Regarding Salient Object Segmentation

Anonymous CVPR submission

Paper ID The Usual Segments

Abstract

Salient object detection and segmentation in natural scenes has attracted much attention and achieved considerable progress in the past three decades. In this paper we review three recent CVPR papers, focusing on different aspects of the salient object segmentation problem.

This paper is divided into two parts. In the first part, we introduce the main claims and statements of the papers, and explain the main ideas of the algorithms. We then discuss the result of experiments conducted in the papers and point out the advantages and limitations as we aim to provide a comprehensive review of solutions to the salient object segmentation problem. In the second part of this paper, we present our replication of the work by Frintrop et al. [1]. We describe our implementation of their work in detail and evaluate it on similar benchmarking datasets.

1. Introduction

Salient object detection in computer vision started with the iNVT model by Itti *et al.* [2] in 1998. The goal of this method was to segment and detect the 'salient' regions of interest. Since then, many new algorithms have been proposed. There are two types of salient object segmentation algorithms: *top-down* methods that are task and target driven, and *bottom-up* methods that are data and feature driven. Bottom-up visual saliency uses low-level cues embedded in the image itself, such as color and intensity. While bottom-up saliency methods are effective, a high-level understanding of the image content is needed in some cases, as discussed in [3].

In this paper, we review three papers presented at CVPR in year 2014 and 2015. Frintrop *et al.*'s work [1] is based on the original saliency method by Itti *et al.* [2]. They modify this biologically-inspired saliency algorithm, achieving comparable performance to current state-of-the-art methods. Greatest performance increases were seen when applying twin scale-space pyramids and an appropriate center-surround ratio. This suggests that most bottom-up methods primarily involve a form of contrast computation.

Saliency propagation method is another popular bottom-up method, where the image is represented as a graph of superpixels. Saliency values are iteratively diffused from the superpixels with known saliency values to unlabeled ones. Chen *et al.* [4] propose a novel scheme to perform propagation using a 'starting simple' strategy. The unlabeled superpixels are ordered by difficulty, and saliency values are then propagated from the easier superpixels to more difficult ones, as determined by prior performance. This approach takes advantage of the knowledge obtained from labeling easy regions so that more difficult regions can be more precisely labeled.

The work of Yin *et al.* [5] focuses on benchmarking state-out-the-art bottom-up saliency methods on common datasets. Their experiments identify design bias in existing benchmarking datasets. Dataset design bias can mislead algorithm design. The authors analyze the causes of design bias and propose a new less-biased dataset, on which they build a novel model that performs well.

2. Related Work

In this section, we review three papers focusing on salient object segmentation presented at CVPR. We will explore the problem, claims, core ideas, advantages and limitations of their experiments.

2.1. Paper 1: Feature Integration Theory based Method

Feature integration theory (FIT) [17] claims that many features are processed in parallel in different areas of the brain to obtain a saliency map. FIT methods simulate eye movements, but they are limited for salient object detection and segmentation. Therefore, many other less biologically-inspired saliency systems have been proposed to address this.

In this paper, Frintrop *et al.* [1] explore the problem of salient object detection. More specifically, are there essential ideas that the biologically-inspired iNVT model is missing? The authors claim that the underlying nature of most saliency systems is contrast computation, and they hypothesize that the original FIT model, iNVT, proposed by Itti *et*

al. [2] is still comparable with current state-of-the-art methods if certain changes are made. The authors claim that this modified algorithm is robust enough to handle various benchmarks, works for real-time applications, and is easy to understand how parameters affect performance. Furthermore, the authors claim that if the modified model performs comparably to other top saliency methods, then the iNVT model did not miss any essential ideas about object saliency.

The improved model, called VOCUS2, has two color and one intensity input channels, and it involves two important changes: twin pyramids (one center pyramid and one surround pyramid) and a flexible center-surround ratio. For each pyramid, a scale-space structure is used as in [6], instead of using simple Gaussian pyramid. By subtracting between pyramids, the ratio of center to surround is not limited to a multiple of two as in iNVT. Another important aspect is equally fusing the feature maps for each channel into one saliency map. Segmentation can be added to achieve better performance. This algorithm is further explored and reproduced in later sections.

