Saliency Detection of Objects in Images (7)

Abhishek Naik, *CS13B030,* K Sai Srinivas, *CS13B039*

Abstract

We study the evolution of various techniques for detecting salient objects in images over the years - from traditional core CV methods to the new state-of-the-art deep learning models. Further, we implement 'Boolean Map Saliency Detection' (BMS) technique, which is a fundamentally simple, not deep, extremely fast, and with performance comparable to the state-of-the-art in almost all saliency metrics. We experiment with some postprocessing techniques to generate better saliency maps. Further, we explore the original algorithm's shortcomings empirically and suggest improvements on each of those fronts.

I. Introduction

SALIENT object detection is an important problem in computer vision. In its most general formulation, a salient object is one that has the most visual attention in an image. Salient objects in an image are usually foreground objects, though that is not always the case. There are two factors that determine an object's visual relevance - the first one is a task independent component which depends on low-level features, like image statistics. The second factor is the use of high-level features to determine the importance of scene relevance in the real world. The second factor is what makes saliency detection hard, since domain knowledge is often required to correctly predict visual attention.

The inherent vagueness in the definition of saliency has resulted in large number of models, each with their own metric of evaluation. This makes is extremely hard to judge which model is better than the other. Hence, a decision was made at the ECCV 2016 saliency tutorial, wherein it was decided that models will be ranked by NSS (Normalized Scanpath Saliency). This requires a proper salient object detection model to return a saliency map, where the value of each pixel is the probability that the pixel belongs to a salient object.

A. Applications

Salient object detection models have a variety of applications including object detection, image and video compression, content based image and video retrieval, visual tracking, and human-robot interaction. Hence, saliency detection has applications in computer vision, graphics and robotics. Some other notable applications include:

- camera auto-focussing
- scene summarization
- motion detection

B. Related Work

Salient object detection models can be broadly split into two categories,

- Rarity/Contrast based models; where low level features like contrast and image-statistics are used to identify salient patches. Information theory is used to measure the improbability of a local patch as a bottom-up saliency cue. However, such models make no use of high-level features at all, and so are doomed to fail, especially for images with multiple foreground objects, or images with complex, off-center foregrounds.
- Learning based models; where machine learning is used to learn saliency. A classifier is trained using a combination of low and high-level features. However, these models require offline learning and may not be very useful in online applications, because saliency map generation is done pixel by pixel. For instance, many models use SVMs and more recently, have started using Convolutional Neural Networks to perform classification. However, learning based models give the best performance, mainly due to the emergence of deep models.

We now look at a few older models before moving on to BMS, a boolean map based saliency model.

1) Boosting top down and bottom up visual features for saliency estimation: [1]

An image is decomposed into low-level attributes such as color, intensity, and orientation across several spatial scales which are then linearly or non-linearly normalized and combined to form a master saliency map. An important element of this theory is the idea of center-surround that defines saliency as distinctiveness of an image region to its immediate surroundings. They take thirty low level features, and four high level features, to get high dimensional feature vectors for pixels. Then a boosted regressor is trained to predict the value of the output.

2) Context-aware saliency detection: [3]

The paper proposes a new kind of saliency, context-aware saliency – which aims at detecting the image regions that represent the scene, as compared to identifying fixation points or detecting the dominant object. The principles kept in mind are,

- Local low-level considerations, including factors such as contrast and color.
- Global considerations, which suppress frequently occurring features, while maintaining features that deviate from the norm.
- Visual organization rules, which state that visual forms may possess one or several centers of gravity about which the form is organized.
- High-level factors, such as human faces.

The model uses these multi scale, high-level features to augment the lower-level features, and by including the immediate context by factoring distance to detected salient regions, they manage to capture a context for the salient object as well.

3) Deep Models: [6]

There are various deep models that give state-of-the-art performance in salient object detection. Most methods used pre-trained CNNs, like AlexNet or VGGNet and modify them appropriately to the context of saliency. There are too many models and architectures to mention, including LSTMs, Recurrent networks, transfer learning approaches etc. Though often giving very good results, deep models require large computational power and lots of training data.

