Efficient Saliency Aggregation via Soft-Voting Evolution

Ling Zheng ^{a,b*}, Shuhan Chen ^b, Xuelong Hu ^b, Lifeng Zhang ^a

^a Kyushu Institute of Technology

1-1, Sensui, Tobata, Kitakyushu, Japan 804-8550

^b Yangzhou University

196, Huayangxilu, Yangzhou, Jiangsu, China 225-127

*Corresponding Author: zlapgx@163.com

Abstract

Efficient saliency detection algorithm plays an important role in computer vision tasks, such as object recognition, visual tracking, image compression, image segmentation and so on. A variety of saliency detection methods have been proposed in recently, which often complement each other. In this paper, we propose a soft-voting evolution saliency aggregation algorithm which combines them together. First of all, we set a threshold TH which is used to segment the background and foreground region. Then, each map can be seen as a voter, and it is voted as foreground when the saliency value is greater than TH. In contrast, it is voted as background when the value is less than TH. Different with previous works, we define a soft-voting weight based on sigmoid function. Besides, iterative operation is necessary to further improve performance. Experiments on three publicly available datasets demonstrate that the proposed algorithm performs favorably against the state-of-the-art methods.

Keywords: Saliency detection, Saliency aggregation, Sigmoid function, Soft-voting Evolution.

1. Introduction

Recent years have witnessed rapidly increasing interest in salient object detection [1]. Saliency detection aimed at detecting the most important part of an image has become a very active topic in computer vision, which measures the low-level stimuli that grabs viewers' attention in the early stage of human vision [2], whose research mainly contains three aspects: eye fixation prediction [3], saliency detection [4], and objectness estimation [5]. In this paper, we focus on saliency detection which aims to make certain regions of an

Fig.1. Individual saliency methods often complement each other. Saliency aggregation can effectively outperform each one of them.

image stand out from their neighbors. It is still challenging to develop efficient saliency detection algorithms which play an important role in the wide range of computer vision tasks, such as object recognition, visual tracking, image compression, image segmentation and so on.

Fortunately, many efficient saliency detection algorithms have been proposed in recent years, most of them can uniformly highlight the salient object in an image on public benchmarks, but not perfectly. When facing complex scenes there still exists a large margin from the ground truth. While each of them has its own advantages and disadvantages. More interestingly, different approaches can often be complementary to each other [6]. As illustrated by Fig.1, different saliency maps usually do not exhibit similar characteristics and each of them only works well for some images or some parts of the image[7], and none of them can handle all the cases, such as Fig.1 some saliency maps are

complementary in measuring saliency. Therefore, the aggregation of these saliency analysis results can outperform each individual one.

There are few literatures about saliency aggregation algorithms we can refer to. In [8], Li et al. proposed a saliency detection scheme whose final step is Bayes integration of two saliency maps generated by dense and sparse reconstruction errors respectively. Focusing on eye fixation prediction, Le Meur et al. Borji et al. [9] proposed two combining strategies: Naive Bayesian evidence accumulation and linear summation, which demonstrated aggregation results outperform than individual ones. Similar to this work in [6], Conditional Random Field was used for saliency aggregation, which not only models the contribution from individual saliency map but also the interaction between neighboring pixels. Regrettably, Conditional Random Field is time consuming due to the training and inference steps.

Totally speaking, individual saliency detection do not have well performance compared with the state-of-the-art. Therefore, there are still a big margin can be explored. Different with the aforementioned studies, we select top five state-of-the-art methods [8, 10] for aggregation and do the job on three benchmark datasets, and try to make the detected salient object more uniform and as close as possible to the ground truth.

In this paper, we present a novel Soft-voting Evolution (SVE) approach for saliency aggregation illustrated in Fig.2. Different with our previous work [7], we don't use Bayes updating as evolution input. We define a soft-voting weight for each saliency map based on sigmoid function for individual saliency map updating. Then, we do five iterations to further improve performance. Finally, our final aggregation result can be obtained by the average of all the evolution outputs.

Fig.2. Diagram of our proposed model.

The rest of the paper is organized as follows. Section 2 introduces three methods used for saliency aggregation, including linear aggregation (LA), Bayes aggregation and our SVE. Section 3 shows the experiment performance of different saliency aggregation approaches. Finally, conclusions are drawn in Section 4.