One advantage of VOCUS2 is its simplicity and clarity, since it is mostly convolution with various Gaussian filters. Simple structure leads to low computational complexity, which means VOCUS2 can run fast and is applicable to mobile devices. Furthermore, the main idea of VOCUS2 is biologically inspired, and there are few parameters to tune. Another advantage is the ability to tune the center-surround ratio to better fit the image instead of being limited to multiples of two. This study showed that the correct center-surround ratio greatly improved performance. Also, equally treating the feature maps from each channel during fusion does not bias the method to a training dataset.

However, there are some limitations for this study of VOCUS2. Since this method is based on low-level image cues, some 'background' objects with sharp color contrast may be labeled as salient objects (false positives), and salient objects with low contrast might not be detected (false negatives). This limitation is common for bottom-up saliency computation method due to the lack of high-level understanding of the input image. Learning-based algorithms might help improve performance. Another limitation of the study is the use of a location prior to assume where salient objects will be. This likely overfits the typical benchmarking datasets, in which salient objects tend to be centered. Additionally, use of segmentation to improve accuracy is limited to the performance of the segmentation used.

Frintrop *et al.* performed multiple evaluations on their algorithm. They evaluated the impact on performance for each modification from iNVT to VOCUS2. The results show that the biggest increase in performance is obtained by introducing twin pyramids structure and adapting center-surround ratio to the image. Since this twin pyramid was used to compute contrasts, this finding validates the authors' claim that the saliency computation problem is primarily contrast. Good results are obtained when the contrast is properly computed. VOCUS2 is further evaluated by comparing with other popular saliency methods. The comparison result shows that the segment-based version of VOCUS2 outperforms all other methods under weighted F-measure on five image datasets and is competitive with other methods as measured by area under the curve (AUC) of the recall-precision curve. These evaluations validated the authors' claim that biologically-inspired iNVT can perform well with some adjustments, even though some of these changes remove biological aspects from iNVT, such as using orientation as an input.

However, their evaluation also showed that another method (DRFI [7]) has a similar f-measure as VOCUS2 while its AUC is 14% more than that of the segment-based VOCUS2. Other aspects of the algorithm should be compared to determine which algorithm performs better. Also, the authors evaluate VOCUS2 on mobile devices by using the CMS dataset [8] and comparing performance of various algorithms. VOCUS2 greatly outperforms other methods. Yet, the CMS dataset contains about 600 frames, which is not much data to conclude that VOCUS2 performs the best. Also, the authors did not explain why the performance drops for all algorithms on the CMS dataset. This paper showed a simple and well-evaluated method for performing bottom-up saliency. The other methods look into different methods for computing bottom-up saliency for object detection.

2.2. Paper 2: Saliency Propagation based Method

Saliency propagation is commonly used in bottom-up saliency computation. Before propagation, the image is segmented into superpixels to form a graph. Prior knowledge is used to initialize saliency value of selected superpixels known as 'seed points'. Next, saliency values are iteratively propagated from labeled superpixels to their unlabeled neighbors. Propagation can be performed by various algorithms, including Random Walk, personalized PageRank [9] and manifold based diffusion [9].

However, saliency propagation is strongly affected by the spatial relationship of the superpixels and may result in inaccurate propagation may occur for inhomogeneous superpixels. Chen *et al.* [4] investigate this problem and propose a solution. They hypothesize that propagation can be optimized by using a 'starting simple' method to remove the effects of inhomogeneous superpixels. This strategy directs the propagation sequence according to the difficulty of labeling a superpixel instead of spatial relationship, with previously learned knowledge helping to improve labeling accuracy for the difficult superpixels. The authors use a novel teaching-to-learn and learning-to-teach scheme to iteratively update the propagation as difficulty increases.

In the algorithm proposed by Chen *et al.*, the input image is segmented into superpixels using SLIC [10], and the Harris corner detector estimates the target object's location by constructing a convex hull. Seed points are initialized by applying background and boundary priors. Then, the difficulty of labeling a certain superpixel is evaluated by *informativity* (the 'entropy' of the superpixel), *individuality* (the contrast to its surrounding superpixels), *inhomogeneity* (whether a superpixel is ambiguous) and *connectivity* (strength of similarity to labeled superpixels). In the teaching-to-learn step, the 'teacher' decide the which superpixels to be labeled based on unlabeled superpixel difficulties and the performance of the 'learner' in the last iteration. In the learning-to-teach step, the 'learner' provides a feedback to the 'teacher'. The process repeats until all superpixels are propagated.

