Non-Local Deep Features for Salient Object Detection

Zhiming Luo[1,2,3], Akshaya Mishra[4], Andrew Achkar[4], Justin Eichel[4], Shaozi Li[1,2]*, Pierre-Marc Jodoin[3]

[1]Department of Cognitive Science, Xiamen University, China

[2]Fujian Key Laboratory of Brain-Inspired Computing Technique and Applications, Xiamen University, China

[3]Department of Computer Science, University of Sherbrooke, Canada

[4]Miovision Technologies Inc., Canada

{zhiming.luo, pierre-marc.jodoin}@usherbrooke.ca, szlig@xmu.edu.cn

{amishra, aachkar, jeichel}@miovision.com

Abstract

Saliency detection aims to highlight the most relevant objects in an image. Methods using conventional models struggle whenever salient objects are pictured on top of a cluttered background while deep neural nets suffer from excess complexity and slow evaluation speeds. In this paper, we propose a simplified convolutional neural network which combines local and global information through a multi-resolution 4×5 grid structure. Instead of enforcing spacial coherence with a CRF or superpixels as is usually the case, we implemented a loss function inspired by the Mumford-Shah functional which penalizes errors on the boundary. We trained our model on the MSRA-B dataset, and tested it on six different saliency benchmark datasets. Results show that our method is on par with the state-of-the-art while reducing computation time by a factor of 18 to 100 times, enabling near real-time, high performance saliency detection.

1. Introduction

Saliency detection aims to mimic the human visual system which naturally separates predominant objects of a scene from the rest of the image. Several applications benefit from saliency detection including image and video compression [14], context aware image re-targeting [25], scene parsing [50], image resizing [3], object detection [44] and segmentation [33].

A salient object is often defined as a region whose visual features differ from the rest of the image and whose shape follows some *a priori* criteria [5]. Traditional methods typically extract local pixel-wise or region-wise features and compare it with global features. The result of that comparison is called a saliency score which is stored in a saliency map. Recently, deep learning has entered the field of saliency detection and quickly established itself as the *de facto* benchmark. Their greatest asset relative to traditional unsupervised approaches is that they can be trained end-to-end using simple optimization functions that combine local and deep features.

While some methods apply a straight forward convolutional neural net (CNN) model [36], others have proposed a model tailored to the saliency detection problem [25, 26, 29, 43, 50]. To achieve state-of-the-art performance, the top performing CNN models require non-trivial steps such as generating object proposals, applying post-processing, enforcing smoothness through the use of superpixels or defining complex network architectures, all the while making predictions far slower than real-time. As such, there remain opportunities to simplify the model architecture and speed up the computation.

In this paper, we show that the overarching objectives of state-of-the-art CNN models (enforcing spatial coherence of the predicted saliency map and using both the local and global features in the optimization) can be achieved with a much simplified non-local deep feature (NLDF) model. Spatial coherence is enforced with a Bayesian loss inspired by the Mumford-Shah (MS) functional [35]. The loss is expressed as the sum of a cross-entropy term and a boundary term. As opposed to conventional implementations of the MS functional, we use non-local features learned by a deep network instead of raw RGB colors. Also, rather than minimizing the boundary length directly (as done by unsupervised MS implementations), we minimize an intersection over union loss computed using predicted and ground truth boundary pixels. This boundary penalty term is shown to contribute significantly to our model's performance.

Our model's network is composed of convolution and deconvolution blocks organized in a 4×5 grid (see Figure 1) where each column of the grid extracts resolution-specific features. Local contrast processing blocks are also used along each resolution axis in order to promote features with strong local contrast. The resulting local and global features are combined into a "score" processing block that gives the final output at half of the input resolution.

*Corresponding author.

Since our method does not rely on superpixels, it is fully convolutional and thus achieves best-in-class evaluation speeds. The NLDF model evaluates an input image in 0.08s, a speed gain of 18 to 100 times as compared to other state-of-the-art deep learning methods, while being on par with state-of-the-art evaluation performance on the MSRA-B[30], HKU-IS[25], PASCAL-S[27], DUT-OMRON[49], ECSSD [48] and SOD [32] benchmark datasets.

The rest of the paper is organized as follows. Section 2 provides an overview of deep learning based saliency detection techniques. Section 3 describes the theory and practical implementation of our NLDF model. Finally, Section 4 discusses the performance of non-local feature model compared to other state-of-the-art saliency detection methods.

2. Related Works

Most previous methods implement an unsupervised model whose goal is to find objects with visual features different than those from the background. Prior efforts have tested simple features such as color and grayscale [2], edges [13] or texture [10], as well as more complex features such as objectness, focusness and backgroundness [18, 46, 9]. The literature offers a wide variety of unsupervised methods working at the pixel level [2], the region level [11], with graph-based methods [15, 45], and with a Bayesian formulation [47]. The reader shall refer to the survey paper by Borji *et al.* [5] for more details on unsupervised methods. While unsupervised methods have their advantages, including simplicity and no need for training, they have been outperformed by machine learning approaches. Although some traditional AI methods such as SVM [41] perform well, deep learning methods, specifically CNN models, have raised the bar and imposed themselves as the unavoidable standard. With CNNs, the saliency problem has been redefined as a labeling problem where feature selection between salient and non-salient objects is done automatically through gradient descent.

CNNs were first developed to perform image classification [22, 4, 24, 23]. These models are made of a series of convolution layers with non-linear activation functions and max pooling operations all the way to a softmax layer which predicts the likelihood of each class. CNN methods are *a priori* unfit to predict a saliency map since their output is a k-D vector (where k is the number of classes), and not an $N \times M$ map (where $N \times M$ is the size of the input image) as one would expect. However, one can alleviate that problem by extracting a square patch around each pixel and use that patch to predict the center pixel's class [12, 16]. In order for these methods to capture a global context that goes beyond the scope of each patch, they process patches taken from different resolutions of the input image.

