Learning Mixed Templates for Object Recognition

Zhangzhang Si[1] Haifeng Gong[1,2] Ying Nian Wu[1] Song-Chun Zhu[1,2]
[1] Department of Statistics, UCLA [2] Lotus Hill Research Institute
{zzsi, hfgong, ywu, sczhu}@stat.ucla.edu

Abstract

This article proposes a method for learning object templates composed of local sketches and local textures, and investigates the relative importance of the sketches and textures for different object categories. Local sketches and local textures in the object templates account for shapes and appearances respectively. Both local sketches and local textures are extracted from the maps of Gabor filter responses. The local sketches are captured by the local maxima of Gabor responses, where the local maximum pooling accounts for shape deformations in objects. The local textures are captured by the local averages of Gabor filter responses, where the local average pooling extracts texture information for appearances. The selection of local sketch variables and local texture variables can be accomplished by a projection pursuit type of learning process, where both types of variables can be compared and merged within a common framework. The learning process returns a generative model for image intensities from a relatively small number of training images. The recognition or classification by template matching can then be based on log-likelihood ratio scores. We apply the learning method to a variety of object and texture categories. The results show that both the sketches and textures are useful for classification, and they complement each other.

1. Introduction

There has been large amount of work on designing image features to represent visual patterns of different types. Primitive features (e.g. [7]) explicitly record locations of edges/bars and are good features for image patches of clean object boundaries. We may generally call them sketch features. In contrast, texture patches tend to be better described by histogram features (e.g. [5], [1]). In terms of object recognition, sketch features are shown to work well on objects with regular shapes, while histogram features seem more suitable for complex objects with small inner structures and moderate deformations.

We propose a model for mixed templates learnable from example images, where each constituent component of a template is a sub-template for a local patch inside the template image lattice. "Convolving" the template on an image provides a response vector on which we can build our statistic model. In the hedgehog example of Fig.1, the image is decomposed into two types of local patches: those with strong edges lying on the object boundaries that tend to be described by image primitives, and those with cluttered structures that are described by local histograms. Image sketches and local histograms compete to explain different local patches of the images.

To learn such a mixed template for a certain image category, we may use discriminative methods or generative models. In many recent papers (e.g. [8][11][14][6]) feature combination is performed towards a discriminative goal by concatenating long feature vectors and learning weights on them. When there are a large number of categories with relatively small number of training images, it is desirable to have a generative learning framework, where the probability density function of the image intensities can be written in the form of a background density multiplied by a likelihood ratio term. In such a framework, different types of features are made comparable to each other by an information gain criterion. We further constrain that the local sketches have little overlap with each other, and so are the local textures. Under such a constraint, the likelihood function can be factorized to make possible a fast and robust estimation of the model parameters and the normalizing constant. This may generalize to many other image features.

The work closest to ours is the active basis model ([12]), which learns a shape template composed of Gabor wavelet elements at different locations. We extend this work to a more general case by modeling both sketch patches and texture patches. We not only identify the commonly shared sketches at different locations and orientations that have strong responses across all the training images, but also commonly shared textures at different locations, so that at each location, the variance of the texture statistics across the training images is small. Before learning the image template, we make the training images roughly aligned.

Contribution. First, we propose a simple method for

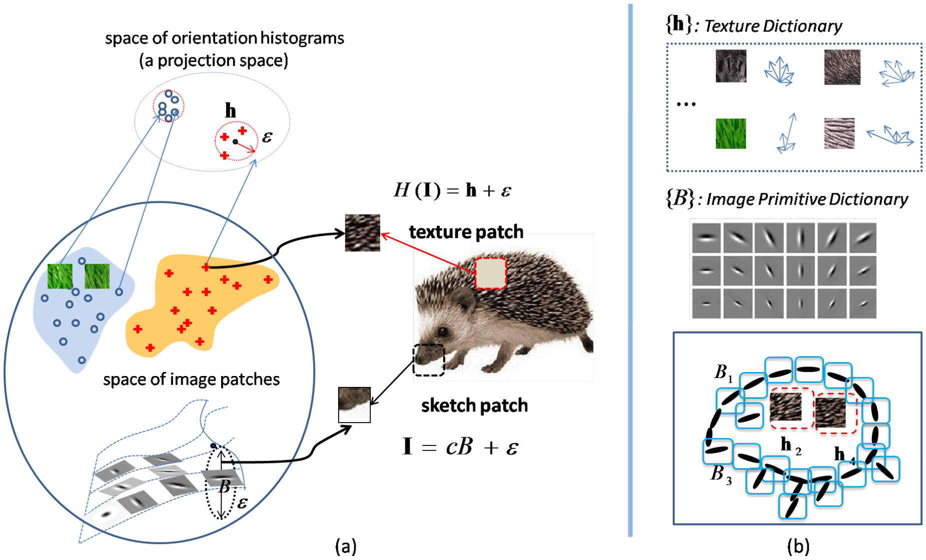


Figure 1. **(a)** A hedgehog image may be seen as a bunch of local image patches, being either sketches or textures. If a local patch $\mathbf{I}$ falls within an ϵ-ball in the image space, i.e. $\mathbf{I} = cB + \epsilon$, where B is a geometric primitive and ϵ is the residual, then it is explained by the primitive B. If a local patch $\mathbf{I}$ falls within an ϵ-ball in the histogram feature space, i.e., $H(\mathbf{I}) = \mathbf{h} + \epsilon$, where H is some histogram statistics, then the local patch is explained by texture "prototype" $\mathbf{h}$. **(b)** Quantization in the image space and histogram feature space provides a primitive dictionary $\{B\}$ and a texture dictionary $\{\mathbf{h}\}$ respectively, which compete to explain observed image patches. A mixed template of hedgehog $T = \{B_1, \mathbf{h}_2, B_3, \mathbf{h}_4, \cdots\}$ is composed of sketches and histogram prototypes explaining local image patches at different locations.

learning mixed templates composed of sketch and texture variables and a common information theoretic criterion to compare and merge these two different types of variables. Second, we evaluate the strengths of sketch and texture variables and their complementarity by applying the learning method to a variety of object and texture categories.

