Salient object detection: a discriminative regional feature integration approach (9)

Santhosh Kumar R, *EE12B101,* and Yogesh B, *EE12B066,*

Abstract

Detection of visually salient image regions is useful for applications like object segmentation, adaptive compression, and object recognition. We consider the saliency map computation as a regression problem. We use supervised learning to map the regional features into a saliency score and fuse the saliency maps across multiple levels to yield the final saliency map. To extract the foreground region, we use this saliency map as an initial estimate to GrabCut algorithm which yields the final foreground segmented image.

I. Introduction

Visual saliency is the perceptual quality that makes an object, person, or pixel stand out relative to its neighbors and thus capture our attention. There are various applications for salient object detection, including object detection and recognition, image compression, image cropping, photo collage, dominant color detection and so on. Recently, lot of heuristic algorithms have been proposed to compute saliency. In [1], saliency estimation is regarded as a regression problem, where a regressor is learnt that directly maps the regional feature vector to a saliency score. There are three main steps involved. The first one is multi-level segmentation, which decomposes the image to multiple segmentations from a fine level to a coarse one. Second, a region saliency computation is done with a random forest regressor that maps the regional features to a saliency score. Last, a saliency map is computed by fusing the saliency maps across multiple levels of segmentations.

We improve the saliency map fusion step in [1] by using sum of pairwise product maps as opposed to using linear regressor. This map is then used as an initial estimate to the iterated Grabcut algorithm whereby foreground regions are extracted.

II. ALGORITHMIC DESCRIPTION

The approach in [1] consists of three stages: multi-level segmentation, region saliency computation and multi-level saliecy fusion. We explain each of the stage below :

A. Multi-level Segmentation

For a given image I, a set of M-level segmentations $S = \{S_1, ...S_M\}$ are computed where S_1 is the finest and S_M is the coarsest. S_1 is computed using oversegmented Graph based Segmentation approach[2] and S_m is computed from S_{m-1} by merging regions of S_{m-1} until a certain threshold is met. Choosing different thresholds give different levels of segmentation.

B. Region Saliency Computation

Each region is represented using three types of features: regional contrast, regional property and regional backgroundness.

1) Regional Contrast Descriptor: Each region is described by a feature vector v containing color and texture. The regional contrast descriptor of a region R is computed as the differences diff(v^R, v^N) between its features and the neighborhood features. Specifically, the difference of the histogram feature is computed as the distribution divergence, and the differences of other features are computed as the absolute elements differences of the vectors. As a result, we get a 26-dimensional feature vector.

2) Regional property descriptor: we consider the generic properties of a region, including appearance and geometric features. The appearance features attempt to describe the distribution of colors and textures in a region, which can characterize the common properties of the salient object and the background. The geometric features include the size and position of a region that may be useful to describe the spatial distribution of the salient object and the background.

3) Regional backgroundness descriptor: Pseudo-background region, defined as 15 pixel border region of each image is extracted for each image and the backgroundness descriptor for each region is computed using pseudo-background region as reference. The backgroundness feature of the region R is then computed as the differences diff(v^R, v^B) between its features v^R and the features v^B of the pseudo-background region.

4) Training: We denote the feature as a vector x. Then the feature x is passed into a random forest regressor f, yielding a saliency score. The random forest regressor is learnt from the regions of the training images, and integrates the features together in a discriminative strategy. The training examples include a set of confident regions R = $\{R_1, ...R_Q\}$ and the corresponding saliency scores A = $\{a_1, a_2...a_Q\}$, which are collected from the multi-level segmentation over a set of images with the ground truth annotation. A region is considered to be confident if the number of the pixels belonging to the salient object or the background exceeds 80% of the number of the pixels in the region, and its saliency score is set as 1 or 0 accordingly.We learn a random forest regressor f from the training data $X = \{x_1, ...x_Q\}$ and the saliency scores $A = \{a_1, ...a_Q\}$.