Several deep visual saliency detection methods use this same patch trick for predicting a saliency map [29, 50, 25, 43]. Zhao *et al.* [50] integrated the global and local context of an image into a single, multi-context network, where the global context helps to model the saliency in the full image, and the local context helps to estimate the saliency of fine-grained, feature rich areas. Li *et al.* [25] developed a computational model using multi-scale deep features extracted by three CNNs and three fully connected layers to define salient regions of an image. Such a complex model was designed to capture the saliency map of objects with various scales, geometry, spatial positions, irregularities, and contrast levels. Wang *et al.* [43] developed a two tier strategy: each pixel is assigned a saliency based upon a local context estimation in parallel to a global search strategy used to identify the salient regions. These two saliency maps are then combined using geodesic object proposal techniques [21].

Another way of having the output resolution of a CNN match the input image resolution is through one (or several) upsampling layer(s). A popular method for doing so is the FCN method by Long *et al.* [31] which adds an upsampling layer at the very end of the network. Saliency detection methods using that approach are among the most accurate ones [7, 36, 26] most likely because they better capture the local and global context than patch-wise methods.

In order to enforce spatial coherence, a large number of methods use pre-computed regions or super pixels [25, 26, 50, 29, 43]. Roughly speaking, the idea is to set the saliency score of a superpixel as the mean saliency score of each pixel located inside of it. Since superpixels can be inaccurate, some methods [43] use several object proposals which they combine afterwards while others use more than one CNN stream [26, 50]. Spacial coherence can also be enforced by using a CRF or mean-field postprocess [26, 23]. The main inconvenience with these approaches is their processing time.

Our approach differs from these methods as it uses a single and fully convolutional CNN. It uses a series of multiscale convolution and deconvolution blocks organized in a novel 5×4 grid. Our CNN model ensures that the output has the right size while capturing the local and global context as well as features at various resolutions. Spatial coherence is enforced with a loss function inspired by the Mumford-Shah model [35] which we adapted to the context of machine learning.

3. Proposed Method

3.1. Model Based Saliency Detection

Salient region detection as well as image segmentation often boils down to the optimization of a non-convex energy function which consists of a data term and a regularization term. An elegant mathematical global model is the cartoon

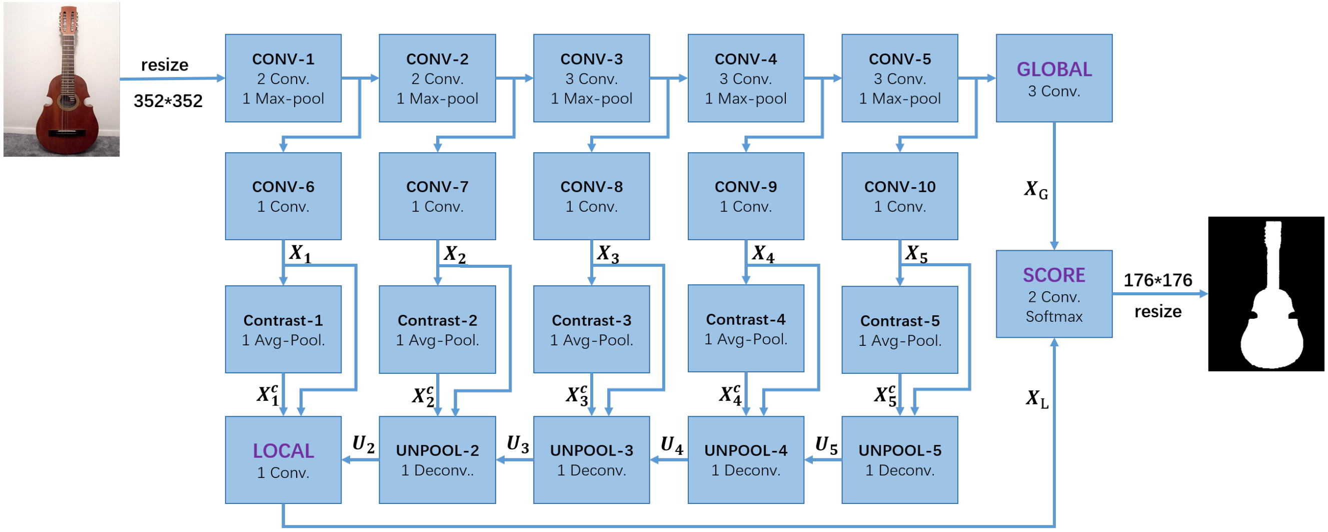

Figure 1. Architecture of our 4 × 5 grid-CNN network for salient object detection.

Mumford-Shah (MS) model [35], whose fitting energy,

$$F^{\text{MS}} = \underbrace{\sum_j \lambda_j \int_{\mathbf{v}\in\Omega_j} \left|I(\mathbf{v}) - u_j\right|^2 d\mathbf{v}}_{\text{data fidelity}} + \underbrace{\sum_j \gamma_j \oint_{\mathbf{v}\in C_j} d\mathbf{v}}_{\text{boundary length}} \tag{1}$$

segments an image I as a set of disjoint piece-wise constant functions u_j, indexed by j. Here, $\Omega \subset R^N$ is an open set representing the image domain, I is the observed image, u_j is the underlying piece-wise constant segmented image, $\mathbf{v}$ is a pixel location, and C is the boundary of the segmented regions. The positive weighting constants λ_j, and γ_j tune the multi-criteria energy function in terms of data fidelity, and total boundary length. From a Bayesian statistical perspective [6, 51], Eq. (1) can be approximated as,

$$F^{\text{MS}} \approx \underbrace{\sum_j \lambda_j \int_{\mathbf{v}\in\Omega_j} \log p_j(I(\mathbf{v}), \mathbf{v}) d\mathbf{v}}_{\text{data fidelity}} + \underbrace{\sum_j \gamma_j \oint_{\mathbf{v}\in C_j} d\mathbf{v}}_{\text{boundary length}}. \tag{2}$$

As there is no analytic solution to Eqs. (1) and (2), the most common unsupervised approaches to optimize these employ level set base curve evolution techniques [8, 42], generalized Bayesian criteria using the variational principle, and simulated annealing [51]. Despite their mathematical elegance, these methods are all iterative in nature, making them sensitive to initial conditions and likely to fail in the presence of noise, background clutter, weak image boundaries or image non-uniformity. Furthermore, poor convergence rates in the iterative solution of the level set limits their utility to non real-time applications.

