Learned Shape-Tailored Descriptors for Segmentation

Naeemullah Khan and Ganesh Sundaramoorthi
King Abdullah University of Science and Technology (KAUST), Saudi Arabia
{naeemullah.khan,ganesh.sundaramoorthi}@kaust.edu.sa

Abstract

We address the problem of texture segmentation by grouping dense pixel-wise descriptors. We introduce and construct learned *Shape-Tailored Descriptors that aggregate image statistics only within regions of interest to avoid mixing statistics of different textures, and that are invariant to complex nuisances (e.g., illumination, perspective and deformations). This is accomplished by training a neural network to discriminate base shape-tailored descriptors of oriented gradients at various scales. These descriptors are defined through partial differential equations to obtain data at various scales in arbitrarily shaped regions. We formulate and optimize a joint optimization problem in the segmentation and descriptors to discriminate these base descriptors using the learned metric, equivalent to grouping learned descriptors. Experiments on benchmark datasets show that the descriptors learned on a small dataset of segmented images generalize well to unseen textures in other datasets, showing the generic nature of these descriptors. We also show state-of-the-art results on texture segmentation benchmarks.*

1. Introduction

Segmentation of an image into textures is a fundamental problem in computer vision, and may play a key role in higher level tasks such as object segmentation, both in human and computer vision. Textures are composed of small tokens, called textons, that may vary by photometric (e.g., illumination) and geometric (e.g., perspective of the camera) nuisances, but are otherwise stationary within the texture. Thus, a natural approach to segment textures is to construct descriptors at each pixel that are invariant to variations of textons within textures and are discriminative of textons in different textures. Such descriptors can then be grouped to form the segmentation.

There are two difficulties with this descriptor grouping approach to segmentation. First, in order to construct invariant descriptors for segmentation, one needs to know the segmentation. This is because invariant descriptors aggregate image statistics from a neighborhood around a pixel, and to be descriptive of the texton within the texture, they must aggregate image statistics only within the texture to which the pixel belongs. Otherwise, statistics from different textured regions are mixed, and such pixels, usually near boundaries, become difficult to group. Second, provided the region in which to aggregate statistics is known, one needs to construct descriptors that are invariant to complex nuisances. The first problem has been addressed by [9]. There, segmentation is formulated as a joint problem of regions of the segmentation and dense invariant descriptors. Those descriptors, called *Shape-Tailored Descriptors*, are defined as solutions of partial differential equations (PDE) within regions of interest, and thus they only aggregate image statistics within regions of interest. The segmentation algorithm consists of an iterative process of updates of the descriptors based on the current segmentation, and updates of the segmentation based on the current descriptors.

In this work, we address the second problem, that of constructing descriptors *invariant* to complex nuisances yet discriminative of textures, while aggregating image statistics only within regions of interest. To do this, we use the *base* shape-tailored descriptors of [9], which are color channels and oriented gradients at various scales defined through PDEs, and learn a function of such base descriptors that is invariant to more complex nuisances than the limited invariances to small contrast and small geometric distortions that the base descriptors possess. By learning a function of base shape-tailored descriptors, we automatically inherit the shape-tailored property, i.e., that the learned descriptors aggregate image statistics only within regions of interest. Thus, they are naturally suited for a joint problem in the segmentation and the descriptors.

The problem we wish to address does not fit into a labeling problem, where one labels each pixel in the image according to pre-defined labels, representing certain categories in a training set. In particular, we do not wish to segment classes of textures (e.g., different types of tree barks or different types of sea shells should not be labeled the same). Since the set of textures in the natural world is enormous, perhaps not even enumerable, it is infeasible to construct a

training set with samples of each texture labeled. Further, we wish to segment textures that are not even in the training set. Instead of associating a class label to each texture, we aim to learn *generic* descriptors, beyond just textures or classes of textures in the training set, by learning a *metric* to discriminate textures by their base shape-tailored descriptors. By learning a metric to discriminate textures, the training set only needs to consist of ground truth segmentations of textures and not class labels. The aim is, by choice of appropriate regularization in the learning method, the metric and hence the descriptor generalizes beyond the training set and learns generic properties of all textures. The learned descriptor is the output of a fully connected neural network whose input is a base shape-tailored descriptor. The metric is formed from a Siemese network [6] composed of the aforementioned neural network, and a weighted $\mathbb{L}^2$ norm between the output of each component of the Siemese network.

Contributions: Our contributions are as follows: **1.** We show how to construct *learned* Shape-Tailored Descriptors, descriptors that aggregate image statistics only within arbitrary shaped regions of interest and are invariant to complex photometric and geometric nuisances yet discriminative for segmentation. The shape-tailored property is necessary so that segmentation by grouping descriptors can be accomplished, and the invariance is needed to segment textures plagued by nuisances. Invariance is accomplished by learning it from training data. **2.** We formulate grouping of Learned Shape-Tailored Descriptors as a joint optimization problem for the segmentation and descriptors, and derive the optimization algorithm. **3.** We test our method on texture segmentation benchmarks, and show state-of-the-art performance.

1.1. Related Work

Segmentation has a vast literature in computer vision, and we will only briefly discuss the most relevant literature. Exising approaches for texture segmentation can be roughly divided into learning based approaches and "hand-crafted" approaches. Further, hand-crafted approaches can be divided into edge-based and region-based approaches. Some region-based approaches attempt to partition the image into regions that have global intensity distributions that are maximally separated [26, 11, 20]. Since spatial relations are lost, other approaches have tried to incorporate spatial relations by considering distributions of pairs or neighborhoods of pixels (e.g., [8]). Larger neighborhoods are described, by for instance the output of Gabor filters at various scales and orientations [15], and grouped in other approaches for texture segmentation [25, 21, 7]. However, such approaches are affected by the problem that describing neighborhoods without knowing or having an estimation of the segmentation is prone to errors as neighborhoods that aggregate statistics across segmentation boundaries are difficult to group. This problem was addressed by [9], who formulated the estimation of descriptors and segmentation as a joint problem. However, the descriptors constructed in [9] were hand-crafted, and do not exhibit invariances to complex nuisances. In [10] instead of a few handcrafted scales of [9] a continuum of scales is considered.