4) Boolean Map Saliency: [10][9]

BMS is a boolean map based model that makes use of global topological cues that can help in perceptual segregation. Various factors like surroundedness, convexity and symmetry are used. The main cue exploited here is surroundedness, which is an enclosure topological relationship between different visual components. The main advantages of this method is its speed and relevance of results to actual human visual attention. The high-level intuition and algorithm is described below.

Despite its simplicity, BMS performs very well on different datasets across all the metrics, consistently featuring in the Top 5 alongside the deep architectures.

C. Dataset(s) used

The dataset that was used for all the experimentation was MSRA10k [7], which contains 10,000 images with pixel-level saliency labeling. Some examples are shown in Figure 3 and Figure 4. We also report results (presented by different authors) of various techniques on the MIT300 dataset [4], which has 300 natural indoor and outdoor scenes, with recorded human-eye movements and the ASD dataset [2], which is an accurate object-contour based ground truth database of 1000 images.

II. ALGORITHMIC DESCRIPTION

A. Brief explanation

1) Boolean Map Generation
   The color channels are enumerated in the given color space and the threshold is sampled at a fixed step size δ within the range of that channel. The color space is rectified before doing this, with a color-whitening step:

$$\bar{x} = \frac{1}{n}\sum_i x_i$$

$$Q = \frac{1}{n}\sum_i x_i x_i^T - \bar{x}\bar{x}^T$$

   where n is the number of pixels in that image.
2) Activation Map Computation
   The surroundedness is computed for each of these boolean maps using a simple Flood Fill algorithm using all the border pixels as the seeds. The resultant activation map $M(B)$ has 1s for all the surrounded pixels and 0 s for the rest.
3) Attention Map Computation
   Dilation and L2-normalization is used for normalizing the resultant activation maps, so that the activation maps with small concentrated active areas will receive more emphasis.
4) Salient Object Detection
   All the attention maps are averaged into a mean attention map P. Now, eye fixation maps are usually highly blurred because of the uncertainty and sparse distribution. However, salient object detection requires object level

segmentation, which means the corresponding saliency map should be high- resolution with uniformly highlighted salient regions and clear-defined region boundaries.
For this purpose, some post-processing is done, involving an opening-by-reconstruction operation followed by a closing-by-reconstruction operation, in order to smooth the saliency maps but keep the boundary details.

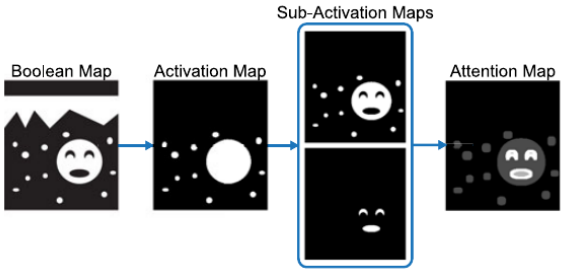


Fig. 1: The pipeline of the BMS algorithm, showing intermediate outputs. Notice that the boolean map is followed by the calculation of surroundedness. This is followed by generating the attention maps using the selected and surrounded part of the attention map, before the application of L2-regularization to give weightage to small concentrated active areas. Finally, an average of all the activation maps is taken to get the eye-fixation result.

B. Pseudocode

The above algorithm is represented compactly in the following pseudocode:

Algorithm 1 Boolean Map Saliency

procedure BMS(I)
  $P \leftarrow$ `ZEROS`(I.size());
  $\hat{I} \leftarrow$ `featureSpaceWhitening`(I);
  for feature maps $\phi_k(I)$: k = 1,2 ..., N **do**
    for $\theta = \min_i \theta_k(I)(i) : \delta : \max_i \theta_k(I)(i)$ **do**
      $B \leftarrow$ `THRESH`($\phi_k(I), \theta$);
      $M \leftarrow$ `computeSurroundedness`(B);
      $A \leftarrow$ `computeAttention`(M);
      $P \leftarrow P + A$;
    end for
  end for
  $P \leftarrow P / \max_i P(i)$;
  $S =$ `postProcess`(P)
  return S;
end procedure

