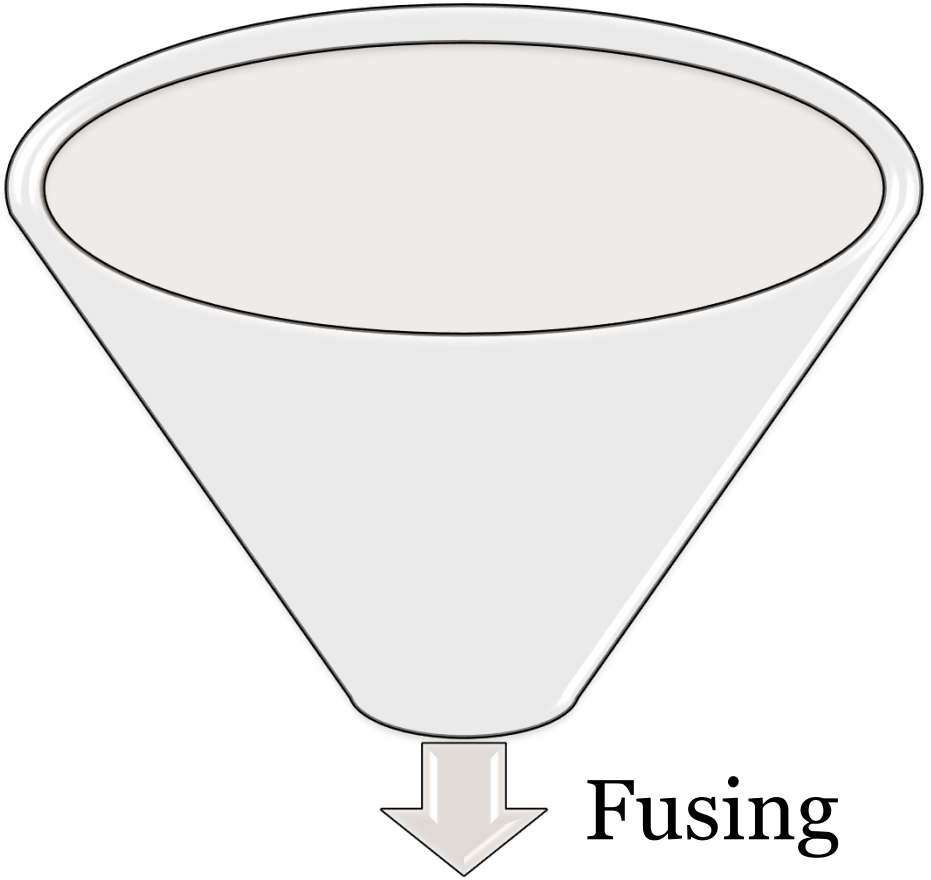
Fusing Generic Objectness and Visual Saliency for Salient Object Detection