2. Saliency Aggregation

Firstly, we make some definition of the symbols which will be used in the following subsections. Let $\{S_i | 1 \leq i \leq m\}$ be the saliency maps generated by different saliency detection algorithms on a given image I , whose saliency value in each map is normalized to $[0, 1]$, G is the corresponding ground truth. Each element $S_i(z)$ in a saliency map denotes the saliency value at pixel z . Our goal is to aggregate these m saliency maps into a final saliency map which can outperform each individual one. In this section, we introduce different aggregation approaches including previous works and our proposed method. Among the previous works, we mainly focus the ones with high performance.

2.1 Linear Aggregation

The simplest aggregation scheme is linear summation which is defined as below:

$$S(z) = \frac{1}{Z} \sum_{i=1}^m w_i(z) \times S_i(z) \quad (1)$$

where w_i is the weighting coefficient, the sum of it is equal to 1 and $w_i \geq 0$. And Z is a constant for normalization in order to ensure the sum of w_i is equal to 1.

Based on this function, we can design various aggregation schemes only by varying the weighting coefficients. The most used is average weights which is uniform and spatial invariant, $w_i = 1/m$. It is verified by Borji [9] and Le Meur [10] and can produce satisfied aggregation results. Here, we modified it as:

$$w_i(z) = \frac{1}{\sum \|S_i(z) - S_p(z)\|^2} \quad (2)$$

It can be explained as that it should be given a large weight if it is more similar with the reference map which is defined as below:

$$S_p = \frac{1}{m} \sum_{i=1}^m S_i \quad (3)$$

2.2 Bayes Aggregation

In our previous work [7], we modified the original Bayes [8] to fit multiple saliency maps. In detail, given m saliency maps, we select S_p as the prior and use each individual one S_i to compute the likelihood.

Then S_p is threshold to obtain its background and foreground regions described by B_p and F_p , respectively. In each region, likelihoods are computed by comparing S_i and S_p in terms of the background and foreground bins at pixel z :

$$P(S_i(z) | B_p) = \frac{N_{b_{B_p}(S_i(z))}}{N_{B_p}}, P(S_i(z) | F_p) = \frac{N_{b_{F_p}(S_i(z))}}{N_{F_p}} \quad (4)$$

where N_{B_p} denotes the number of pixels in the background region and $N_{b_{B_p}(S_i(z))}$ is the number of pixels whose saliency value fall into the background bin $b_{B_p}(S_i(z))$,

while N_{F_p} and $N_{b_{F_p}(S_i(z))}$ are denoted for foreground region. Consequently, the posterior probability is computed as output:

$$SB_i(z) = \frac{S_p(z)P(S_i(z) | F_p)}{S_p(z)P(S_i(z) | F_p) + (1 - S_p(z))P(S_i(z) | B_p)} \quad (5)$$

2.3 Soft-Voting Evolution

In [19], Multi-layer Cellular Automata is proposed for multiple saliency map aggregation. The synchronous updating rule is defined as:

$$l(S_i^{t+1}) = l(S_i^t) + \sum_{k=1, k \neq i}^m \text{sign}(S_k^t - \lambda_k) \cdot 0.15 \quad (6)$$

in which l denotes natural logarithm. However, it is not efficient. In order to integrate the individual saliency map together, a hard-voting evolution method [7] has been proposed in our previous work. This idea is intuitive and easy to implement. However, it is quite rough to approximate the contribution of each individual component, and Bayes updating is also time consuming. Thus, we propose a soft-voting based saliency aggregation algorithm to improve it. First of all, we set a threshold TH which is used to segment the background and foreground region. Then, each map can be seen as a voter, and it is voted as foreground when the saliency value is greater than TH. In contrast, it is voted as background when the saliency is less than TH. In our study,

the threshold TH is simply generated by Otsu. In addition, each voter is given a weight λ , and λ is set between [0.1, 0.2]. To better capture the contribution of each saliency map, we introduce sigmoid function to define our soft-voting weight:

$$S_i^{t+1} = S_i^t + \sum_{k=1, k \neq i}^m \frac{1}{1 + \exp(-\Phi(S_k^t))} \lambda$$

$$\Phi(S_k^t) = \begin{cases} 10 \cdot \left(\frac{S_k^t}{T} - 1 \right), & S_k^t < T \\ 10 \cdot \frac{S_k^t - T}{1 - T}, & S_k^t \geq T \end{cases} \quad (7)$$