This 'starting simple' strategy is a novel approach for saliency propagation with intuitive ideas rooted in psychology. It also involves well-studied diffusion laws from physics. Another advantage of this method is that it results in cleaner saliency maps, while minimally affecting how saliency propagation is performed. Furthermore, this algorithm automates a process that would normally be manually performed by looking at four attributes of each superpixel. However, this method is limited by the performance of segmentation result and prior knowledge used to initialize the seed points. Most propagation-based methods suffer from this disadvantage since they use segmentation and prior knowledge to initialize seed points before performing propagation. Two other limitations in performance arise when the salient object looks similar to the background or if the salient object is not located within the convex hull. So, the performance is sometimes limited by the assumptions used. Another limitation of the method is its long computation time, especially to achieve better results.

The experiments validated the effectiveness of the proposed algorithm. It outperforms other algorithms under the measurement of weighted precision, recall and f-measure. Parametric sensitivity was examined, indicating that the proposed algorithm has few parameters to adjust, meaning it is robust. The experiments showed how the Gaussian kernel width, the main parameter to tune, affects performance. The authors' claim that their 'starting simple' strategy would reduce background clutter and compare favorably to other methods was validated. However, the authors did not explore what parts of this new algorithm contributed most to the improved performance and did not optimize the code for efficiency. Like the other two papers, this study focused on computing saliency maps using a bottom-up approach, but this method converts the image into a graph and computes saliency using iterative propagation, providing an interesting contrast to the methods and image representations presented in the other papers.

2.3. Paper 3: Fixation Prediction and Dataset Bias

Saliency computation has two specific tasks: fixation prediction and salient object segmentation. In fixation prediction, subjects' eye fixations are recorded and the goal is to predict human eye gaze. In salient object segmentation, algorithms generate a pixel-level saliency map that matches a manually labeled salient object mask.

In Yin *et al.* [5], the authors claim that existing saliency algorithms focus on only one of these tasks and overlook the other. This can become a problem that leads to biased datasets, which can lead to false results when testing algorithms and can also bias algorithm design due to overfitting. The authors also claim to have created a saliency algorithm that performs comparably with other common methods as well as a dataset that is less biased.

Yin *et al.* explore the connection between fixation prediction and salient object segmentation using a novel datset with both features annotated. They showed that human subjects are consistent in defining salient objects. In the study, seven algorithms were benchmarked on five widely used datasets, and they found that many of these datasets were biased, as determined by four assessment indicators (*local color contrast*, *global color contrast*, *local gPB boundary strength* and *object size*). The authors hypothesized that using the correlation between fixations and salient objects can generate datasets and algorithms with less dataset design bias.

The core idea of the proposed algorithm is to obtain a set of object candidates using unsupervised segmentation algorithm and then use fixation data to evaluate the saliency of object candidates. Based on evaluation on multiple datasets, the proposed algorithm performs comparably with other salient object detection algorithms. Furthermore, the PASCAL-S dataset proposed by the authors proved to be unbiased and consistent according to evaluation in the paper. These results validate the authors' claims about their novel method and dataset.

An advantage of this study is that the benefits of using fixation prediction with segmentation are clearly shown. This provides a way to avoid biasing data and results. However, the algorithm requires human fixation data, which is not always available. Another limitation is that the algorithm only performs as well as the segmentation used. Furthermore, if the algorithm generates a fixation pattern, the run time may be quite long. Therefore, evaluation and comparison of the run time of this model should be evaluated on datasets besides PASCAL-S. As with the first two papers, this method focuses on bottom-up salient object detection. However, it focuses on inherent biases in datasets and how utilizing eye fixation patterns can reduce this bias. All three papers provide different perspectives on salient object detection, spanning a variety of features and image representations to achieve better performance.

3. Implementation of VOCUS2

We introduced the core ideas of VOCUS2 in Sec. 2.1. In this section we implement the VOCUS2 algorithm proposed by Frintrop *et al.* [1] based on the traditional FIT-based model [2]. Fig. 1 shows an overview of our VOCUS2 implementation. We first describe each step of this structure compare it with [2]. In Sec. 3.2 we add in local prior knowledge and then add pre-segmentation in Sec. 3.3.