Edge-based approaches (e.g., [1]) attempt to locate edges as a response to a filter bank. [1] use a filter bank of Gabor filters among other hand-crafted filters. Such responses are then post-processed to fill gaps, and generate a segmentation. Learning-based approaches to edge detection have been shown to achieve better results [24, 23, 12]. Such approaches have used deep learning to derive a probability that a pixel belongs to boundaries between segments. While these approaches achieve impressive results, still a difficulty remains in generating the segmentation from edges, which still rely on hand-crafted approaches [1] and the problem remains not fully solved. Alternatively, region-based approaches, like our method, solve directly for the regions, and avoid this problem. However, they have the problem of selecting the correct number of regions, which cannot be fully addressed without a hierarchy of segmentations. Our approach addresses one of the problems in region-based approaches, that of learning descriptors to group. Our approach is the first to address this problem to the best of our knowledge.

There has been recent interest in methods for semantic segmentation using deep learning [3, 4, 14]. These approaches aim to label each pixel in the image as semantically distinct objects from a pre-defined set of objects. Such approaches achieve impressive results. However, they are limited to object classes in the training set, and it would be difficult to apply this approach to our problem of texture segmentation, as the set of textures that we wish to segment is probably not even enumerable.

2. Learning Shape-Tailored Descriptors

In this section, we describe our approach to learning descriptors that are descriptive of neighborhoods around a pixel within a specific region of interest, while having invariance to complex photometric and geometric nuisances. We first review Shape-Tailored Descriptors [9], which are descriptors invariant to minor photometric and geometric nuisances, and are computed only from image information within a region of interest. We then describe how to use these "base" descriptors to learn such shape-tailored descriptors that are invariant to more complex nuisances, such as illumination change, shading, etc.

2.1. Base Shape-Tailored Descriptors

Let Ω be the domain of the image, $J_j : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^+$ for $j = 1, \ldots, N_c$ (N_c is the number of channels) be

channels of the image, these could be for instance, color and oriented gradients. Let $R \subset \Omega$ be a region of interest, which can have arbitrary shape. Shape-Tailored Descriptors are defined as the solution of Poisson-type partial differential equations (PDE):

$$\begin{cases} u_{ij}(x) - \alpha_i \Delta u_{ij}(x) = J_j(x) & x \in R \\ \nabla u_{ij}(x) \cdot N = 0 & x \in \partial R \end{cases}, \quad (1)$$

where ∇ is the gradient, Δ is the Laplacian, ∂R is the boundary of R, N is the unit outward normal to ∂R, $i = 1, \ldots, N_s$ and N_s is the number of scales, and $\alpha_i \in \mathbb{R}^+$ are the scales. The solution of the PDE can be shown to be the minimizer of the energy $E = \int_R (J_j(x) - u_{ij}(x))^2 \, \mathrm{d}x + \alpha_i \int_R |\nabla u_{ij}(x)|^2 \, \mathrm{d}x$. The solution is thus a balance between fidelity to the image and smoothness, with α_i larger implying more smoothness. We set $\mathbf{u} : R \to \mathbb{R}^{N_s N_c}$ as the vector of all components of scales and channels:

$$\mathbf{u}(x) = (u_{11}(x), \ldots, u_{1N_c}(x), \ldots, u_{N_s 1}(x), \ldots, u_{N_s N_c}(x))^T.$$

The u_{ij} are smoothed channels of the image and since the PDE is defined in a specific region R, no image information outside R is used to determine u_{ij}. This is important in region-based approaches to segmentation, as aggregating image information across segmentation boundaries mixes unrelated statistics and then such descriptors are difficult to group. Due to the smoothing, the descriptors exhibit invariance to small geometric transformations. However, they are not in general invariant to more complex geometric transformations or complex photometric transformations, such as illumination change. Therefore, in the next section, we use the descriptors above and learn more invariant descriptors. Since these learned descriptors are built from the descriptors above, they inherit the shape-tailored property.

2.2. Metric and Descriptor Learning

In this section, we learn a function, $f : \mathbb{R}^n \to \mathbb{R}^m$ where $n = N_s \times N_c$ and $m > 0$, from the space of base Shape-Tailored Descriptors to another vector space, with better invariance properties. In other words, f takes in $\mathbf{u}(x) \in \mathbb{R}^n$ at a particular pixel and returns a descriptor with m components. We choose f to be the output of a fully-connected neural network. Since we will eventually use the descriptor to discriminate between descriptors of different regions, we learn f by learning a Siemese neural network [6] designed to discriminate descriptors of different segmentation regions. Thus, the overall system is a symmetric function, $D : \mathbb{R}^n \times \mathbb{R}^n \to [0, 1]$, with a value of 1 indicating the descriptors are from different segmentation regions, and 0 indicating the descriptors are from the same region. The architecture is shown in Figure 1. The network takes in two different base descriptors $\mathbf{u}(x)$ and $\mathbf{v}(x)$, each are input to f and the output are two descriptors with m-components,

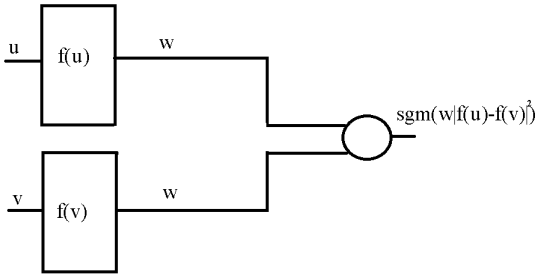

Figure 1. **Siamese network for metric learning.** u and v are descriptors, the training label is zero when u and v are descriptors at pixels belonging to same region and the label is one otherwise.

then a weighted $\mathbb{L}^2$ norm of the difference of the descriptors are computed, followed by a sigmoid function. Note that for descriptors at pixels x and y, the metric from the Siemese network is defined as

$$\begin{aligned} D(\mathbf{u}(x), \mathbf{v}(y))^2 &:= \|f(\mathbf{u}(x)) - f(\mathbf{v}(y))\|_w^2 \\ &= \sum_{i=1}^{m} w_i |f(\mathbf{u}(x))_i - f(\mathbf{v}(y))_i|^2, \quad w_i \geq 0 \end{aligned} \quad (2)$$

where w_i, $i = 1, \ldots, m$ are weights, and $f(\mathbf{u}(x))_i$ is the i^{th} component of $f(\mathbf{u}(x))$. Figure 2 shows a few components of STLD and the learned descriptors. We can observe that learned descriptors are more invariant to intrinsic and extrinsic nuisances of complex textures.