The above pipelines is more-or-less the same as proposed by the authors in [9]. Since their wrappers and functions were written in Matlab versions which are no longer supported, we ported it all to `Python` and `C++`. This was quite tricky due to the lack of a direct mapping between the framework in the Matlab and Python versions of OpenCV. Furthermore, we have added a final `postProcess` stage to the pipeline, which we believe is essential in obtaining better saliency maps. The post-processing stage is described in more detail in the following subsection.

C. Post-Processing

Proper post-processing algorithms are necessary to convert the fixation maps into proper saliency maps. We describe the post-processing algorithm in greater detail.

The output returned, P is an eye fixation map. Models with salient object detection have a different emphasis when compared to models for eye fixation prediction. Saliency maps should be

- High resolution maps, where the salient object should be presented in sharp contrast, without any untoward blurring. Thus salient maps need to have clearly defined image boundaries.
- uniformly highlighted, unlike eye fixation maps, the whole salient object must be highlighted

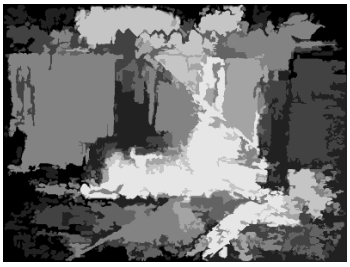
Post processing is done by performing an opening-by-reconstruction operation followed by closing-by-reconstruction operator. The opening operator is a morphological operator that is used for removal of noise or small particles in a

grayscale image. It is erosion followed by dilation, the net effect being to preserve shapes similar to the opening kernel or that can completely contain the opening kernel. Performing opening leads to loss of some foreground information, but gives a high resolution map. In order to get the missing pixels, we perform morphological reconstruction, which starts from seed pixels and grows in a flood fill fashion, to include complete connected components. Closing by reconstruction is then done to fill in the gaps in the foreground. The morphological closing operation is dilation followed by erosion. The effect of this is to preserve background elements that have similar shape to the structuring element. Thus we see how applying these morphological operators leads to a high resolution, filled salient map.

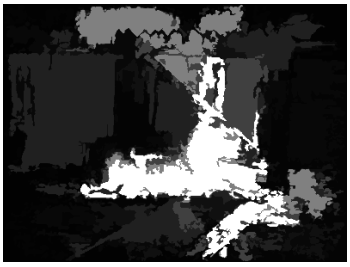
However, this alone is not enough as the generated saliency maps have many grey levels, and are often blurred (as seen in Figure 2(b)). Two solutions are proposed, the first is to simply threshold the saliency map, using say, Otsu's multilevel thresholding. This method, at times, does not preserve the boundary of the salient object. In order to maintain the image boundary, we perform exponential contrast enhancement. This takes care of the problem of too many grey levels, without losing the boundary information. To demonstrate it's working, consider the following example : Suppose the resulting saliency map has pixels intensities in the range of 20-100 (highest being 255). We empirically observe that an exponential stretching of these 20-100 values among 0-255 works better than something like a linear scaling. It is like performing a softmax with a number other than e (we observe 1.1 works the best) The output of performing such an operation is seen in Figure 2(c).

Here is an example of how proper post processing can generate better saliency maps.

(a) The input image.



(b) Saliency map obtained by the authors' BMS algorithm.



(c) After our additional post-processing.

Fig. 2: Here is a comparison of the highlights the advantages of a proper post-processing. This is a particularly hard example because of the uniformity of color in the object and the background. The authors' BMS algorithm results in a saliency map with a lot of grey-levels, which fails to identify the 'salient' sheep. On the other hand, our post-processing (described earlier) results in a saliency map with a better contrast and detection of the sheep.