C. Multi-level saliency fusion

After region saliency computation, each region has a saliency value. For each level, we assign saliency value of each region to its contained pixels, thereby generating M saliency maps $\{A_1....A_M\}$. In [1], the saliency maps are fused using linear regressor.

But, linear regressor did not produce very good results because low resolution maps gave nonzero scores for many false regions which could not be driven to zero using a linear regressor. Instead, we combined the maps by using sum of pairwise products over all saliency maps and suitably normalizing the resulting map. This improved the resulting saliency map.

$$I(x,y) = \sum_{i=1}^{i=N} \sum_{j=i+1}^{j=N} \frac{I_i(x,y) * I_j(x,y)}{N^2}$$

where I is the final saliency map, I_i is the i^{th} map from multi-level output, N is the number of maps.

D. Foreground extraction

GrabCuts[3] are very good methods for object segmentation given an initial estimate. K-means or Gaussian Mixture models are used for learning colour distributions and graph cuts are used to infer segmentation. These two steps are carried iteratively until there is convergence. So, the saliency map which was obtained in the previous method was thresholded using Otsu thresholding. Dilation was performed so as to increase the bounding region. This was then passed as an initial estimate to GrabCut which gave foreground segmented image.

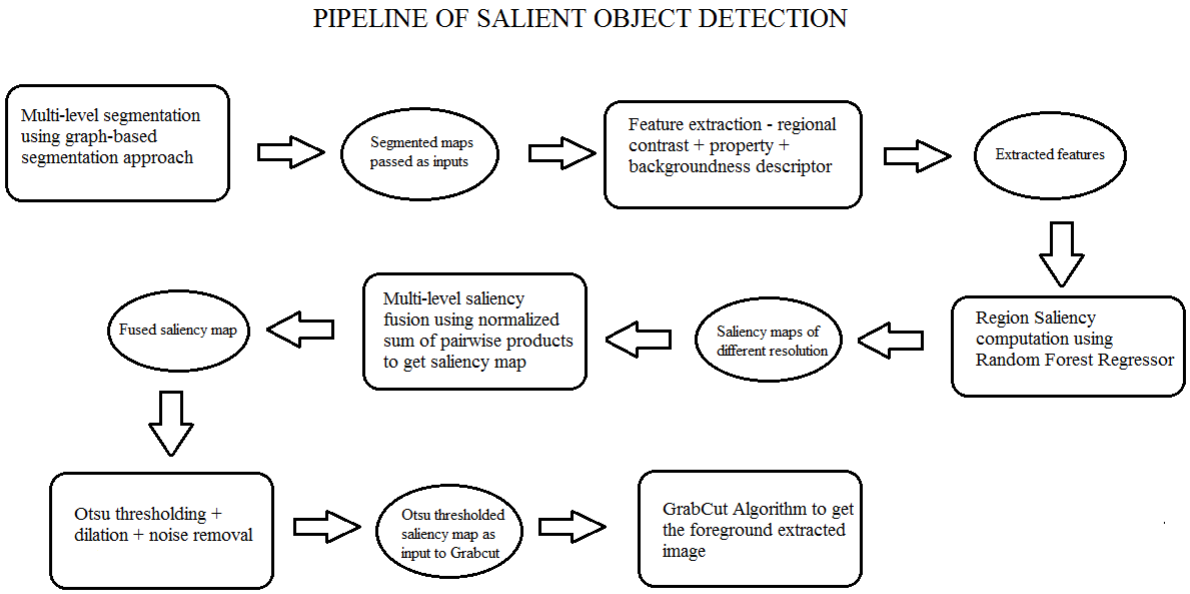


Fig. 1: Flow diagram

E. New training approach attempted

Since the existing algorithm[1] did not perform well on cluttered images, we tried to incorporate clutter information while training. So we had to look for features that represent clutter score. Clutter regions generally have more high frequency content.

