Learning a Dictionary of Shape Epitomes with Applications to Image Labeling

Liang-Chieh Chen[1], George Papandreou[2], and Alan L. Yuille[1,2]
Departments of Computer Science[1] and Statistics[2], UCLA
lcchen@cs.ucla.edu, {gpapan, yuille}@stat.ucla.edu

Abstract

The first main contribution of this paper is a novel method for representing images based on a dictionary of shape epitomes. *These shape epitomes represent the local edge structure of the image and include hidden variables to encode shift and rotations. They are learnt in an unsupervised manner from groundtruth edges. This dictionary is compact but is also able to capture the typical shapes of edges in natural images. In this paper, we illustrate the shape epitomes by applying them to the image labeling task. In other work, described in the supplementary material, we apply them to edge detection and image modeling.*

We apply shape epitomes to image labeling by using Conditional Random Field (CRF) Models. They are alternatives to the superpixel or pixel representations used in most CRFs. In our approach, the shape of an image patch is encoded by a shape epitome from the dictionary. Unlike the superpixel representation, our method avoids making early decisions which cannot be reversed. Our resulting hierarchical CRFs efficiently capture both local and global class co-occurrence properties. We demonstrate its quantitative and qualitative properties of our approach with image labeling experiments on two standard datasets: MSRC-21 and Stanford Background.

1. Introduction

In this paper, we propose a novel representation for local edge structure based on a dictionary of shape epitomes, which were inspired by [12]. This dictionary is learnt from annotated edges and captures the mid-level shape structures. By explicitly encoding shift and rotation invariance into the epitomes, we are able to accurately capture object shapes using a compact dictionary of only five shape epitomes. In this paper, we explore the potential of shape epitomes by applying them to the task of image labeling. Most modern image labeling systems are based on Conditional Random Fields (CRFs) [18, 20] for integrating local cues with neighborhood constraints. Image segments are typically represented in the pixel domain [9, 17, 26], or in the domain of superpixels (a region of pixels with uniform statistics) [6, 7, 10, 11, 21, 23].

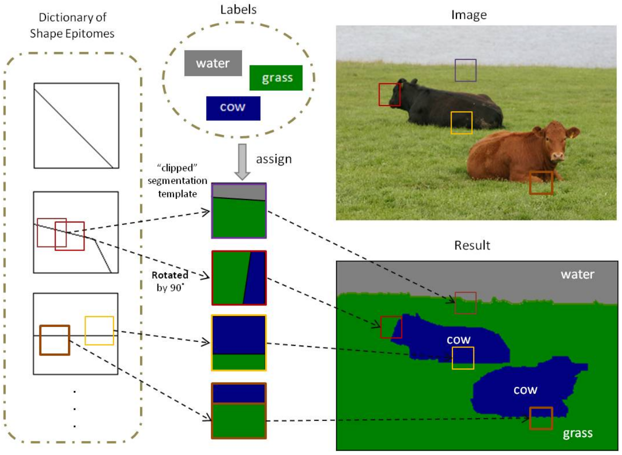


Figure 1. Proposed dictionary of *Shape Epitomes* in the context of image labeling. Segmentation templates are generated from the shape epitomes, by specifying the values of the hidden variables. Image labels are assigned to the regions within the templates, and thus the local relationship between object classes is explicitly modeled. Note the rotation and shift-invariance illustrated in the second and third shape epitome, respectively.

One motivation for shape epitomes was the success of segmentation templates for image labeling [27]. These templates also represent the local edge structure but differ from pixels and superpixels because they represent typical edges structures, such as L-junctions, and hence provide a prior model for edge structures. Each patch in the image was encoded by a particular segmentation template with semantic labels assigned to the regions specified by the template, as illustrated in Fig. 1. Segmentation templates, like superpixels, have computational advantages over pixel-based approaches by constraining the search process and also allow enforcing label consistency over large regions. Compared to superpixels, segmentation templates do not make early decisions based on unsupervised over-segmentation and, more importantly, explicitly enumerate the possible spatial configurations of labels making it easier to capture local relations between object classes. See Table 1 for a comparison

	Pixel	Superpixel	Template
Computation	−	+	+
Flexibility	+	+	+
Long-Range	−	+	+
Explicit Configuration	−	−	+

Table 1. General comparison between representations from the aspects of *Computation*, *Flexibility* (better align with object shapes), *Long-Range* consistency, and ability to *Explicitly* model the local *configuration* of objects. We improve the flexibility of template-based representation by learning a dictionary of shape epitomes.

summary.

But those segmentation-templates [27] have limitations. Firstly, they were hand-specified. Secondly, there were not invariant to shift and rotation which implies that a very large number of them would be needed to give an accurate representation of edge structures in the images (Zhu *et al* [27] used only thirty segmentation-templates which meant that they could only represent the edges very roughly).

Each shape epitome can be thought of a set of segmentation-templates which are indexed by hidden variable corresponding to shift and rotation. More precisely, a shape epitome consists of two square regions one inside the other. The hidden variable allows the inner square region to shift and rotate within the the bigger square, as shown in Fig. 1. The hidden variable specifies the shift and rotation. In the current paper, each shape epitome corresponds to $81 \times 4 = 324$ segmentation-templates. Hence, as we will show, a small dictionary of shape epitomes is able to accurately represent the edge structures (see Sec. 4.3.1). Intuitively the learned dictionary captures generic mid-level shape-structures, hence making it transferable across datasets. By explicitly encoding shift and rotation invariance, our learned epitomic dictionary is compact and only uses five shape epitomes. We also show that shape epitomes can be generalized to allow the inner square to expand which allow the representation to deal with scale (see Sec. 4.3.4).

