Application of Conditional Random Fields to Pixel-Level Salient Object Detection Through Local, Regional and Global Features

Jimmy Lin
Australian National University
Canberra, Australia
linxin@gmail.com

Christopher Claoué-Long
Australian National University
Canberra, Australia
u5183532@anu.edu.au

Abstract

Making use of OpenCV, DARWIN and the MSRA dataset, we detect the saliency of by extracting local , regional and global saliency features at the pixel level, then combine those features with pre-fitted weights derived by logistic regression. The Conditional Random Field is then constructed to capture the spatial continuity of the saliency. The importance ratio of the combined unary and pairwise terms is determined by cross-validation. Based on the binary mask inferred by the Conditional Random Field, we ultimately apply a winner-takes-all algorithm to output a single bounding rectangle to label the detected salient object, by which the performance of our approach is evaluated through boundary-displacement error.

1. Introduction and Motivation

A long-standing problem in the field of computer vision is how to calculate the saliency of objects in an image – that is, their prominence in the image when compared to their background and surrounds. This property is important for image recognition since it leads to the ability to single out individual areas of an image, for example in automatic image cropping and visual attention simulation in robotics, and 3D surface reconstruction in augmented reality displays.

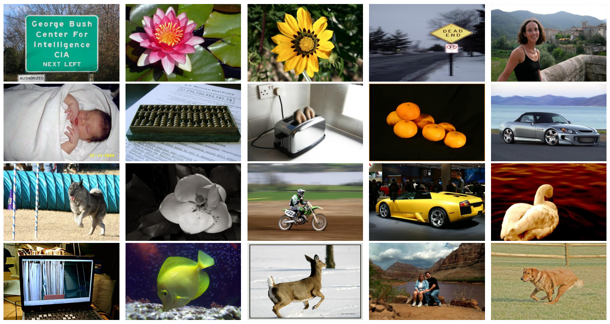


Fig. Example images from MSRA dataset B

2. Related Works

Most existing visual attention approaches are based on the bottom-up computational framework, where visual attention is supposed to be driven by low-level stimulus in the scene, such as color intensity and contrast. These approaches consist of the following three steps: The first step is feature extraction in which multiple low-level visual features, such as intensity, color, orientation, texture, and motion, are extracted from the image at multiple scales. The second step is saliency computation. The saliency is computed by a center-surround operation, self-information, or graph-based random walk using multiple features. After normalization and linear/nonlinear combination, a master map or a saliency map is computed to represent the saliency of each image pixel. Last, a few key locations on the saliency map are identified by winner-take-all, or inhibition-of- return, or other nonlinear operations.

3. Our Approach

We base the essence of our salient object detection on several suppositions: it is likely for stuff with high contrast boundary to its near neighbours to be salient object, large ”block” distinction relative to located surrounds, intensive colour distribution compared to all other color component in candidate image and space continuity of saliency.

3.1. Formulation

To formulate the salient object detection problem, we incorporate the high-level concept of the salient object into the process of saliency map computation. As can be observed in Fig, people naturally pay more attention to salient objects in images, such as a person, a face, a car, an animal, or a road sign. Therefore, salient object detection can be formulated as a binary labeling problem that separates a salient object from the background.

For each pixel x of given an image I, the binary mask A_x indicate whether this pixel x belongs to the salient object (1) or not (0). Thus, our objective can be transformaed to derive the corresponding saliency map A, indicating binary salient property of one pixel.

Build up a probabilistic model $P(A|I) = \frac{1}{Z}e^{-E(A|I)}$, where $\frac{1}{Z}$ is the normalising factor, and $E(A|I)$ is the energy function incorporating both unary and pairwise potentials between pixels.

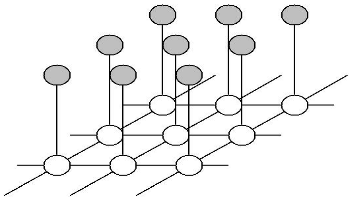


Fig.2 graph for Conditional Random Field

Formally, the energy function can be represented as

$$E(A|I) = \sum_x S_{unary}(a_x, I) + \lambda_0 \sum_{x,x'} S_{pair}(a_x, a_{x'}, I)$$

where λ is the relative weight between the summary of multiple unary features and pairwise features.

