Deep Contrast Learning for Salient Object Detection

Guanbin Li Yizhou Yu
Department of Computer Science, The University of Hong Kong
{gbli, yzyu}@cs.hku.hk

Abstract

Salient object detection has recently witnessed substantial progress due to powerful features extracted using deep convolutional neural networks (CNNs). However, existing CNN-based methods operate at the patch level instead of the pixel level. Resulting saliency maps are typically blurry, especially near the boundary of salient objects. Furthermore, image patches are treated as independent samples even when they are overlapping, giving rise to significant redundancy in computation and storage. In this paper, we propose an end-to-end deep contrast network to overcome the aforementioned limitations. Our deep network consists of two complementary components, a pixel-level fully convolutional stream and a segment-wise spatial pooling stream. The first stream directly produces a saliency map with pixel-level accuracy from an input image. The second stream extracts segment-wise features very efficiently, and better models saliency discontinuities along object boundaries. Finally, a fully connected CRF model can be optionally incorporated to improve spatial coherence and contour localization in the fused result from these two streams. Experimental results demonstrate that our deep model significantly improves the state of the art.

1. Introduction

Visual saliency aims at identifying the most visually distinctive parts in an image, and has received increasing interest in recent years. Though early work primarily focused on predicting eye-fixations in images, research has shown that salient object detection, which emphasizes object-level integrity of saliency prediction results, is more useful and can serve as a pre-processing step for a variety of computer vision and image processing tasks including content-aware image editing [3], object detection [36], image classification [45], person re-identification [4] and video summarization [32]. Despite recent progress, salient object detection remains a challenging problem that calls for more accurate solutions.

Results from perceptual research [11, 37] indicate that visual contrast is the most important factor in visual saliency. Various conventional saliency detection algorithms based on local or global contrast cues [7, 48] have been successfully developed. In previous work, visual contrast is exemplified by contrast in various types of handcrafted low-level features (e.g., color, intensity and texture) at the pixel or segment level. Though handcrafted features tend to perform well in standard scenarios, they are not sufficiently robust for all challenging cases. For example, local contrast features may fail to detect homogenous regions inside salient objects while global contrast suffers from complex background. Although machine learning based saliency models have been developed [31, 20, 29, 33], they are primarily for integrating different handcrafted features [20] or fusing multiple saliency maps generated from different methods [33].

To obtain more robust features than handcrafted ones for salient object detection, deep convolutional neural networks (CNNs) have recently been employed, achieving substantially better results than previous state of the art [25, 49, 43]. In addition to improved robustness, features extracted using CNNs contain more high-level semantic information since those CNNs were typically pre-trained on datasets for visual recognition tasks. However, in all these methods, CNNs are all operated at the patch level instead of the pixel level, and each pixel is simply assigned the saliency value of its enclosing patch. As a result, saliency maps are typically blurry without fine details, especially near the boundary of salient objects. Furthermore, all image patches are treated as independent data samples for classification or regression even when they are overlapping. As a result, these methods usually have to run a CNN at least thousands of times (once for every patch) to obtain a complete saliency map. This gives rise to significant redundancy in computation and storage, and makes both training and testing very space and time consuming. For example, training a patch-oriented CNN model for saliency detection takes over 2 GPU days and requires hundreds of gigabytes of storage for the 5000 images in the MSRA-B dataset.

In this paper, inspired by a recent trend of developing fully convolutional neural networks for pixel labeling prob-

lems [30, 6, 46], we propose an end-to-end deep contrast network to overcome the aforementioned limitations of recent CNN-based saliency detection methods. Here, "end-to-end" means that our deep network only needs to be run on the input image once to produce a complete saliency map with the same pixel resolution as the input image. Our deep network consists of a pixel-level fully convolutional stream and a segment-level spatial pooling stream. In the fully convolutional stream, we design a multi-scale fully convolutional network (MS-FCN), which takes the raw image as input and directly produces a saliency map with pixel-level accuracy. Our MS-FCN can not only generate effective semantic features across different scales, but also capture subtle visual contrast among multi-scale feature maps for saliency inference. The segment-level spatial pooling stream generates another saliency map at the superpixel level by performing spatial pooling and saliency estimation over superpixels. This stream extracts segment-wise features very efficiently from MS-FCN by masking an intermediate feature map computed for the entire image. The saliency maps from both streams are fused at the end.

In summary, this paper has the following contributions:

- We introduce an end-to-end deep contrast network for salient object detection. It consists of a fully convolutional stream and a segment-wise spatial pooling stream. A training scheme is designed to learn the weights in both streams of this deep network. The fused saliency map from these two streams is further refined with a fully connected CRF for better spatial coherence and contour localization.

- We propose a multi-scale fully convolutional network as the first stream in our deep contrast network to infer a pixel-level saliency map directly from the raw input image. This model can not only infer semantic properties of salient objects, but also capture visual contrast among multi-scale feature maps.

- We also design a segment-wise spatial pooling stream as the second stream in our framework. This stream efficiently extracts segment-wise features, and accurately models visual contrast between regions and saliency discontinuities along region boundaries.

2. Related Work

Salient object detection can be performed either in a bottom-up fashion using low-level features [13, 1, 29, 22, 38, 48, 20, 50, 7] or in a top-down fashion via the incorporation of high-level knowledge [21, 5, 16, 40, 28, 18, 27]. Since this paper is focused on visual saliency based on deep learning, we discuss relevant work in this context below.

