhu, Y. Chen, A. Torralba, W. Freeman, and A. Yuille, "Part and appearance sharing: Recursive compositional models for multi-view multi-object detection," in *CVPR*, 2010.
[15] N. Ahuja and S. Todorovic, "Learning the taxonomy and models of categories present in arbitrary images," in *ICCV*, 2007.
[16] S. Fidler and A. Leonardis, "Towards scalable representations of object categories: Learning a hierarchy of parts," in *CVPR*, 2007.
[17] C.-E. Guo, S.-C. Zhu, and Y. N. Wu, "Primal sketch: integrating structure and texture," *Computer Vision and Image Understanding*, pp. 5–19, 2007.
[18] S. D. Pietra, V. D. Pietra, and J. Lafferty, "Inducing features of random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.
[19] S.-C. Zhu, Y. N. Wu, and D. B. Mumford, "Minimax entropy principle and its applications to texture modeling," *Neural Computation*, vol. 9(8), no. 2, pp. 1627–1660, 1997.
[20] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, p. 2002, 2000.
[21] J. H. Friedman, "Exploratory projection pursuit," *Journal of American Statistics Association*, vol. 82, no. 397, pp. 249–266, 1987.
[22] E. B. Sudderth, A. B. Torralba, W. T. Freeman, and A. S. Willsky, "Describing visual scenes using transformed objects and parts," *IJCV*, vol. 77, no. 1-3, pp. 291–330, 2008.
[23] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, pp. 119–139, 1997.
[24] V. N. Vapnik, *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.
[25] F. Fleuret, "Fast binary feature selection with conditional mutual information," *Journal of Machine Learning Research*, vol. 5, p. 15311555, 2004.
[26] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "Svm-knn: Discriminative nearest neighbor classification for visual category recognition," in *CVPR*, 2006.
[27] A. Bissacco, M.-H. Yang, and S. Soatto, "Detecting humans via their pose," in *NIPS*, 2007.
[28] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in *ICCV*, October 2007.
[29] A. Opelt, A. Pinz, and A. Zisserman, "Learning an alphabet of shape and appearance for multi-class object detection," *International Journal of Computer Vision*, 2008.
[30] X. Ma and W. E. L. Grimson, "Learning coupled conditional random field for image decomposition with application on object categorization," in *CVPR*, 2008.
[31] P. Gehler and S. Nowozin, "On feature combination for multiclass object detection," in *ICCV*, 2009.
[32] C. Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz, "Exploring features in a bayesian framework for material recognition," in *CVPR*, 2010.
[33] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Object recognition with cortex-like mechanisms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 411–426, 2007.
[34] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.
[35] B. Heisele, P. Ho, and T. Poggio, "Face recognition with support vector machines: global versus component-based approach," in *ICCV*, 2001.
[36] R. A. Epstein, W. E. Parker, and A. M. Feiler, "Two kinds of fmri repetition suppression? evidence for dissociable neural mechanisms," *Journal of Neurophysiology*, vol. 99, pp. 2877–2886, 2008.
[37] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.
[38] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories," in *CVPR Workshop on Generative-Model Based Vision*, 2004.



Zhangzhang Si received a BS degree in Computer Science from Tsinghua Univ. in 2006 and a PhD degree of Statistics from UCLA in 2011. He is a postdoc researcher in the Center of Image and Vision Science at UCLA, where his research focuses on deformable image models for object recognition and unsupervised learning. He received a Marr Prize honorary nomination in 2007 with Y. N. Wu *et al.* .

Song-Chun Zhu received a BS degree from the Univ. Sci.& Tech. of China in 1991 and a PhD degree from Harvard University in 1996. He is a professor with the Department of Statistics and the Department of Computer Science at UCLA. His research interests include computer vision and learning, statistical modeling, and stochastic computing. He received a number of honors, including the David Marr Prize in 2003 with Z. Tu *et al.* , the J.K. Aggarwal prize from the Intl Association of Pattern Recognition in 2008, the Marr Prize honorary nominations in 1999 and 2007 with Y.N. Wu *et al.* , a Sloan Fellowship in Computer Science in 2001, a US National Science Foundation Early Career Development Award in 2001, and an US Office of Naval Research Young Investigator Award in 2001. In 2005, he founded, with friends, the Lotus Hill Institute for Computer Vision and Information Science in China as a nonprofit research organization (www.lotushill.org). He is a Fellow of IEEE.