2.3. Training Data

The training data to train the network is generated from ground truth segmentations of images in the training set of images. Given a training image, we compute base Shape-Tailored Descriptors from the ground truth segmentation. For any pair of pixels x and y in adjacent ground truth regions or the same region in the same image, we form the training data as

$$D(\mathbf{u}_l(x), \mathbf{u}_k(y)) = \begin{cases} 0 & x, y \in R_l, l = k \\ 1 & x \in R_l, y \in R_k, l \neq k \end{cases},$$

where $\mathbf{u}_l(x)$ is the base Shape-Tailored Descriptor computed within R_l at x and $\mathbf{u}_k(x)$ is the base shape-tailored descriptor computed within R_k at y. Note we only choose adjacent regions since during segmentation, only discriminating between adjacent regions will be needed.

During segmentation at test time, we will solve a joint problem for the base descriptors $\mathbf{u}$ and the regions of the segmentation. The method iteratively updates the regions and the base descriptors. Thus, the metric D also needs to discriminate shape-tailored descriptors, when the descriptors are not computed on the ground truth segmentation. To this end, we perturb the ground truth segmentations by dilations and erosions to form regions $\tilde{R}_l$, and compute base

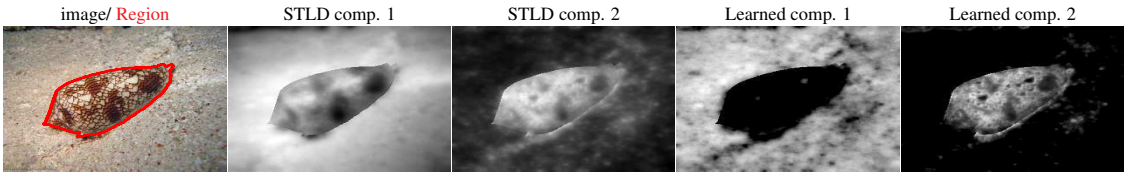

Figure 2. **Comparison of STLD and Learned Descriptors.** (Image/Region) shows the image, the red contour marks the boundary between regions. (STLD comp.1, STLD comp. 2) are two components of the base Shape-Tailored Local Descriptor, where statistics are aggregated in foreground and background separately. (Learned comp. 1, Learned comp. 2) are components of the learned descriptor. Notice that different components of the learned descriptors are active for differently textured regions, making segmentation easier and precise. Also, notice that the Learned comp. 1 is more invariant to illumination change in the background then STLD comp. 1 and 2

shape-tailored descriptors $\tilde{\mathbf{u}}$ within the perturbed regions. This simulates possible base-descriptors anticipated during test time. We augment the training data with these descriptors as follows:

$$D(\tilde{\mathbf{u}}_l(x), \tilde{\mathbf{u}}_k(y)) = \begin{cases} 0 & x, y \in R_l, l = k \\ 1 & x \in R_l, y \in R_k, l \neq k \end{cases},$$

where x and y are in the same or adjacent ground truth regions R_l and R_k. Note that the descriptors are computed in the perturbed regions, whereas the distance above is defined according to the ground-truth regions where pixels belong.

3. Segmentation

In this section, we describe our method for segmentation by using the invariant descriptors and the metric learned in the previous section.

3.1. Optimization Problem

We assume that the image consists of N_r regions with a constant learned shape-tailored descriptor in each region. We design an optimization problem for segmentation to be optimal when the regions are placed so that the learned shape-tailored descriptors are nearly constant within the regions. Let $\mathbf{u}^i(x) \in \mathbb{R}^n$ denote the base shape-tailored descriptor within region R_i, and let $\mathbf{a}^i \in \mathbb{R}^m$ be the constant learned shape-tailored descriptor representing the region, which is unknown. The energy for segmentation is as follows:

$$E(\{R_i\}_{i=1}^{N_r}) = \sum_{i=1}^{N_r} \int_{R_i} \|f(\mathbf{u}^i(x)) - \mathbf{a}^i\|_w^2 \, \mathrm{d}x + \beta \int_{\partial R_i} \mathrm{d}s, \tag{3}$$

where there are N_r regions, $\beta > 0$, and the second term above is to induce spatial regularity of the segmentation and consists of penalizing boundary length ($\mathrm{d}s$ is the arc-length element). The first term measures how similar the learned shape-tailored descriptor at each pixel within a region is to a constant vector $\mathbf{a}^i$. Thus, the optimal regions will be such that the regions have nearly constant learned descriptors within regions. This energy can be seen as a generalization of the energies considered by [17, 5].