We propose shape epitomes as a general purpose representation for edge structures (i.e. a mid-level image description). In this paper we illustrate them by applying them to the image labeling task. In the supplementary material we show how they can be also used for edge detection and for local appearance modeling. For image labeling, we consider three increasingly more complex models, which adapt current CRF techniques for shape epitomes. We use patches at a single fine resolution whose shape is encoded by a segmentation template (i.e. a shape epitome with hidden variable specified). The patches are overlapping, thus allowing neighbors to directly communicate with each other and find configurations which are consistent in their area of overlap (Model-1). We explore two enhancements of this basic model: Adding global nodes to enforce image-level consistency (Model-2) and also further adding an auxiliary node to encourage sparsity among active global nodes, i.e., encourage that only few object classes occur within an image (Model-3). We conduct experiments on two standard datasets, MSRC-21 and Stanford Background, obtaining promising results.

Related work. Our model is based on the success of several works. First, the ability to generate an image from a condensed epitomic representation [12]. We leverage on this idea to learn a dictionary of shape epitomes. Each segmentation template is generated within a *shape epitome*. This encodes the shift-invariance into the dictionary, since a segmentation template is able to move within a shape epitome. Besides, we encode rotation invariance by allowing the shape epitome to rotate by 0, 90, 180, and 270 degrees.

Second, the potential of using template-based representation and overlapped patches. It has been shown that learning the generic patterns capturing statistics over large neighborhoods can be beneficial for image denoising [24] and image labeling [15]. Besides, finding the mutual consensus between neighboring nodes by using overlapped patches [28] has shown to be effective. Similar ideas have been applied to image labeling [16]. However, they did not learn a dictionary for object shapes.

Third, the power of introducing simple global nodes for image labeling. Ladicky et al. [19] introduced global nodes that can take values from the predefined label set L and a "free" label. There is no energy cost, when the global node takes the free label. Gonfaus et al. [8] proposed a *harmony* model to generalize the idea by allowing the global node to take labels from the power set of L. However, it is computationally challenging to find the most probable state for the global node from the power set. Then, Lucchi et al. [22] proposed the Class Independent Model (CIM) to decompose the global node into $|L|$ global nodes. Our model moves further based on the CIM by encoding the image-level co-occurrence, and adding an auxiliary node to encourage the sparsity of active global nodes, similar to [4].

Structure. Sec. 2 describes our method of learning a dictionary of shape epitomes, and Sec. 3 describes our model based on CRF. Experimental results are discussed in Sec. 4.

2. Learning a dictionary of shape epitomes

In this section, we present our algorithm for learning the dictionary of shape epitomes from annotated images.

To learn the generic dictionary, we use the BSDS500 dataset [1], which provides ground truth of object boundaries. Given that, we extract $M \times M$ patches around the shape boundaries (called shape patches). We cluster these shape patches using affinity propagation [5] to build our shape epitomes (note that the size of shape patches is the same as that of shape epitomes). The segmentation templates are of smaller size $m \times m$ ($m < M$) than the shape epitomes, and are generated as sub-windows of them. By generating the segmentation template from a larger shape epitome, we are able to explicitly encode shift-invariance into the dictionary, as illustrated in Fig. 1. Therefore, one shape epitome compactly groups many segmentation templates which are shifted versions of each other.

Clustering by affinity propagation requires a similarity measure $F(P_1, P_2)$ between two $M \times M$ shape patches P_1 and P_2. We induce $F(P_1, P_2)$ from another similarity measure $F_T(T_1, T_2)$ between two $m \times m$ segmentation templates T_1 and T_2 extracted from P_1 and P_2, respectively. Specifically, let $T(i,j)$ denote the segmentation template extracted from P and centered at (i,j), with $(0,0)$ being the center of P. We define the similarity between the two shape patches P_1 and P_2 to be

$$F(P_1,P_2) = \max_{\frac{m-M}{2} \leq i,j \leq \frac{M-m}{2}} \frac{1}{2}[F_T(T_1(i,j),T_2(0,0)) + F_T(T_1(0,0),T_2(-i,-j))] , \quad (1)$$

as illustrated in Fig. 2. We employ the *covering* of the template T_1 by the template T_2 [1] as the similarity measure $F_T(T_1, T_2)$ between them:

$$F_T(T_1,T_2) = \frac{1}{|T_2|} \sum_{r_2 \in T_2} |r_2| \max_{r_1 \in T_1} \frac{|r_1 \cap r_2|}{|r_1 \cup r_2|} ,$$

where r_1 and r_2 are the regions in templates T_1 and T_2, respectively, and $|r|$ is the area of region r. Note that $F_T(T_1, T_2)$ and consequently $F(P_1, P_2)$ range from 0 (no similarity) to 1 (full similarity).