To address these issues, we propose a supervised deep convolutional network whose loss approximates the MS functional with the sum of a cross entropy data fidelity term between the ground truth and estimated saliency and a boundary loss term:

$$F^{\text{MS}} \approx \underbrace{\sum_j \lambda_j \int_{\mathbf{v}\in\Omega_j} \mathrm{H}_j(y(\mathbf{v}), \hat{y}(\mathbf{v}))}_{\text{cross entropy}} + \underbrace{\sum_j \gamma_j (1 - \mathrm{IoU}(C_j, \hat{C}_j))}_{\text{boundary IoU loss}} \tag{3}$$

where H_j is the total cross entropy between ground truth (y) and estimated ($\hat{y}$) saliency map of all pixels ($\mathbf{v}$) inside region Ω_j, and $\mathrm{IoU}(C_j, \hat{C}_j)$ is the intersection over union between the pixels on the true boundary C_j and the pixels on the estimated boundary $\hat{C}_j$. Note that since our method implements a supervised version of the MS functional, the use of the IoU allows our method to learn a higher level *a priori* term, i.e. a term that learns to penalize erroneous boundaries instead of minimizing the total boundary length.

3.2. Network Architecture

Here we provide a deep convolutional network architecture whose goal is to learn discriminant saliency features (our model is shown in Figure 1). As mentioned in Sec. 2, good saliency features must account for both the local and global context of an image and incorporate details from various resolutions. To achieve this goal, we have implemented a novel grid-like CNN network containing 5 columns and 4 rows. Here, each column is geared toward the extraction of features specific to a given input scale. The input I to our model (on the left) is an 352×352 image and the output (on the right) is a 176×176 saliency map which we resize back to 352×352 with a bilinear interpolation.

The first row of our model contains five convolutional blocks derived from VGG-16 [39] (CONV-1 to CONV-

Table 1. Details of the proposed deep convolutional network for predicting salient objects (S: Stride, Pad: zero padding).

Block	Layer	Kernel	S	Pad	Output
CONV-1	2 conv	3*3	1	Yes	352*352*64
	max-pool	2*2	2	Yes	176*176*64
CONV-2	2 conv	3*3	1	Yes	176*176*128
	max-pool	2*2	2	Yes	88*88*128
CONV-3	3 conv	3*3	1	Yes	88*88*256
	max-pool	2*2	2	Yes	44*44*256
CONV-4	3 conv	3*3	1	Yes	44*44*512
	max-pool	2*2	2	Yes	22*22*512
CONV-5	3 conv	3*3	1	Yes	22*22*512
	max-pool	2*2	2	Yes	11*11*512
CONV-6	conv	3*3	1	Yes	176*176*128
CONV-7	conv	3*3	1	Yes	88*88*128
CONV-8	conv	3*3	1	Yes	44*44*128
CONV-9	conv	3*3	1	Yes	22*22*128
CONV-10	conv	3*3	1	Yes	11*11*128
UNPOOL-5	deconv	5*5	2	Yes	22*22*128
UNPOOL-4	deconv	5*5	2	Yes	44*44*256
UNPOOL-3	deconv	5*5	2	Yes	88*88*384
UNPOOL-2	deconv	5*5	2	Yes	176*176*512
LOCAL	conv	1*1	1	No	176*176*640
GLOBAL	conv-1	5*5	1	No	7*7*128
	conv-2	5*5	1	No	3*3*128
	conv-3	3*3	1	No	1*1*128
SCORE	conv-L	1*1	1	No	176*176*2
	conv-G	1*1	1	No	1*1*2

5). As shown in Table 1, these convolution blocks contain a max pooling operation of stride 2 which downsamples their feature maps $\{X_1, ..., X_5\}$ by a factor of 2, e.g. $\{176 \times 176, 88 \times 88, ..., 11 \times 11\}$. The last and rightmost convolution block of the first row computes features X_G that are specific to the global context of the image.

The second and third row is a set of ten convolutional blocks, CONV-6 to CONV-10 for row 2 and Contrast-1 to Contrast-5 for row 3. The aim of these blocks is to compute features (X_i) and contrast features (X_i^C) specific to each resolution. The contrast features capture the difference of each feature against its local neighborhood favoring regions that are either brighter or darker than their neighbors.

The last row is a set of deconvolution layers used to upscale the features maps from 11×11 (bottom right) all the way to 176×176 (bottom left). These UNPOOL layers are a means of combining the feature maps (X_i, X_i^C) computed at each scale. The lower left block constructs the final local feature maps X_L. The SCORE block has 2 convolution layers and a softmax to compute the saliency probability by fusing the local (X_L) and global (X_G) features. Further details of our model are given in Table 1.

3.2.1 Non-Local Feature Extraction

Multi-Scale local features: As shown in the second row of Figure 1, the convolutional blocks CONV-6 to CONV-10 are connected to the VGG-16 CONV-1 to CONV-5 processing blocks. The goal of these convolutional layers is to learn multi-scale local feature maps $\{X_1, X_2, ..., X_5\}$. Each convolution block has a kernel size 3×3 and 128 channels.