So, the idea was to extract response to Gabor filter banks at different frequency levels and use that as features. A large number of filter banks was chosen and best features among them was selected using PCA. We trained the model using these additional features. But unfortunately we didn't get better results as compared to the normal DRFI code.

III. OUTPUT

Experiments were performed on different datasets and results were obtained. Some of the results are shown below.

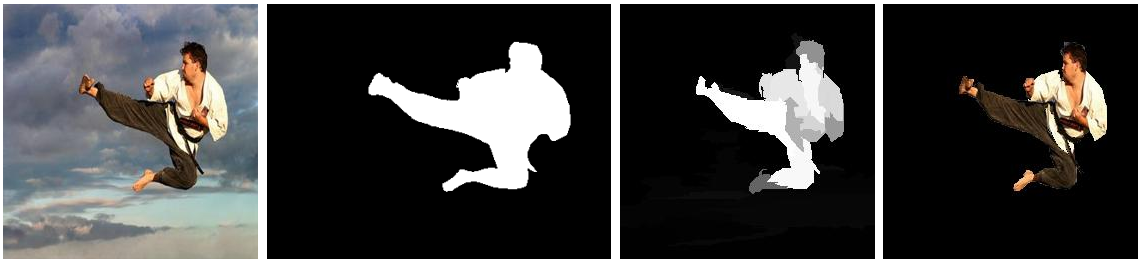


Fig. 2: From left: source image, ground truth, saliency map obtained, foreground extracted object

Fig. 3: From left: source image, ground truth, saliency map obtained, foreground extracted object

This is a comparison of DRFI and our improved implementation. It can be seen that the saliency maps obtained by our implementation is superior when compared to DRFI.

It can be seen the our implementation supresses the false positives when compared to DRFI.