3.1. Basic Structure of VOCUS2

3.1.1 Feature Channels

Since color is important to visual attention, the image is converted into 3 channels: intensity, red vs. green and blue vs. yellow. The color space of input image can be denoted as $(R,\ G,\ B)$, so the intensity channel is computed by $I = \frac{1}{3}(R+G+B)$, the red-green channel by $RG = R-G$ and blue-yellow channel by $BY = B - \frac{R+G}{2}$.

3.1.2 Scale Space Pyramids

In FIT-based saliency models, an appropriate computation of center-surround contrast is key to obtaining accurate saliency maps. A sophisticated scale space twin pyramid is used in VOCUS2 to allow for a flexible center-surround contrast ratio tailored to each image. The twin pyramid consists of a center pyramid and a surround pyramid. Both of the pyramids have multiple layes (octaves) and multiple scales in each layer, as in [6]. In our implementation, using 6 layers with 2 scales per layer achieved qualitatively similar saliency maps to [1].

To create the center pyramid for each channel, each channel image is convolved with a gaussian defined by σ_c (the center part of the center-surround ratio). For each layer, the image in layer l at scale s is down-sampled by factor of 2^{l-1}. The surround pyramid is then directly computed from the by smoothing each image in the center pyramid by a gaussian defined by the center-surround ratio $\sigma_x = \sqrt{\sigma_c^2 + \sigma_s^2}$. The advantage of using twin pyramids is that the center-surround ratio can be fine tuned to fit the image. Frintrop *et al.* showed that this twin peak adaption was the most significant improvement by VOCUS2 over [2].

3.1.3 Contrast Pyramid

After computing the center and surround pyramids, the contrast pyramids for each channel can be computed. The contrast computation is divided into two parts: on-off contrast $\mathbf{X} = \mathbf{C} - \mathbf{S}$ and off-on contrast $\mathbf{Y} = \mathbf{S} - \mathbf{C}$ for all 3 color channels. The on-off contrast and off-on contrast correspond to bright objects on a dark background and a dark object on a bright background, respectively. This corresponds to two types of cells involved in human vision.

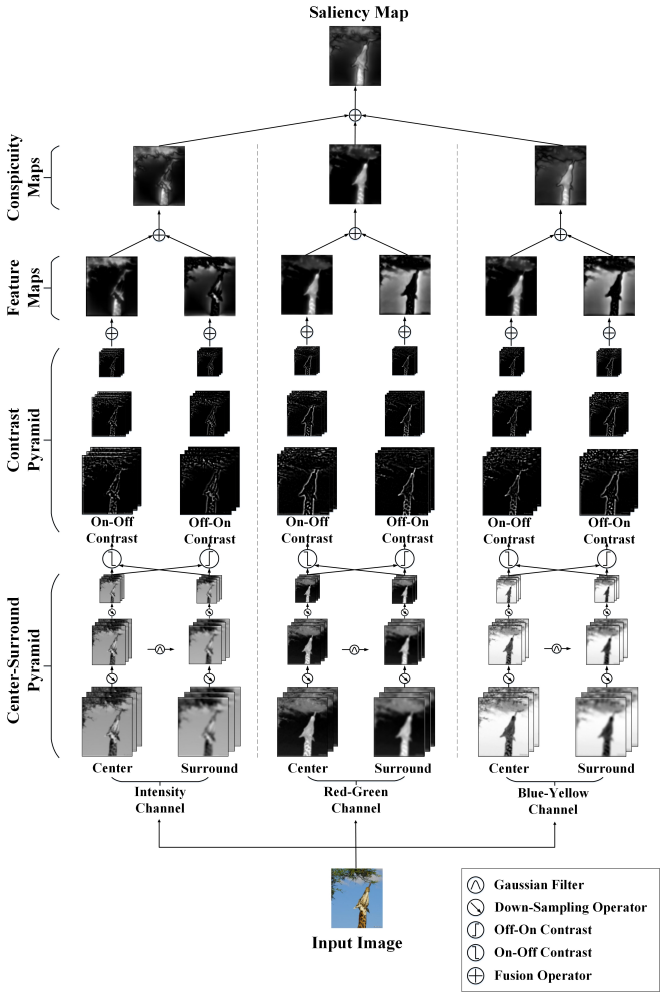

Figure 1. Overview of the basic structure of VOCUS2.