Directly applying affinity propagation results in many similar shape epitomes because simple horizontal or vertical boundaries are over-represented in the training set. We follow [13] and grow the dictionary incrementally, ensuring that each newly added shape epitome is separated from previous ones by at least distance t, as follows:

1. *Clustering.* Apply affinity propagation to find one shape epitome that contains the most members (i.e., the largest cluster) in current training set.
2. *Assigning.* For each shape patch in training set, assign it to the shape epitome found in step 1, if their distance, defined as $1 - F(P_1, P_2)$, is smaller than t.
3. *Update.* Remove the shape patches that are assigned to the shape epitome from the current training set.
4. Repeat until no shape patch is left in the training set.

3. Adapting CRFs for segmentation templates

Having learned the dictionary of shape epitomes, we now proceed to show how we can build models for image labeling on top of it. We propose three models by adapting current CRF models to the template-based representation.

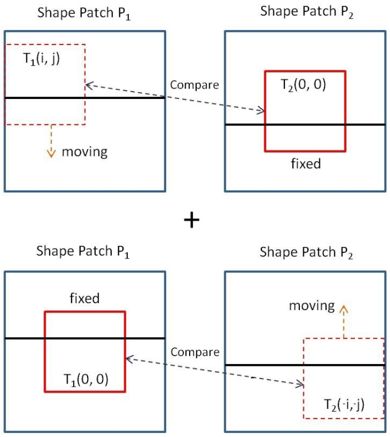


Figure 2. The similarity measure between two shape patches. The optimal value of shift variables (i, j) is shown for this example.

The problem of image labeling in this context can be formulated as follows. Given an image I, we represent it by a set of overlapped $m \times m$ patches. The goal is to encode each patch by a segmentation template, and by assigning labels (from a categorical set L) to each region in the segmentation template. Specifically, the labeling assignment $\boldsymbol{x}$ is represented by both segmentation template and labels. That is, $\boldsymbol{x} = \{x_i\}_{i \in \mathcal{V}}$ with $x_i = \{s_i, \boldsymbol{l}_i\}$, where $\mathcal{V}$ is the set of patches, s_i and $\boldsymbol{l}_i$ denote the type of segmentation template and object labeling, respectively. Note that $\boldsymbol{l}_i$ is a vector, whose length is the number of regions within the segmentation template. For example, $\boldsymbol{l}_i = (cow, grass)$ means that label cow and label grass are assigned to the first region and second region within segmentation template s_i. We call our models SeCRF, short for Shape epitome CRF.

3.1. Model 1: One-level SeCRF

We first introduce a flat model, which is represented by a graph with a single layer $\mathcal{G} = \{\mathcal{V}_l, \mathcal{E}_l\}$, as shown in Fig. 3(a). Each node corresponds to a patch region, and it is encoded by both the type of segmentation template and the labels assigned to it. The image region represented by node i (i.e., i-th patch) is denoted by $R(i)$.

The energy of $\boldsymbol{x}$ given image I is given by:

$$E(\boldsymbol{x}|I) = E_1(\boldsymbol{x};\alpha_1) + E_2(\boldsymbol{x};\alpha_2) + E_3(\boldsymbol{s};\alpha_3) + E_4(\boldsymbol{l};\boldsymbol{\alpha_4}) + E_5(\boldsymbol{x};\boldsymbol{\alpha_5}) \quad (2)$$

where $\boldsymbol{\alpha}$ is the model parameters. Note we suppress the dependency on the image I in subsequent equations. Each energy term is defined below.

The first term $E_1(\boldsymbol{x}; \alpha_1)$ is the data term which accumulates the pixel features with respect to certain type of segmentation template and labels assigned to the correspond-

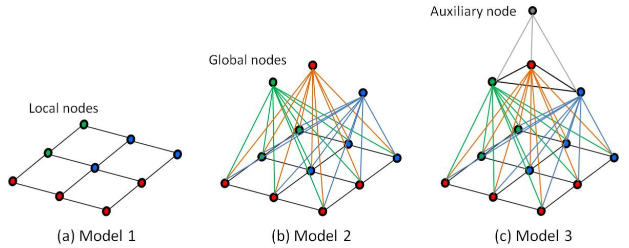


Figure 3. Adapting CRFs for segmentation templates. (a) Model 1 uses only a single layer of local nodes. (b) Model 2 adds global nodes to encode global consistency, similar to [22] (but the energy value is soft in our model). (c) Model 3 encodes the pairwise co-occurrence between global nodes, and adds an auxiliary node to encourage the sparsity of active global nodes.

ing regions. We set $E_1(\boldsymbol{x};\alpha_1) = -\alpha_1 \sum_{i\in\mathcal{V}_l} \psi_1(x_i)$, and

$$\psi_1(x_i) = \frac{1}{|R(i)|} \sum_{p\in R(i)} \log Pr(x_i^p|I)$$

where we define x_i^p as the labeling of pixel p in the region of segmentation template s_i. The value $Pr(x_i^p|I)$ is computed by a strong classifier with features (e.g., filter bank responses) extracted within a region centered at position p.