Yasin KAVAK

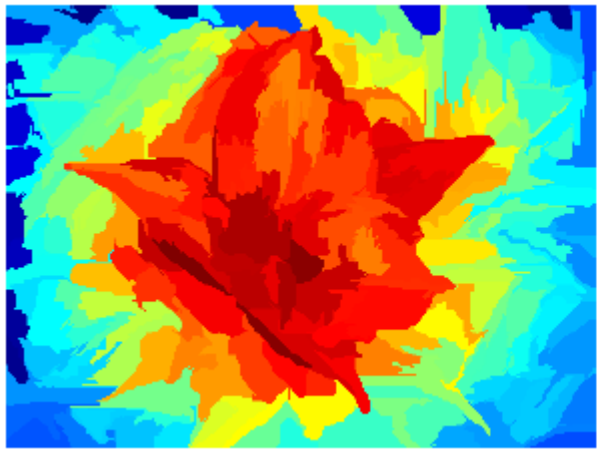
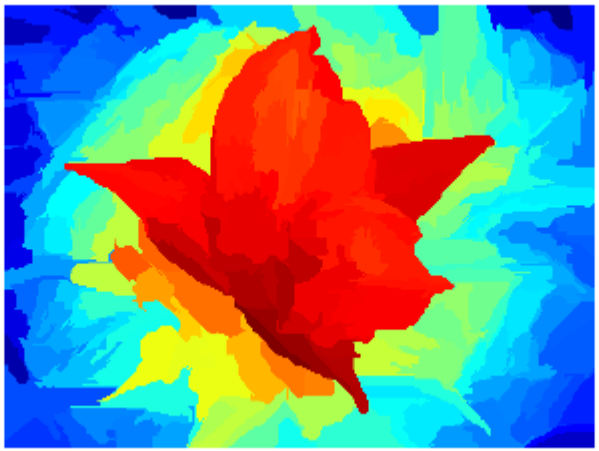
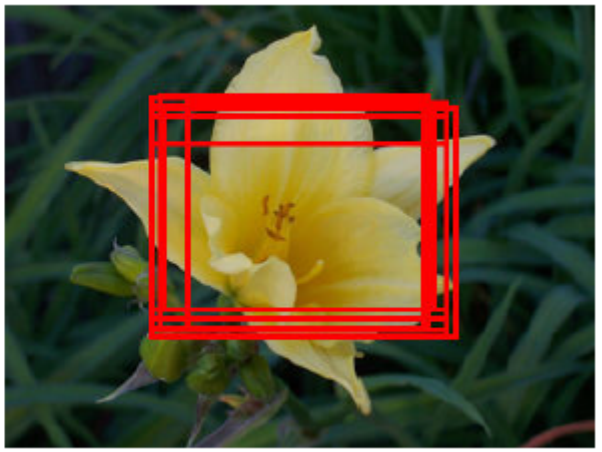
06/12/2012

Citation 1: Salient Object Detection: A Benchmark



Fusing

for Salient Object Detection

INDEX

(Related Work)

- [3] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, pages 73–80, 2010.
- [5] S. Goferman, L. Zelnik Manor, and A. Tal. Context-aware saliency detection. In *CVPR*, pages 2376–2383, 2010.

- AIM
- Fusion
  - Saliency, Objectness, Interaction, Optimization
- Experiment
- Conclusion