3.2. Optimization Algorithm

If we minimize in $\mathbf{a}^i$, we see that the optimizer is $\mathbf{a}^i = 1/|R_i| \cdot \int_{R_i} f(\mathbf{u}^i(x)) \, \mathrm{d}x$ where $|R_i|$ denotes the area of R_i, i.e., the average value of the learned descriptor within the region. Since the energy above is non-convex in the regions, as the descriptor $\mathbf{u}^i$ depends on R_i non-linearly and f is non-convex, we use a gradient descent to optimize the energy. The gradient with respect to the boundary of R_i of the i^{th} term, using techniques from [9], is

$$(\|f(\mathbf{u}^i) - \mathbf{a}^i\|_w^2 + \kappa_i) N_i + (\mathrm{tr}[(D\mathbf{u}^i)^T D\hat{\mathbf{u}}^i] + (\mathbf{u}^i - \mathbf{J})^T A^{-1} \hat{\mathbf{u}}^i) N_i \tag{4}$$

where κ_i is the signed curvature of ∂R_i, N_i is the inward normal to ∂R_i, tr is the trace, D is the derivative, A is a diagonal matrix of size n with diagonal entries $(\alpha_1, \ldots, \alpha_1, \ldots, \alpha_{N_s}, \ldots, \alpha_{N_s})$, $\mathbf{J} = (J_1, \ldots, J_{N_c}, \ldots, J_1, \ldots, J_{N_c})^T$ is a vector of size n, and $\hat{\mathbf{u}}^i$ satisfies the PDE

$$\begin{cases} \hat{\mathbf{u}}^i(x) - A\Delta\hat{\mathbf{u}}^i(x) = 2\nabla f(\mathbf{u}^i(x))[f(\mathbf{u}^i(x)) - \mathbf{a}^i] & x \in R_i \\ \nabla\hat{\mathbf{u}}^i(x) \cdot N_i = 0 & x \in \partial R_i \end{cases}.$$

Note that the first term in (4) arises from the variation of the integrals as the boundary is deformed, and the second term arises from the variation of the descriptor as the boundary is changed. The gradient ∇f, which involves the neural network, can be approximated numerically. However, for simplicity of implementation, we neglect the variation of the descriptor since the numerical algorithm will involve only small changes of the boundary at each iteration and the descriptors $\mathbf{u}^i$ do not change much, and so the term is negligible.

To implement the gradient descent numerically, we represent the regions with relaxed indicator or "level-set" functions $\phi_i : \Omega \to [0, 1]$, $i = 1, \ldots, N_r$. R_j is the region where ϕ_j achieves the maximum over all $i = 1, \ldots, N_r$. We can then convert the boundary evolution into an evolution of ϕ_i analogous to level set methods [19]. In order to extend the evolution beyond just the boundary, we extend the terms in the gradient to a band around the boundary. Computing the full gradient of the energy and neglecting

variation of the descriptor terms, our algorithm to minimize the energy is given in Algorithm 1.

Algorithm 1 Grad. Descent of Learned STLD Energy

Input: An initialization of ϕ_i
repeat
  Set regions: $R_i = \{x \in \Omega : i = \text{argmax}_j \phi_j(x)\}$
  Compute dilations, $D(R_i)$, of R_i
  Compute $\mathbf{u}^i$ in $D(R_i)$, compute $\mathbf{a}^i = 1/|R_i| \cdot \int_{R_i} \mathbf{u}^i(x)\,\mathrm{d}x$.
  Compute band pixels $B_i = D(R_i) \cap D(\Omega \backslash R_i)$
  Compute $G_i = \|f(\mathbf{u}^i(x)) - \mathbf{a}^i\|_w^2$ for $x \in B_i$. f is evaluated from the neural network.
  Update pixels $x \in D(R_i) \cap D(R_j)$ as follows:

$$\phi_i^{\tau+\Delta\tau}(x) = \phi_i^{\tau}(x) - \Delta\tau(G_i(x) - G_j(x))|\nabla\phi_i^{\tau}(x)| + \Delta\tau \cdot \beta\kappa_i|\nabla\phi_i^{\tau}(x)|. \quad (5)$$

  Update all other pixels as

$$\phi_i^{\tau+\Delta\tau}(x) = \phi_i^{\tau}(x) + \Delta\tau \cdot \beta\kappa_i|\nabla\phi_i^{\tau}(x)|.$$

  Clip between 0 and 1: $\phi_i = \max\{0, \min\{1, \phi_i\}\}$.
until regions have converged

4. Experiments

Datasets: We use four different datasets to test our method. We use the Real World Texture Dataset and Brodatz Synthetic Dataset introduced in [9]. The first consists of 256 total real-world textured images with two dominant textures. 128 images are used for training and 128 for testing. The Brodatz Synthetic Dataset consists of 200 images of two textured regions of various shapes. We also use the Graz Segmentation dataset [22], which consists of 243 images of real-world textured objects with multiple objects per image. Finally, we use the Berkeley Segmentation Dataset [1], which consists of 200 training and test images, and 100 validation images, and various numbers of textured objects in each image. Each of the datasets exhibit complex nuisances, such as illumination, shading, perspective effects, etc.

Architecture Details: We use a Siamese twin network, where each component has two fully connected layers. We test the sensitivity to number of hidden layers and hidden units later. Our input base shape-tailored descriptor is a 40 dimensional descriptor (RGB channels, gray scale and four oriented gradients at 5 scales, $\alpha = (10, 20, 30, 40, 50)$). The output descriptor f of the Siamese network is same size as the number of hidden units used. The sigmoid of the (learned) weighted difference of the two twins is used to compute the metric D of a pair of descriptors.

Results on Real-World Texture Dataset: We use 128 images in the training set to train our network and test on the 128 images in the test set. This gives us 9153732 training pairs of descriptors. We initialize our method by a 5×5 standard block tessellation, with random labels (out of 1 or 2) chosen for each block. Quantitative results are in Table 1. We compare to hand-crafted Shape-Tailored Descriptors (STLD) [9], to non-STLD (the descriptors in (1) when $R = \Omega$ the whole image), learned non-STLD (non-STLD base descriptors used to learn invariant descriptors through the Siemese network), and other methods. non-STLD hand-crafted performs the worst, followed by learned non-STLD, then hand-crafted STLD performs better, and the learned STLD (our approach) performs the best. This shows that both properties of shape-tailored and learning invariance are necessary to achieve the best results. Figure 4 shows some visual comparisons of our approach to handcrafted STLD.