The pairwise feature $S(a_x, a_{x'}, I)$ exploits the spatial relationship between two adjacent pixels. It can be viewed as a "penalty" for labelling adjacent pixels the same or differently.

$$S(a_x, a_{x'}, I) = |a_x - a_{x'}| \cdot e^{-\beta d_{x,x'}}$$

where x, x' represent two adjacent pixels, $d_{x,x'}$ is the L2-norm (standard norm) representing the colour difference between the two pixels, and $\beta = (2\langle||I_x - I_{x'}||^2\rangle)^{-1}$ is a robust parameter to weight the colour contrast.

The unary potential for combination of three features is specified as

$$S_{unary}(a_x, I) = \sum_{k=1}^{K} \lambda_k \cdot F_k(a_x, I)$$

where λ_k is the weight of the k^{th} feature conforming to $\sum_{k=1}^{K} \lambda_k = 1$.

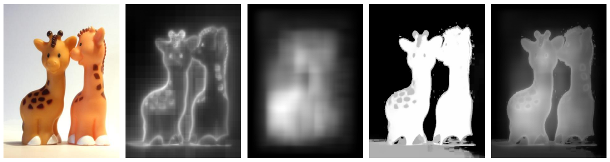


Fig.3 Original Image and Preview of feature maps
Left to Right: (b) multiscale contrast (c) center surround histogram
(d) color spatial distribution (e) composed unary potential

The value of each feature $F_k(a_x, I)$ comes from a normalised feature map $f_k(x, I) \in [0, 1]$, and for each pixel:

$$F_k(a_x, I) = \begin{cases} f_k(x, I), & a_x = 0 \\ 1 - f_k(x, I), & a_x = 1 \end{cases}$$

3.2. Feature Extraction

Feature Extraction, widely acknowledged as the most significant component of computer vision task, represents how we want the computer to interpret the raw iamges. In this project, we just focus on three critical features, each of which is capable of capturing the saliency individually but in various level of scope. They are respectively Multiscale Contrast, Center Surround Histogram and Color Spatial Distribution.

3.2.1 Multiscale Contrast

Constrast is commonly utilised as local feature because the contrast operator simulates the human visual receptive fields. Specifically, it captures the point that salient object tends to have tremendous contrast to the surroundings in its boundary (not vice versa). Since we may have no preknowledge about the size of salient object, it is usual to compute the contrast at multiple scale.

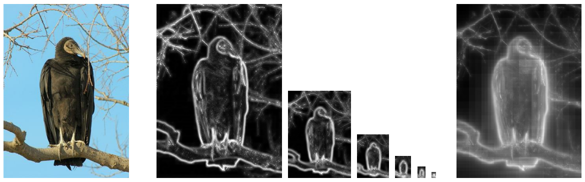


Fig. Pyramid of Multiscale Contrast.
(a) leftmost: Original Image. (b) rightmost: Multiscale Feature Map.
(c) immediate images: multiscale pyramids from level 1 to 6.

Hence, define the multiscale constrast to be a contrast map from the linear combination of image contrast at all levels of an N-level gaussian image pyramid, using the pixels x in the image I:

$$f_c(x, I) = \sum_{n=1}^{N} \sum_{x' \in W(x)} ||I^n(x) - I^n(x')||^2$$

where W(x) is a window that delineates which area to consider for neighbouring pixels to compare contrast values.

In our implementation, we choose the total number of pyramid level N to be 6 and the size of the window to be 9×9.

It is evident that the Multiscale Contrast give high distinction between the boundary and non-boundary region. This provides us a precise description of where the boundary of salient object exist in the output binary mask. For those salient object has relatively large contrast within its body, this feature works perfectly, such as the bird in the second image and the house in the third image. As to the pixels within the salient objects, they are usually given low marks, this weakness can be complemented by the regional feature - Center Surround Histogram, which will be introduced latter.

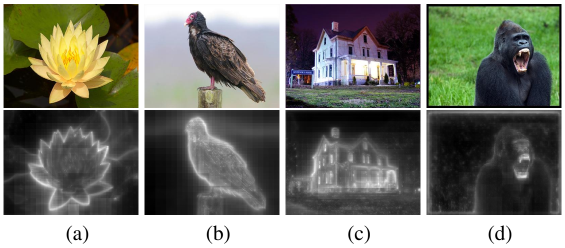


Fig. Feature Maps of Multiscale Contrast under various scenes

But there are some drawbacks that are tough to avoid. First and foremost, boundary of all objects are highlighted, which is obviously not desired. We only wish to give high mark to pixels that are relavant to salient object, rather than all boundaries in one image. This would possibly lead to some ”saliency leak” in the ultimate result and deteriorate detection precision since some pixels outside the true salient object are likely to be considered as a part of that object.

