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 an object in an image by extracting local, regional and global saliency features at the pixel level, then combine those features with pre-fitted weights derived by logistic regression. A conditional random field is then constructed to capture the spatial continuity of the salient object, with a weighting ratio of combined unary and pairwise terms that is determined by cross-validation **LOGISTIC REGRESSION?**. The binary mask inferred by the conditional random field is then used to output a single bounding rectangle around a salient area through a winner-takes-all algorithm to label the detected salient object, by which the performance of our approach is evaluated through mean 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. Human vision can accomplish this quite easily, since the human brain has mechanisms to direct attention to the main objects in the perceived image. This visual attention has been studied by researchers in multiple fields, from physiology and psychology to computer vision. The property of detecting a salient object is important for the latter because it leads to the ability to single out individual areas of an image, for example in automatic image cropping and vision simulation in robotics, as well as 3D surface reconstruction in augmented reality displays.

2. Related Works

Most existing approaches are based on a bottom-up computational framework, where the computer's simulation of visual attention is driven by low-level stimuli in the scene. These approaches employ three steps.
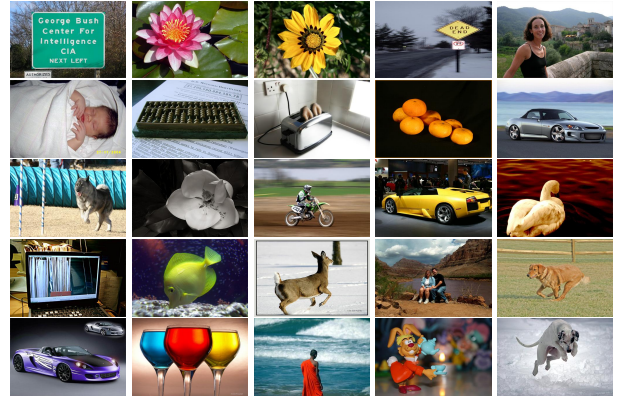

Figure 1. Example images from MSRA dataset B

The first step is feature extraction, where multiple low-level visual features such as intensity, colour, orientation, texture, and motion are extracted from the image at multiple scales.

The second step is saliency computation. The saliency is computed by a centre-surround operation, self-information, or graph-based random walk using multiple features. After normalisation and linear/nonlinear combination, a master or saliency map is computed to represent the saliency of each pixel in the image.

The last step is label output, where 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

In our approach, we base our salient object detection on several suppositions that mimic how human vision differentiates salient objects from the rest of the perceived field of view. As can be observed in **REDO FIGURES**, people naturally pay more attention to salient objects in images, such as a person, a face, a car, an animal, or a road sign. This is because of several reasons.

First of all, it is likely that pixels with a high contrast difference to their near neighbours to be part of a salient object, since they could represent a contour or boundary around an object. This is similar to how the human brain discovers boundaries between objects in its visual field by detecting differences in light intensity**CITATION PSYCHBOOK**.

Salient objects are more often than not quite distinct from their local surrounding region. Calculating the distance between an object and its surrounds can therefore give us information about how salient that object is. If we apply this concept over the entire image, we can gather information about how different various areas are to their surrounds and work out the likelihood that parts of an image are salient.

Finally, humans perceive prominence in objects through how distinct they are from their surrounds. Indeed, salient objects often demonstrate a marked difference in colour to the rest of a scene. Therefore, the more widely distributed a colour is in an image, the less likely it is that the salient object will contain that colour. This global colour distribution can therefore be used to describe the saliency of the pixels in an image.

3.1. Formulation

To turn the problem of salient object detection into a mathematical formulation, we incorporate these three high-level concepts of a salient object into the process of creating a saliency map. Salient object detection can be formulated this way as a binary labelling task that separates a salient object from the background through multiple operations.

For each pixel x of given an image I, the binary mask A_x must indicate whether it belongs to the salient object (1) or not (0). Our objective is to compute this mask A in order to show the location of the salient object in the image.

To do this, we build up a probabilistic model

$$P(A|I) = \frac{1}{Z} e^{-E(A|I)}$$

to determine the probability of a binary mask over an image with $\frac{1}{Z}$ as the normalising factor, and $E(A|I)$ as an energy function incorporating both unary and pairwise potentials between pixels in the image
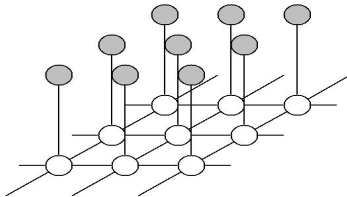

Figure 2. A Conditional Random Field
White nodes are the objects, grey nodes are their potentials

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 sum of multiple unary and pairwise features.