Robustness to Initialization, Training Data, and Architecture: First we test sensitivity to initialization. We vary the box tessellation from 3×3 to 5×5. Results are in Table 2. They show that the method is robust to initialization. Now we test the sensitivity to the architecture in our approach and the learned non-STLD approach. To this end, we vary the architecture of our network by changing the number of hidden units. Results are shown in Table 3. With two layers, the performance is mostly stable as the number of hidden layers are changed. We also show results with 3 and 4 layers with 41 hidden units. Performance degrades somewhat, and we believe this to be an overfit. We now test sensitivity to the number of training images, results are in Table 4. The results do not deteriorate much as we vary the number of images. Table 6 shows the results against training by varying the number of dilation of the ground truth during training phase.

Results on Synthetic Texture Dataset: We test our method on synthetic Brodatz with the previous network. Table 7 shows results, and our method performs the best by a wide margin. We also perform an experiment to test the performance of learned descriptors from STLD and non-STLD descriptors. Results are shown in Figure 3, which shows that STLD outperforms non-STLD by a significant margin.

Results on Graz and BSDS 500 Dataset: We now test our method against hand-crafted STLD on Graz and BSD500. These experiments provide more verification that the network is learning a generic property for segmentation over STLD. We initialize methods with a Voronoi partition of the seed points provided in Graz. For BSD500, we initialize the segmentation with 20 different initializations each with different random number of boxes of random sizes. The best result of the random initializations is chosen as the segmentation and the results are reported in Table 8. It shows that the learned STLD better capture properties of textures than the hand-crafted STLD.

Real-World Texture Dataset

	Contour		Region metrics					
	F-meas.		GT-cov.		Rand. Index		Var. Info.	
	ODS	OIS	ODS	OIS	ODS	OIS	ODS	OIS
Learned (ours)	**0.65**	0.65	**0.92**	**0.92**	**0.92**	**0.92**	**0.43**	**0.43**
Learned(non-STLD)	0.53	0.53	0.89	0.89	0.89	0.89	0.47	0.47
STLD	0.58	0.58	0.86	0.86	0.88	0.88	0.63	0.63
non-STLD	0.20	0.20	0.83	0.83	0.84	0.84	0.79	0.79
mcg [2]	0.51	0.54	0.74	0.82	0.77	0.85	0.80	0.66
gPb [1]	0.53	0.57	0.81	0.84	0.82	0.85	0.82	0.78
Kok.[12]	0.64	**0.66**	0.56	0.56	0.56	0.57	0.92	0.92
CTF[10]	0.60	0.60	0.91	0.91	0.91	0.91	0.45	0.45
CB [8]	0.54	0.56	0.75	0.80	0.79	0.84	0.81	0.76
SIFT	0.13	0.13	0.54	0.54	0.58	0.58	1.50	1.50
Entropy [7]	0.19	0.19	0.74	0.74	0.76	0.76	1.00	1.00
Hist-5 [18]	0.17	0.17	0.67	0.67	0.72	0.72	1.25	1.25
Hist-10 [18]	0.16	0.16	0.67	0.67	0.72	0.72	1.26	1.26
Chan-Vese [5]	0.19	0.19	0.73	0.73	0.76	0.76	1.07	1.07
LAC [13]	0.14	0.14	0.54	0.54	0.58	0.58	1.51	1.51
Global Hist [16]	0.14	0.14	0.66	0.66	0.68	0.68	1.16	1.16

Table 1. **Results on Texture Segmentation Datasets**. Algorithms are evaluated using contour and region metrics. Higher F-measure for the contour metric, ground truth covering (GT-cov), and rand index indicate better fit to the ground truth, and lower variation of information (Var. Info) indicates a better fit to ground truth.

Initialization

	Contour		Region metrics					
	F-meas.		GT-cov.		Rand. Index		Var. Info.	
	ODS	OIS	ODS	OIS	ODS	OIS	ODS	OIS
5by5	0.62	0.62	0.91	0.91	0.91	0.91	0.44	0.44
4by4	0.61	0.61	0.91	0.91	0.91	0.91	0.44	0.44
3by3	0.61	0.61	0.91	0.91	0.91	0.91	0.44	0.44

Table 2. **Insensitivity to Initialization**. The results remain similar as we vary the box tessellation initialization for segmentation.

Comparison of Learned STLD against pre-trained VGG Descriptor: Table 5 provides the results of descriptors obtained from convolutional layers of pre-trained VGG network. Descriptors from VGG (CNNs) are not shape-tailored and hence suffer from the same problems as "non-STLD" descriptors, i.e., they do not aggregate data within objects of interest, thus blurring the boundaries between objects, resulting in erroneous segmentation.

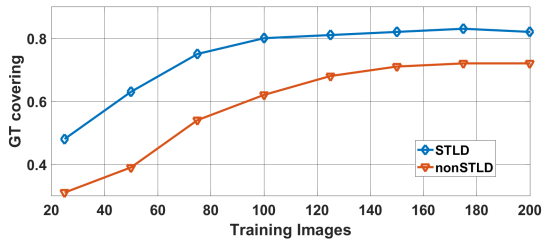

Figure 3. **Comparison of Training with STLD and non-STLD desp. on synthetic dataset.** The training images are varied form 25 to 200 and result of GT covering metric is reported. Learned STLD performs better then non-STLD for all training sizes and requires less no. of images for best possible performance.

Network Architecture

	Contour		Region metrics					
	F-meas.		GT-cov.		Rand. Index		Var. Info.	
	ODS	OIS	ODS	OIS	ODS	OIS	ODS	OIS
61 units	0.60	0.60	0.91	0.91	0.90	0.90	0.45	0.45
51 units	0.61	0.61	0.91	0.91	0.91	0.91	0.45	0.45
41 units	0.62	0.62	0.91	0.91	0.91	0.91	0.44	0.44
31 units	0.60	0.60	0.91	0.91	0.91	0.91	0.45	0.45
21 units	0.57	0.57	0.90	0.90	0.90	0.90	0.48	0.48
3 Layers	0.55	0.55	0.89	0.89	0.89	0.89	0.48	0.48
4 Layers	0.54	0.54	0.88	0.88	0.88	0.88	0.52	0.52
61 units	0.52	0.52	0.89	0.89	0.89	0.89	0.48	0.48
51 units	0.51	0.51	0.88	0.88	0.88	0.88	0.48	0.48
41 units	0.53	0.53	0.89	0.89	0.89	0.89	0.47	0.47
31 units	0.49	0.49	0.88	0.88	0.88	0.88	0.49	0.49
21 units	0.46	0.46	0.87	0.87	0.87	0.87	0.51	0.51
3 Layers	0.54	0.54	0.87	0.87	0.87	0.87	0.55	0.55
4 Layers	0.53	0.53	0.87	0.87	0.87	0.87	0.58	0.58

Table 3. **Performance vs. Architecture**. The top half of the table shows the performance for learned STLD descriptor (ours) and the bottom part show the results for learned non-STLD descriptor. We have varied the number of hidden units in the two-layer network, and the number of layers from 3-4 with 41 units.