Recently, machine learning and artificial intelligence have been revolutionized by deep convolutional neural networks, which have set new state of the art on a number of visual recognition tasks, including image classification [24], object detection [15], scene classification [47] and scene parsing [12], closing the gap to human-level performance. There have also been attempts to apply deep learning to salient object detection. Li *et al.* [25] trained a deep neural network for deriving a saliency map from multiscale features extracted using deep convolutional neural networks. Wang *et al.* [43] adopted a deep neural network (DNN-L) to learn local patch features for each centered pixel. In [49], both global context and local context are utilized and integrated into a deep learning based pipeline for saliency detection. However, all these methods treat local image patches as independent training and testing samples. Since sharing computation among overlapping patches is not considered, there is a great deal of redundancy in feature computation, which gives rise to high computational cost for both training and testing. This limitation can be potentially overcome by recent end-to-end deep networks, which have been proven a success in semantic segmentation [30, 6]. However, directly applying existing fully convolutional network architecture to salient object detection would not be most appropriate because a standard fully convolutional model is not particularly good at capturing subtle visual contrast in an image. Therefore, our paper focuses on discovering high-level visual contrast in an end-to-end mode, and experimental results demonstrate that our proposed deep model can significantly improve the current state of the art. This paper can be viewed as the first piece of work that aims to discover visual contrast information inside an image using end-to-end convolutional neural networks.

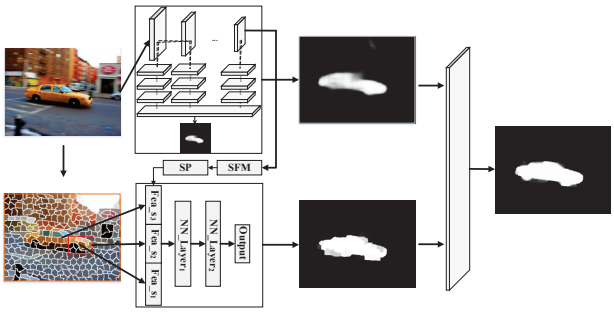

Figure 1. Two streams of our deep contrast network.

3. Deep Contrast Network

As shown in Fig. 1, the architecture of our deep contrast network for salient object detection consists of two complementary components, a fully convolutional stream

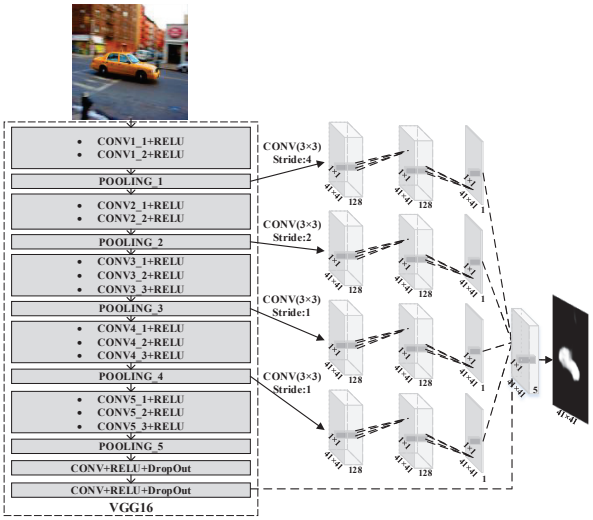

Figure 2. The architecture of multi-scale fully convolutional network.

and a segment-wise spatial pooling stream. The fully convolutional stream is a multi-scale fully convolutional network (MS-FCN), which generates a saliency map S_1 with one eighth resolution of the raw input image by exploiting visual contrast across multiscale convolutional layers. The segment-wise spatial pooling stream generates a saliency map at the superpixel level by performing spatial pooling and saliency estimation over individual superpixels. The saliency maps from both streams are fused at the end through an extra convolutional layer with 1×1 kernels in our deep network to produce the final saliency map. The weights in this fusion layer are learned during training.

3.1. Multi-Scale Fully Convolutional Network

In the fully convolutional stream, we aim to design an end-to-end convolutional network that can be viewed as a regression network mapping an input image to a pixel-level saliency map. To conceive such an end-to-end architecture, we have the following considerations. First, the network should be deep enough to produce multi-level features for detecting salient objects at different scales. Second, the network should be able to discover subtle visual contrast across multiple maps holding deep features at different scales. Last but not the least, fine-tuning an existing deep model is much desired since we do not have enough training images to train such a deep network from scratch.

We chose VGG16 [41] as our pre-trained network and modified it to meet our requirements. To re-purpose it into a dense image saliency prediction network, the two fully connected layers of VGG16 are first converted into convolutional ones with 1×1 kernel as described in [30]. However directly evaluating the resulting network in a convolutional manner yields a very sparse prediction map with a 32-pixel stride since the original VGG16 network has 5 pooling layers each of which has stride 2. To make the prediction map denser, we skip subsampling in the last two max-pooling layers to maintain an 8-pixel stride after the last pooling layer. To retain the original receptive field in the convolutional layers that follow, we use the "hole algorithm" to introduce zeros to increase the size of their convolutional kernels. The "hole algorithm", which is also called à trous algorithm, was originally developed for efficient computation of the undecimated wavelet transform [34], and has recently been implemented in Caffe [6, 26] to efficiently compute dense CNN feature maps at any target subsampling rate without introducing any approximation. This hole algorithm helps us keep the kernels intact, and a convolution now sparsely samples the input feature map using a stride of 2 or 4 pixels (2-pixel stride in the three convolutional layers after the penultimate pooling layer and 4-pixel stride in the last two converted 1×1 convolutional layers after the final pooling layer). For our experiments, we followed the implementation of the published DeepLab code [6] and added the option to sparsely sample the underlying feature map to the 'im2col' function. 'im2col' is a function implemented in Caffe to convert multi-channel feature maps to vectorized patches for improving the efficiency of convolutions.