Bad vulnerability to the outlier pixels within the body of salient object is another problem of multiscale contrast. Look at the chimpanzee in the fourth image, the tooth is tremendously contrasted to its surroundings, far beyond the contrast of the body of chimpanzee to the grass. Since each entry of output feature map is quantitatively normalised, the contrast of the body of chimpanzee to the grass is comparatively trivial. However, the tooth may be too narrow to be observed by the human receptive field and thus not labelled as the salient object. That is, this flaw may result in the low recall comparing to the ground truth data.

3.2.2 Center Surround Histogram

As shown in the Fig. , the local feature - multiscale contrast only partially detects salient objects since it is only sensitive to the boundaries. Out of the purpose of detecting the whole block of objects, we make use of another static salient feature, which captures the regional information of saliency and can be computed using various low-level features via a center-surround operation.

Create a colour RGB histogram for both the rectangle and the surrounding frame, which has the same area with the center rectangle, with a certain resolution (number of “bins” for each colour). And then measure the distinguishability of the area centered at each pixel x by calculating the χ^2 value between the two histograms to obtain the differences between the rectangle and the surrounding frame. Do this for multiple aspect ratios, and keep the largest χ^2 value:

$$\begin{aligned} R(x) &= \arg\max_{R(x)} \chi^2(R(x), R_s(x)) \\ &= \arg\max_{R(x)} \frac{1}{2} \cdot \sum_{i \in bins} \frac{(hist_{R(x)_i} - hist_{R_s(x)_i})^2}{hist_{R(x)_i} + hist_{R_s(x)_i}} \end{aligned}$$

Here, we traverse through five template of rectangle with different aspect ratio $\{0.5, 0.75, 1.0, 1.25, 1.5\}$ for determining the most distinct pair of center and surround rectangle centered at each pixel x. Besides, to reduce the demands on computational resources, the size range of the rectangle is discretized to be 12 ratios $\{0.18, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75\}$ with regards to $min(w, h)$, which is the minimal value of width and height of the processed image.

The center-surround histogram feature at each pixel x is finally calculated by:

$$f_h(x, I) \propto \sum_{x' | x \in R(x')} w_{xx'} \chi^2(R(x'), R_s(x'))$$

Note that the center-surround histogram feature reflects the distinguishability of each pixel, which is assigned from the distinguishability of the most outstanding area centered at each pixel of one image, with the weight of spatial closeness to the center of that area.

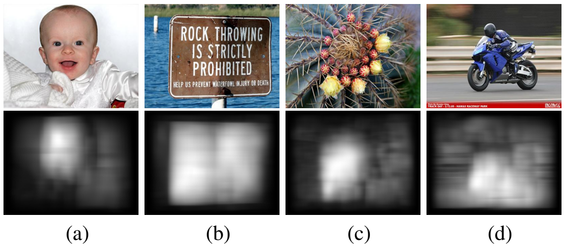


Fig. Feature Maps of Center-Surrond Histograms under various scenes

As we can see from above Fig, the whole salient block within one image is given distinguishable emphasis comparing to its surrounds in the output feature map.

3.2.3 Color Spatial Distribution

The goal of using Color Spatial Distribution feature is to take into account the global saliency information, that is, the information about how widely the colors occurring in one image are distributed over the global scope. By creating Gaussian Mixture Components and fitting them to pixels of the whole image, the variance of each color component can be captured and we give penality to the pixels with high-variance color according to the supposition that the colour of salient object in one image tends to have intensive spatial distribution.

According to the theory of Machine Learning, each pixel is associated to a colour component with the probability

$$P(c|I_x) = \frac{\omega_c \mathcal{N}(I_x|\mu_c, \Sigma_c)}{\sum_c \omega_c \mathcal{N}(I_x|\mu_c, \Sigma_c)}$$

where ω_c is the weight, μ_c is the mean colour, Σ_c is the covariance, and $\mathcal{N}(I_x|\mu_c, \Sigma_c)$ is the multivariate normal distribution of the c^{th} component.