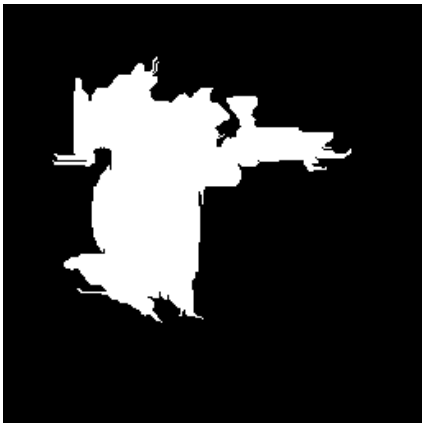
(a) RGB imput

(b) Ground truth

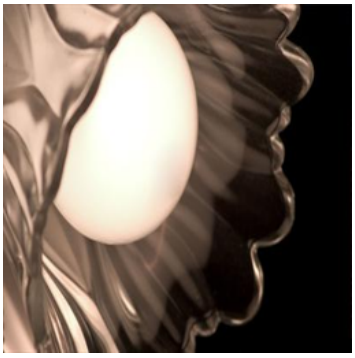
(c) DRFI Saliency map

(d) Improved saliency map

(e) DRFI segmented output



(f) Improved segmented output



(a) RGB imput

(b) Ground truth

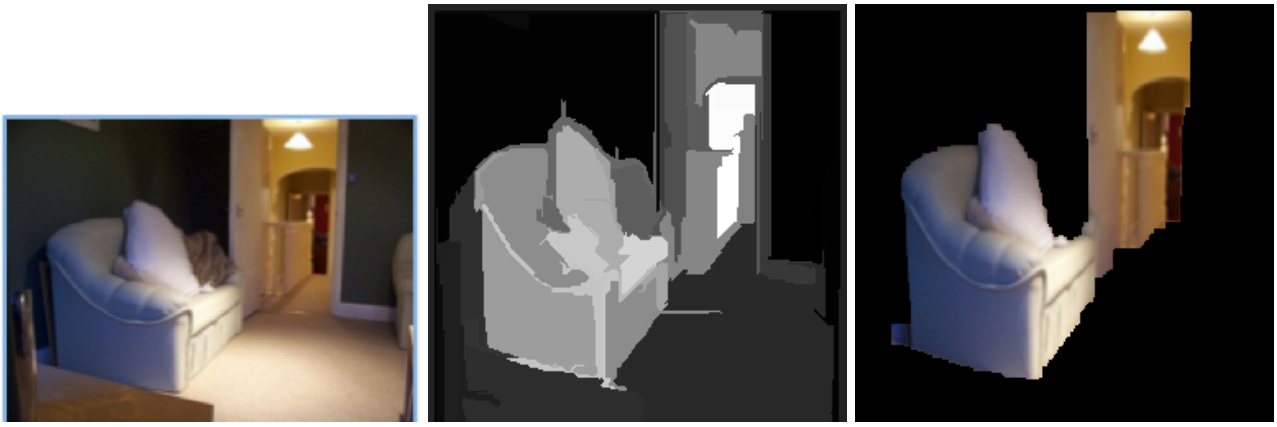
(c) DRFI Saliency map

(d) Improved saliency map

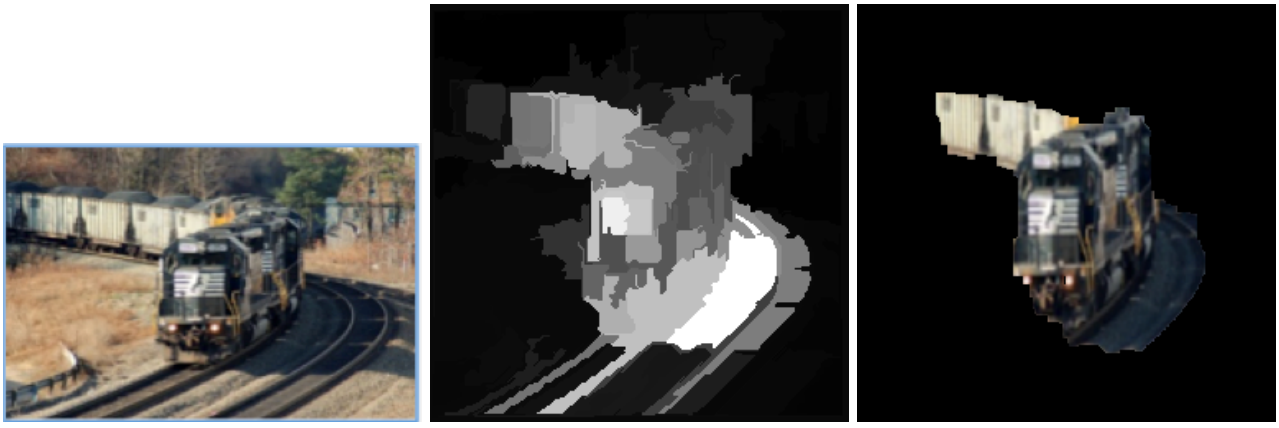
(e) DRFI segmented output

(f) Improved segmented output

Some more results with a few failure cases have been shown below.



(a) RGB input (b) Saliency map (c) Foreground image



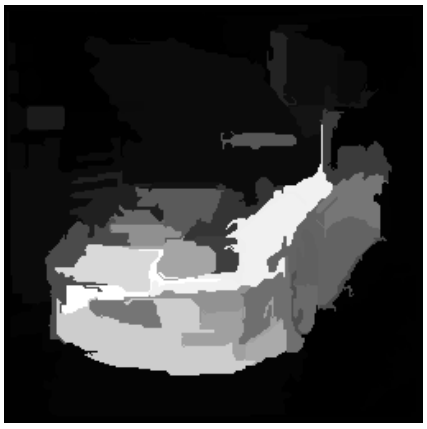
(a) RGB input (b) Saliency map (c) Foreground image