VGG16 has five pooling and downsampling layers, each of which has an increasingly larger receptive field containing contextual information. To design a deep network that is capable of discovering visual contrast crucial in saliency inference, we further develop a multiscale version of the above fully convolutional extension of VGG16. As shown in Fig. 2, we connect three extra convolutional layers to each of the first four max-pooling layers of VGG16. The first extra layer has 3×3 kernels and 128 channels, the second extra layer has 1×1 kernels and 128 channels, and the third extra layer (output feature map) has a 1×1 kernel and a single channel. To make the output feature maps of the four sets of extra convolutional layers have the same size ($8\times$ subsampled resolution), the stride of the first layer in these four sets are set to 4, 2, 1, and 1, respectively. Although the resulting four output maps have the same size, they are generated using receptive fields with different sizes and hence represent contextual features at 4 different scales. We further stack these four feature maps together with the final output map of the above end-to-end extension. The stacked feature maps (5 channels) are fed into a final convolutional layer with a 1×1 kernel and a single output channel, which is the inferred saliency map. The sigmoid activation function is used in the final layer. Although the output saliency map is of $8\times$ subsampled resolution, they are smooth enough and allow us to use simple bilinear interpolation to make their resolution the same as that of the

original input image at a negligible computational cost. We call this resized saliency map S_1.

Note that the method in [30] does not use the "hole algorithm" and produces very coarse maps (subsampled by a factor of 32), which motivate the use of trained deconvolution layers. The incorporation of deconvolution layers significantly increases the complexity and training time of their network. Experimental results also show that convolution with the "hole algorithm" can generate better results than trained deconvolution layers [6].

3.2. Segment-Level Saliency Inference

Salient objects often have irregular shapes and the corresponding saliency map has discontinuities along object boundaries. Our multiscale fully convolutional network operates at a subsampled pixel level without explicitly modeling such saliency discontinuities. To better model visual contrast between regions and visual saliency along region boundaries, we design a segment-wise spatial pooling stream in our network.

We first decompose the raw input image into a set of superpixels, and call each superpixel a segment. A mask is computed for every segment in the feature map generated from the last true convolutional layer (Conv5_3) of MS-FCN as follows. Since each activation in Conv5_3 is controlled by a receptive field in the input image, we first project every activation to the center of its receptive field as in [14, 9]. For each segment in the input image, we first generate a binary mask with the same size as its bounding box. In this mask, pixels inside the segment are labeled '1' while others are labeled '0'. Each label in the binary mask is first assigned to the nearest center of receptive field and then backprojected onto Conv5_3. Thus, each activation in Conv5_3 collects multiple binary labels backprojected from its receptive field. The collected binary labels at each activation are first averaged and then thresholded by 0.5, yielding a corresponding binary segment mask on Conv5_3, where pixels within the segment can be easily identified according to this mask. Note that feature maps generated from Conv5_3 have 8-pixel strides in our MS-FCN instead of 32-pixel ones in the original VGG16 network since subsampling was skipped in the last two max-pooling layers as described in Section 3.1. Therefore, the resolution of the feature map generated from Conv5_3 is sufficient for segment masking.

Since segments on Conv5_3 have variable size, to produce a fixed-length feature vector, we further perform spatial pooling (SP) over a fixed grid as with [17]. We divide the bounding box of a segment on Conv5_3 into $h \times w$ cells. Let the size of the bounding box be $H \times W$. Spatial pooling is performed within each cell with $H/h \times W/w$ pixels. Afterwards, the aggregated feature vector of each segment has $h \times w \times C$ dimensions, where C is the number of channels of the feature map generated by Conv5_3.

To discover segment-level visual contrast, for each segment, we obtain three spatially aggregated feature vectors from three nested and increasingly larger windows, which are respectively the bounding box of the considered segment, the bounding box of its immediate neighboring segments, and the entire map from Conv5_3 (with the considered segment masked out to indicate the position of the segment in the map). Finally, the three aggregated feature vectors are concatenated and fed into two fully connected layers. The output of the second fully connected layer is fed into the output layer, which uses the sigmoid function to perform logistic regression to produce a distribution over binary saliency labels. We call the saliency map generated in this way S_2.

This segment-wise spatial pooling stream of our network is in fact an accelerated version of the method in [25]. Although they share similar strategies for multiscale feature extraction, our method is much more efficient because convolutional feature maps only need to be computed once for the entire image and afterwards, local features for thousands of segments from the same image can be masked out instantaneously. Moreover, our model also achieves better results as segment features are extracted from our multiscale fully convolutional network, which has been fine-tuned for salient object detection, instead of from the original VGG16 model for image classification.