3.1.4 Feature Fusion

To obtain the final saliency map, images in the contrast pyramids need to be fused together. For each contrast pyramid, the images are up-sampled to the finest resolution and added up across different layers and scales. Thus, we have six feature maps (two types of contrast for each of the three channels). Next, each channel's feature maps are fused to obtain a conspicuity map, and the final saliency map is generated by fusing the three conspicuity maps.

There are various options for the fusing operation, such as non-linear weighting used in [2]. However, we followed the VOCUS2 alogirthm and used the arithmetic mean to fuse the maps. This treats the three color channels equally. This is the full method for our implementation of the basic VOCUS2 method. Further extensions of this basic method are explored in the next sections.

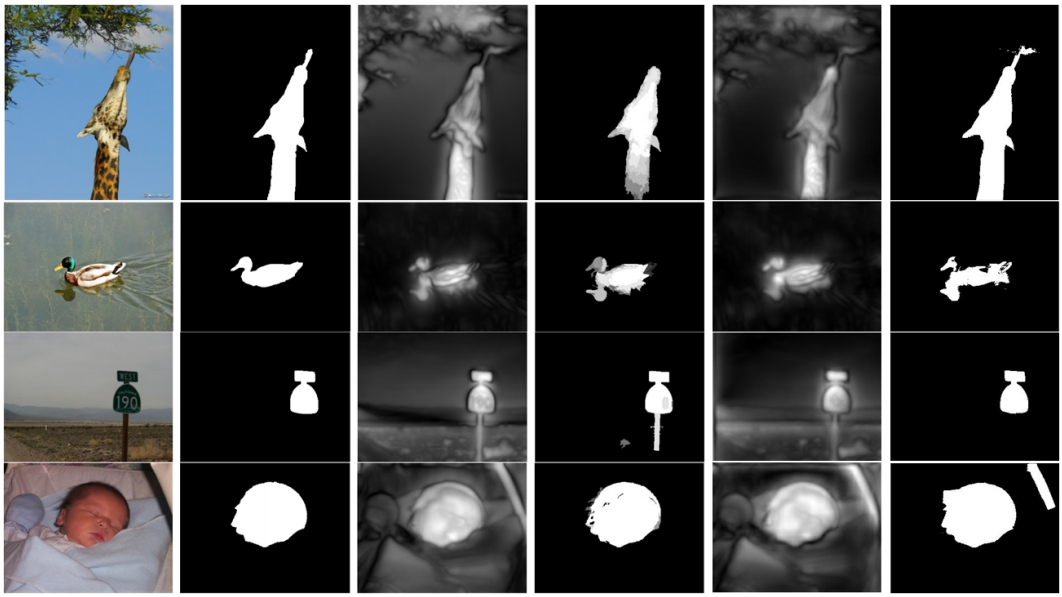

Figure 2. Comparison of example saliency maps. From left to right: Original input image, ground truth, basic saliency map of VOCUS2 in the paper, segment-based saliency map of VOCUS2 in the paper, basic saliency map of our implementation, segment-based saliency map of our implementation

3.2. Location Prior

In order to deal with arbitrary input image, a robust saliency computation model should not rely on prior location knowledge. However, according to Yin *et al.* [5], many popular datasets for saliency computation are biased toward image center. To achieve better performance on current benchmarking datasets, an optional center bias term is introduced:

$$G_{lp}(x,y) = \exp(-\frac{d^2}{2\sigma^2}) \quad (1)$$

where $d = \|(x,y) - (x_c,y_c)\|$ is the distance between the pixel (x,y) and the image center (x_c,y_c). Therefore, the center biased saliency map $\hat{s}(x,y)$ is computed as

$$\hat{s}(x,y) = s(x,y) \cdot G_{lp}(x,y) \quad (2)$$

where $s(x,y)$ is the original saliency map. Implementing a location prior allowed us to further compare model performance with the original VOCUS2 model in Sec. 4.

3.3. Segment Based Method

So far, this model has created pixel-level precise saliency maps, but for some applications, it is necessary to compute an accurate object boundary. By combining an image segmentation algorithm and the pixel-level precise saliency map computed so far, a segment-based saliency map can be computed.