2. A uniform design of sketch and texture variables

In this paper we adopt a design where sketch and texture features are both defined on Gabor filtered images (or S1 maps as in [10]). To fix notation, let $\mathcal{O}$ be a quantized set of orientations, and Λ be the image lattice of a template.

Sketch by local maximum pooling. We use Gabor wavelets as the primitive dictionary, and use $B_{x,y,o,s}$ to denote the Gabor wavelet element located at position (x, y), orientation o and scale s. Let $\Lambda_j \subset \Lambda$ be a region inside image template. Consider local image patch $\mathbf{I}_{\Lambda_j}$ centered at (x, y) and explained by a Gabor wavelet element $B_{x,y,o,s}$ with an additive residual:

$$\mathbf{I}_{\Lambda_j} = c \cdot B_{x+\Delta x, y+\Delta y, o+\Delta o, s} + \epsilon,$$

where c denotes the coefficient, $(\Delta x, \Delta y, \Delta o)$ are the local perturbations of location and orientation of $B_{x,y,o,s}$. The reason we need to incorporate $(\Delta x, \Delta y, \Delta o)$ is that the edge segments in the training images are not exactly aligned, and there can be shape deformations.

Then the sub-template for local patch $\mathbf{I}_{\Lambda_j}$ is the primitive B, and we may define the *sketch* variable (feature response) for $\mathbf{I}_{\Lambda_j}$ as

$$\mathbf{r}^{(sk)}(\mathbf{I}_{\Lambda_j}; o, s) = \max_{\Delta x, \Delta y, \Delta o} \mathrm{s}(|\langle \mathbf{I}_{\Lambda_j}, B_{x+\Delta x, y+\Delta y, o+\Delta o, s} \rangle|^2). \quad (1)$$

where $s()$ is a sigmoid-like transformation, i.e., a monotone function that increases from 0 to a saturation level. The response $|\langle \mathbf{I}_{\Lambda_j}, B \rangle|^2$ is normalized globally by the average response over the whole image template lattice Λ. The local maximum pooling is proposed by [10] as a possible function of complex cells in V1.

Texture by local average pooling. A locally normalized orientation histogram of Gabor responses describes the relative frequency of edges at different orientations within a local patch, without explicitly specifying where the edges are. The orientation histogram pooled over the region Λ_j is defined as:

$$H(\mathbf{I}_{\Lambda_j}) = (h_1, ..., h_{|\mathcal{O}|}),$$

where $\sum_o h_o = 1$ and

$$h_o \propto \sum_{(x,y) \in \Lambda_j} \mathrm{s}(|\langle \mathbf{I}, B_{x,y,o,s} \rangle|^2).$$

We may entertain two types of texture variables (feature responses). One is simply

$$\mathbf{r}^{(tex)}(\mathbf{I}_{\Lambda_j}; o, s) = h_o,$$

which is similar to sketch variable in Eq.(1).

The other type is slightly more complex. If the local image patch $\mathbf{I}_{\Lambda_j}$ is explained by a "prototype" orientation histogram $\mathbf{h}$ with additive residual ϵ:

$$H(\mathbf{I}) = \mathbf{h} + \epsilon,$$

then the texture sub-template for local patch $\mathbf{I}_{\Lambda_j}$ is the prototype orientation histogram $\mathbf{h}$ and we model how well $\mathbf{h}$ explains $\mathbf{I}_{\Lambda_j}$ by the squared Euclidean distance

$$r^{(tex)}(\mathbf{I}_{\Lambda_j}) = \|H(\mathbf{I}_{\Lambda_j}) - \mathbf{h}\|^2. \tag{2}$$

The connection between the above orientation histogram and SIFT/HoG features is obvious. The orientation histogram computed at a small scale would resemble the orientation histogram of gradients. Since the local areas to pool orientation histograms are subject to selection, an image template made of orientation histogram features $(H(\mathbf{I}_{\Lambda_1}), ..., H(\mathbf{I}_{\Lambda_k}))$ may be considered a customizable version of HoG.

3. Learning mixed templates

We introduce the learning algorithm for extracting a mixed template from example images $\{\mathbf{I}_1, ..., \mathbf{I}_n\}$, where each sub-template is either a Gabor wavelet element or a prototype histogram. Let $r_1(\mathbf{I}), ..., r_k(\mathbf{I})$ be the corresponding feature responses from image $\mathbf{I}$, as defined in Eq.(1) and (2). Our model is built on these responses.