Sub Presentation
What is an Object

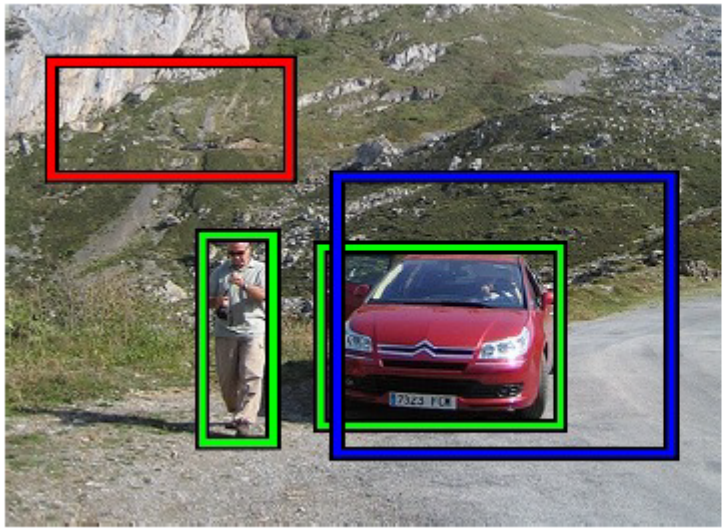
What is an Object

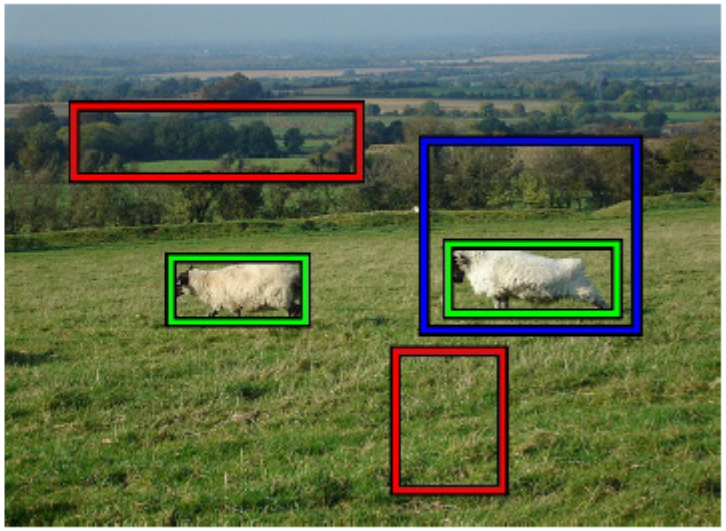
- AIM: a **generic** objectness *measure*, quantifying how likely it is for an image window to contain an object of any class

Distinctive Characteristics:

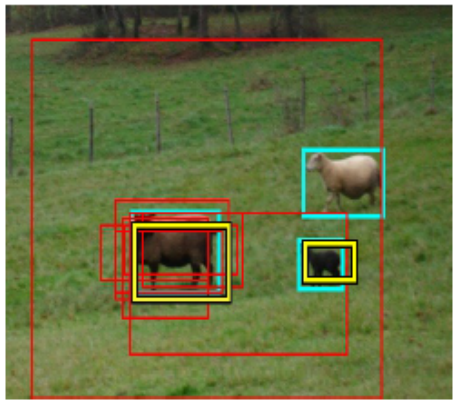
- (a) a well-defined closed boundary in space;
- (b) a different appearance from their surroundings
- (c) sometimes it is unique within the image and stands out as salient