The second term is used to encourage the consistency between neighboring nodes in their area of overlap. For a pixel that is covered by both node i and j, we encourage node i to assign the same label to it as node j. The consistency is defined by using the Hamming distance:

$$E_2(\boldsymbol{x};\alpha_2) = -\alpha_2 \sum_{(i,j)\in\mathcal{E}_l} \psi_2(x_i,x_j)$$

where

$$\psi_2(x_i,x_j) = \frac{1}{|O(i,j)|} \sum_{p\in O(i,j)} \delta(x_i^p = x_j^p)$$

where $O(i,j)$ is the overlapped region between nodes i and j, and $\delta(x_i^p = x_j^p) = 1$ if $x_i^p = x_j^p$, and zero, otherwise. In our experiments, we use 4-neighborhood.

The third term encodes the generic prior of segmentation templates. Specifically, we binarize the type of s_i to be either 1, meaning that it contains some type of shapes, or 0, meaning that it contains no shape.

$$E_3(\boldsymbol{s};\alpha_3) = -\alpha_3 \sum_{i\in\mathcal{V}_l} \log Pr(s_i)$$

The fourth term $E_4(\boldsymbol{x};\boldsymbol{\alpha}_4)$ is used to model the co-occurrence of two object classes within a segmentation template. Note that parameter $\boldsymbol{\alpha}_4$ is a 2-D matrix, indexed by u and v, ranging over the label set L.

$$E_4(\boldsymbol{l};\boldsymbol{\alpha}_4) = -\sum_{i\in\mathcal{V}_l} \sum_{u,v=1,\ldots,|L|} \alpha_4(u,v)\psi_4(u,v,\boldsymbol{l}_i)$$

where $|L|$ is the total number of object classes, and $\psi_4(u,v,\boldsymbol{l}_i)$ is an indicator function which equals one when both object classes u and v belong to $\boldsymbol{l}_i$.

The fifth term $E_5(\boldsymbol{x};\boldsymbol{\alpha}_5)$ models the spatial relationship between two classes within a segmentation template. We model only the "above" relationship. For example, we encourage sky to appear above road, but not vice versa.

$$E_5(\boldsymbol{x};\boldsymbol{\alpha}_5) = -\sum_{i\in\mathcal{V}_l} \sum_{u,v=1,\ldots,|L|} \alpha_5(u,v)\psi_5(u,v,x_i)$$

where $\psi_5(u,v,x_i)$ is an indicator function which equals one when object class m is above class n within a certain segmentation template. Note that for some segmentation template that does not have the "above" relationship (e.g., a template with vertical boundary), this term is not used.

3.2. Model 2: Two-level SeCRF

Motivated by the Class Independent Model (CIM) in [22], we add $|L|$ independent global nodes $\{\mathcal{V}_g\}$ to enforce image-level consistency, as shown in Fig. 3(b). A global node encodes the absence or presence of a object class in the image (i.e., $y_i \in \{0,1\}, \forall i \in \mathcal{V}_g$), and it is densely connected to every local node. We denote the set of edges connecting global nodes and local nodes as $\{\mathcal{E}_{lg}\}$, and then labeling assignment $\boldsymbol{x} = \{\{x_i\}_{i\in\mathcal{V}_l} \cup \{y_i\}_{i\in\mathcal{V}_g}\}$. An extra global-local energy term is added to Equation 2 with each global node y_j having a 2-D matrix parameter $\boldsymbol{\alpha}_6^j$:

$$E_6(\boldsymbol{x};\boldsymbol{\alpha}_6) = -\sum_{(i,j)\in\mathcal{E}_{lg}} \sum_{u=1}^{|L|}\sum_{v=0}^{1} \alpha_6^j(u,v)\psi_6(u,v,x_i,y_j)$$

where

$$\psi_6(u,v,x_i,y_j) = \begin{cases} \frac{1}{|R(i)|} \sum_{p\in R(i)} \delta(x_i^p = u), & \text{if } y_j = v \\ 0, & \text{otherwise} \end{cases}$$

Note that our Model 2 differs from CIM in two parts. First, the value of function ψ_6 is proportional to the number of pixels whose labels are u in the node x_i. This formulation is different from the energy cost used in the original CIM, which is either zero or one (i.e., a hard value). On the contrary, we formulate this energy cost as a soft value between zero and one. Second, our local nodes are based on overlapped segmentation templates (not superpixels) so that neighbors can directly communicate with each other. Furthermore, unlike the robust P^n model [14], our penalty depends on the region area within a segmentation template, and thus it is a function of the segmentation template type.

3.3. Model 3: Three-level SeCRF

We further refine Model 2 by adding image-level classification scores to the unary term of global nodes [25]. Specifically, we train $|L|$ SVM classifiers to predict the presence or absence of object classes, following the pipeline of [2]. The unary energy for global nodes is then defined as follows.

$$E_7(\boldsymbol{y};\alpha_7) = -\alpha_7 \sum_{i\in\mathcal{V}_g} C(y_i|I)$$

where $C(y_i|I)$ is the output of i-th classifier.

The independency among global nodes in Model 2 ignores the co-occurrence between object classes in the image level. Hence, we add edges $\{\mathcal{E}_g\}$ to connect every pair of global nodes, and define an energy term on them:

$$E_8(\boldsymbol{y};\boldsymbol{\alpha}_8) = -\sum_{(i,j)=e\in\mathcal{E}_g}\sum_{u,v=0}^{1} \alpha_8^e(u,v)\delta(y_i=u, y_j=v)$$

where $\boldsymbol{\alpha}_8^e$ depends on the specific edge $e=\{i,j\}$ that connects two different global nodes, y_i and y_j.