(a) RGB input

(b) Saliency map

(c) Foreground image

(a) RGB input



(b) Saliency map

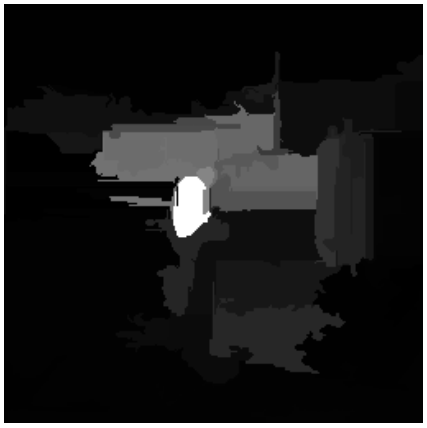
(c) Foreground image

(a) RGB input

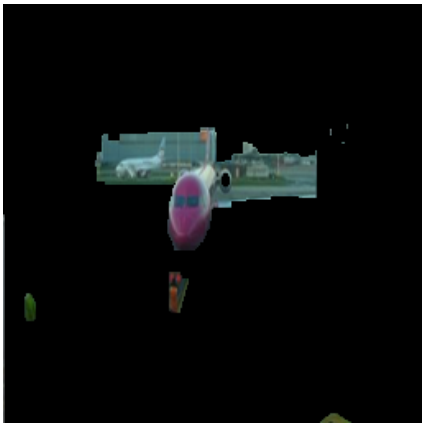
(b) Saliency map

(c) Foreground image

(a) RGB input

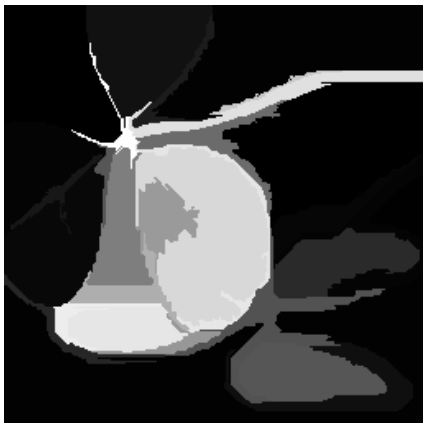


(b) Saliency map

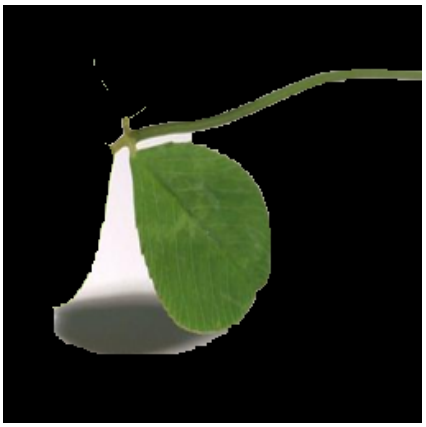


(c) Foreground image

(a) RGB input



(b) Saliency map



(c) Foreground image

(a) RGB input (b) Saliency map (c) Foreground image

The Precision-Recall curve is plotted by taking the saliency image, thresholding the saliency map at different levels. True positives are measured as the number of white and black pixels common to both ground truth and saliency map obtained. Likewise, false positives are the pixels which are different in both the images.