Desired Behaviour

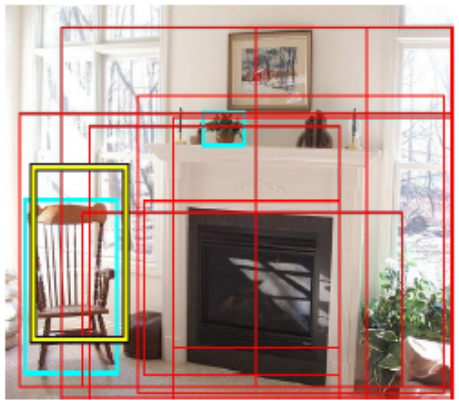




samples

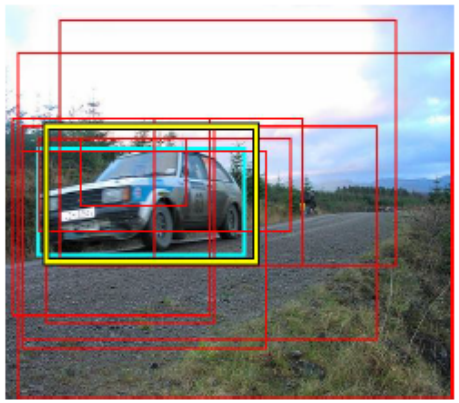


MS + CC + SS



MS + CC + SS

MS + CC + SS



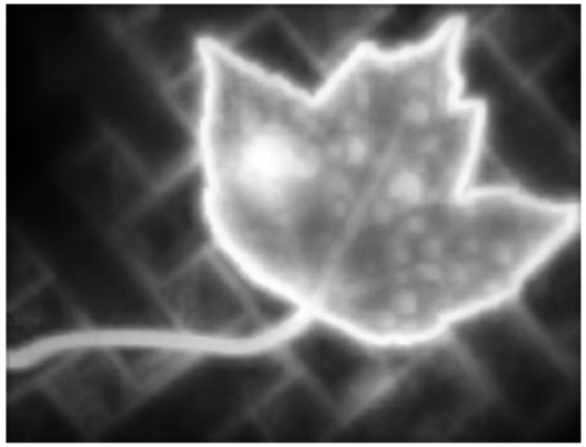
MS + CC + SS

Context-Aware Saliency Detection

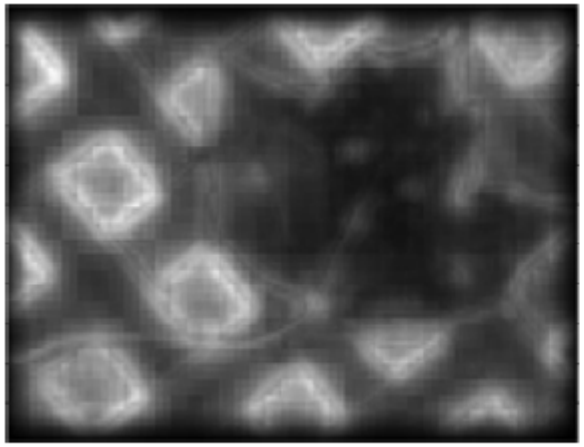
- We propose a new type of saliency – context-aware saliency – which aims at detecting the image regions that represent the scene. This definition differs from previous definitions whose goal is to either **identify fixation points** or **detect the dominant object**.

- Local-global single-scale saliency
- Multi-scale saliency enhancement
- Including the immediate context
- High-level factors