3.1. Feature pursuit

Let $f(\mathbf{I})$ be the target image distribution where positive exmaples $\{\mathbf{I}_1, ..., \mathbf{I}_n\}$ are sampled from. Let $q(\mathbf{I})$ be a reference image distribution, such as the uniform distribution on all the natural images, from which we have a sample of negative examples $\{\mathbf{J}_1, ..., \mathbf{J}_N\}$. The learning method selects most informative features and construct a probability density function $p(\mathbf{I})$. Let $r_1(\mathbf{I}), \cdots, r_k(\mathbf{I})$ be the responses of selected features, in principle we would like to let our model agree upon dimensions $(r_1, \cdots, r_k)$ with the target distribution, i.e. $p(r_1, \cdots, r_k) = f(r_1, \cdots, r_k)$, where $p(r_1, \cdots, r_k)$ is the distribution of $(r_1, \cdots, r_k)$ under $p(\mathbf{I})$, and $f(r_1, \cdots, r_k)$ is the distribution of $(r_1, \cdots, r_k)$ under $f(\mathbf{I})$.

Among all such models $\{p_k : p_k(r_1, \cdots, r_k) = f(r_1, \cdots, r_k)\}$, the one with minimum Kullback-Leibler divergence to q would have the following form [9]:

$$p_k(\mathbf{I}) = q(\mathbf{I})\lambda(r_1, ..., r_k), \tag{3}$$

where $\lambda(r_1, ..., r_k)$ is the likelihood ratio,

$$\lambda(r_1, ..., r_k) \triangleq f(r_1, ..., r_k)/q(r_1, ..., r_k).$$

$p_k(\mathbf{I})$ is a modification of $q(\mathbf{I})$ in the following sense. (1) We change the distribution of $r_1, ..., r_k$ from $q(r_1, ..., r_k)$ to $f(r_1, ..., r_k)$. (2) We keep the conditional distribution of the remaining dimensions given $r_1, ..., r_k$ to be the same.

However, we cannot select all the features at once. So we adopt the following sequentia0l pursuit scheme: We start from the reference distribution $p_0(\mathbf{I}) = q(\mathbf{I})$, and at the j-step, we select a feature r_j , so that $p_j(\mathbf{I}) = p_{j-1}(\mathbf{I})p_j(r_j)/p_{j-1}(r_j)$, where $p_j(r_j)$ is pooled from positive training images, and $p_{j-1}(r_j)$ is the distribution of r_j under existing model $p_{j-1}(\mathbf{I})$. In each step, we choose the feature r_j so that the Kullback-Leibler divergence $\mathcal{K}(p_j(r_j)\|p_{j-1}(r_j))$ is the largest. $\mathcal{K}(p_j(r_j)\|p_{j-1}(r_j))$ measures the information gain after adding the new feature r_j . It also measures the decrease in the divergence from the true distribution $f(\mathbf{I})$.

The above process is essentially a generalized version of projection pursuit ([4]). We do not have to know $q(\mathbf{I})$ explicitly. Moreover, if we enforce that the selected features have little overlap, then we may simply assume that $p_{j-1}(r_j) = q(r_j)$ as an approximation. So we only need to pool $q(r_j)$ from natural images. That means the likelihood ratio can be factorized, and $p_k(\mathbf{I}) = q(\mathbf{I}) \prod_{j=1}^{k} [p_j(r_j)/q(r_j)]$.

3.2. Log-linear model

If the selected features have little overlap, we may assume a factorized log-linear form for p_k/q, so that:

$$p_k(\mathbf{I}) = q(\mathbf{I}) \prod_{j=1}^{k} \left[\exp\{\lambda_j r_j(\mathbf{I})\} z_j^{-1}\right],$$

where $z_j = E_q\left[\exp\{\lambda_j r_j(\mathbf{I})\}\right]$ is the normalizing constant for the j-th term. And $\mathcal{K}(p_k\|q)$ can be written as:

$$\mathcal{K}(p_k||q) = \sum_{j=1}^{k} (\lambda_j E_p[r_j] - \log z_j). \tag{4}$$

This suggests a step-wise algorithm that selects one feature at a time, which maximizes the marginal KL divergence between $p_j(r_j)$ and $q(r_j)$. $q(r_j)$ can be estimated off-line.

The *template matching score* (Fig.2) for any observed image $\mathbf{I}$ is then

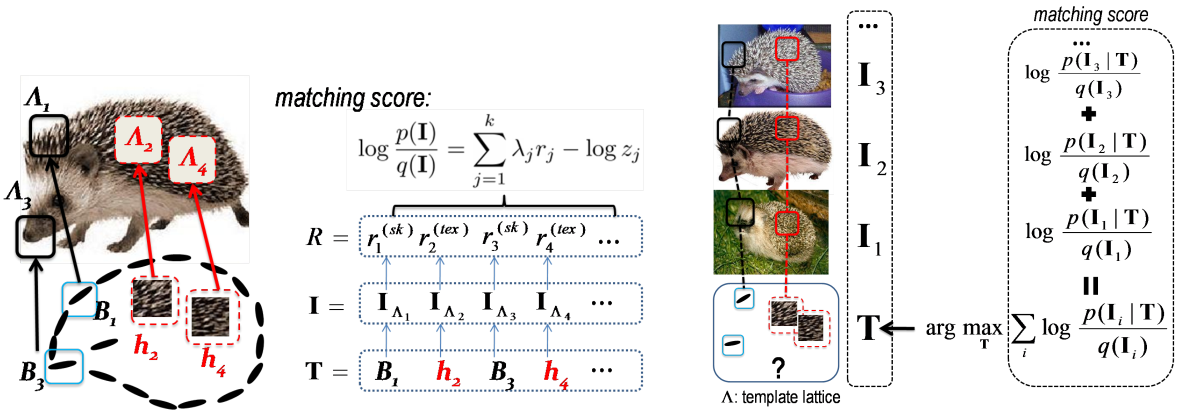
$$\text{Score}(\mathbf{I}) = \log \frac{p_k(\mathbf{I})}{q(\mathbf{I})} = \sum_{j=1}^{k} (\lambda_j r_j(\mathbf{I}) - \log z_j), \tag{5}$$

which can make binary decisions given a threshold.