For our implementation, we over-segmented the input image using SLIC, which was shown in [10] to perform better than Mean Shift, used by Frintrop *et al.*. Our implementation over-segments the image into 128 segments. To improve future run-time, saliency map computation could run in parallel with segmentation. After segmentation, local saliency map maxima are detected and are considered as seed points for the following region growing process. Region growing is done according to the saliency value, and limited by threshold corresponding to the saliency value of the seed points (i.e. local maxima). In this case we used a threshold of 20 percent similarity. For each grown region, we select the segments that overlap more than 30 percent with the region to form a proposal, following what Frintrop *et al.* do in their work. The proposals are then ranked based on their average saliency and the top 50 percent of the proposals are selected to obtain the final saliency map. This method combines the segmentation result and the pixel-level precise saliency map to generate a segment-based one with much more accurate object boundary.

4. Evaluation and Discussion

In this section, we evaluate the performance of our implementation of VOCUS2 using the same evaluation measures and datasets used by Frintrop *et al.* Two different measurements are presented: recall-precision curves and weighted f-measure. We also discuss some failed cases and the impact of tuning model parameters.

Performance was evaluated on five datasets: MSRA-10k [14], SED1 and SED2 [15], ECSSD [16] and PASCAL-S [5], which are the same datasets Frintrop *et al.* used in their work. Note that PASCAL-S is the dataset proposed in the third paper we critiqued, which claims to be more unbiased. We also use the same parameters that are mentioned

in the paper: the twin Gaussian pyramids have 6 layers with 2 scales in each layer, and the center-surround ratio is set to be $3:13$. Some example saliency maps are shown in Fig. 2.

4.1. Precision-Recall Curve Measurement

Since the saliency map is pixel-precise, a popular measure is thresholding the saliency map with an increasing value $k \in [0, 255]$. This creates a sequence of binary saliency maps that are then compared with the ground truth, generating recall and precision values. Fig. 3 shows the recall-precision curve for our implementation compared to [1] averaged across all five datasets. It can be seen that our implementation is performs similarly to [1], justifying the correctness of our implementation.

4.2. Weighted f-measure Measurement

Although the recall-precision measurement is still commonly used, it does have limitations [18]. The weighted f-measure has been recently proposed to overcome these flaws, so we also evaluated out implementation using this measure.

In [1], the weighted f-measure was averaged across all images from the five datasets, achieving an approximate value of 0.3200. Evaluating this same measure on our implementation, resulted in a score of 0.3287, which is quite similar. To achieve this average, the weighted f-measure score was averaged across all images in all five datasets. Average performance for each dataset varied from 0.2720 on the ECSSD dataset to 0.5026 on the SED1 dataset. Adding in the center bias, the weighted f-measure score improved to 0.3800, which is close to the 0.3500 score seen in [1]. Our implementation appears to perform consistent with [1], with some slight discrepancies. However, these slight differences in performance are similar to the differences seen when the authors replicated the original model from Itti *et al.* [2]. Slight differences can arise due to recent updates in the datasets themselves.

Please note that we compared our implementation with that in the VOCUS2 paper to justify our method's correctness. Comparisons with other saliency computation methods can be found in [1].

4.3. Discussion on several interesting cases

After verifying the accuracy of our implementation, we ran it on various images to understand the limitations of VOCUS2. For a few images, the segment-based saliency map was clearly incorrect while the pixel-precise saliency map appeared more accurate. This was found to have been caused by inaccurate segmentation that led to some segments being mistakenly classified as background. From this, it is clear that segmentation performance strongly influences the final saliency map as previously mentioned

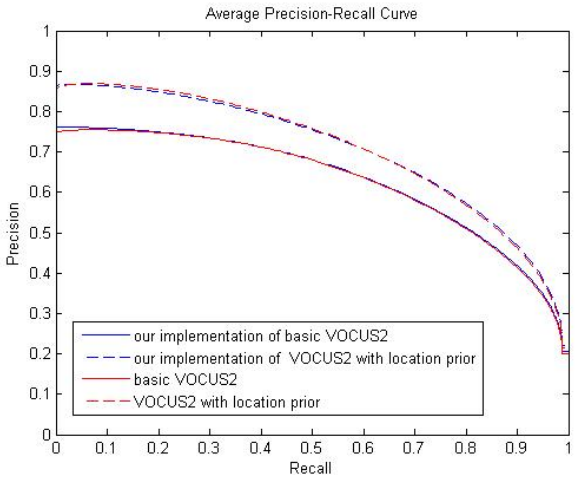

Figure 3. Precision-recall curves. The results are averaged over MSRA-10k, SED1, SED2, ECSSD, and PASCAL-S datasets.