3.3. Deep Contrast Network Training

Given training images and their superpixels, we first train the neural network in the second stream alone to obtain its initial weights. Segment features are extracted using the original VGG16 network pre-trained over the ImageNet dataset [10]. After this initialization, we fine-tune the two streams of our deep contrast network in an alternating manner. We first fix the parameters in the second stream and train the first stream for one epoch. During this process, the weights for fusing the saliency maps (S_1 and S_2) from the two streams as well as the parameters in the multiscale fully convolutional network are updated using stochastic gradient descent. Then we fix the parameters in the first stream and fine-tune the neural network in the second stream for one epoch using groundtruth saliency maps. Segment features are extracted using the updated VGG16 network embedded in the first stream. We typically alternate the above two steps 8 times (16 epochs in total) before the whole fine-tuning process converges.

The loss function for fine-tuning the deep contrast network (the first stream) and the fusing weights is the cross entropy between the ground truth and the fused saliency

map (S):

$$L = -\beta_i \sum_{i=1}^{|I|} G_i \log P\left(S_i = 1 | I_i, W\right) - \left(1-\beta_i\right) \sum_{i=1}^{|I|} \left(1-G_i\right) \log P\left(S_i = 0 | I_i, W\right), \quad (1)$$

where G is the groundtruth label, W denotes the collection of all network parameters in MS-FCN and the fusion layer, β_i is a weight balancing the number of salient pixels and unsalient ones, and $|I|$, $|I|_-$ and $|I|_+$ denote the total number of pixels, unsalient pixels and salient pixels in image I, respectively. Then $\beta_i = \frac{|I|_-}{|I|}$ and $1-\beta_i = \frac{|I|_+}{|I|}$. When fine-tuning the second stream, its parameters are updated by minimizing the squared prediction errors accumulated over all segments from all training images.

4. The Complete Algorithm

4.1. Superpixel Segmentation

We aim to decompose the input image into non-overlapping segments. In this paper, we use a slightly modified version of the SLIC algorithm [2], which uses geodesic image distance [8] during K-means clustering in the CIELab color space. As discussed in [44], geodesic distance based superpixels can guarantee connectivity while well preserve edges in the image. In our experiments, we have found that the final saliency detection performance does not vary much when the number of superpixels is between 200 and 300. And the performance becomes slightly worse when the number of superpixels is fewer than 200 or more than 300.

4.2. Spatial Coherence

Since both streams in our deep contrast network assign saliency scores to individual pixels or segments without considering the consistency of saliency scores among neighboring pixels and segments, we propose a pixel-wise saliency refinement model based on a fully connected CRF [23] to improve spatial coherence. This model solves a binary pixel labeling problem, and employs the following energy function,

$$E\left(L\right) = -\sum_i \log P\left(l_i\right) + \sum_{i,j} \theta_{ij}\left(l_i, l_j\right), \quad (2)$$

where L represents a binary label (salient or not salient) assignment for all pixels. $P(l_i)$ is the probability of pixel x_i having label l_i, which indicates the likelihood of pixel x_i being salient. Initially, $P(1) = S_i$ and $P(0) = 1 - S_i$, where S_i is the saliency score at pixel x_i from the fused saliency map S. $\theta_{ij}\left(l_i, l_j\right)$ is a pairwise potential and defined as follows,

$$\theta_{ij} = \mu\left(l_i, l_j\right)\left[\omega_1 \exp\left(-\frac{\left\|p_i - p_j\right\|^2}{2\sigma_\alpha^2} - \frac{\left\|I_i - I_j\right\|^2}{2\sigma_\beta^2}\right) + \omega_2 \exp\left(-\frac{\left\|p_i - p_j\right\|^2}{2\sigma_\gamma^2}\right)\right], \quad (3)$$

where $\mu\left(l_i, l_j\right) = 1$ if $l_i \neq l_j$, and zero otherwise. θ_{ij} involves two kernels. The first kernel depends on pixel positions (p) and pixel intensities (I). This kernel encourages nearby pixels with similar colors to take similar saliency scores. The degree of influence by color similarity and spatial closeness is controlled by three parameters (σ_α and σ_β), respectively. The second kernel aims at removing small isolated regions.

Energy minimization is based on a mean field approximation to the CRF distribution, and high-dimensional filtering can be utilized to speed up the computation. In this paper, we use the publicly available implementation of [23] to minimize the above energy, and it takes less than 0.5 second on an image with 300×400 pixels. At the end of energy minimization, we generate a saliency map using the posterior probability of each pixel being salient. We call the generated saliency map S_{crf}. As shown in Fig. 3, the saliency maps generated from the proposed method without CRF are fairly coarse and the contours of salient objects may not be well preserved. The proposed saliency refinement model can not only generate smoother results with pixelwise accuracy but also well preserve salient object contours. A quantitative study of the effectiveness of the saliency refinement model can be found in Section 5.3.2.

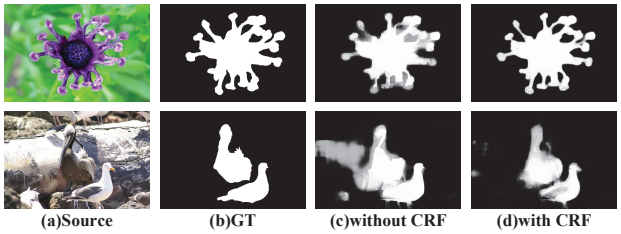

Figure 3. Comparison of saliency detection results with and without CRF.