Training Images

	Contour		Region metrics					
	F-meas.		GT-cov.		Rand. Index		Var. Info.	
	ODS	OIS	ODS	OIS	ODS	OIS	ODS	OIS
128 images	0.62	0.62	0.91	0.91	0.91	0.91	0.44	0.44
100 images	0.59	0.59	0.90	0.90	0.90	0.90	0.47	0.47
75 images	0.58	0.58	0.90	0.90	0.90	0.90	0.49	0.49
50 images	0.54	0.54	0.89	0.89	0.89	0.89	0.52	0.52

Table 4. **Varying Number of Images in Training**. We vary the number of training images and report the results.

VGG Descriptors in Segmentation

	Contour		Region metrics					
	F-meas.		GT-cov.		Rand. Index		Var. Info.	
	ODS	OIS	ODS	OIS	ODS	OIS	ODS	OIS
Learned (ours)	**0.65**	**0.65**	**0.92**	**0.92**	**0.92**	**0.92**	**0.43**	**0.43**
Learned(non-STLD)	0.53	0.53	0.89	0.89	0.89	0.89	0.47	0.47
VGG conv3 (256 dim)	0.49	0.49	0.84	0.84	0.84	0.84	0.67	0.67
VGG conv4 (512 dim)	0.44	0.44	0.79	0.79	0.80	0.80	0.77	0.77
VGG conv2 & 3 (384 dim)	0.47	0.47	0.86	0.86	0.87	0.87	0.63	0.63

Table 5. **Comparison of Learned Descriptors with pre-trained VGG descriptor.** The output of Convolutional layers of VGG network is used as dense descriptor for segmentation and is compared with our learned descriptors (detailed exp. in supp.).

Levels from each images in Training

	Contour		Region metrics					
	F-meas.		GT-cov.		Rand. Index		Var. Info.	
	ODS	OIS	ODS	OIS	ODS	OIS	ODS	OIS
50 steps	0.62	0.62	0.91	0.91	0.91	0.91	0.44	0.44
40 steps	0.60	0.60	0.91	0.91	0.90	0.90	0.46	0.46
30 steps	0.59	0.59	0.90	0.90	0.90	0.90	0.47	0.47
20 steps	0.60	0.60	0.91	0.91	0.90	0.90	0.46	0.46
10 steps	0.58	0.58	0.90	0.90	0.90	0.90	0.49	0.49

Table 6. **Varying no. of dilations for Images in Training**. We vary the number of dilation of the ground truth and report the effect on performance, higher no of dilations means more data per image.

5. Conclusion

We have shown how to construct learned-Shape-Tailored Descriptors for texture segmentation. The descriptors have two key properties. First, they are shape-tailored so that