3.3. Modeling sketch features

Let B be a Gabor wavelet explaining local patch Λ_j. The marginal distribution $p(r^{(sk)}) = p(\mathrm{s}(|\langle \mathbf{I}_\Lambda, B\rangle|^2))$ is:

$$p(r^{(sk)}) \quad = \quad q(r^{(sk)}) \exp\{\lambda r^{(sk)}\} z^{-1}. \tag{6}$$


Λ_1
Λ_2
Λ_3
Λ_4
B_1
B_3
h_2
h_4
matching score:
$\log\frac{p(\mathbf{I})}{q(\mathbf{I})}=\sum_{j=1}^{k}\lambda_j r_j-\log z_j$
$R = r_1^{(sk)}\ r_2^{(tex)}\ r_3^{(sk)}\ r_4^{(tex)}\ \dots$
$\mathbf{I} = \mathbf{I}_{\Lambda_1}\ \mathbf{I}_{\Lambda_2}\ \mathbf{I}_{\Lambda_3}\ \mathbf{I}_{\Lambda_4}\ \dots$
$\mathbf{T} = B_1\ h_2\ B_3\ h_4\ \dots$
...
$\mathbf{I}_3$
$\mathbf{I}_2$
$\mathbf{I}_1$
$\mathbf{T}$
?
Λ: template lattice
matching score
...
$\log\frac{p(\mathbf{I}_3\,|\,\mathbf{T})}{q(\mathbf{I}_3)}$
+
$\log\frac{p(\mathbf{I}_2\,|\,\mathbf{T})}{q(\mathbf{I}_2)}$
+
$\log\frac{p(\mathbf{I}_1\,|\,\mathbf{T})}{q(\mathbf{I}_1)}$
||
arg $\max_{\mathbf{T}}\sum_i\log\frac{p(\mathbf{I}_i\,|\,\mathbf{T})}{q(\mathbf{I}_i)}$

(a) (b)

Figure 2. Template matching and template learning. (a) Matching the mixed template to a hedgehog image. Each component of the template is matched to a local patch and produces a response r calculated for sketch and texture in Eq.(1) and (2) respectively. The mixed template gives each observed image a template matching score that takes the form of log-likelihood ratio, which is a linear combination of individual feature responses. (b) Learning the mixed template. The best template to explain a set of images $\{\mathbf{I}_1, ..., \mathbf{I}_n\}$ is the one that scores highest on them, i.e., the one that achieves maximum likelihood. Sketch features and texture features compete to be better explanations for local patches across image examples, and the information gain in Eq.(10)(14) measures their contributions to the template. By projecting image patch $\mathbf{I}_{\Lambda_j}$ to one dimensional response r_j , we are able to get robust estimation for model parameter λ_j and the normalizing constant z_j.

We need to find λ such that:

$$E_p[r;\lambda] = \bar{r} \triangleq \frac{1}{n}\sum_{i=1}^{n} r(\mathbf{I}_i) \approx E_f[r]. \quad (7)$$

Given the log-linear form we are able search efficiently for λ by a simple look-up table. For a grid of possible values of λ: $(\lambda^{(1)}, ..., \lambda^{(M)})$ in ascending order, we estimate their associated z's and $E_p[r]$'s by importance sampling on a set of random natural images (negative examples) $\{\mathbf{J}_1, ..., \mathbf{J}_N\}$:

$$z^{(l)} \approx \frac{1}{N}\sum_{i=1}^{N} e^{\lambda^{(l)} r(\mathbf{J}_i)}, \quad (8)$$

$$E_p[r;\lambda^{(l)}] \approx \frac{1}{N}\sum_{i=1}^{N}\left[r(\mathbf{J}_i)e^{\lambda^{(l)} r(\mathbf{J}_i)}\right]\frac{1}{z^{(l)}}. \quad (9)$$

Then we look up $\bar{r}$ in the table to find the best λ. This Monte-Carlo approach is reasonable because in our design r is one dimensional. So a moderate sample size would be able to provide a robust estimate for z.

Following the above analysis, among all pairs of (B, Λ_j), we select the one that has maximum *information gain*, defined as the marginal KL divergence

$$\begin{aligned}\text{gain}(B, \Lambda_j) &= \mathcal{K}(p(r^{(sk)})||q(r^{(sk)}))\\ &\approx \frac{1}{n}\sum_{i=1}^{n}\lambda\cdot \text{s}(|\langle \mathbf{I}_{i,\Lambda_j}, B\rangle|^2) - \log z. \end{aligned}\quad (10)$$

where the estimation of λ and z is already explained. For a sketch feature λ is positive, so the above information gain selects the $r^{(sk)}$ with the largest sample mean.