5. Experimental Results

5.1. Experimental Setup

5.1.1 Datasets

We evaluate the performance of our method on five public datasets: MSRA-B [29], PASCAL-S [27], DUT-OMRON [48], HKU-IS [25] and SOD [35]. The MSRA-B dataset contains 5,000 images with a variety of image

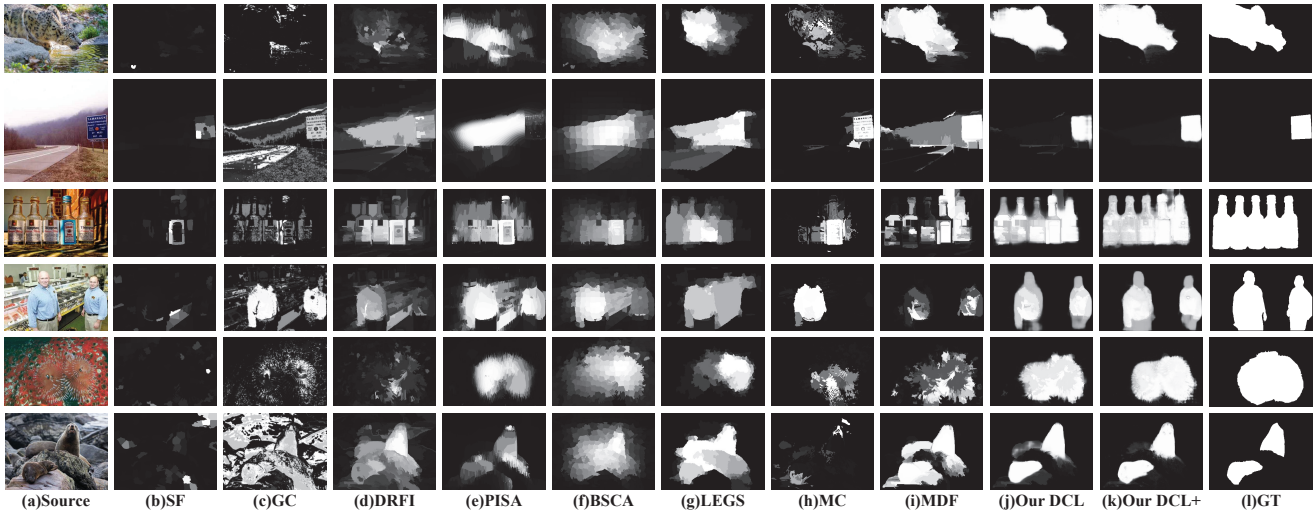

Figure 4. Visual comparison of saliency maps generated from state-of-the-art methods, including our DCL and DCL$^+$. The ground truth (GT) is shown in the last column. DCL$^+$ consistently produces saliency maps closest to the ground truth.

Data Set	Metric	SF	GC	DRFI	PISA	BSCA	LEGS	MC	MDF	FCN	DCL	DCL$^+$
MSRA-B	maxF	0.700	0.719	0.845	0.837	0.830	0.870	**0.894**	0.885	0.864	**0.905**	**0.916**
	MAE	0.166	0.159	0.112	0.102	0.130	0.081	**0.054**	0.066	0.096	**0.052**	**0.047**
HKU-IS	maxF	0.590	0.588	0.776	0.753	0.723	0.770	0.798	0.861	**0.867**	**0.892**	**0.904**
	MAE	0.173	0.211	0.167	0.127	0.174	0.118	0.102	**0.076**	0.087	**0.054**	**0.049**
DUT-OMRON	maxF	0.495	0.495	0.664	0.630	0.617	0.669	**0.703**	0.694	0.681	**0.733**	**0.757**
	MAE	0.147	0.218	0.150	0.141	0.191	0.133	**0.088**	0.092	0.131	**0.084**	**0.080**
PASCAL-S	maxF	0.493	0.539	0.690	0.660	0.666	0.752	0.740	0.764	**0.793**	**0.815**	**0.822**
	MAE	0.240	0.266	0.210	0.196	0.224	0.157	0.145	0.145	**0.128**	**0.113**	**0.108**
SOD	maxF	0.516	0.526	0.699	0.660	0.654	0.732	0.727	0.785	**0.795**	**0.829**	**0.832**
	MAE	0.267	0.284	0.223	0.223	0.251	0.195	0.179	**0.155**	0.158	**0.129**	**0.126**

Table 1. Comparison of quantitative results including maximum F-measure (larger is better) and MAE (smaller is better). The best three results are shown in **red**, **blue**, and **green** , respectively.

contents. Most of the images has a single salient object. PASCAL-S was built using the validation set of the PASCAL VOC 2010 segmentation challenge. It contains 850 images with the ground truth labeled by 12 subjects. We threshold the masks at 0.5 to obtain binary masks as suggested in [27]. Dut-OMRON contains 5,168 challenging images, each of which has one or more salient objects and relatively complex backgrounds. We have noticed that many saliency annotations in this dataset may be controversial among different human observers. As a result, none of the existing saliency models has achieved a high accuracy on this dataset. HKU-IS is another large dataset containing 4447 challenging images, most of which have either low contrast or multiple salient objects. The SOD dataset contains 300 images and it was originally designed for image segmentation. Many images in this dataset have multiple salient objects with low contrast. All the datasets contain manually annotated groundtruth saliency maps. To facilitate a fair comparison against other methods, we divide the MSRA-B dataset into three parts as in [20, 25], 2500 for training, 500 for validation and the remaining 2000 images for testing. To test the adaptability of trained saliency models to other different datasets, we use the models trained on the MSRA-B dataset and test them over all other datasets.