Contrast features: Saliency is the distinctive quality of a foreground object which makes it stand out from its surrounding background. Saliency features must thus be uniform inside the foreground objects and within the background but at the same time be different between foreground and background areas. In order to capture this kind of contrast information, we added a contrast feature associated to each local feature X_i. Each contrast feature X_i^c is computed by subtracting X_i from its local average. The kernel size of the average pooling is 3×3

$$X_i^c = X_i - \text{AvgPool}(X_i). \tag{4}$$

Note that such contrast feature is similar in spirit to that of Achanta *et al.* [2] which computes the difference between the pixel RGB color and the global average color of the image. It is even closer to that of Liu and Gleicher [28] which computes contrast features from a Gaussian image pyramid. However, our approach is different as our features are learned and not predefined.

Deconvolution features: Since the size of the final output is 176×176, we use a series of deconvolution layers to increase the size of the precomputed features maps X_i and X_i^C. Instead of increasing the feature maps by a ratio of $\{2, 4, 8, 16\}$ as suggested by Long *et al.* [31] which results in coarse feature maps, we adopt a step-wise upsampling procedure as showed in the third row in Figure 1. At each UNPOOL processing block, we upsample the previous feature maps by a factor of 2. The resulting unpooled feature map U_i is computed by combining the information of its local feature X_i, local contrast feature X_i^c, and the previous block's unpooled feature U_{i+1}

$$U_i = \text{UNPOOL}(X_i, X_i^c, U_{i+1}). \tag{5}$$

The UNPOOL operation is implemented with a deconvolution layer with a stride of 2 and a 5×5 kernel. The input is the concatenation of X_i, X_i^C and U_{i+1} . The number of feature channels of U_i is equal to the sum of X_i and U_{i+1}.

Local feature maps: We use a convolution layer with a kernel size 1×1 to get the final local feature maps X_L. The input of that layer is the concatenation of X_1, X_1^C and U_2

$$X_L = \text{CONV}(X_1, X_1^c, U_2). \tag{6}$$

The number of feature channels of X_L is equal to the sum of X_1 and U_2. Note that we tried using another UNPOOL operation to increase the size of X_L from 176×176 to 352×352, but found that this operation doubles the computation time without measurably improving accuracy.

Capturing global context: Detecting salient objects in an image requires the model to capture the global context of the image before assigning saliency to individual small regions. To account for this, we added three convolutional layers after the CONV-5 block to compute the global feature X_G. The first two convolutional layers have a kernel size of 5, and the last convolutional is 3. All three layers have 128 features channels.

3.3. Cross Entropy Loss

The final saliency map is computed as a linear combination of the local features X_L and global features X_G using two linear operators (W_L, b_L) and (W_G, b_G). The softmax function is used to compute the probability for each pixel of being salient or not.

$$\hat{y}(\mathbf{v}) = p\big(y(\mathbf{v}) = c\big) = \frac{e^{W_L^c X_L(\mathbf{v}) + b_L^c + W_G^c X_G + b_G^c}}{\sum_{c' \in \{0,1\}} e^{W_L^{c'} X_L(\mathbf{v}) + b_L^{c'} + W_G^{c'} X_G + b_G^{c'}}} \tag{7}$$

The cross-entropy loss function

$$H_j(y(\mathbf{v}), \hat{y}(\mathbf{v})) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c \in \{0,1\}} (y(\mathbf{v}_i) = c)\big(\log(\hat{y}(\mathbf{v}_i) = c)\big) \tag{8}$$

is used to minimize the first data term in Eq. (2).

3.4. IoU Boundary Loss

Motivated by the significant applications of Dice loss or IoU boundary loss in medical image segmentation [53, 40, 34], our proposed method approximates the penalty on boundary length of Eq. (1) using an IoU boundary loss term. To compute the boundary loss, we approximate the saliency map gradient magnitude (and hence the boundary pixels) using a Sobel operator followed by a $\tanh$ activation. The $\tanh$ activation projects the gradient magnitude of saliency maps to a probability range of $[0, 1]$. Given the gradient magnitude of saliency maps $\hat{C}_j$ and gradient magnitude of true saliency maps C of region j, the Dice or IoU boundary loss can be computed as

$$\text{IoU Loss} = 1 - \frac{2|C_j \cap \hat{C}_j|}{|C_j| + |\hat{C}_j|}, \tag{9}$$

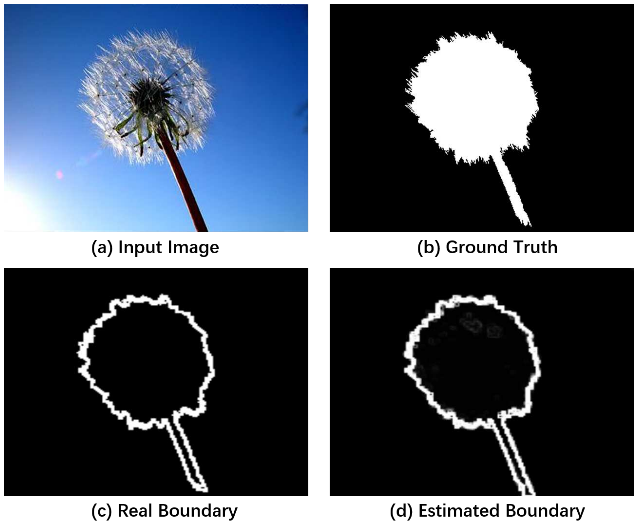

Figure 2. A single input image (a) together with its groundtruth saliency (b) and boundary (c) is used to train a model only containing the IoU boundary loss term in Eq. (3). The estimated boundary (d) after training for 200 iterations is in excellent agreement with the true boundary.

which has range [0, 1]. Our whole boundary overlapping loss computation procedure is end-to-end trainable, and an example is shown in Figure 2. Please note that the intersection is implemented using a point-wise multiplication operator.

4. Experimental Results

4.1. Benchmark Datasets

We have evaluated the performance of our method (NLDF) on six different public benchmark datasets: MSRA-B [30], HKU-IS [25], DUT-OMRON [49], PASCAL-S [27], ECSSD [48] and SOD [32].