As shown in Fig. 3(c), we also add an auxiliary node $\mathcal{V}_a$ (then, $\boldsymbol{x}=\{\{x_i\}_{i\in\mathcal{V}_l}\cup\{y_i\}_{i\in\mathcal{V}_g}\cup\{z_i\}_{i=\mathcal{V}_a}\}$). This node favors sparsity among global nodes (similar to [4]) by introducing a set of edges $\{\mathcal{E}_{ga}\}$ from $\{\mathcal{V}_g\}$ to $\mathcal{V}_a$. Specifically, $\mathcal{V}_a$ is a dummy node, which can take only one meaningless state. We define an energy term on $\{\mathcal{E}_{ga}\}$ to encourage only few global nodes to be active as follows.

$$E_9(\boldsymbol{y}, z_j;\alpha_9) = -\alpha_9 \sum_{(i,j)\in\mathcal{E}_{ga}} \delta(y_i=0)$$

where $\delta(y_i=0)$ equals one when the global node y_i is off. This energy term has the effect of biasing the global nodes.

4. Experiments

In this section, we first show the results of learning a dictionary of shape epitomes following the methods described in Sec. 2. We then use this dictionary for image labeling using the SeCRF models of Sec. 3.

4.1. Learned dictionary of shape epitomes

We learn the dictionary of shape epitomes from shape patches extracted from the BSDS500 dataset [1]. In the experiment, we fix the size of a shape patch to be 25×25, and the size of segmentation template 17×17, namely $M=25$ and $m=17$. After applying affinity propagation incrementally with distance $t=0.05$, the first 10 shape epitomes are shown at the top of Fig. 4.

For computational and modeling purposes it is desirable to have a compact dictionary consisting of only few shape epitomes. We have found that the first 5 shape epitomes contain most of the shape patches in the training set of BSDS500, and the cluster size decreases very quickly.

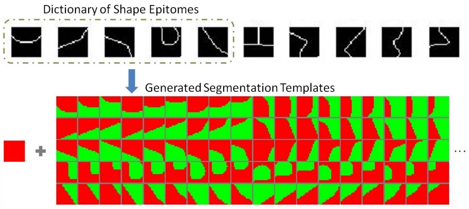

Dictionary of Shape Epitomes
Generated Segmentation Templates
+
...

Figure 4. Top row: first 10 shape epitomes learned by our method. Bottom: a flat segmentation template (i.e., no shape) and some others generated from the first 5 shape epitomes. Note that some of them are generated from the rotated shape epitomes.

In our setting, a segmentation template is allowed to move within a shape epitome for each horizontal and vertical displacement up to ±4 pixels. We define *stride* as the step-size for horizontal/vertical displacement. For example, if stride $=4$, we can generate 9 templates from each shape epitome, only considering the nine templates $T(i,j)\ \forall i,j\in\{-4,0,4\}$ at all four possible orientations (0, 90, 180 and 270 degrees), ending up with $45=9\times5$ templates per epitome. In total, there are $181(5\times45+1)$ segmentation templates, including the flat one that contains no shape. On the other hand, if stride $=1$, we use every template within a shape epitome, resulting in 1621 $(81\times5\times4+1)$ segmentation templates.

Using this compact dictionary of 5 shape epitomes suffices to accurately encode the ground truth segmentations in our datasets, as demonstrated in Sec. 4.3.1. As one can observe in Fig. 4, our generated segmentation templates cover the common boundary shapes, such as vertical/horizontal edges, L-junctions, and U-shapes. The learned dictionary thus captures generic mid-level shape-structures and can be used across datasets. We emphasize that we learn it on the BSDS500 dataset and use it unadapted for image labeling on MSRC-21 and Stanford Background datasets.

4.2. Implementation details for image labeling

MAP Inference. We use loopy belief propagation (LBP) to minimize the energy function in Equation 2. We prune the unpromising states by rejecting the unlikely proposals whose E_1 data terms are too high, similar to [27]. We fix the number of states per node to be 100, since in our experiments adding more states only improve the performance marginally at the sacrifice of computation time.

Learning the parameters. We use the same structure-perceptron algorithm [3] as HIM [27], because we would like to have a direct comparison with it by emphasizing on the representation part of our model, not learning.

Fusion of predicted labels. The traditional Conditional Random Field models directly assign an object class label to each pixel in the image. On the contrary, our model uses overlapped patches, and each patch is encoded by a segmentation template and by labels assigned to the regions in the template. The number of patches that will cover the same pixel depends on the size of overlap between patches. We set the overlap size to be $(m-1)/2$ pixels in all experiments. To find the labels for every pixel, we fuse the predicted labels for each pixel by letting the patch having the minimal unary energy ($E_1 + E_3 + E_4 + E_5$) determine the final result of the covered pixel, since the pairwise term E_2 already encourages consistency.

4.3. Results

For image labeling, we experiment on two datasets: (1) The MSRC-21 with 591 images and $|L| = 21$ classes, using the original splitting (45% for training, 10% for validation, and 45% for testing) from [26]. (2) The Stanford Background dataset [10] consisting of 715 images and $|L| = 8$ classes, which we randomly partition into training set (572 images) and test set (143 images). Note in all the experiments, we fix $M = 25$, and $m = 17$ except in Sec. 4.3.4.