For each fitted gaussian color component c, we compute its horizontal variance $V_h(c)$ as follows,

$$V_h(c) = \frac{1}{|X|_c} \sum_x p(c|I_x) \cdot |x_h - M_h(c)|^2$$

where x_h is horizontal coordinate of pixel x, $|X|_c$ is normalising factor and $M_h(c)$ is the mean of the gaussin component:

$$|X|_c = \sum_x p(c|I_x)$$

$$M_h(c) = \frac{1}{|X|_c} \sum_x p(c|I_x) \cdot x_h$$

Vertical variance $V_v(c)$ is defined similiarly, and we equally combine the horizental and vertical variance to derive the unnormalised composite variance $V_h(c)$:

$$V'(c) = V_h(c) + V_v(c)$$

Then employ the min-max approach to normalise the composite covariance:

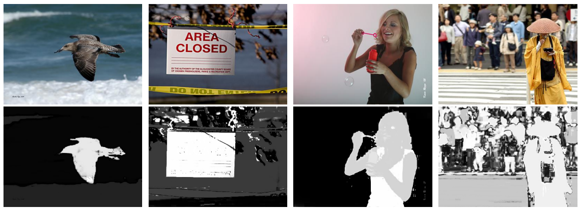
$$V(c) = \frac{V'(c) - min\big(V'(c)\big)}{max\big(V'(c)\big) - min\big(V'(c)\big)}$$

where $V(c)$ is the normalised composite covariance of the c^{th} component, contained between 0 and 1.

The final colour spatial distribution feature for pixel x is defined as a weighted sum of its color intensiveness:

$$f_s(x, I) \propto \sum_c p(c|I_x) \cdot (1 - V(c))$$

The feature map $f_s(\cdot, I)$ is also normalized to the range $[0, 1]$. The following figure shows color spatial-distribution feature maps of several example images. The salient objects are well covered by this global feature.

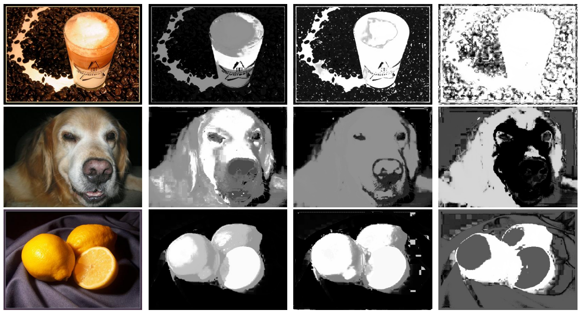


(a) (b) (c) (d)

Fig. Feature Maps of Color Spatial Distribution under various scenes

It is evident that the global feature gains a huge success in mark up the saliency when the background is monotonous. As illustrated by the first three images in the above figure, the flying stuff is distinguished from the bichrome background - blue ocean and white tide, the sign card under the background of ocean and tree leaves and in the third image the girl body before the fully white wall. However, in the case of colorful background, color spatial distribution, with a high probability, fails to distinguish the salient object in the image. The Fig.. (d) demonstrates us this undesired property of the global feature.

One variation in our implementation is to create the component model from only a subset of the pixels in the image. The pixels in the subset is picked up with equal spatial distance. Specifically, we apply the "pydown" module in Opencv to dilute the pixels participating this unsupervised learning task. Such manipulation will not sacrifice too much accuracy, but greatly reduce the running time of fitting the Mixture of Gaussians since the number of pixels provided for the fitting halves.



(a) original (b) no pydown (c) one-level pydown (d) two-level pydown

Fig. Examples for global feature using diluted images

We test three levels of pixel dilution: no pydown, one-level pydown and two-level pydown. One-level pydown means reducing both of the width and height of one image to half of the original size, such that only a quarter of pixels are available for fitting the mixture of gaussians. From the above figure, it is evident that one-level pydown, to some extent, increases the quality of this global feature by marking the salient block softly or obscurely rather than in a strict way. Extraction for two-level pydown indicated in Fig. (d) is even faster, but generally it lost the effectiveness of this global feature. Hence, in our implementation of color spatial distribution, we employ the one-level pydown as a trick to both enhance quality of global feature and improve extraction time performance.

Besides, the maximum number of iterations is limited (100) and the convergence criterion is lowered (10^{-1}) in order to reduce the time taken to compute this feature. This outcome of this global feature, after such simplification, will not get deteriorated since we only care about capturing approximate component location, rather than precisely maximum likelihood of the coloured pixels.

The number of gaussian components is another trade-off between the quality of feature extraction and computational cost. Our implementation uses five gaussians to softly capture color components in one image.

3.3. Learning

Those three features aforementioned have their own strengthes and weaknesses in different areas. For example, the . It goes without saying that incorporating all three features into the unary potential of the CRF model to complement each other. It would be oversimplistic and unpersuasive to treat three features in equal weight since one feature perhaps may be stronger than other two features and ought to be assigned with higher importance. Hence, one effective and reasonable approach is to adjustably determine the optimal weight to combine three features with the help of certain machine learning algorithm.

In this project, we implement logistic regression to decide the optimal weight under the help of training data.