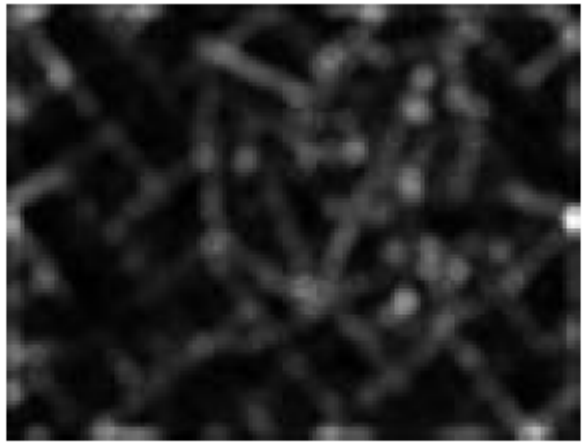
(a) Input



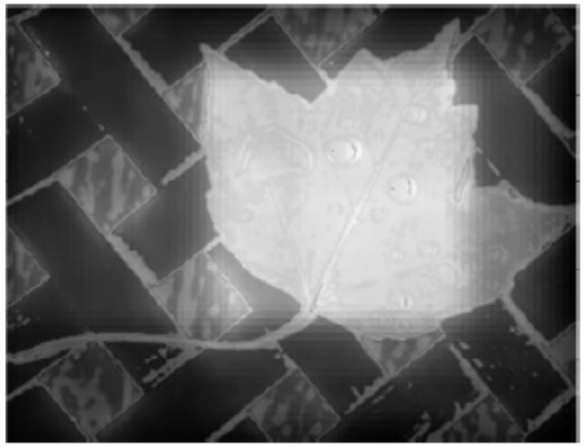
(e) Our context-aware



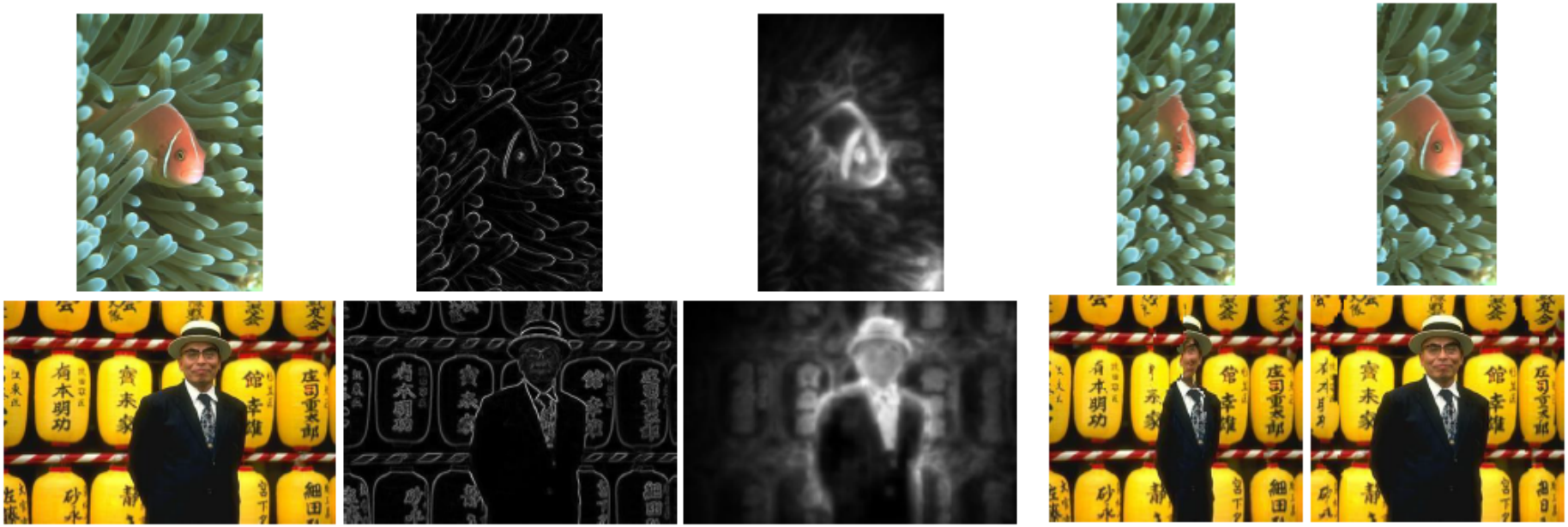