4.3.1. Encoding the ground truth

The ground truth provided by the datasets contains the true labeling for each pixel, not the true states of segmentation template type with regions labeled. This experiment is designed to see if our learned dictionary of shape epitomes can accurately encode the ground truth. We estimate the true states of the local nodes by selecting the pairs of segmentation template type and labeling (i.e., find true $x_i = (s_i, \boldsymbol{l}_i)$) that have maximum overlap with the true pixel labels. For MSRC-21 dataset, our result shows that this encoding of ground truth results in 0.27% error in labeling image pixels, while HIM [27] reported 2% error. This shows that our learned dictionary of shape epitomes is flexible enough to more accurately encode the MSRC ground truth than the hand-crafted dictionary of [27].

Here, we show the advantage of using our learned dictionary of shape epitomes over *directly* learning a dictionary of segmentation templates (in the latter case, the training shape patches have size $m \times m$ instead of $M \times M$) by conducting experiments on the Stanford Background dataset, which provides more detailed object boundaries. We propose to compare those two dictionaries in terms of the error of encoding the ground truth, when given the same covered areas, which is equivalent to learning the same number of parameters. Suppose the size of the dictionary of shape epitomes is K_E, and the size of the dictionary of segmentation templates is K_T. Given K_E, to cover the same areas, we select $K_T = 25^2/17^2 K_E$. As shown in Fig. 5, our learned dictionary of shape epitomes attains better performance than the dictionary of segmentation templates when given the same number of parameters.

4.3.2. Image labeling: MSRC-21 dataset

We generate 9 segmentation templates from each of the 5 shape epitomes in the labeling experiments (i.e., 181 templates totally). In a first set of experiments we directly compare our models with HIM [27]. We use the same boosting-based data term as HIM, provided by the authors, the main difference between HIM and our model lying in the representation part. As shown in Fig.7, our learned dictionary encodes the object shapes better than the hand-crafted dictionary used by HIM. Furthermore, both our Model 2 and Model 3 attain better performance than HIM (see Table 3).

We also compare our model with the recent method of [9] which incorporates powerful non-local patch similarity. We have used the same boosting-based data term as [9], as implemented in the Darwin software library[1]. As shown in Table 3, our Model 3 attains similar performance to [9], although we do not use non-local cues at the patch level.

4.3.3. Image labeling: Stanford background dataset

In this experiment, we use the data term provided by the Darwin software library. The results for the Stanford Background dataset are shown in Fig. 8. We achieve comparable results with other state-of-the-art models. Specifically, our segmentation template-based Model 3 performs better than the more complicated model of [10], which builds on a dynamic superpixel representation and incorporates both semantic and geometric constraints in a slow iterative inference procedure. We also perform better than the hierarchical semantic region labeling method of [23]. Our models perform somewhat worse than the long-range model of [9] (unlike the MSRC case), and the segmentation tree model of [21], which however employs different image features.

4.3.4. Scaling the segmentation templates

Here, we show that our learned dictionary can generate different sizes of segmentation templates, while attaining good performance on the Stanford Background dataset. Specifically, we explore the effect of varying the size of generated segmentation templates as the dictionary of shape epitomes is fixed. First, we explore the effect by encoding the ground truth. The size varies from $m = \{13, 17, 21, 25\}$. The stride variable is also changed to generate different number of segmentation templates from the dictionary. As shown in Fig. 6, the error is consistently decreased when m or stride is smaller. Second, we extract spatially equally 9 segmentation templates from the dictionary for different m (all resulting in 181 templates), and apply our Model 1 based on these templates to label the test images, as shown in Table 2. These results show that our proposed representation: shape epitomes is also able to handle scale effects without relearning the dictionary.

5. Conclusion

In this paper, we introduced shape epitomes and showed that they could efficiently encode the edge structures in the MSRC and Stanford Background datasets. This efficient encoding is due to their ability to represent local shifts and

[1] http://drwn.anu.edu.au, version 1.2

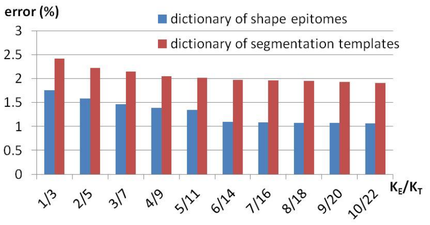


Figure 5. Error (%) of encoding ground truth of Stanford Background dataset, when using a dictionary of K_E shape epitomes or a dictionary of K_T segmentation templates.

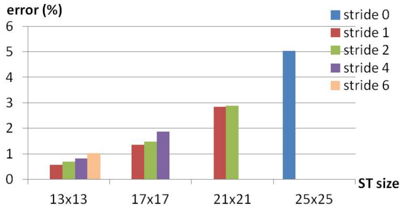


Figure 6. Error (%) of encoding ground truth of Stanford Background dataset. The dictionary of shape epitomes is fixed. The size of generated templates is different, and so is the stride.