5.1.2 Evaluation Criteria

We evaluate the performance using precision-recall (PR) curves, F-measure and mean absolute error (MAE). The precision and recall of a saliency map is computed by converting a continuous saliency map to a binary mask using a threshold and comparing the binary mask against the ground truth. The PR curve of a dataset is obtained from the average precision and recall over saliency maps of all images in the dataset. The F-measure is defined as

$$F_\beta = \frac{(1+\beta^2)\cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}, \qquad (4)$$

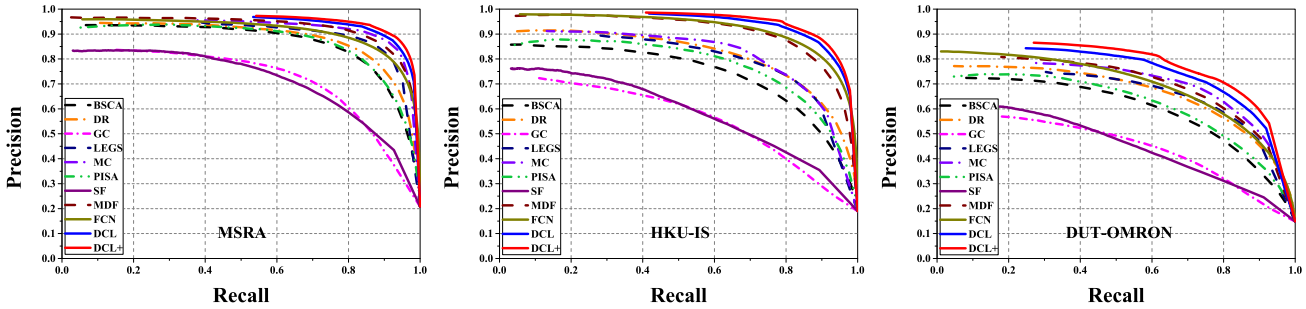

Figure 5. Comparison of precision-recall curves of 11 saliency detection methods on 3 datasets. Our DCL and DCL$^+$ (DCL with CRF) consistently outperform other methods across all the testing datasets. Note that MC [49] and LEGS [43] are overrated on the MSRA-B dataset and LEGS [43] is also overrated on the PASCAL-S dataset.

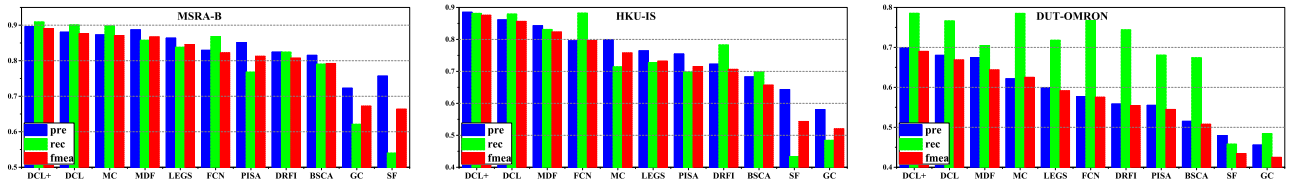

Figure 6. Comparison of precision, recall and F-measure (computed using a per-image adaptive threshold) among 11 different methods on 3 datasets.

where β^2 is set to 0.3 to weigh precision more than recall as suggested in [1]. We report the maximum F-measure (maxF) computed from the PR curve. We also report the average precision, recall and F-measure using an adaptive threshold for generating a binary saliency map. The adaptive threshold is determined to be twice the mean value of a saliency map. In addition, MAE [38] represents the average absolute per-pixel difference between an estimated saliency map and its corresponding ground truth. MAE is meaningful in evaluating the applicability of a saliency model in a task such as object segmentation.

5.1.3 Implementation

Our proposed deep contrast network has been implemented on the basis of Caffe [19], an open source framework for CNN training and testing. We resize all the images to 321×321 pixels for training, and set the initial learning rate to 0.01 for all newly added layers with one channel and 0.001 for all other layers. The momentum parameter is set to 0.9 and the weight decay is 0.0005. For the segment-level stream, the number of superpixels is set to 400 with 3 different scales (200, 150 and 50 respectively). A 2×2 grid is used for spatial pooling over each segment. Thus the aggregated feature for each segment has 6144 dimensions, and this feature is further fed into two fully connected layers each of which has 300 neurons. The parameters of the fully connected CRF are determined through cross validation as in [23] on the validation set and finally the paramters of w_1, w_2, σ_α, σ_β, and σ_γ are set to 3.0, 5.0, 3.0, 50.0 and 3.0 respectively in our experiments.

We use DCL to denote our saliency model based on deep contrast learning only without CRF-based post-processing, and DCL$^+$ to denote the saliency model that includes CRF-based refinement. While it takes around 25 hours to train our deep contrast network using the MSRA-B dataset, it only takes 1.5 seconds for the trained model (DCL) to detect salient objects in a testing image with 400x300 pixels on a PC with an NVIDIA Titan Black GPU and a 3.4GHz Intel processor. Note that this is far more efficient than the latest deep learning based methods which treat all image patches as independent data samples for saliency regression. CRF-based post-processing requires additional 0.8 second per image. Experimental results will show that DCL alone without CRF-based post-processing already outperforms existing state-of-the-art methods.