(b) Local [24]



(c) Global [7]



(d) Local-global [13]



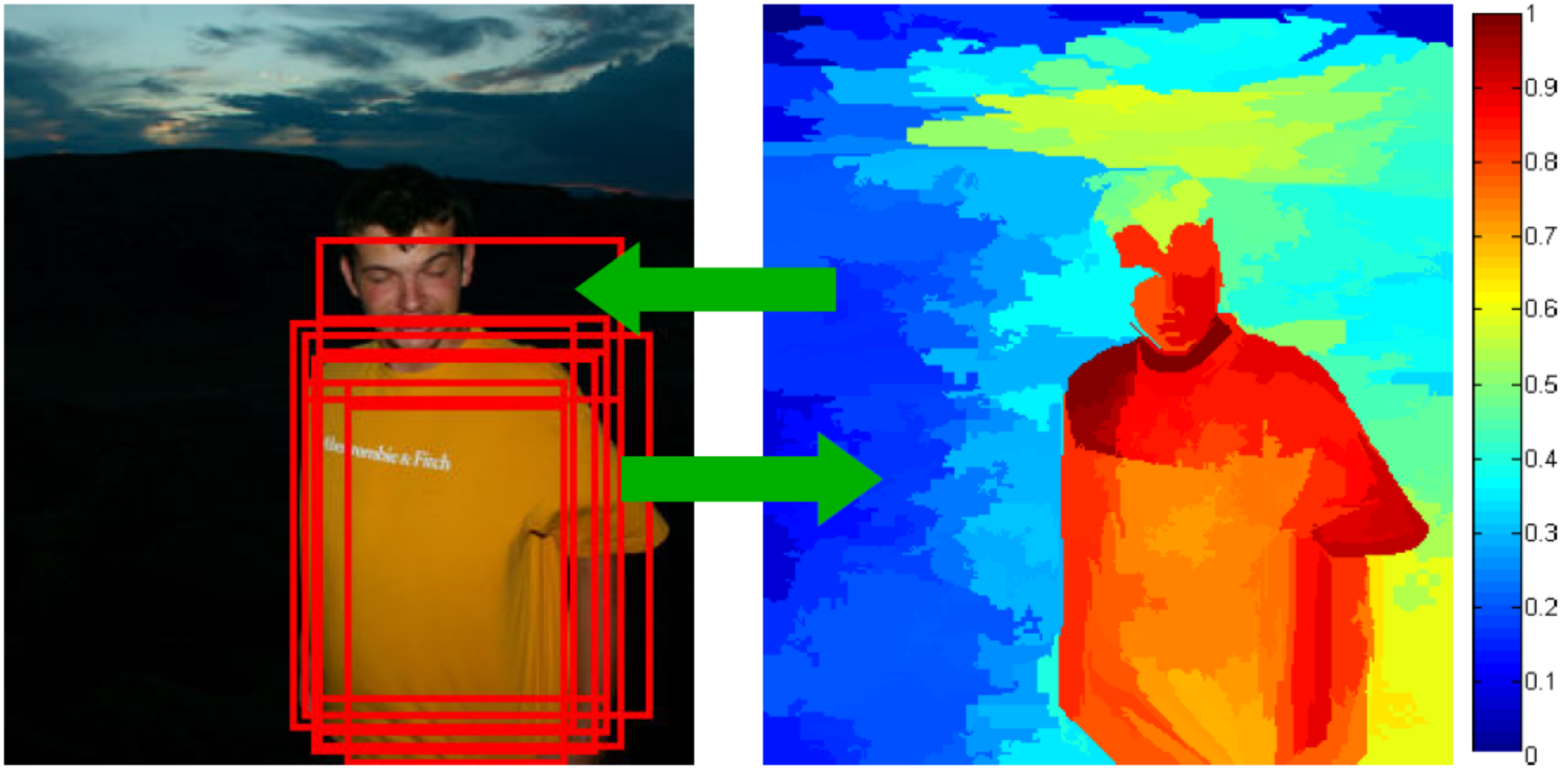
Input | Saliency of [19] | Our saliency | Results of [19] | Our result