3.4. Modeling texture features

The marginal distribution on $r^{(tex)}$ is:

$$p(r^{(tex)}) = \frac{1}{z}q(\|H - \mathbf{h}\|^2)e^{\lambda\|H-\mathbf{h}\|^2}. \quad (11)$$

$\mathbf{h}$ is an additional parameter. The optimal $\mathbf{h}^*$ is obtained by averaging over positive training images:

$$\mathbf{h}^* = \frac{1}{n}\sum_{i=1}^{n} H(\mathbf{I}_i). \quad (12)$$

Although λ and the corresponding information gain can be computed similarly to Eq.(10) ($\lambda < 0$ for texture feature), in this paper, we simply assume the following Gaussian-like form

$$p(r^{(tex)}) = \frac{1}{z}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\{-\frac{1}{2\sigma^2}r^{(tex)}\}q(r^{(tex)}), \quad (13)$$

where σ^2 is estimated by $\sum_{i=1}^{n}\|H(\mathbf{I}_i) - \mathbf{h}^*\|^2/n$, and z is approximated by Monte Carlo method.

The information gain for a histogram feature $(\mathbf{h}, \Lambda_j)$ is then defined as:

$$\begin{aligned}\text{gain}(\mathbf{h}, \Lambda_j) &= \mathcal{K}(p(r^{(tex)})||q(r^{(tex)}))\\ &\approx -\log z - \log\sqrt{2\pi\sigma^2} - \frac{1}{2\sigma^2}\frac{1}{n}\sum_{i=1}^{n}\|H(\mathbf{I}_i) - \mathbf{h}\|^2\\ &= -\log z - \log\sqrt{2\pi\sigma^2} - 1/2. \end{aligned}\quad (14)$$

That histogram features with the least σ^2 are more informative.

An alternative way to model the texture feature is

$$p(H_o(\mathbf{I})) = q(H_o(\mathbf{I})) \exp\{\lambda_o \cdot H_o(\mathbf{I})\} z_o^{-1}, \qquad (15)$$

where each orientation bin $H_o(\mathbf{I})$ is treated as one feature response. The learning of such a model may proceed similarly as the learning of the sketch model.

3.5. Decouple sketch and texture by adaptive background

If we already include a local texture feature that describes a local patch, then if we want to add a local sketch r_j within the same patch, then in $p_j(\mathbf{I}) = p_{j-1}(\mathbf{I}) p_j(r_j)/p_{j-1}(r_j)$, the $p_{j-1}(r_j)$ should account for the local texture variable, in other words, we should let $p_{j-1}(r) = q(r)\exp(-\lambda r)/z(\lambda)$ for some λ, so that $E_\lambda(r)$ matches the local average of the Gabor filter responses (after sigmoid transformation). p_{j-1} fitted in this way then serves as an adaptive background model for the newly selected sketch variable r_j (Fig.3). In other words, we should measure the local maximum against the local average, in order to decouple sketch and texture.

Currently we adopt the adaptive background in our implementation of template matching, where the averages are pooled over either global or local image lattice for each testing image, at different orientations. We then score the strength of a sketch variable against this adaptive background in the testing stage.

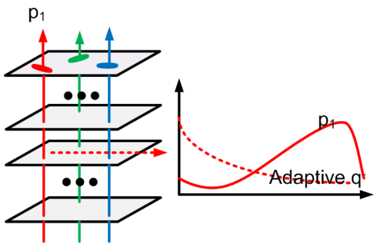


Figure 3. Sketch arises from adaptive textural background. For each image $\mathbf{I}$, adaptive q's are pooled over the Gabor responses for different orientations. Such adaptive q's capture texture information in image $\mathbf{I}$. Each $p(r^{(sk)})$ is paired with an adaptive $q(r^{(sk)})$ at the same orientation.

The stepwise learning algorithm for mixed image templates is described in table 1.

4. Experiment

4.1. Classification experiments

Single scale mixed templates. Firstly, we study the sketch/texture combination for a single scale of Gabor elements in the context of classifying objects from background clutters. In Fig.5, to compare the sketch-only, texture-only and mixed templates, the area under ROC curve (AUC) is averaged over cross validation runs and plotted against the number of positive training examples.

```
Algorithm. Stepwise pursuit for sketches and textures.
Let template $T = empty$ .
For each scale of Gabor filter $s$:
  Compute maps of Gabor response for example images $S_i(x,y,o) = \mathrm{s}(|\langle \mathbf{I}_i, B_{x,y,o,s}\rangle|^2)$
  Pursuit of sketch features:
  Repeat:
    (1) Select $(x,y,o)$ with the largest mean response (locally maximized): $\frac{1}{n}\sum_{i=1}^n \max_{(x',y',o')\in\partial(x,y,o)} S_i(x',y',o')$;
    (2) Compute information gain by Eq.(10).
    (3) Add $(B_{x,y,o,s},(x,y,o,s),gain)$ to $T$, and inhibit nearby positions in each image: for image $\mathbf{I}_i$ find the deformed location $(x_i,y_i,o_i)$, then set $S_i(x',y',o') = 0$ if $(x',y',o')$ is in a small neighborhood of $(x_i,y_i,o_i)$.
  until gain is smaller than a threshold.
  Pursuit of texture features:
  For each radius $\delta$ of local region (to pool histogram):
    (1) Compute maps of local histograms $H_i(x,y,o)$ normalized over orientations.
    $H_i(x,y,o) \propto \sum_{|x'-x|<\delta,|y'-y|<\delta} S_i(x,y,o)$
    (2) Compute average histogram map $\bar{H} \leftarrow \frac{1}{n}\sum_{i=1}^n H_i$.
    (3) Compute variance map:
    $V(x,y) \leftarrow \frac{1}{n}\sum_{i=1}^n \|H(x,y,:) - \bar{H}(x,y,:)\|^2$.
    (4) Set inhibition map $\eta(x,y)$ to be all zeros.
    Repeat:
      (5.1) Select the non-inhibited position $(x,y)$ with smallest variance $V(x,y)$.
      (5.2) Calculate information gain by Eq.(14).
      (5.3) Add $(\bar{H}(x,y,:),(x,y,s,l),gain)$ to $T$, and inhibit nearby positions by setting every $\eta(x',y') = 1$ if $|x'-x| < l$ and $|y'-y| < l$.
    until gain is smaller than a threshold.
Output:
The template $T$ with components sorted by information gain.
```