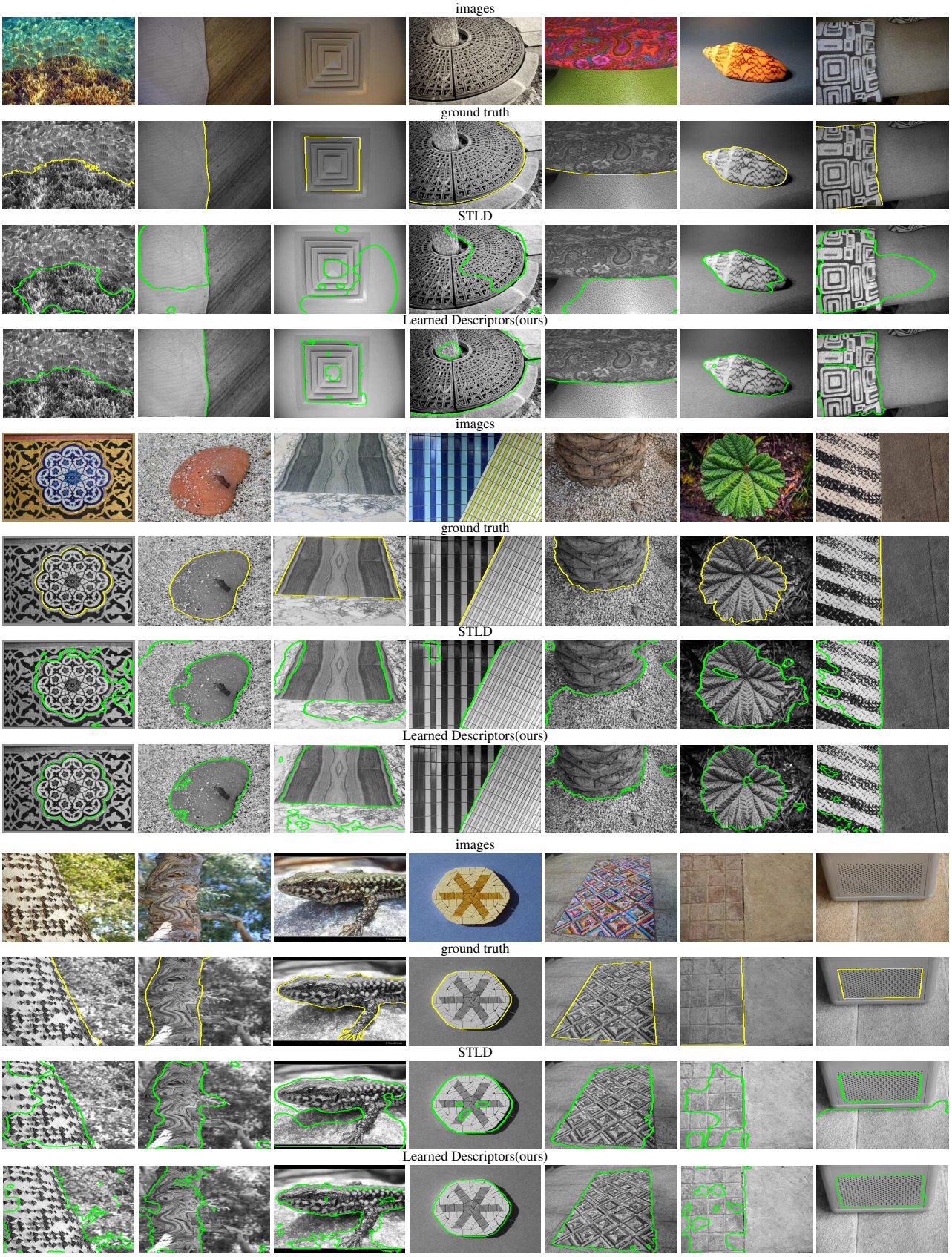

Figure 4. **Sample representative results on Real-World Texture Dataset**. *We compare the Learned Descriptors (ours) and STLD.*

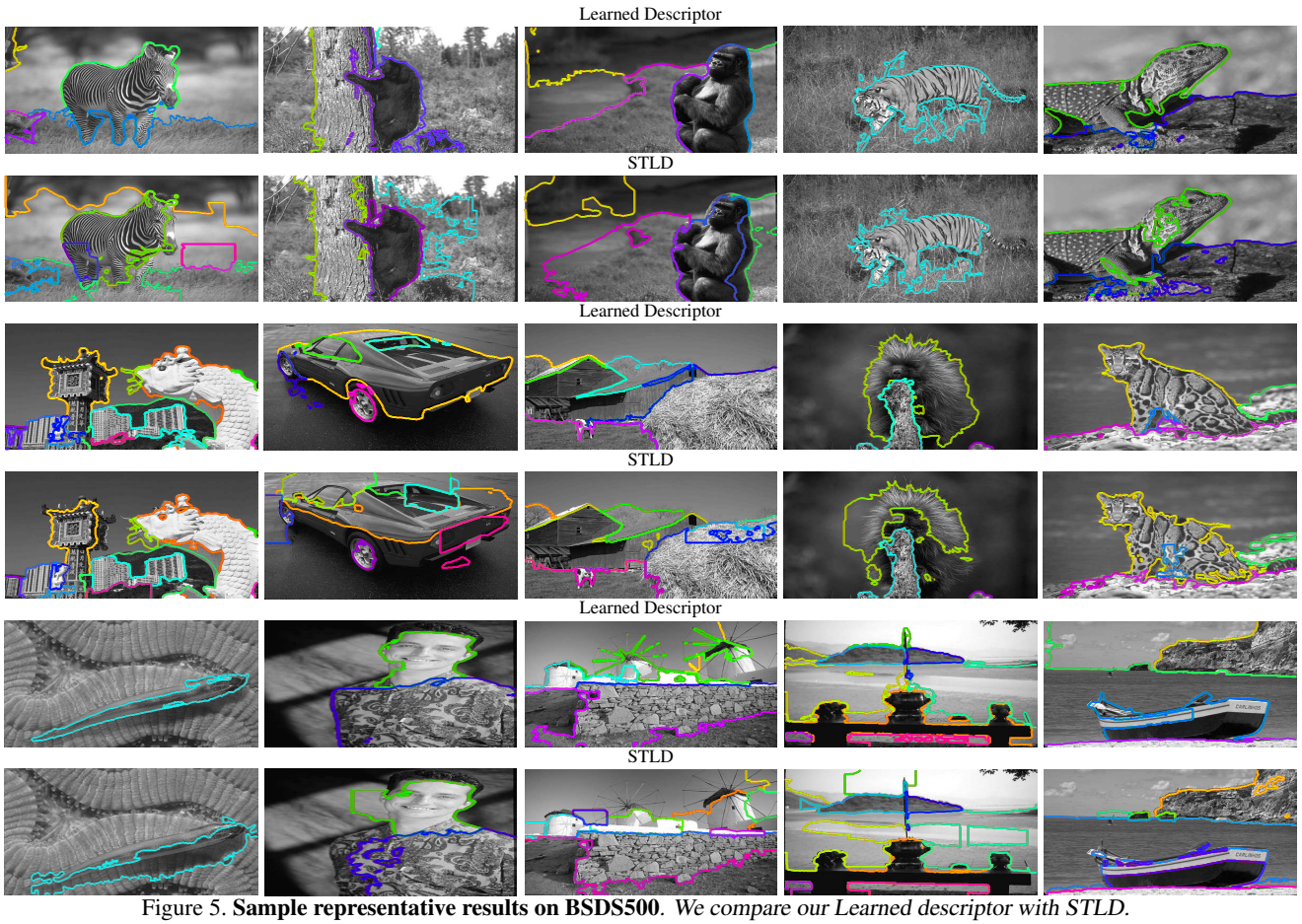

Figure 5. **Sample representative results on BSDS500**. *We compare our Learned descriptor with STLD.*

Synthetic Dataset

	Contour		Region metrics					
	F-meas.		GT-cov.		Rand. Index		Var. Info.	
	ODS	OIS	ODS	OIS	ODS	OIS	ODS	OIS
Learned (ours)	**0.45**	**0.45**	**0.90**	**0.90**	**0.89**	**0.89**	**0.46**	**0.46**
STLD	0.41	0.41	0.87	0.87	0.86	0.86	0.53	0.53
non-STLD	0.18	0.18	0.84	0.84	0.84	0.84	0.65	0.65
gPb [1]	0.40	0.38	0.79	0.81	0.79	0.82	0.75	0.73
CB [8]	0.30	0.29	0.75	0.77	0.76	0.79	1.09	1.08
SIFT	0.11	0.11	0.70	0.70	0.70	0.70	1.07	1.07
Entropy [7]	0.13	0.13	0.75	0.75	0.75	0.75	0.91	0.91
Hist-5 [18]	0.32	0.32	0.67	0.67	0.68	0.68	1.10	1.10
Hist-10 [18]	0.32	0.32	0.65	0.65	0.67	0.67	1.15	1.15
Chan-Vese [5]	0.19	0.19	0.72	0.72	0.72	0.72	0.95	0.95
LAC [13]	0.14	0.14	0.72	0.72	0.70	0.70	1.14	1.14
Global Hist [16]	0.28	0.28	0.75	0.75	0.75	0.75	0.79	0.79

Table 7. **Results on Synthetic Texture Segmentation Dataset**. See Table 1 caption for details on the measures.