MSRA-B: contains 5000 images, and is widely used for visual saliency detection. Most of the images have one salient object and a pixel-wise ground truth [17].

HKU-IS: contains 4447 images, most of which have low contrast and multiple salient objects. This dataset has been split into 2500 training images, 500 validation images and the remaining 1447 test images.

DUT-OMRON: contains 5168 challenging images, each of which contains one or more salient objects with a relatively cluttered background.

PASCAL-S: contains 850 natural images which were built from the validation set of the PASCAL-VOC 2010 segmentation challenge. This dataset contains both pixel-wise saliency ground truth and eye fixation ground truth labeled by 12 subjects.

ECSSD: contains 1000 images with complex structure acquired from the Internet. The ground truth masks were labeled by 5 subjects.

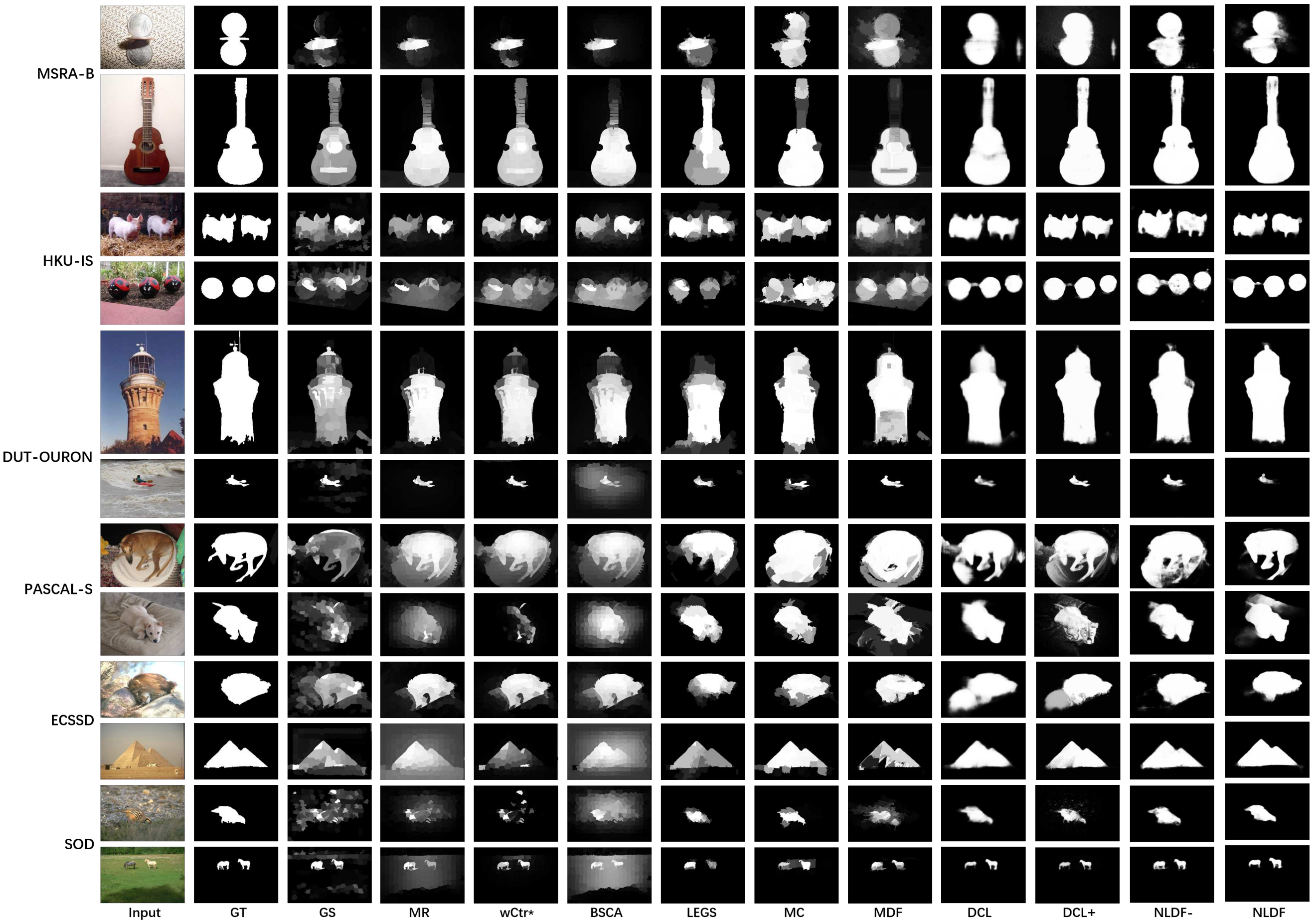

Figure 3. Saliency maps produced by the GS [46], MR [49], wCtr* [52], BSCA [38], LEGS [43], MC [50], MDF [25] and DCL [26] methods compared to our NLDF method. The NLDF maps provides clear salient regions and exhibit good uniformity as compared to the saliency maps from the other deep learning methods (LEGS, MC, MDF and DCL). Our method is also more robust to background clutter than the non-deep-learning methods (GS, MR, wCtr* and BSCA).

SOD: contains 300 images originally designed for image segmentation. Many images contain multiple salient objects with low contrast and overlapping boundaries.

4.2. Implementation and Experimental Setup

Our NLDF model was implemented in TensorFlow [1]. The weights in the CONV-1 to CONV-5 blocks were initialized with the pretrained weights of VGG-16 [39]. All the weights of newly added convolution and deconvolution layers were initialized randomly with a truncated normal ($\sigma = 0.01$), and the biases were initialized to 0. The Adam optimizer [19] was used to train our model with an initial learning rate of 10^{-6}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The λ_j and γ_j in Eq. (3) were set to 1.