Table 1. Stepwise pursuit for sketches and textures

We test on four categories: human head/shoulder, cat head, swine head and hedgehog (each with about 100 positive examples) for binary classification versus a common set of 600 random negative images. We resize images to have an area of about 120 × 120 pixels while keeping the aspect ratio unchanged. Both sketch and texture features are represented by Gabor filters of 17 × 17 pixels. And for Gabor filters we use the same parameters as in [12]. Sketches are allowed to move 6 pixels at most and $\pi/15$ in orientation. The radius of local texture is 10 pixels. When we fit the adaptive background model $p_{j-1}(r)$ for sketch features (see sec.3.5), we pool from a local neighborhood of the same radius. An threshold of 0.4 is used as a stopping criterion for both sketch and texture features. The selected Gabor filters are enforced to overlap no larger than a threshold ($|\langle B_i, B_j \rangle|^2 < 0.1$). Local histograms are also only allowed to overlap 25% of the area ($|\Lambda_i \cup \Lambda_j| < 0.25 \max(|\Lambda_i|, |\Lambda_j|)$).

Among the four categories the first two are relatively easy (AUC $\approx 99\%$), and the rest are of moderate difficulty. The combined model (mixed template) is able to provide a significant improvement over sketch/texture-only templates for all the four categories in terms of AUC on the testing data (for most of the training sample sizes). We also experimented with a log linear model on $H(\mathbf{I})$ (Eq.15) with local adaptive background for sketch features as explained in section 3.5, which gives competitive results to its Gaussian counterpart (Eq.11).

To see which sketch and texture features are selected and why they complement with each other, we display the learned mixed templates from 10 examples in Fig.6 for pig head and hedgehog, and their matching results on the training examples.

Multi-scale mixed templates. We also extend the experiments to mixed templates on multi-scales Gabors and 100+ image categories. The dataset includes 60 object categories and 41 homogeneous texture categories. Part of the categories are selected from Caltech-101([3]) and CUReT texture database ([2]). The dataset is made reasonably difficult by object categories easy to confuse, e.g., 18 categories of animal faces, and some similar texture categories.

Instead of using different scales of Gabors, we change the image lattice size (or area) $|\Lambda| =$ width $\times$ height to be 100^2, 150^2, 200^2, but keep Gabors on the same scale, 17^2 pixels. We also vary the neighborhood size used to pool orientation histograms from 11^2, 21^2 to 41^2. Images are transformed to grayscale, and are resized to have the specified image area while preserving the original aspect ratio.

For training and testing, in each category we randomly select 15 examples as training positives, and the rest (at most 50) are used for testing (around 4200 images are used for testing in total). To compute λ and z for sketch features, we use an independent random sample from all categories. The Gaussian-like form is used for texture models. We use a universal threshold 0.1 on information gain as the stopping criterion of feature selection. On average about 200 features are selected per category (or per template).

We evaluate the learned templates in one-versus-all classifications and for each template we measure its average precision (the area under precision-recall curve). Each box-and-whisker diagrams in Fig.7 describes the distribution of average precisions over all categories. The mixed template performs observably better than the individual sketch or texture templates. In Fig.8 we show several examples of mixed templates learned from training images. Though templates are multi-scale, we only show a single scale for the clearness of illustration.

Speed. Learning one mixed template of multiple scales from 15 images takes within one minute after feature convolution. It can be made even faster by sub-sampling pixels.

4.2. Image complexity and feature competition

Image categories of different intrinsic complexities live inside the whole image space. Categories with clear shape patterns have low intrinsic complexities, while cluttered texture categories span large intrinsic dimensions. Based on the above experiments, we study the relationship between this complexity and the relative importance of sketch and texture features by the classification task.