5.2. Comparison with the State of the Art

We compare our saliency models (DCL and DCL$^+$) against eight recent state-of-the-art methods, including SF [38], GC [7], DRFI [20], PISA [42], BSCA [39], LEGS [43], MC [49] and MDF [25]. The last three are the latest deep learning based methods. For fair comparison, we use either the implementations or the saliency maps provided by the authors. In addition, we also train a fully convolutional neural network (FCN) (the FCN-8s network proposed in [30]) for comparison. To train the FCN saliency model, we simply replace its last softmax layer with a sigmoid cross-entropy layer for saliency inference, and fine-tune the revised model using the training sets in the aforementioned saliency datasets.

A visual comparison is shown in Figure 4. As can be seen, our method generates more accurate saliency maps in various challenging cases, e.g., objects touching the image boundary (the first two rows), multiple disconnected salient objects (the middle two rows) and low contrast between object and background (the last two rows). It is necessary to point out that the performance of MC [49] is overrated on the MSRA-B dataset and the performance of LEGS [43] is overrated on both the MSRA-B dataset and the PASCAL-S dataset because most testing images in the corresponding datasets were used as training samples for the publicly released trained models of MC and LEGS used in our comparison.

Our method significantly outperforms all existing salient object detection algorithms across the aforementioned public datasets in terms of PR curve (Fig. 5) and average precision, recall and F-measure (Fig. 6). Refer to the supplemental materials for the results on the PASCAL-S and SOD datasets. Moreover, we report a quantitative comparison w.r.t. maximum F-measure and MAE in Table 1. Our complete model (DCL$^+$) improves the maximum F-measure achieved by the best-performing existing algorithm by 3.5%, 5.0%, 7.7%, 7.6% and 6.0% respectively on MSRA-B (skipping MC and LEGS on this dataset), HKU-IS, DUT-OMRON, PASCAL-S (skipping LEGS on this dataset) and SOD. And at the same time, our model lowers the MAE by 28.8%, 35.5%, 9.1%, 25.5% and 18.7% respectively on MSRA-B (skipping MC and LEGS on this dataset), HKU-IS, DUT-OMRON, PASCAL-S (skipping LEGS on this dataset) and SOD. We can also see that our model without CRF (DCL) significantly outperforms all evaluated salient object detection algorithms across all the considered datasets. Our model also significantly outperforms the FCN adapted from a model originally designed for semantic segmentation [30] because we explicitly perform deep contrast learning, which is critical for saliency detection.

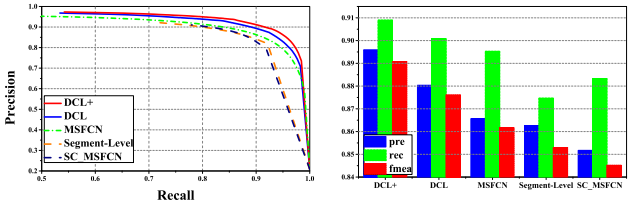

Figure 7. Componentwise efficacy of the proposed deep contrast network and the effectiveness of the CRF model.

5.3. Ablation Studies

5.3.1 Effectiveness of Deep Contrast Network

Our deep contrast network consists of a fully convolutional stream and a segment-wise spatial pooling stream. To show the effectiveness and necessity of these two components, we compare the saliency map S_1 generated from the first stream (MS-FCN), the saliency map S_2 from the second segment-level stream and the fused saliency map from S_1 and S_2 (DCL) using testing images in the MSRA-B dataset. As shown in Fig. 7, the fused saliency map (DCL) consistently achieves the best performance on average precision, recall and F-measure, and the fully convolutional stream (MS-FCN) has more contribution to the fused result than the segment-wise spatial pooling stream. These two streams are complementary to each other, and our trained deep contrast network is capable of discovering and understanding subtle visual contrast among multi-scale feature maps as well as between neighboring segments. To demonstrate the effectiveness of MS-FCN, we also generate saliency maps from the last scale of MS-FCN (the best performing scale) for comparison. The last scale of MS-FCN is in fact the fully convolutional version of the original VGG16 network. As shown in Fig. 7, this single scale of MS-FCN (called SC_MSFCN) performs much worse than the complete version of MS-FCN in terms of the PR curve as well as the average precision, recall and F-measure.

5.3.2 Effectiveness of CRF

In Section 4.2, a fully connected CRF is incorporated to improve the spatial coherence of the saliency maps from our deep contrast network. To validate its effectiveness, we have also evaluated the performance of our final saliency model with and without the CRF using the testing images in the MSRA-B dataset. The results are also shown in Fig. 7. It is evident that the CRF improves the accuracy of our model.

6. Conclusions

In this paper, we have introduced an end-to-end deep contrast network for salient object detection. Our deep network consists of two complementary components, a pixel-level fully convolutional stream and a segment-level spatial pooling stream. A fully connected CRF model can be optionally incorporated to further improve spatial coherence and contour localization in the fused result from these two streams. Experimental results demonstrate that our deep model can significantly improve the state of the art.

Acknowledgment

The first author is supported by Hong Kong Postgraduate Fellowship.