For fair comparison with other methods, we followed the experimental setup of [17], dividing the MSRA-B dataset into 3 parts: 2500 images for training, 500 images for validation and the remaining 2000 images for testing. The training and validation sets were combined together to train our model with horizontal flipping as data augmentation. The inputs were resized to 352×352 for training. With an NVIDIA Titan X GPU, it takes $\sim$ 9 hours to finish the whole training procedure for 20 epochs with a single image batch size. Without further optimization, this trained model was used to compute the saliency maps of the other datasets.

4.3. Evaluation Criteria

Precision-recall (PR) curves, F_β and mean absolute error (MAE) were used as metrics to evaluate the performance of saliency detection. The PR curve is computed by binarizing the saliency maps under different probability thresholds ranging from 0 to 1 and comparing against the ground truth. As for the F_β measure, it is defined as,

$$F_\beta = \frac{(1+\beta^2) \cdot \text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}. \qquad (10)$$

where $\beta^2 = 0.3$ to emphasize precision over recall as suggested in [2]. We report the maximum F-Measure computed

Table 2. Quantitative performance of our model on six benchmark datasets compared with the GS [46], MR [49], wCtr* [52], BSCA [38], LEGS [43], MC [50], MDF [25] and DCL [26]models. The latter four are deep learning methods and the former are not. The F_β and MAE metrics are defined in the text.

Dataset	Metric	GS	MR	wCtr*	BSCA	LEGS	MC	MDF	DCL	DCL+	**NLDF-**	**NLDF**
MSRA-B	max F_β	0.777	0.824	0.820	0.830	0.870	0.894	0.885	0.905	**0.916**	0.912	0.911
	MAE	0.144	0.127	0.110	0.130	0.081	0.054	0.066	0.052	**0.047**	0.048	0.048
HKU-IS	max F_β	0.682	0.715	0.726	0.723	0.770	0.798	0.861	0.892	**0.904**	0.874	0.902
	MAE	0.167	0.174	0.141	0.174	0.118	0.102	0.076	0.054	0.049	0.060	**0.048**
DUT-OMRON	max F_β	0.557	0.610	0.630	0.616	0.669	0.703	0.694	0.733	**0.757**	0.724	0.753
	MAE	0.173	0.187	0.144	0.191	0.133	0.088	0.092	0.084	**0.080**	0.085	**0.080**
PASCAL-S	max F_β	0.624	0.666	0.659	0.666	0.756	0.740	0.764	0.815	0.822	0.804	**0.831**
	MAE	0.224	0.223	0.201	0.224	0.157	0.145	0.145	0.113	0.108	0.116	**0.099**
ECSSD	max F_β	0.661	0.736	0.716	0.758	0.827	0.822	0.832	0.887	0.901	0.886	**0.905**
	MAE	0.206	0.189	0.171	0.183	0.118	0.106	0.105	0.072	0.075	0.075	**0.063**
SOD	max F_β	0.601	0.619	0.632	0.634	0.707	0.688	0.745	0.795	0.801	0.776	**0.810**
	MAE	0.266	0.273	0.245	0.266	0.215	0.197	0.192	**0.142**	0.153	0.161	0.143

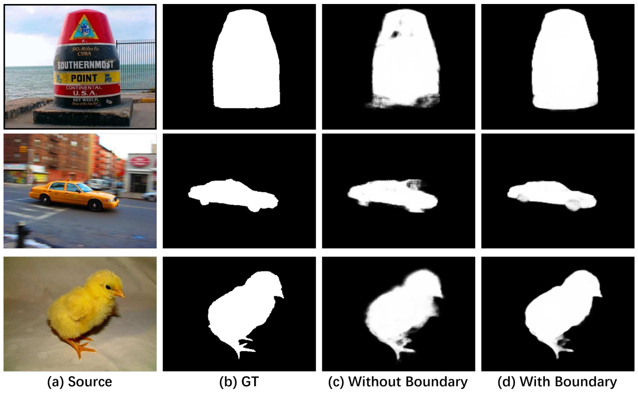

Figure 4. Visual comparison of saliency detection results with and without the boundary loss term in Eq. (2).

from the PR curve. MAE [37] is computed as the average pixel-wise absolute difference between the estimated saliency map S and its corresponding ground truth L,

$$\mathrm{MAE} = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} \left| S(x,y) - L(x,y) \right|. \tag{11}$$

where W and H is the width and height of a given image.

4.4. Effectiveness of the Boundary Loss Term

In addition to our NLDF model, we also trained a model, denoted as NLDF-, which only contains the cross-entropy loss term and excludes the boundary loss term [see Eq. 3]. As shown in Figure 4, the saliency maps generated from NLDF- are fairly coarse and the boundary of the salient objects are not well preserved. As shown in last two columns of Table 2, this qualitative decrease in performance is also mirrored in the quantitative results. The inclusion of the boundary loss in NLDF as compared to NLDF- accounts for increases in max F_β of 2.1% to 4.4% and decreases in MAE of 5.8% to 20.0% on HKU-IS, DUT-OMRON, PASCAL-S, ECSSD and SOD datasets. Little change is observed for MSRA-B, an expected result, since training and testing samples are drawn from a similar pool of images. Significantly, these results illustrate that the boundary loss term directly enhances the generality of NLDF, making it more robust to variations in input types.

4.5. Comparison with the State of the Art

We quantitatively compared our NLDF method with several recent state-of-the-art methods: Geodesic Saliency (GS) [46], Manifold Ranking (MR) [49], optimized Weighted Contrast (wCtr*) [52], Background based Single-layer Cellular Automata (BSCA) [38], Local Estimation and Global Search (LEGS) [43], Multi-Context (MC) [50], Multiscale Deep Features (MDF) [25] and Deep Contrast Learning (DCL) [26]. LEGS, MC, MDF and DCL are the latest deep learning based saliency detection methods. Note that since part of the HKU-IS dataset was used to train the MDF model [25], we only compute the evaluation metrics on the testing set of HKU-IS. Also the MDF only provided 200 pre-compute saliency maps on SOD dataset, we use the same subset for evaluation.