Template size	13 × 13	17 × 17	21 × 21
Global	76.9	76.7	76.3

Table 2. Reuse the dictionary of shape epitomes with different size of generated templates on Stanford Background dataset.

rotations explicitly. The dictionary of shape epitomes were learnt from BSDS500 dataset. Next we explored the use of shape epitomes for CRF models of image labeling. The proposed SeCRF model can attain comparable results with other state-of-the-art models. Our supplementary material shows other applications of shape epitomes to edge detection and local appearance modeling.

Acknowledgements

This work was supported by ONR N00014-12-1-0883, ONR N000014-10-1-0933, and NIH Grant 5R01EY022247-03.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 2011.

[2] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. *BMVC*, 2011.

[3] M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. *ACL*, 2002.

[4] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. *International Journal of Computer Vision*, 2012.

[5] B. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 2007.

[6] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. *ICCV*, 2009.

[7] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. *CVPR*, 2008.

[8] J. Gonfaus, X. Boix, J. Van De Weijer, A. Bagdanov, J. Serrat, and J. Gonzalez. Harmony potentials for joint classification and segmentation. *CVPR*, 2010.

[9] S. Gould. Multiclass pixel labeling with non-local matching constraints. *CVPR*, 2012.

[10] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. *ICCV*, 2009.

[11] X. He, R. Zemel, and D. Ray. Learning and incorporating top-down cues in image segmentation. *ECCV*, 2006.

[12] N. Jojic, B. Frey, and A. Kannan. Epitomic analysis of appearance and shape. *ICCV*, 2003.

[13] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. *ICCV*, 2005.

[14] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 2009.

[15] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order mrfs. *CVPR*, 2009.

[16] P. Kontschieder, S. Buló, H. Bischof, and M. Pelillo. Structured class-labels in random forests for semantic image labelling. *ICCV*, 2011.

[17] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with gaussian edge potentials. *NIPS*, 2011.

[18] S. Kumar and M. Hebert. Discriminative random fields. *IJCV*, 2006.

[19] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Associative hierarchical CRFs for object class image segmentation. *ICCV*, 2009.

[20] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML*, 2001.

[21] V. Lempitsky, A. Vedaldi, and A. Zisserman. A pylon model for semantic segmentation. *NIPS*, 2011.

[22] A. Lucchi, Y. Li, X. Boix, K. Smith, and P. Fua. Are spatial and global constraints really necessary for segmentation? *ICCV*, 2011.

[23] D. Munoz, J. Bagnell, and M. Hebert. Stacked hierarchical labeling. *ECCV*, 2010.

[24] S. Roth and M. Black. Fields of experts. *IJCV*, 2009.

[25] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. *CVPR*, 2008.

[26] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 2009.

[27] L. Zhu, Y. Chen, Y. Lin, C. Lin, and A. Yuille. Recursive segmentation and recognition templates for image parsing. *PAMI*, 2012.

[28] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. *ICCV*, 2011.

		building	grass	tree	cow	sheep	sky	airplane	water	face	car	bicycle	flower	sign	bird	book	chair	road	cat	dog	body	boat	**Average**	**Global**
Exp1	Pxl-Cls	59.4	95.8	85.2	73.8	74.4	91.6	80.3	66.3	82.9	63.3	84.5	59.3	49.7	40.4	82.4	65.3	73.8	61.3	37.8	65.5	17.6	67.2	75.9
use same data	Model 1	62.7	96.2	87.9	78.7	78.6	92.6	83.7	66.8	85.5	69.2	87.1	63.9	53.6	45.3	85.7	71.8	75.4	66.5	41.9	68.4	17.5	70.4	78.1
term as HIM	Model 2	66.2	97.7	88.9	88.0	85.7	91.9	82.8	72.6	85.7	80.1	89.8	66.6	64.0	54.1	90.4	74.1	78.9	60.3	53.8	71.3	15.3	74.2	81.4
	Model 3	69.1	97.7	88.5	86.5	84.0	91.6	82.7	70.7	85.6	80.4	90.3	68.5	62.5	67.6	90.7	73.0	79.0	73.3	50.9	69.8	13.1	75.0	81.7
	HIM [27]	66.5	96.2	87.9	82.3	83.3	91.4	80.7	65.7	89.0	79.0	91.9	78.5	69.9	44.5	92.6	80.3	78.2	77.6	41.2	71.9	13.1	74.1	81.2
Exp2	Pxl-Cls	55.9	95.9	82.0	77.1	71.1	90.3	72.4	69.1	79.7	54.3	78.7	62.2	42.5	38.8	64.3	58.0	84.4	59.4	39.8	64.4	27.8	65.2	74.5
use data term	Model 1	63.9	96.9	86.6	81.9	75.2	91.9	76.1	72.8	81.3	59.9	84.4	65.8	45.5	41.7	66.1	61.9	87.7	64.0	43.4	67.5	29.4	68.7	77.6
from Darwin	Model 2	71.6	98.3	91.9	90.1	76.1	94.5	68.3	78.3	82.2	57.3	84.7	74.0	44.7	35.8	73.6	55.4	88.7	67.0	44.9	61.8	23.6	69.7	80.4
software library	Model 3	73.0	98.1	92.3	91.4	78.4	94.4	70.6	77.2	82.5	60.2	86.2	73.4	48.4	35.3	76.8	60.3	89.0	68.0	44.6	63.1	22.2	70.7	81.1
	[9]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	71.1	81.0
some	[19]	80	96	86	74	87	99	74	87	86	87	82	97	95	30	86	31	95	51	69	66	9	75	86
state-of-the-art	[23]	63	93	88	84	65	89	69	78	74	81	84	80	51	55	84	80	69	47	59	71	24	71	78
models	[8]	60	78	77	91	68	88	87	76	73	77	93	97	73	57	95	81	76	81	46	56	46	75	77
	DPG [22]	65	87	87	84	75	93	94	78	83	72	93	86	70	50	93	80	86	78	28	58	27	76	80
	[17]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	78.3	86.0