The unary potential, combining the three pixel 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 cross-validated **LOGISTIC REGRESSION?** weight of the kth feature such that $\sum_{k=1}^{K} \lambda_k = 1$.

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 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.
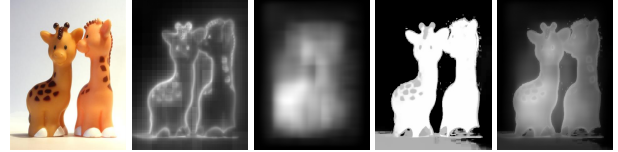

Figure 3. Original Image and Preview of feature maps

Left to Right: Original Image, Multiscale Contrast Map, Centre-Surround Histogram, Colour Spatial Distribution, Composed Unary Potentials

The final value of each feature $F_k(a_x, I)$ comes from a normalised feature map $f_k(x, I) \in [0, 1]$, where 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 a computer vision task, represents how we want the computer to interpret raw images. In this project, we focus on three features, each of which is capable of capturing saliency individually but in different level of scope. They are respectively multiscale contrast, centre-surround histograms and colour-spatial distribution.

3.2.1 Multiscale Contrast

Constrast is commonly used as local feature because it simulates the human visual receptive fields. It acts on the fact that the boundary of salient objects tend to have a marked contrast to the surrounding region. Since we may have no prior knowledge about the size of salient object, it is usual to compute the contrast at multiple scales to then incorporate back into one map, since this will demarcate the various boundaries in the image. This multiscale contrast map is thus a linear combination of image contrast at all levels of an N-level gaussian image pyramid, using the pixels x in the image I. Formally, this amounts to calculating

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

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

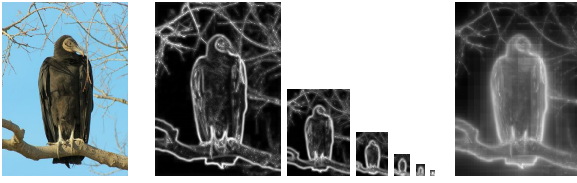**TODO ADD more technical analysis about why we should have multiple scales.**

Figure 4. Multiscale Contrast example.
Leftmost: Original Image. Rightmost: Multiscale Feature Map.
Immediate images: multiscale pyramids from level 1 to 6.

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.

As can be seen in **CONTRAST FIGURE**, the derived multiscale contrast map provides a high distinction between boundary and non-boundary regions. This provides us a precise description of where the boundary of a salient object could exist in the output binary mask. When a salient object also has high contrast inside its boundaries, this feature works perfectl, such as the bird in **FIG** and the house in **FIG**.

But there are some drawbacks that are tough to avoid. The boundaries of all objects are highlighted, which is not desirable when searching for a single salient object. This is because we need to give a high probability to marking pixels that are relevant to the salient object rather than every boundary possible. If this doesn't happen, there is the possibility of a "saliency leak" into non-salient areas of the final result that could deteriorate the detector's precision.

Another disadvantage to only using multiscale contrast is that the inner regions of salient objects are poorly represented. For example in the gorilla in **FIG**, the teeth are
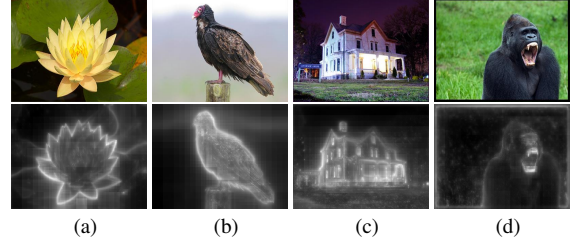

(a) (b) (c) (d)

Figure 5. Feature Maps of Multiscale Contrast

labelled as higher contrast to their surroundings compared to that of the body of the gorilla to the grass. Since each entry of the output feature map is quantitatively normalised, the contrast of the gorilla's body to the grass is rendered trivial compared to the contrast of the tooth. The tooth is not what the human receptive field would label as salient because it is a smaller part of the animal, and thus it should not be labelled as the salient object. This flaw may result in low recall comparing to the ground truth data when contrast is used as the sole method to distinguish a salient object.