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *Computer vision and pattern recognition, 2009. cvpr 2009.*

ieee conference on, pages 1597–1604. IEEE, 2009. 2, 7

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels. Technical report, 2010. 5

[3] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. In *ACM Transactions on graphics (TOG)*, volume 26, page 10. ACM, 2007. 1

[4] S. Bi, G. Li, and Y. Yu. Person re-identification using multiple experts with random subspaces. *Journal of Image and Graphics*, 2(2), 2014. 1

[5] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai. Fusing generic objectness and visual saliency for salient object detection. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 914–921. IEEE, 2011. 2

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014. 2, 3, 4

[7] M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S. Hu. Global contrast based salient region detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(3):569–582, 2015. 1, 2, 7

[8] A. Criminisi, T. Sharp, C. Rother, and P. Pérez. Geodesic image and video editing. 5

[9] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3992–4000, 2015. 4

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 4

[11] W. Einhäuser and P. KoÈnig. Does luminance-contrast contribute to a saliency map for overt visual attention? *European Journal of Neuroscience*, 17(5):1089–1097, 2003. 1

[12] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1915–1929, 2013. 2

[13] D. Gao and N. Vasconcelos. Bottom-up saliency is a discriminant process. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–6. IEEE, 2007. 2

[14] R. Girshick. Fast r-cnn. In *International Conference on Computer Vision (ICCV)*, 2015. 4

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014. 2

[16] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *TPAMI*, 34(10):1915–1926, 2012. 2

[17] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Computer Vision–ECCV 2014*, pages 346–361. Springer, 2014. 4

[18] Y. Jia and M. Han. Category-independent object-level saliency detection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1761–1768. IEEE, 2013. 2

[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014. 7

[20] P. Jiang, H. Ling, J. Yu, and J. Peng. Salient region detection by ufo: Uniqueness, focusness and objectness. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1976–1983. IEEE, 2013. 1, 2, 6, 7

[21] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *ICCV*, 2009. 2

[22] D. Klein, S. Frintrop, et al. Center-surround divergence of feature statistics for salient object detection. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2214–2219. IEEE, 2011. 2

[23] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *arXiv preprint arXiv:1210.5644*, 2012. 5, 7

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[25] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1, 2, 4, 5, 6, 7

[26] H. Li, R. Zhao, and X. Wang. Highly efficient forward and backward propagation of convolutional neural networks for pixelwise classification. *arXiv preprint arXiv:1412.4526*, 2014. 3

[27] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014*

IEEE Conference on, pages 280–287. IEEE, 2014. 2, 5, 6

[28] R. Liu, J. Cao, Z. Lin, and S. Shan. Adaptive partial differential equation learning for visual saliency detection. In *CVPR*, 2014. 2

[29] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(2):353–367, 2011. 1, 2, 5

[30] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2014. 2, 3, 4, 7, 8

[31] S. Lu, V. Mahadevan, and N. Vasconcelos. Learning optimal seeds for diffusion-based salient object detection. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2790–2797. IEEE, 2014. 1

[32] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li. A user attention model for video summarization. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 533–542. ACM, 2002. 1

[33] L. Mai, Y. Niu, and F. Liu. Saliency aggregation: a data-driven approach. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1131–1138. IEEE, 2013. 1

[34] S. Mallat. *A wavelet tour of signal processing*. Academic press, 1999. 3

[35] 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 *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 416–423. IEEE, 2001. 5

[36] V. Navalpakkam and L. Itti. An integrated model of top-down and bottom-up attention for optimizing detection speed. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2049–2056. IEEE, 2006. 1

[37] D. Parkhurst, K. Law, and E. Niebur. Modeling the role of salience in the allocation of overt visual attention. *Vision research*, 42(1):107–123, 2002. 1

[38] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 733–740. IEEE, 2012. 2, 7

[39] Y. Qin, H. Lu, Y. Xu, and H. Wang. Saliency detection via cellular automata. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 110–119, 2015. 7

[40] X. Shen and Y. Wu. A unified approach to salient object detection via low rank matrix recovery. In *CVPR*, 2012. 2

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

[42] K. Wang, L. Lin, J. Lu, C. Li, and K. Shi. Pisa: Pixelwise image saliency by aggregating complementary appearance contrast measures with edge-preserving coherence. *Image Processing, IEEE Transactions on*, 24(10):3019–3033, Oct 2015. 7

[43] L. Wang, H. Lu, X. Ruan, and M.-H. Yang. Deep networks for saliency detection via local estimation and global search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3183–3192, 2015. 1, 2, 7, 8

[44] P. Wang, G. Zeng, R. Gan, J. Wang, and H. Zha. Structure-sensitive superpixels via geodesic distance. *International journal of computer vision*, 103(1):1–21, 2013. 5

[45] R. Wu, Y. Yu, and W. Wang. Scale: Supervised and cascaded laplacian eigenmaps for visual object recognition based on nearest neighbors. In *CVPR*, 2013. 1

[46] S. Xie and Z. Tu. Holistically-nested edge detection. *arXiv preprint arXiv:1504.06375*, 2015. 2

[47] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and Y. Yu. Hd-cnn: Hierarchical deep convolutional neural networks for large scale visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2740–2748, 2015. 2

[48] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3166–3173. IEEE, 2013. 1, 2, 5

[49] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1265–1274, 2015. 1, 2, 7, 8

[50] W. Zhu, S. Liang, Y. Wei, and J. Sun. Saliency optimization from robust background detection. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2814–2821. IEEE, 2014. 2