Input | Result of [19] | Result of [13] | Our result

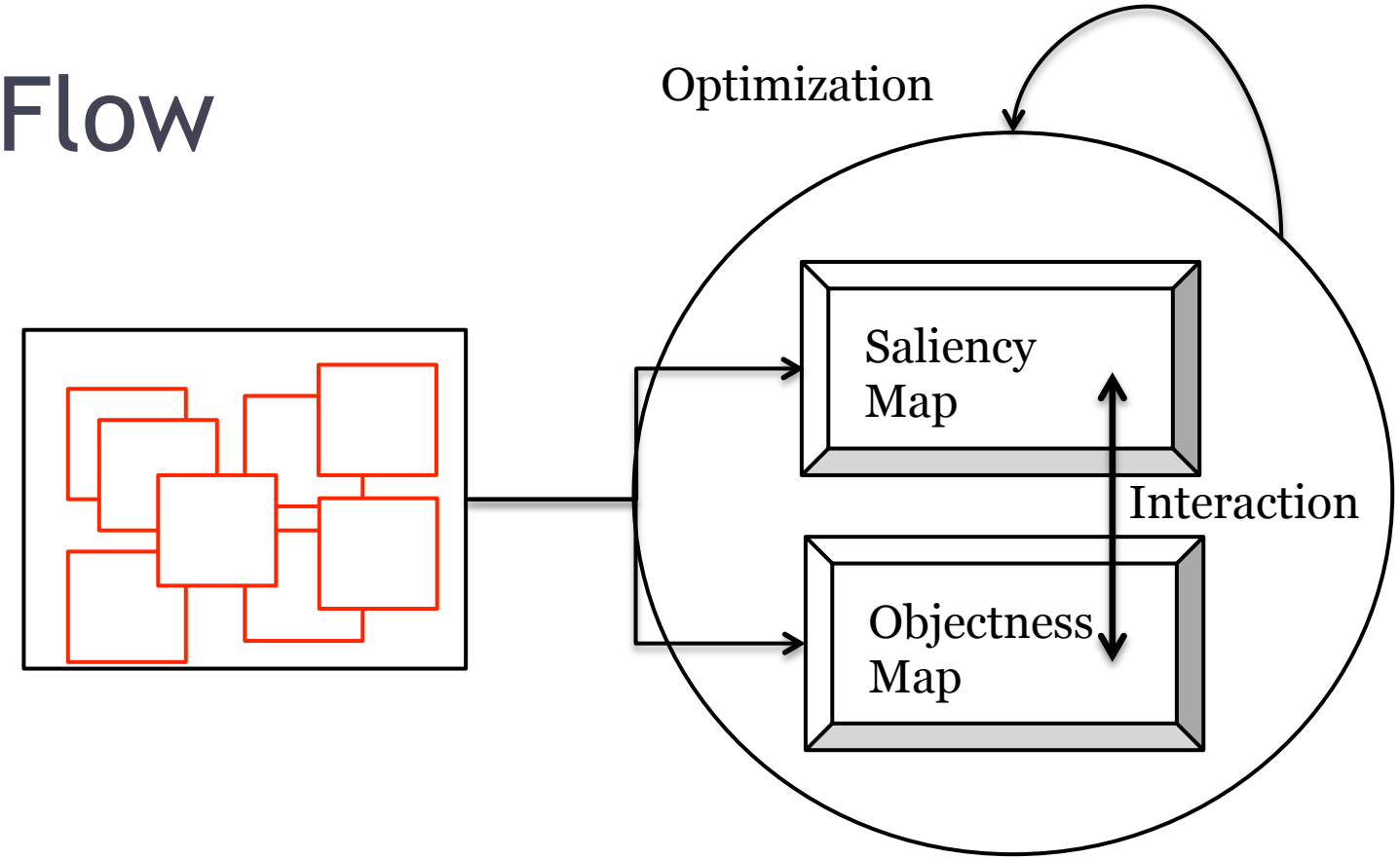
AIM

- Salient Object Detection
- Define The Relation Between Objectness and Saliency
- Improved Saliency and Objectness Results Seperately

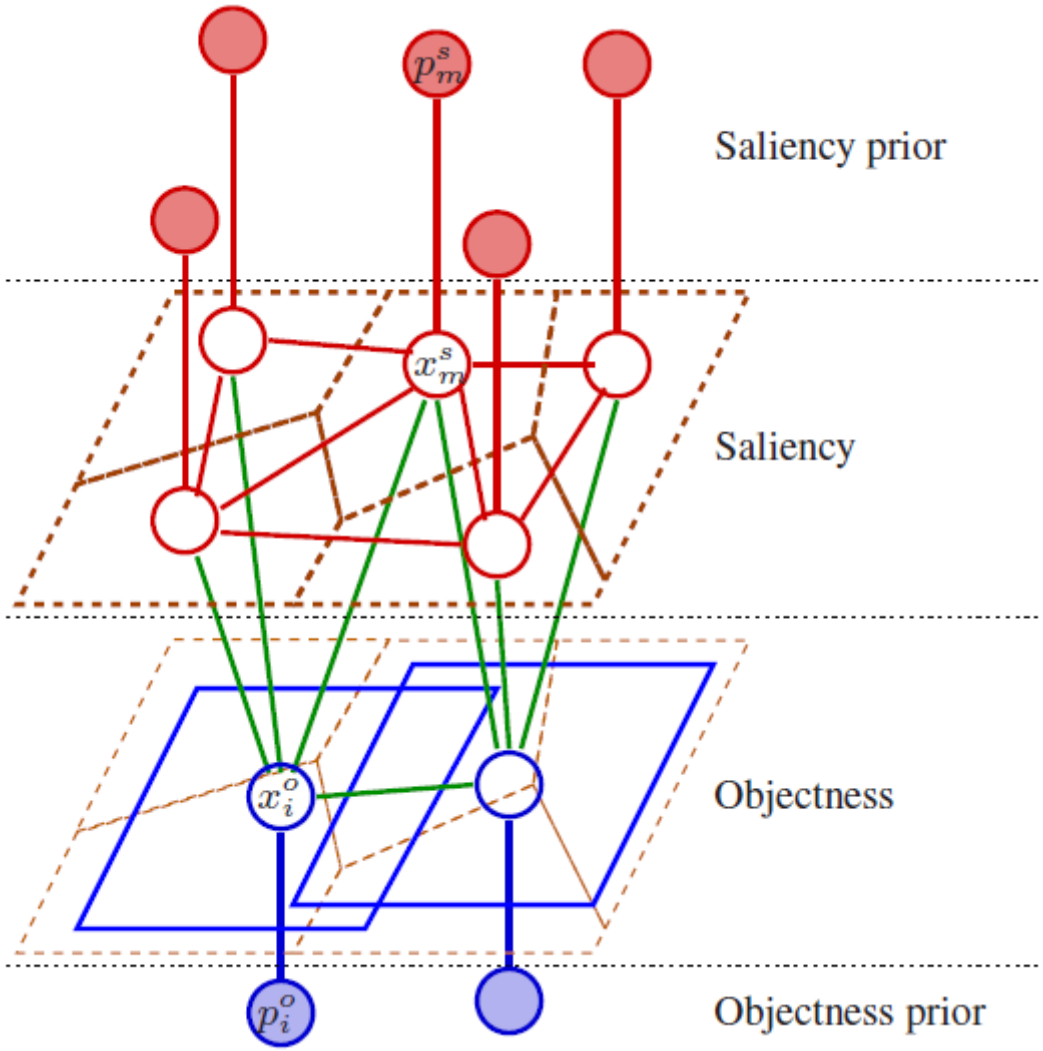
- By coupling visual saliency and generic objectness into a unified framework, the proposed approach can not only yield good performance of detecting salient objects in a scene but also concurrently improve the quality of both the saliency map and the objectness estimations.