3.2.2 Centre-Surround Histogram

As shown in 5 , 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:

$$R(x) = \underset{R(x)}{\arg\max} \chi^2(R(x), R_s(x))$$

$$= \underset{R(x)}{\arg\max} \frac{1}{2} \cdot \sum_{i \in bins} \frac{(hist_{R(x)_i} - hist_{R_s(x)_i})^2}{hist_{R(x)_i} + hist_{R_s(x)_i}}$$

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 centre and surround rectangle centred at each pixel x. Besides, to reduce the demands on computational resources, the size range of the rectangle is reduced to 12 discrete 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 centre-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 centre-surround histogram feature reflects the distinguishability of each pixel, which is assigned from the distinguishability of the most outstanding area centred at each pixel of one image, with the weight of spatial closeness to the centre of that area.
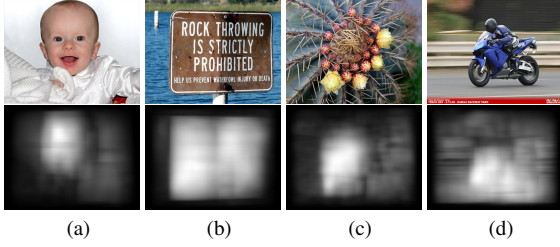

(a)   (b)   (c)   (d)

Figure 6. Feature Maps of Centre-Surround Histograms

As we can see from Fig.6, the whole salient block within one image is given distinguishable emphasis comparing to its surrounds in the output feature map.

However, unlike multiscale contrast, the center surround histogram does not provide accurate description for the boundary of one object, which is an inherent drawback of this regional feature. Fortunately, as mentioned previously, the local feature, to some extent, would complement the regional feature at the boundary description.

3.2.3 Colour Spatial Distribution

The goal of using colour spatial distribution feature is to take into account the global saliency information, that is, the information about how widely the colours occurring in one image are distributed over the global scope.

First of all, we create the Gaussian Mixture Components $\{c, \mu_c, \Sigma_c\}$ with regard to the color component c and fitting these components, making use of pixels of the whole image. According to the Gaussian Mixtures Model, 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 cth component.

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

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

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

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

Vertical variance $V_v(c)$ is defined similarly, and we equally combine the horizontal 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

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

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

And then we give penalty to the pixels with high-variance colour, according to the supposition that the colour of salient object in one image tends to be intensive in the spatial aspect. Thus, the ultimate 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 colour spatial distribution feature maps of several example images. The salient objects are well covered by this global feature.
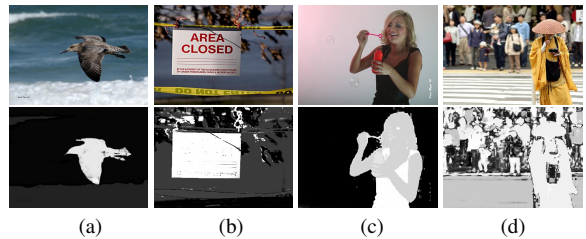

(a)   (b)   (c)   (d)

Figure 7. Feature Maps of Color Spatial Distribution

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.7 (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.
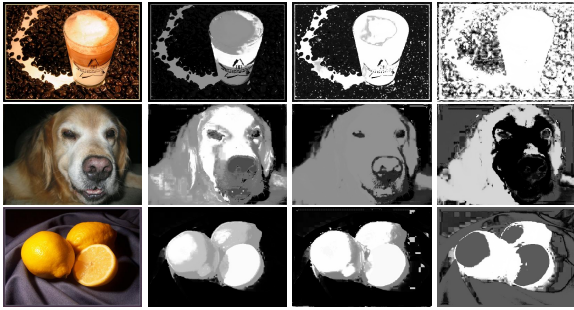
(a) original (b) no pydown (c) one-level pydown (d) two-level pydown

Figure 8. 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.8 (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 multiscale contrast provides no emphasis on the central part of salient object even though we compute the contrast at multiple scales to enlarge the scope the detected boundary, while the center surround histogram does highlight the salient block, but the salient boundary is not distinguishably resulted.