In comparison to the top performing method, DCL+, [26] an extension of DCL that uses a fully-connected CRF [20] as a post-processing step to refine the saliency map, we find that NLDF attains nearly identical (or better) performance across the board (see Table 2). That this is achieved without a significant post-processing step means that the execution time and implementation complexity are greatly reduced. The computation time reported in [26] for DCL is 1.5 s per (300 × 400) image and an additional 0.8 s for CRF post-processing (DCL+). In comparison, our NLDF method only requires 0.08 s per image on a Titan X GPU. This substantial speedup enables nearly real-time salient object detection while also delivering state-of-the-art performance.

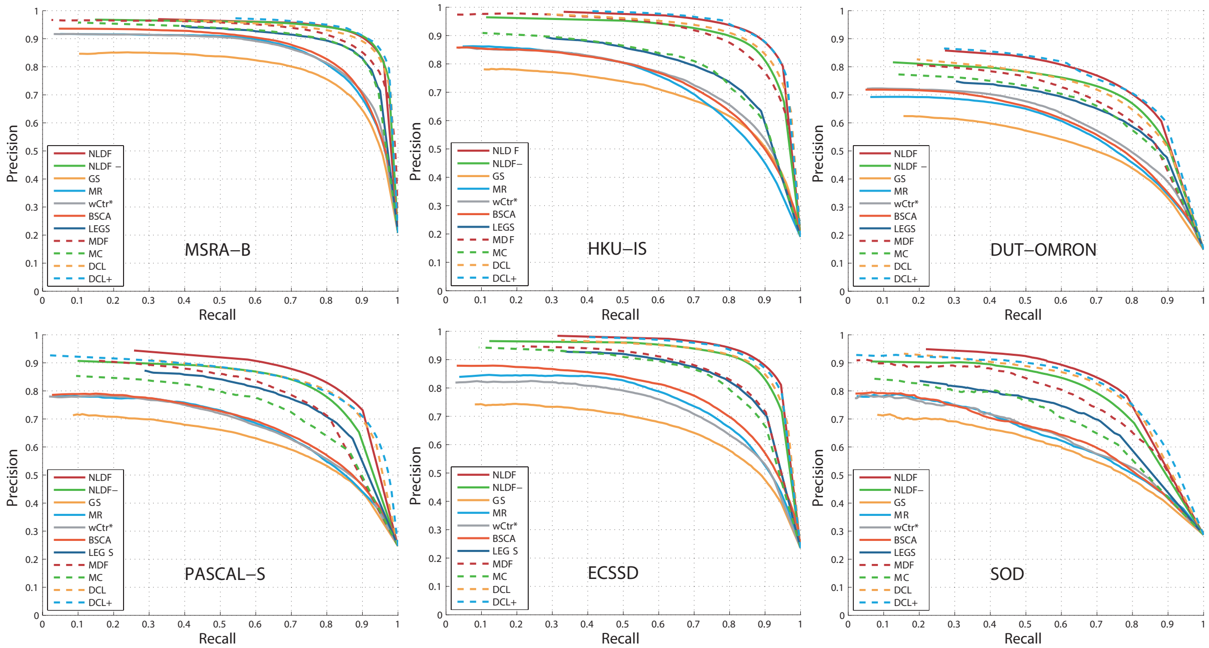

Figure 5. Precision-recall curves for our model compared to GS [46], MR [49], wCtr* [52], LEGS [43], BSCA [38], MDF [25], MC [50] and DCL [26] evaluated on the MASR-B, HKU-IS, DUT-OMRON, PASCAL-S, ECSSD and SOD benchmark datasets. Our NLDF model can deliver state-of-the-art performance on all six datasets.

A visual comparison of the saliency maps is provided in Figure 3. All saliency maps of other methods were either provided by the authors or computed using the authors' released code. Precision-recall curves are shown in Figure 3 and the maximum F_β and MAE scores are in Table 2. As shown in Table 2, our NLDF model achieves superior quantitative max F_β, MAE and PR performance across the board when compared to GS, MR, wCtr*, BSCA, LEGS, MC, MDF and DCL. NLDF also surpasses DCL+ more times than not in max F_β and MAE and exhibits equivalent or better PR curves.

We also compared the average computation time with other four leading deep learning methods for generating the saliency map of one images in Table 3. On a Titan Black GPU, our approach is 18 to 100 times faster than existing methods.

Table 3. Inference time of leading deep learning methods.

	LEGS	MC	MDF	DCL	DCL+	**NLDF**
s/img	2	1.6	8	1.5	2.3	**0.08**

5. Conclusion

The integration of local and global features has already been shown to be a powerful mechanism for saliency detection. Here we took this approach one step further by adding a boundary loss term to the typical cross entropy loss, in effect implementing the Mumford-Shah functional in a deep neural net framework and training it end to end. The resulting model achieves state of the art performance across multiple saliency detection benchmark datasets, does not use any special pre- or post-processing steps and computes saliency maps 18 to 100 times faster than competing systems.

6. Acknowledgments

This work is supported by the Nature Science Foundation of China (No.61572409, No.61402386 & No.61571188), Fujian Province 2011 Collaborative Innovation Center of TCM Health Management and Collaborative Innovation Center of Chinese Oolong Tea Industry Collaborative Innovation Center (2011) of Fujian Province, FQRNT Team research project (No.172083), National Research Council Canada IRAP and National Sciences and Engineering Research Council of Canada ENGAGE.