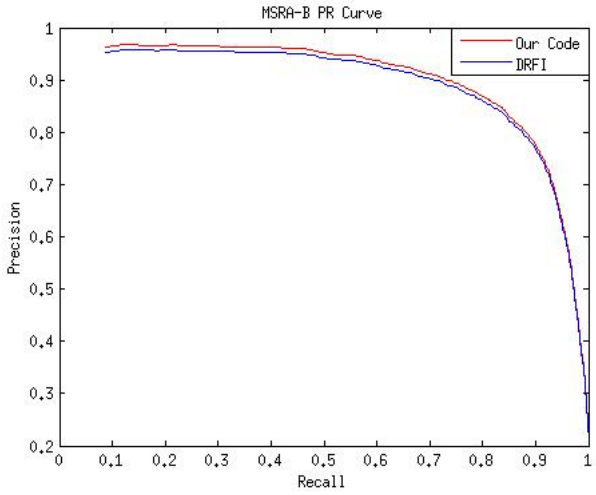


Fig. 14: PR curve on MSRA-B dataset

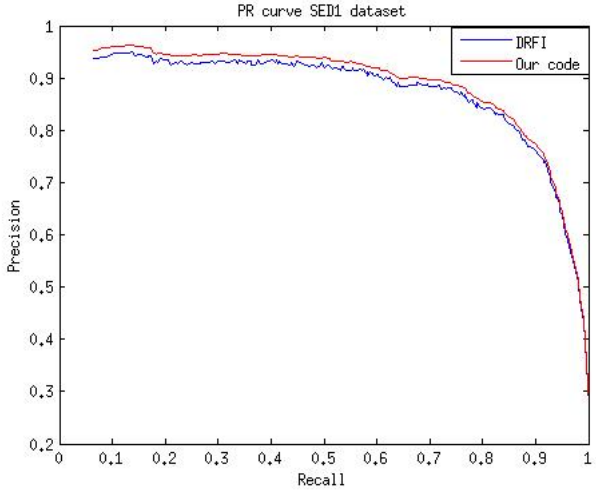


Fig. 15: PR curve on SED1 dataset

In the above two test cases, we can see that the saliency map computation results in some false positive regions. So, simple thresholding for foreground extraction will not work always. But, we can see that grabcut results in very clear output. We can also observe the finer details being highlighted at the boundaries.

A. Observation

Saliency map obtained from [1] was good enough, but it was not performing well on cluttered background images. Moreover, many false regions were getting highlighted, so a simple Otsu thresholding did not give good results for foreground extraction. We then retrained the model using clutter as features. Since, in cluttered images, high frequency components are generally high, we introduced a bank of Gabor filters at various frequency levels and used it as additional features to retrain our model. But, the new model did not give significant improvements.

Next, we tried our approach for combining saliency maps which was based on normalized sum of pairwise products of saliency maps. This method improved the results. Intensity of false regions that were highlighted by linear regressor came down. Better performance on cluttered images were obtained because in cluttered images, high low resolution maps indicate many false regions.

For the foreground extraction, a comparison on Otsu thresholding on output of [1] and Grabcut was made. In many images, the output of [1] when thresholded gave many false regions as foreground. Whereas, Grabcut showed significant performance improvement. Results close to ground truth were obtained on many images. Grabcut was also benefitial in extracting very fine details around the boundaries of objects.

The Precision recall curves have been plotted for both DRFI and our implementation. It can be observed that our implementation is performing better than DRFI on both the datasets.

IV. Conclusion

In this project, we address the salient object detection problem using discriminative regional feature integration approach. The success of this approach stems from the fact that we are integrating a lot of regional features instead of heuristically computing the saliency maps. Moreover combining saliency maps of different levels by suitable means boosted confidence of salient regions thereby producing good results. Finally, foreground extraction using Grabcut gave great improvements in foreground extraction.

REFERENCES

[1] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, Shipeng Li, *Salient Object Detection: A Discriminative Regional Feature Integration Approach*. In CVPR, pages 2083-2090, 2013

[2] P. F. Felzenszwalb and D. P. Huttenlocher., *Efficient graph based image segmentation*. International Journal of Computer Vision, 59(2):167-181, 2004.

[3] Carsten Rother and Vladimir Kolmogorov and Andrew Blake, *"GrabCut – Interactive Foreground Extraction using Iterated Graph Cuts*. ACM TRANSACTIONS ON GRAPHICS, 23:pages 309-314, 2004.

[4] R. Achanta, S.S.Hemami, F.J.Estrada, and S.Susstrunk., *Frequency-tuned salient region detection.* In CVPR, pages 1597-1604, 2009

[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 ICCV, pages 914921, 2011.

[6] B. Fernando, E.Fromont, D. Muselet, and M. Sebban., *Discriminative feature fusion for image classification.* In CVPR, pages 34343441, 2012.