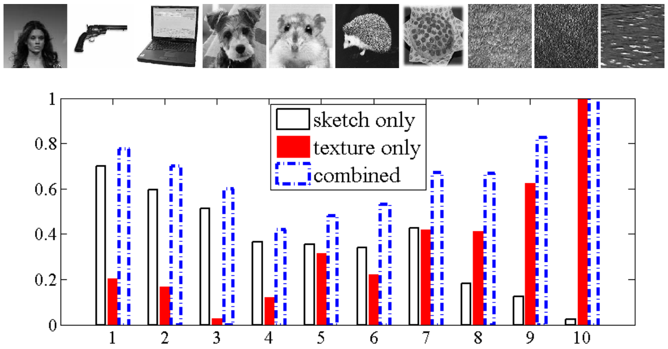
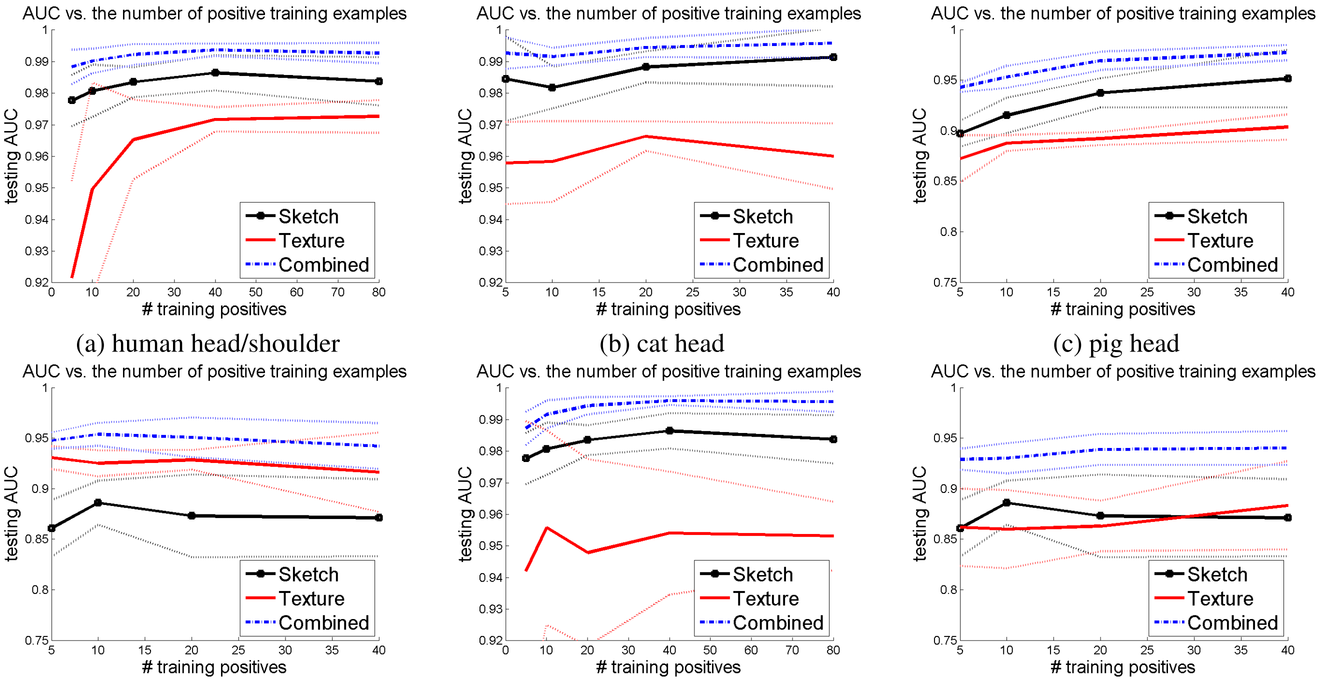


Figure 4. Performance of sketch/texture vs. perceived complexity. **Top:** image categories. We asked human subjects to rank 10 object/texture categories by their perceived complexity. These categories are human head and shoulder, pistol, laptop, dog head, mouse head, hedgehog, pizza and three texture categories. **Bottom:** average precisions (AP) of object categories (ordered as the plot on the top), for image templates using only sketch features, only texture features, and both. Combination of sketch and texture features benefits the most for categories that lie in the mid-complexity area.

Human subjects are asked to rank 10 categories of images by their perceived image complexities. Average precisions of ten categories are shown for sketch-only, texture-only and mixed templates in Fig.4, which illustrates how sketch features are less and less important as the perceived image complexity increases. The mixed template performs constantly better than both sketch-only and texture-only templates, over the whole range of image complexity. It is also indicated in the figure that the combination of sketch and texture benefits most for the image categories at the "mid-complexity" zone, i.e., categories that have neither a clean boundary nor a homogeneous texture pattern.

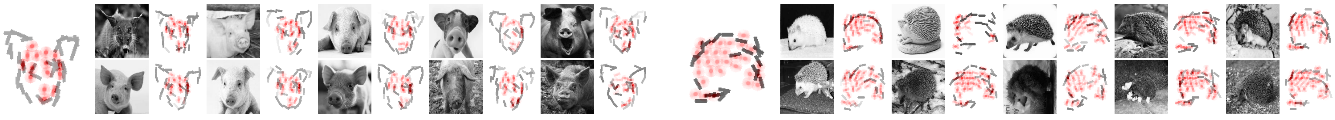
We show the competition of sketches vs. textures in learning mixed templates from four categories (Fig.9). It is observed that the relative importance of texture features correlates positively with the complexity of the image category. Information gains for sketch and texture features are on the same scale, making it possible to compare the two



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

(d) hedgehog (e) head/shoulder (local adaptive) (f) hedgehog (local adaptive)

Figure 5. 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 (a)-(d) the Gaussian texture model (Eq.11) is used. In (e) and (f), the log-linear texture model (Eq.15) is used and is combined with locally adaptive sketch feature responses.



(a) pig head (b) hedgehog

Figure 6. Matching the mixed templates to images. For each of the two figures, on the left is the learned mixed template. Black bars denote sketch features and red dots denote texture features. On the right are matched templates on images. The black bars denote the deformation of the sketches and their responses on observed images. The red dots in the right figure denote the texture features fired on these images. The red blob is generated by a weighted superposing of bar symbols at all orientations, where the weights are coordinates in the orientation histogram. The local texture could be strongly oriented or directionless, depending on different object categories and locations.

completely different types of features.

5. Conclusion

In this paper, we propose a simple and uniform design of sketch and texture variables or features, where both descriptors are extracted from common maps of Gabor filter responses, and can be decoupled by adaptive backgrounds. We start with atomic descriptors of patches with pure sketches or textures, then more complex model can be developed by composing them.