III. OUTPUT

A. Observations

1) Elementary cases: When there is a single object, having a good contrast with a relatively uniform background, BMS works perfectly, giving near-perfect NSS scores. Some examples are shown in Figure 3. This is because:

- The fundamental pillar of BMS is Gesalt's principle of surroundedness, according to which areas which can be seen as surrounded by others tend to be perceived as figures. A single, whole object fits this criterion perfectly.
- A high contrast with the background enables a larger number of boolean maps to capture the disparity in the color space, resulting in a higher confidence in the saliency map.

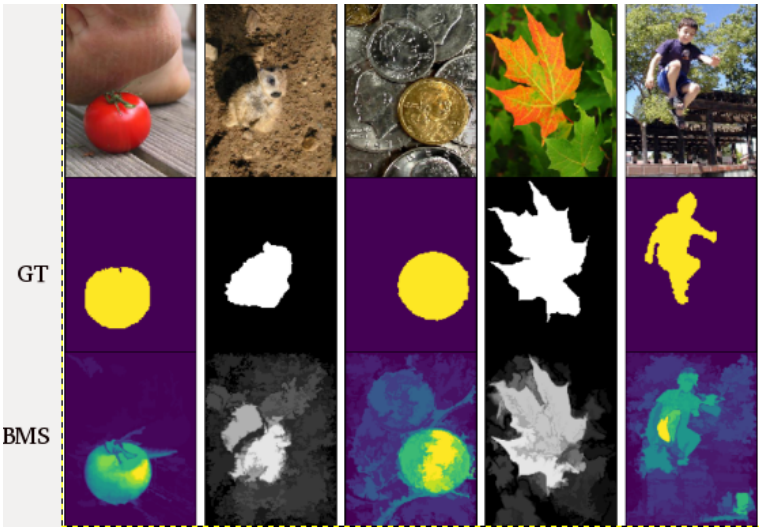


Fig. 3: Examples of successful detection of salient objects by our modified BMS on the MSRA10k dataset [7]

Notice that in the 2nd and 5th example of Figure 3, there is significant illumination variation and background clutter. BMS still performs quite decently in both cases.

2) Failure cases: Experiment have revealed that BMS fails in two cases,

- Failure of the notion of surroundedness[8]
  When the salient object is not adequately 'surrounded' by other object, the algorithm might fail to recognize the object. There are also scenarios where texture change is mistaken as surroundedness, resulting in improper segmentation of a whole salient object. The effects are mainly seen when the salient object is a ring, or is a multi textured object such as a human being. (Examples 1 and 2 in Figure 4)
- When there is a context to the images
  Since BMS does not use any high-level features, the algorithm does not understand the importance of the objects detected via low level features. This results in missing salient objects that cannot be found out by using purely low-level features. Sometimes this can manifest as improper importance assignment to different foreground objects, where the algorithm can pick a less important foreground object as the more salient one. Thus BMS sometimes fails when there are multiple salient objects in an image. (Examples 3 and 5 in Figure 4)

Fig. 4: Examples of failures in salient object by our modified BMS on the MSRA10k dataset [7]

B. Performance

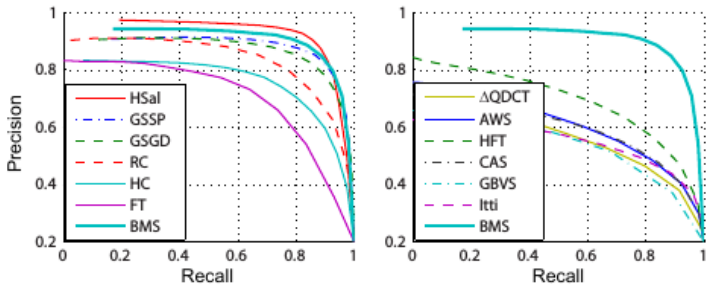


Fig. 5: Precision-Recall curves on the ASD dataset [2], (courtesy [10])which comprises of 1000 images and ground-truth segmentation masks. BMS is compared with six (then) state-of-the-art salient object detection methods (HSal, GSSP, GSGD, RC, HC and FT), as well as some leading models for eye fixation prediction. They binarize the saliency maps at a fixed threshold and compute the average precision and recall (PR) for each method. By varying the threshold of binarization, the PR curve is obtained for each method.