In order to get better performance, N iterations are needed. N is set to 5 in our experiment. The final saliency map S^N is produced as below:

$$S^N = \frac{1}{m} \sum_{i=1}^m S_i^N \quad (8)$$

3. Performance Evaluation

We select top five state-of-the-art salient object detection methods for aggregation as mentioned in [11], including DSR [8], MR [12], MC [13], RBD [14], DRFI [15], whose codes or results can be acquired from the authors' personal websites. There are four aggregation schemes tested which are denoted as: LA, Bayes, HVE and our SVE.

3.1 Datasets

We evaluate our method on three different types of benchmark datasets including: ASD [16], ECSSD [17] and DUT-OMRON [12]. There are some samples chosen from three data sets as illustrated by Fig.3. ASD includes 1000 images selected from the MSRA database. Most images in it

Fig.3. The sample images chosen from three datasets.

have only one salient object and there are usually strong contrast between objects and backgrounds. ECSSD consists of a large number of semantically meaningful but structurally complex natural images. DUT-OMRON is a challenging dataset that contains 5,168 images with complex backgrounds

3.2 Evaluation Measures

Precision-Recall (PR) curve and Mean Absolute Error (MAE) are evaluated in our experiments.

PR curve: Given corresponding masks, the precision and recall rate are defined as bellows:

$$\text{Precision} = \frac{|M \cap G|}{|M|}, \text{Recall} = \frac{|M \cap G|}{|G|}$$

where M is the binary object mask generated by thresholding corresponding saliency map and G is the corresponding ground truth. A fixed threshold changing from 0 to 255 is used for thresholding. On each threshold, a pair of precision/recall scores are computed, and are finally combined to form a PR curve to describe the performance at different situations.

MAE: PR curve does not consider the true negative saliency assignments. For a more comprehensive comparison, MAE is further introduced to evaluate the performance between the saliency map S and the ground truth G , which is defined as:

$$\text{MAE} = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H |S(i, j) - G(i, j)|$$

where W and H are the width and the height of the saliency map, respectively. Lower MAE value indicates better performance. This measure is also found complementary to PR curves [14]. As described in [11], we draw our conclusions mainly based on PR curves, and also report MAE scores for comprehensive comparisons and for facilitating specific application requirements.

F-measure: Usually, neither Precision nor Recall can comprehensively evaluate the quality of a saliency map. To this end, the F-measure is proposed as a weighted harmonicmean of them with a non-negative weight β :

$$F_{\beta} = \frac{(1 + \beta^2) \text{Precision} \times \text{Recall}}{\beta^2 \text{Precision} + \text{Recall}}$$

As suggested by many salient object detection works [18], β^2 is set to 0.3 to raise more importance to the Precision value. The reason for weighting precision more than recall is

that recall rate is not as important as precision.

Table 1. Average running time on ASD

Method	LA	Bayes	HVE	SVE
Time(s)	0.037	0.100	0.128	0.082

Fig.4. Comparison of PR curves, MAE and F-measure results of various aggregation methods on different datasets

Fig.5. Comparison results of various aggregation methods

3.3. Experiment Results

In this paper, we evaluate our method by comparing PR curve, MAE and F-measure about the four methods including LA, Bayes, HVE and SVE on three different types of benchmark datasets as shown in Fig.4. From the left to right: first one is PR curve which more close to the top is better; Second one is MAE with lower value indicates better performance; The last one is the histogram about comparison of the three indicators including F-measure, Precision and Recall, actually F-measure is more high more better.

Extensive experiments on three benchmark datasets demonstrate that the proposed SVE algorithm performs favorably against the state-of-the-art saliency aggregation methods. It is worth to say that our SVE gets better recall value than the other method as shown below by Fig.4 which means it can detect much more salient regions and it is important for some applications such as multiple salient objects detection. The visual results of various aggregation methods are shown in Fig.5. Note that the methods used for aggregation are all superpixel based which has good characteristic of edge-preserving, thus it will not meet the noisy problem on edges of objects.