Graz Dataset

	Contour		Region metrics					
	F-meas.		GT-cov.		Rand. Index		Var. Info.	
	ODS	OIS	ODS	OIS	ODS	OIS	ODS	OIS
Learned STLD	0.42	0.42	0.76	0.76	0.82	0.82	1.02	1.02
STLD	0.34	0.34	0.70	0.70	0.77	0.77	1.21	1.21

BSD500 Dataset

	Contour		Region metrics					
	F-meas.		GT-cov.		Rand. Index		Var. Info.	
ODS	OIS	ODS	OIS	ODS	OIS	ODS	OIS	
Learned STLD	0.66	0.66	0.67	0.67	0.86	0.86	1.54	1.54
STLD	0.56	0.56	0.57	0.57	0.79	0.79	1.99	1.99
gPb [1]	0.71	0.74	0.59	0.65	0.81	0.85	1.65	1.47

Table 8. **Graz and BSDS 500 Dataset Results** See Table 1 caption for details on the measures. Comparison on Graz dataset is between STLD and learned descriptors. On BSDS we compare STLD and learned STLD against state-of-the-art on region metric.

they are computed by aggregating image statistics only within regions of interest so they do not mix statistics across texture boundaries. Second, they exhibit invariances to complex nuisances, which was accomplished by learning descriptors derived from base hand crafted shape-tailored descriptors using neural networks. Experiments have shown that the learned descriptors are able to better cope with nuisances than hand-crafted shape-tailored descriptors. Currently our method requires the number of regions as initialization; we plan to address this in future work by considering a hierarchical approach. The focus of this work has been to construct learned shape-tailored descriptors.

Acknowledgements

This research was funded by KAUST OCRF-2014-CRG3-62140401 and VCC.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):898–916, 2011. 2, 5, 6, 8

[2] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 328–335, 2014. 6

[3] A. Arnab, S. Jayasumana, S. Zheng, and P. H. S. Torr. Higher order potentials in end-to-end trainable conditional random fields. *CoRR*, abs/1511.08119, 2015. 2

[4] V. Badrinarayanan, A. Handa, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *CoRR*, abs/1505.07293, 2015. 2

[5] T. F. Chan and L. A. Vese. Active contours without edges. *Image processing, IEEE transactions on*, 10(2):266–277, 2001. 4, 6, 8

[6] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, June 2005. 2, 3

[7] B.-W. Hong, S. Soatto, K. Ni, and T. Chan. The scale of a texture and its application to segmentation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 2, 6, 8

[8] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. Crisp boundary detection using pointwise mutual information. In *Computer Vision–ECCV 2014*, pages 799–814. Springer, 2014. 2, 6, 8

[9] N. Khan, M. Algarni, A. Yezzi, and G. Sundaramoorthi. Shape-tailored local descriptors and their application to segmentation and tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3890–3899, 2015. 1, 2, 4, 5

[10] N. Khan, B.-W. Hong, A. Yezzi, and G. Sundaramoorthi. Coarse-to-fine segmentation with shape-tailored continuum scale spaces. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 6

[11] J. Kim, J. W. Fisher III, A. Yezzi, M. Çetin, and A. S. Willsky. A nonparametric statistical method for image segmentation using information theory and curve evolution. *Image Processing, IEEE Transactions on*, 14(10):1486–1502, 2005. 2

[12] I. Kokkinos. Surpassing humans in boundary detection using deep learning. *CoRR*, abs/1511.07386, 2015. 2, 6

[13] S. Lankton and A. Tannenbaum. Localizing region-based active contours. *Image Processing, IEEE Transactions on*, 17(11):2029–2039, 2008. 6, 8

[14] F. Liu, G. Lin, and C. Shen. CRF learning with CNN features for image segmentation. *CoRR*, abs/1503.08263, 2015. 2

[15] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *JOSA A*, 7(5):923–932, 1990. 2

[16] O. Michailovich, Y. Rathi, and A. Tannenbaum. Image segmentation using active contours driven by the bhattacharyya gradient flow. *Image Processing, IEEE Transactions on*, 16(11):2787–2801, 2007. 6, 8

[17] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5):577–685, 1989. 4

[18] K. Ni, X. Bresson, T. Chan, and S. Esedoglu. Local histogram based segmentation using the wasserstein distance. *International Journal of Computer Vision*, 84(1):97–111, 2009. 6, 8

[19] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988. 4

[20] G. Peyré, J. Fadili, and J. Rabin. Wasserstein active contours. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 2541–2544. IEEE, 2012. 2

[21] C. Sagiv, N. A. Sochen, and Y. Y. Zeevi. Integrated active contours for texture segmentation. *IEEE transactions on image processing*, 15(6):1633–1646, 2006. 2

[22] J. Santner, T. Pock, and H. Bischof. Interactive multi-label segmentation. In *Proceedings 10th Asian Conference on Computer Vision (ACCV), Queenstown, New Zealand*, November 2010. 5

[23] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2

[24] S. Xie and Z. Tu. Holistically-Nested Edge Detection. *ArXiv e-prints*, apr 2015. 2

[25] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry. Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding*, 110(2):212–225, 2008. 2

[26] S. C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(9):884–900, 1996. 2