Table 3. MSRC labeling results. Pixel-wise classification rates are provided for each category. **Global** accuracy refers to the pixel-wise classification rate averaged over the whole dataset, and **Average** accuracy refers to the mean of all object class classification rates. The Pxl-Cls model is the pixel-wise classifier, whose output is integrated in our models.

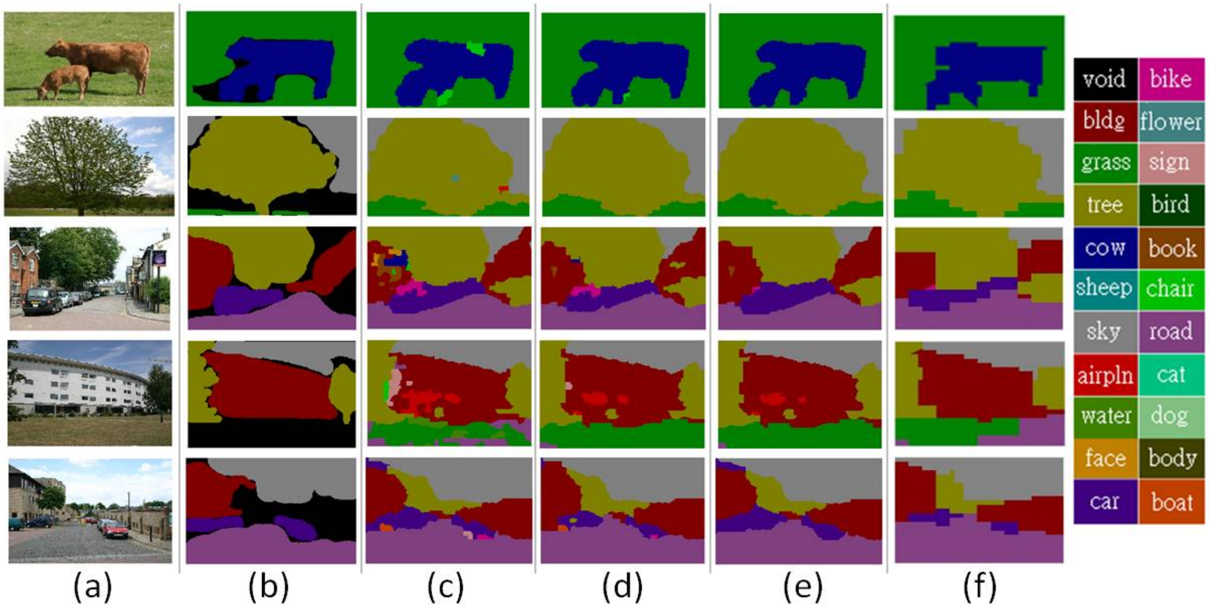
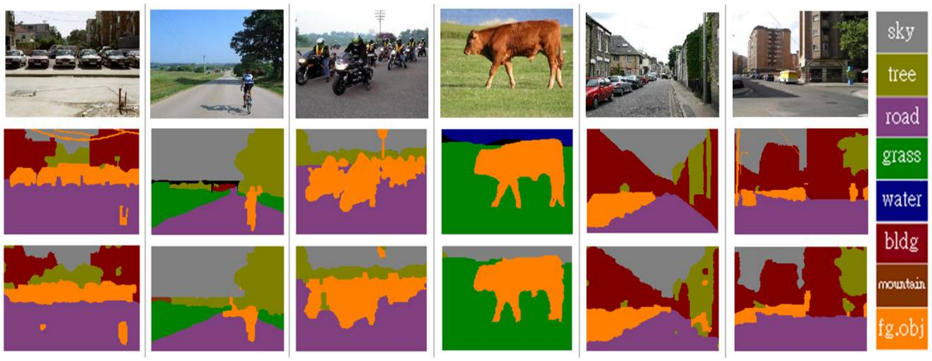


Figure 7. Qualitative results for the MSRC dataset. (a) Original image. (b) Ground truth. (c) Model 1. (d) Model 2. (e) Model 3. (f) HIM (excerpted from [27]). Note that our models capture object shapes more accurately than the HIM.



(a) (top) Original image. (middle) Ground truth. (bottom) Model 3. Note that our model is able to capture object shapes, especially the cow shape in the fourth column.

Method	**Global**
Pxl-Cls	73.9
Model 1	76.7
Model 2	77.2
Model 3	77.4
[10]	76.4
[23]	76.9
[9]	79.6
[21]	81.9

(b) **Global** accuracy is the pixel-wise classification rate averaged over dataset.

Figure 8. Qualitative and quantitative results on the Stanford Background dataset.