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.
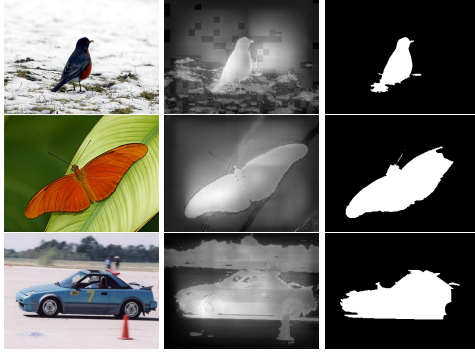
3.4. CRF Inference

To infer the maximum likelihood assignment for pixel-wise 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 penalty 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 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 ($\lambda = 8$)

Figure 9. Examples for CRF inference by α-algorithm

4. Result Evaluation

We randomly pick up 1000 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 used directly in a multitude of applications. However, in order to evaluate the accuracy of our approach, we make use of OpenCV's findContours algorithm to output a bounding box around the salient object, based on the derived pixel-wise binary mask. The dimensions of this bounding box is output as a label for each image into a text file holding all resultant labels for that directory.
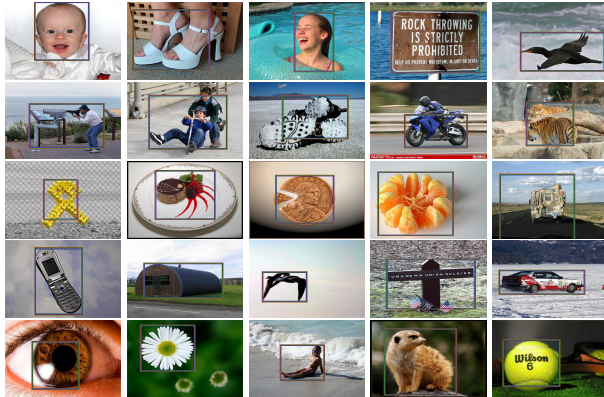

Figure 10. Examples images for bounding box output

4.2. Evaluation Criteria

The main functionality of evaluation criteria is to measure the effectiveness of our approach. In our project, evaluation will be applied, as well, in the cross validation while determining the energy function parameter λ_0. But what kind of criteria could effectively capture the goodness of

our model? As to this project, we utilise the both the region-based and boundary-based measurements proposed by Tie. Liu.

4.2.1 Region-based measurement

Precision and Recall, both taking significant roles in the information retrival as performance indicator, are utilised in this project to evaluate the correctness of our output rectangle.

The precision represents the percentage of pixels that are correctly detected in ground truth, which is the ratio of the number of pixels in overlapped region to pixel number contained in the ground truth box. And the recall reflects the percentage of pixels that are correctly detected in resulted detection, which is the ratio of pixel number the overllapped region contained to the number of pixels.
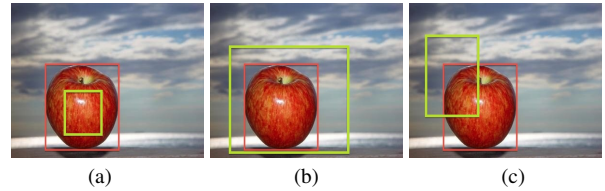

Figure 11. Examples with resulted detection and ground truth
Red Box: ground truth label. Green Box: resulted detection

It is evident that all threes detections in Fig.11 is far from satisfying. The salient detection in the leftmost image, the precision is 100% while the recall is small. Such detection is not desired because of the failure to detect all pixels within the true salient object. On top of that, the detection in the middle image has 100% recall, however, the precision is low in that a large portion of non-salient pixels are returned as the detection result.

For an overall performance measurement, we use the F-measure indicator with $\alpha = 0.5$, which is the weighted harmonic mean of precision and recall:

$$F_{0.5} = \frac{1.5 \times Precision \times Recall}{0.5 \times Precision + Recall}$$

4.2.2 Boundary-based measurement

The Boundary Displacement Error (BDE) measures the average of positional difference of ground truth and resulted detection.
Add brief math description of BDE here?

5. Conclusion and Discussion

In this project, we have indicated a supervised approach for salient object detection which is formulated as a binary labeling problem using a set of local, regional, and global salient object features.

However, out of the limitation of time, one flaw of our project is the lack of comparison with other approaches using the numerical evaluation result, which may tremendously enhance the persuasiveness of the approache utlised in this project.

Multiple object detection?

There are still several remaining issues for further investigation. **multiple**

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] Suzuki, S. and Abe, K., Topological Structural Analysis of Digitized Binary Images by Border Following. *CVGIP 30 1, pp 32-46 (1985)*.

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