References

[1] M. Abadi and et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 6

[2] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *Proc. CVPR*, 2009. 2, 4, 6

[3] R. Achanta and S. Süsstrunk. Saliency detection for content-aware image resizing. In *Proc. ICIP*, 2009. 1

[4] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013. 2

[5] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A survey. *arXiv preprint arXiv:1411.5878*, 2014. 1, 2

[6] T. Brox and D. Cremers. On local region models and a statistical interpretation of the piecewise smooth mumford-shah functional. *Int. J. Comput. Vision*, 84(2):184–193, Jun. 2009. 3

[7] N. Bruce, C. Catton, and S. Janjic. A deeper look at saliency: Feature contrast, semantics, and beyond. In *Proc. CVPR*, 2016. 2

[8] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Trans. Image Process.*, 10(2):266–277, 2001. 3

[9] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai. Fusing generic objectness and visual saliency for salient object detection. In *Proc. ICCV*, 2011. 2

[10] Z. Chen, Y. Liu, B. Sheng, J.-N. Liang, J. Zhang, and Y.-B. Yuan. Image saliency detection using gabor texture cues. *Multimedia Tools and Applications*, pages 1–16, 2015. 2

[11] C.Li, Y.Yuan, W.Cai, Y.Xia, and D.Feng. Robust saliency detection via regularized random walks ranking. In *Proc. CVPR*, 2015. 2

[12] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1915–1929, 2013. 2

[13] C. Guo and L. Zhang. A simple method for detecting salient regions. *Pattern Recognition*, 42(11):2363–2371, 2009. 2

[14] C. Guo and L. Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE Trans. Image Process.*, 19(1):185–198, Jan 2010. 1

[15] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *Proc. NIPS*, 2006. 2

[16] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle. Brain tumor segmentation with deep neural networks. *Medical Image Analysis*, 35:18–31, 2017. 2

[17] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *Proc. CVPR*, 2013. 5, 6

[18] P. Jiang, H. Ling, J. Yu, and J. Peng. Salient region detection by ufo: Uniqueness, focusness and objectness. In *Proc. ICCV*, 2013. 2

[19] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[20] V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Proc. NIPS*, 2011. 7

[21] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *Proc. ECCV*, 2014. 2

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012. 2

[23] S. S. Kruthiventi, V. Gudisa, J. H. Dholakiya, and R. Venkatesh Babu. Saliency unified: A deep architecture for simultaneous eye fixation prediction and salient object segmentation. In *Proc. CVPR*, 2016. 2

[24] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 2

[25] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *Proc. CVPR*, 2015. 1, 2, 5, 6, 7, 8

[26] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *Proc. CVPR*, 2016. 1, 2, 6, 7, 8

[27] Y. Li, X. Hou, C. Koch, J. Rehg, and A. Yuille. The secrets of salient object segmentation. In *Proc. CVPR*, 2014. 2, 5

[28] F. Liu and M. Gleicher. Region enhanced scale-invariant saliency detection. In *Proc. ICME*, 2006. 4

[29] N. Liu, J. Han, D. Zhang, S. Wen, and T. Liu. Predicting eye fixations using convolutional neural networks. In *Proc. CVPR*, 2015. 1, 2

[30] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H. Shum. Learning to detect a salient object. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(2):353–367, 2011. 2, 5

[31] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, 2015. 2, 4

[32] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. ICCV*, 2001. 2, 5

[33] P. Mehrani and O. Veksler. Saliency segmentation based on learning and graph cut refinement. In *Proc. BMVC*, 2010. 1

[34] F. Milletari, N. Navab, and S.-A. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. *arXiv preprint:1606.04797*, 2016. 5

[35] 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. 1, 2, 3

[36] J. Pan, E. Sayrol, X. G. i Nieto, K. McGuinness, and N. O'Connor. Shallow and deep convolutional networks for saliency prediction. In *Proc. CVPR*, 2016. 1, 2

[37] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *Proc. CVPR*, 2012. 7

[38] Y. Qin, H. Lu, Y. Xu, and H. Wang. Saliency detection via cellular automata. In *Proc. CVPR*, 2015. 6, 7, 8

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint:1409.1556*, 2014. 3, 6

[40] A. A. Taha and A. Hanbury. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC medical imaging*, 15(1):29, 2015. 5

[41] N. Tong, H.Lu, X. Ruan, and M.-H. Yang. Salient object detection via bootstrap learning. In *Proc. CVPR*, 2015. 2

[42] L. A. Vese and T. F. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *Int. J. Comput. Vision*, 50(3):271–293, 2002. 3

[43] L. Wang, H. Lu, X. Ruan, and M. Yang. Deep networks for saliency detection via local estimation and global search. In *Proc. CVPR*, 2015. 1, 2, 6, 7, 8

[44] P. Wang, J. Wang, G. Zeng, J. Feng, H. Zha, and S. Li. Salient object detection for searched web images via global saliency. In *Proc. CVPR*, 2012. 1

[45] Q. Wang, W. Zheng, and P. Robinson. Grab: Visual saliency via novel graph model and background priors. In *Proc. CVPR*, 2016. 2

[46] Y. Wei, F. Wen, W. Zhu, and J. Sun. Geodesic saliency using background priors. In *Proc. ECCV*, 2012. 2, 6, 7, 8

[47] Y. Xie, H. Lu, and M.-H. Yang. Bayesian saliency via low and mid level cues. *IEEE Trans. Image Process.*, 22(5):1689–1698, 2013. 2

[48] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *Proc. CVPR*, 2013. 2, 5

[49] C. Yang, L. Zhang, H. Lu, X. Ruan, and M. Yang. Saliency detection via graph-based manifold ranking. In *Proc. CVPR*, 2013. 2, 5, 6, 7, 8

[50] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *Proc. CVPR*, 2015. 1, 2, 6, 7, 8

[51] S. C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(9):884–900, Sept. 1996. 3

[52] W. Zhu, S. Liang, Y. Wei, and J. Sun. Saliency optimization from robust background detection. In *Proc. CVPR*, 2014. 6, 7, 8

[53] K. H. Zou, S. K. Warfield, A. Bharatha, C. M. Tempany, M. R. Kaus, S. J. Haker, W. M. Wells, F. A. Jolesz, and R. Kikinis. Statistical validation of image segmentation quality based on a spatial overlap index 1: Scientific reports. *Academic radiology*, 11(2):178–189, 2004. 5