Saliency Models	AUC-Judd ↑	SIM ↑	EMD ↓	AUC-Borji ↑	shuffled AUC ↑	CC ↑	NSS ↑
DeepFix (Proposed)	**0.87**	**0.67**	**2.04**	0.80	0.71	**0.78**	**2.26**
SALICON [55]	**0.87**	0.60	2.62	**0.85**	**0.74**	0.74	2.12
Mr-CNN [34]	0.77	0.45	4.33	0.76	0.69	0.41	1.13
Deep Gaze I [32]	0.84	0.39	4.97	0.83	0.66	0.48	1.22
BMS[49]	0.83	0.51	3.35	0.82	0.65	0.55	1.41
eDN [33]	0.82	0.41	4.56	0.81	0.62	0.45	1.14
Context Aware Saliency [54]	0.74	0.43	4.46	0.73	0.65	0.36	0.95
Judd Model [45]	0.81	0.42	4.45	0.80	0.60	0.47	1.18
GBVS [4]	0.81	0.48	3.51	0.80	0.63	0.48	1.24

Fig. 6: Metrics in comparison with the newer Deep methods (courtesy [5]), on the MIT300 test set [4]. As can be seen, there is a lot of variation among the scores and the ranks, which makes it hard to judge which is the better model

IV. Conclusion

We see that BMS works the best in situations where there is a single prominent foreground object, but gives a good performance in a wide variety of situations - from background clutter to illumination variation. However, we have identified the scenarios wherein it fails, and we propose the following modifications to the existing algorithm.

- Usage of high level features
  Using high-level features like face-detection will allow for better salient object detection. There are multiple ways to incorporate this, one could simply use a heuristic to combine many such handcrafted features to improve attention map generation, or one could use machine learning to augment the generated bitmaps and their respective weights.
- Better Low level feature extraction
  Usage of linear combinations of various color spaces may capture the complementary topological structure of the scene. Feature maps like contrast/edges etc are not suitable for measuring surroundedness because they tend to lose the topological structure of the scene by only sparsely highlighting certain local patterns (e.g. edges and corners)
- Texture independent saliency map generation.
  Recently, it has been observed via the ImageNet challenge, that texture-agnostic techniques result in better accuracies for some classes (like teapots), where the texture only contributes in an aesthetic sense. Such techniques can used in this context as well, for better performances.

References

[1] Ali Borji. Boosting bottom-up and top-down visual features for saliency estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 438–445. IEEE, 2012.
[2] Antón Garcia-Diaz, Xosé R Fdez-Vidal, Xosé M Pardo, and Raquel Dosil. Saliency from hierarchical adaptation through decorrelation and variance normalization. *Image and Vision Computing*, 30(1):51–64, 2012.
[3] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1915–1926, 2012.
[4] Tilke Judd, Frédo Durand, and Antonio Torralba. A benchmark of computational models of saliency to predict human fixations. 2012.
[5] Srinivas SS Kruthiventi, Kumar Ayush, and R Venkatesh Babu. Deepfix: A fully convolutional neural network for predicting human eye fixations. *arXiv preprint arXiv:1510.02927*, 2015.
[6] Matthias Kümmerer, Lucas Theis, and Matthias Bethge. Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. *arXiv preprint arXiv:1411.1045*, 2014.

[7] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE Transactions on Pattern analysis and machine intelligence*, 33(2):353–367, 2011.
[8] Elizabeth S Spelke. Principles of object perception. *Cognitive science*, 14(1):29–56, 1990.
[9] J Zhang and S Sclaroff. Exploiting surroundedness for saliency detection: A boolean map approach. *IEEE transactions on pattern analysis and machine intelligence*, 38(5):889, 2016.
[10] Jianming Zhang and Stan Sclaroff. Saliency detection: A boolean map approach. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.