in Sec. 2.2. However, Frintrop *et al.* [1]showed that VOCUS2 with segmentation outperformed the basic and location prior implementations for weighted f-measure and AUC for recall-precision curves, so segmentation appears to provide tangible overall benefits despite limitations for some images.

Another limitation we found was not detecting parts of salient objects as salient due to low contrast with the background. For example, the lower body of a person in one image was not determined to be salient, while the upper part was found computed to be salient. However, they are both part of the person, which was the salient object in the image. VOCUS2 does not have high-level information about the image to determine this, though. This shows that a major limitation of VOCUS2, and bottom-up saliency computation in general, is a lack of high-level understanding of the image components.

5. Conclusion

In the first part of this paper, three recent CVPR papers were critiqued. All three studies focused on bottom-up saliency computation for object detection, by either contrast computation, propagation across nodes in a graph, or using subject fixation patterns. In the second part, we describe our implementation of VOCUS2. Our algorithm was then evaluated using recall-precision curves and the weighted f-measure as in the paper by Frintrop *et al.* [1]. Similarities between our evaluation and the papers suggest that our implementation is equivalent. After evaluation, limitations of pre-segmentation and bottom-up methods in general are explored.

References

[1] S. Frintrop, T. Werner, and G. M. García. Traditional Saliency Reloaded: A Good Old Model in New Shape. In *CVPR*, pages 82-90. IEEE, 2015.

[2] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. In *TPAMI, IEEE*, 20(11):1254-1259, 1998.

[3] J. Yang and M. Yang. Top-down visual saliency via joint CRF and dictionary learning. In *CVPR*, pages 2296-2303. IEEE, 2012.

[4] C. Gong, D. Tao, W. Liu, S.J. MayBank, M. Fang, K, Fi and J. Yang. Saliency Propagation from Simple to Difficult. In *CVPR*, pages 2531-2539. IEEE, 2015.

[5] Y. Li, X. Hou, C. Koch, J.M. Rehg and A.L. Yuille. The Secrets of Salient Object Segmentation. In *CVPR*, pages 4321-4328. IEEE, 2015.

[6] D. G. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision (IJCV)*, 60(2):91-110, 2004.

[7] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *CVPR*, pages 2082-2090. IEEE, 2013.

[8] S. Frintrop, G. M. García, and A. B. Cremers. A cognitive approach for object discovery. In *CVPR*, pages 2329 - 2334. IEEE, 2014.

[9] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In *Advances in Neural Information Processing Systems*, 16: 169-176, 2004.

[10] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC: superpixels compared to state-of-the-art superpixel methods. In *TPAMI*, 34(11): 2274-2282. IEEE, 2012.

[11] A. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tunned salient region detection. In *CVPR*, pages 1597-1604. IEEE, 2009.

[12] J. M. Wolfe and T. S. Horowitz. What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience*, (5):1-7, 2004;

[13] L. Hurvich and D. Jameson. An opponent-process theory of color vision. *Psychological review*, 49(10), 2009.

[14] M. M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S. M. Hu. Global contrast based salient region detection. In *TPAMI*, 37(3): 569-582. IEEE, 2015.

[15] A. Alpert, M. Galun, R. Basri, and A. Brandt. Image segmentation by probabilistic bottom-up aggregation and cue integration. In *CVPR*, pages 1-8. IEEE, 2007.

[16] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical Saliency Detection. In *CVPR*, pages 1155-1162. IEEE, 2013.

[17] A. M. Treisman and G. Gelade. A feature integration theory of attention. In *Cog. Psych*, 12, 1980.

[18] R. Margolin, L. Zelnik-Manor, and A. Tal. How to evaluate foreground maps? In *CVPR*, pages 248-255. IEEE, 2014