The running time test is carried out in a 64-bit PC equipped of an i7-4790k 4.00 GHz CPU and 16 GB RAM. All the tested codes are provided by the authors and run unchanged in MATLAB R2015a. The average running time per image of different approaches on the ASD dataset are listed in Table 1. As can be seen, our SVE is faster than HVE and Bayes.

4. Conclusions

In this paper, we propose a saliency aggregation algorithm called SVE. Specially, we firstly define a soft-voting weight based on sigmoid function. Then each saliency map is improved based on our soft-voting evolution. Benefit from SVE, we get the final saliency map by average which outperforms the others. Experimental results show that our proposed approach performs favorably against other state-of-the-art saliency aggregation models on three publicly available databases and each component of the proposed algorithm improves the performance of saliency maps. It is also very efficient and favors the real applications.

Acknowledgements

The authors would like to express their sincere thanks to the anonymous reviewers for their invaluable suggestions

and comments to improve this paper. This work was supported by University Science Research Project of Jiangsu Province (15KJB510032), and “Lv Yang Jin Feng” Program of Yangzhou China.

References

- (1) A.Borji, D.N.Sihite, and L.Itti: “Salient object detection: A benchmark”, In ECCV, 2012.
- (2) ITTI L, KOCH C: “Computational modeling of visual attention”, In Nature Reviews Neuroscience 2, pp.194-203, 2001.
- (3) Hou, X., Zhang, L.Saliency Detection: “A Spectral Residual Approach”, In: CVPR, 2007.
- (4) Cheng, M.-M., Zhang, G.-X., Mitra, N. J., Huang, X., Hu, S.-M.: “Global Contrast Based Salient Region Detection”, In CVPR, 2011.
- (5) Cheng, M.-M., Zhang, Z., Lin, W.-Y., Torr, P.BING: “Binarized Normed Gradients for Objectness Estimation at 300fps”, In CVPR, 2014.
- (6) Mai, L., Niu., Y., Liu., F:Saliency Aggregation: “A Data-driven Approach”, In: CVPR,2013.
- (7) Ling Zheng, Shuhan Chen, Xuelong Hu: “Saliency Aggregation via Hard-Voting Evolution”, In ICEMI,pp.1232-1237 , 2015.
- (8) Li, X., Lu, H., Zhang, L., Ruan, X., Yang, M.-H.: “Saliency Detection via Dense and Sparse Reconstruction”, In ICCV, 2013.
- (9) Borji, A., Sihite, D. N., Itti, L.: Salient Object Detection: “A Benchmark”, In ECCV, 2012.
- (10)Meura, O. L., Liu, Z.: Saliency Aggregation: “Does Unity Make Strength? ”, In: ACCV, 2014.
- (11)Borji, A., Cheng, M.-M., Jiang, H., Li, J.: “Salient Object Detection: A Benchmark”. In Xiv eprint , 2015.
- (12)Yang, C., Zhang, L., Lu, H., Ruan, X., Yang, M.-H.: “Saliency Detection via Graph-Based Manifold Ranking”, In: CVPR, 2013.
- (13)Jiang, B., Zhang, L., Lu, H., Yang, M.-H., Yang, C.: “Saliency Detection via Absorbing Markov Chain”, In: ICCV, 2013.
- (14)Zhu, W., Liang, S., Wei, Y., Sun, J.: “Saliency Optimization from Robust Background Detection”. In: CVPR, 2014.
- (15)T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y.Shum.: “Learning to detect a salient object”. IEEE TPAMI, vol. 33, no. 2, pp. 353–367 ,2011
- (16)Achantay, R., Hemamiz, S., Estraday, S.: “Frequency-tuned Salient Region Detection”, In: CVPR, 2009.

- (17) Yan, Q., Xu, L., Shi, J., Jia, J.: “Hierarchical Saliency Detection”,In: CVPR, 2013.
- (18) R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk.: “Frequency-tuned salient region detection”,In: CVPR ,2009.
- (19) Yao Qin, Huchuan Lu, Yiqun Xu and He Wang.: “Saliency Detection via Cellular Automata”, In CVPR 2015.