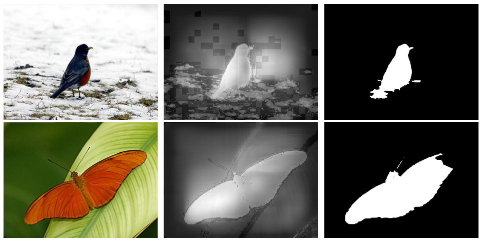
3.4. CRF Inference

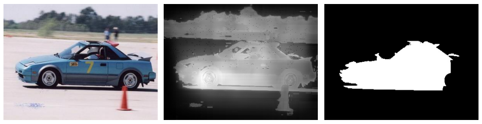
To infer the maximum likelihood assignment for pixelwise variables under the CRF framework, the usual message passing algorithm would be intractable. Therefore, we turn to the α-expansion inference, which successively segments all and non-α pixels with graph cuts and the algorithm will change the value of α at each iteration with graph-cut algorithm. By this means, inferring maximum likelihood assignment for binary saliency variable, or equivalently, minimal energy function would be much faster.

But how to determine the parameter λ_0, which indicates to what extent, relative to the unary term (combined feature map), we ought to consider the pairwise potential in deciding the binary saliency of one pixel? And this parameter would influence the smoothness of resulting binary mask since pairwise term can be regarded as the penality to adjacent pixels with different labels. Our solution is to determine the optimal λ_0 by applying the cross validation on λ_0 at various magnitudes.

TODO:add diagram and graph to illustrate our cross validation on various lambda

By cross validation above, the optimal parameter λ_0 we use for evaluating the binary mask is $\lambda_0 = 8$.





(a) Raw Image (b) Combined Unary Map (c) Binary Mask ($\lambda = 8$)
Fig. Examples for CRF inference by α-algorithm

4. Result Evaluation

We randomly pick up 500 images from MSRA dataset B to form the training set. And we choose another 500 images to form the testing set.

4.1. Bounding Box

The binary mask, derived from the CRF inference, can be directly used for various applications. However, in order to evaluate the effectiveness of our approach, we make use of ... algorithm to output a bounding rectangle to box the salient object, based on the derived pixelwise binary mask.

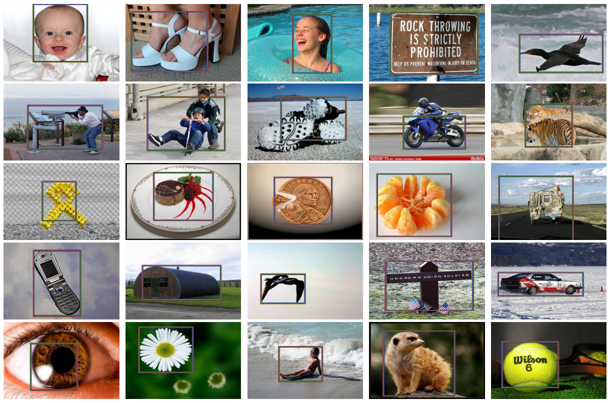


Fig. Examples images for bounding box output

4.2. Evaluation Criteria

4.2.1 Precision & Recall

4.2.2 Boudary Displacement Error

5. Conclusion and Discussion

References

[1] Liu, Tie, et al. "Learning to detect a salient object." *Computer Vision and Pattern Recognition, 2007. CVPR07. IEEE Conference on. IEEE, 2007.*

[2] Liu, Tie, et al. "Learning to detect a salient object." *Pattern Analysis and Machine Intelligence, IEEE Transactions on 33.2 (2011): 353-367.*

[3] Itti, Laurent, Christof Koch, and Ernst Niebur. "A model of saliency-based visual attention for rapid scene analysis." *Pattern Analysis and Machine Intelligence, IEEE Transactions on 20.11 (1998): 1254-1259.*

[4] Ma, Yu-Fei, and Hong-Jiang Zhang. ”Contrast-based image attention analysis by using fuzzy growing.” *Proceedings of the eleventh ACM international conference on Multimedia. ACM, 2003*.

[5] L. Itti. Models of Bottom-Up and Top-Down Visual Attention. PhD thesis, *California Institute of Technology Pasadena, 2000*.

[6] Boykov, Yuri, Olga Veksler, and Ramin Zabih. ”Fast approximate energy minimization via graph cuts.” *Pattern Analysis and Machine Intelligence, IEEE Transactions on 23.11 (2001): 1222-1239*.

[7] Gould, Stephen. ”DARWIN: A Framework for Machine Learning and Computer Vision Research and Development.” *Journal of Machine Learning Research 13 (2012): 3533-3537*.