We also adopt a stepwise procedure to automatically select and combine sketch and texture variables based on an information theoretical criterion. The classification exper-

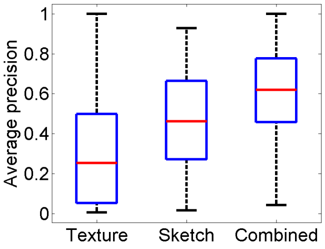


Figure 7. Box plot of average precisions. Each box shows max/min, 25% and 75% percentiles and the median of average precisions on 100+ object/ texture categories.

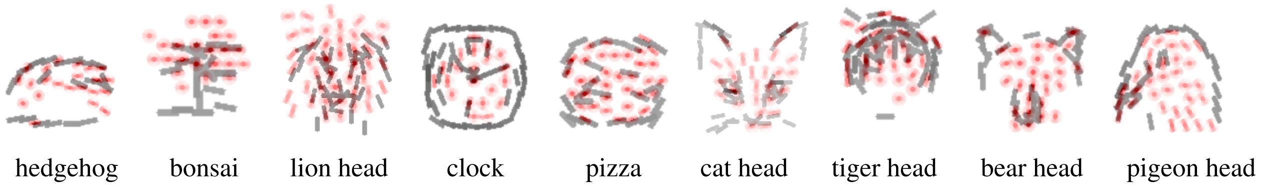


Figure 8. Learned mixed templates of several object categories. Bold black bars denote sketches, while red blobs denote local textures described by orientation histograms. For illustration purpose, we only show sketches/textures of a single scale and vary the (relative) Gabor scales and information gain thresholds for different categories. We use a threshold around 1.35 for sketches and around 1.8 for textures.

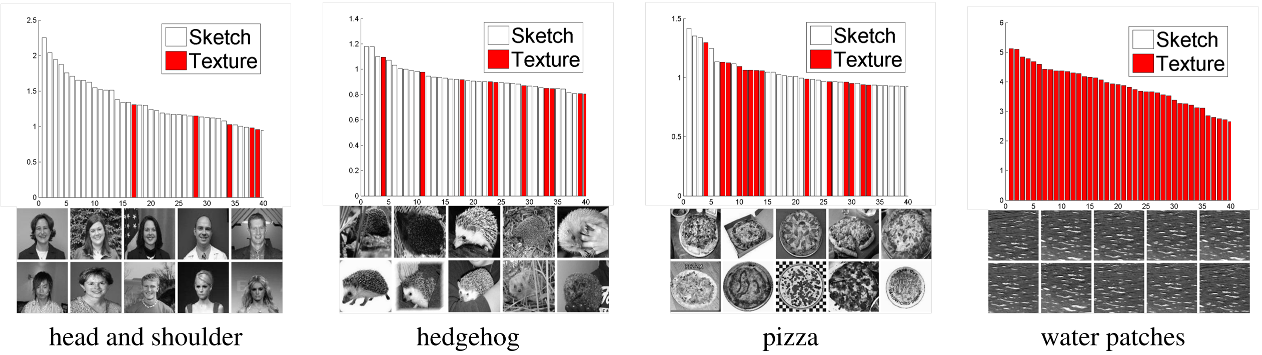


Figure 9. Competition of sketch and texture features. Each figure plots the information gains of the first 40 selected features, ranked in descending order. Hollow black/white bars: information gains of selected sketch features; Solid red bars: information gains of selected texture features. For image categories with clear and regular shape, e.g., head/shoulder, sketch features dominate the information gain. As there are more cluttered structures inside objects, texture features begin to make a bigger contribution. This is seen clearly from the feature competition for hedgehog, pizza and the water patches cropped from a pond image.

iments on various categories verified our hypothesis that sketch and texture variables complement with each other.

Reproducibility page: Source code and data can be found at www.stat.ucla.edu/∼zzsi/mixed_template.html.

Acknowledgement. The authors are supported NSF grants IIS-0713652 and DMS-0707055. The data used in this paper is from the Lotus Hill Institute dataset [13] under the support of NSF China grant 60832004 and 863 project 2008AA01Z126.

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1

[2] K. Dana, B. Van-Ginneken, S. Nayar, and J. Koenderink. Reflectance and texture of real world surfaces. *ACM Transactions on Graphics (TOG)*, 1999. 6

[3] 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. 6

[4] J. H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, 82(397):249–266, 1987. 3

[5] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004. 1

[6] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *IJCV*, 43(1):7–27, June 2001. 1

[7] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607 – 609, June 1996. 1

[8] A. Opelt, A. Pinz, and A. Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. *IJCV*, 2008. 1

[9] S. D. Pietra, V. D. Pietra, and J. Lafferty. Inducing features of random fields. *TPAMI*, 1997. 3

[10] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Object recognition with cortex-like mechanisms. *TPAMI*, 29:411–426, 2007. 2

[11] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *ICCV*, October 2007. 1

[12] Y. N. Wu, Z. Si, C. Fleming, and S.-C. Zhu. Deformable template as active basis. In *ICCV*, October 2007. 1, 5

[13] Z. Yao, X. Yang, and S.-C. Zhu. Introduction to a large scale general purpose groundtruth dataset: methodology, annotation tool, and benchmarks. In *Proc. 6th Int'l Conf on Energy Minimization Methods in CVPR (EMMCVPR)*, Aug 2007. 8

[14] H. Zhang, A. C. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006. 1