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

Flow



Fusion


p_m^s
Saliency prior
x_m^s
Saliency
x_i^o
Objectness
p_i^o
Objectness prior

Method

$$F(\mathbf{x}^s, \mathbf{x}^o) = F_s(\mathbf{x}^s) + F_o(\mathbf{x}^o) + \Delta(\mathbf{x}^s, \mathbf{x}^o) \qquad (1)$$

- P superpixels and Q potential object windows
- Saliency x_m^s
- Objectness x_i^o

- Fs includes the energy affected only by saliency
- Fo contains the energy affected only by objectness
- Δ models the interactions between saliency and objectness

Saliency Energy [using 5]

$$F_s(\mathbf{x}^s) = \sum_m (p_m^s - x_m^s)^2 + \lambda_s \sum_{m,n \in \mathcal{E}} w_{m,n}(x_m^s - x_n^s)^2 \quad (2)$$

- Weight of the smoothness term λ_s
- Set containing the pairs of adjacency superpixels $\mathcal{E}$
- Affinity between superpixels m and n given by :

$$w_{m,n} = \sum_{(k,l) \in B_{m,n}} \exp\left(-\sigma \|\mathbf{v}_k - \mathbf{v}_l\|^2\right) \quad (3)$$

- $\mathbf{v}_k$ and $\mathbf{v}_l$ are respectively the RGB values of pixels k and l
- $B_{m,n}$ adjacent pixels pairs accross superpixels m,n
- Similar saliency for similar superpixels!

Objectness Energy [using 3]

$$F_o(\mathbf{x}^o) = \lambda_o \sum_i (p_i^o - x_i^o)^2 \qquad (4)$$

- Weight of the objectness energy λ_o
- Prior knowledge p_i^o about the objectness of each window i

- However, among other image features, the detector also uses the saliency cue. It implies that a direct application of such an objectness detector would be inappropriate to our formulation. We exploit the fact that the detector is formed by a naive Bayes model where each cue is considered independently, and **modify it by removing the saliency cue** in all our experiments.

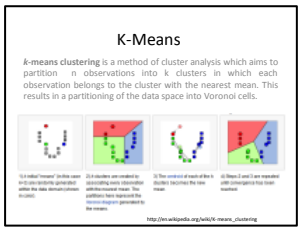
Interaction Energy

- **Definition 1** *Given a window* i, *its object-level saliency* **ci ∈ [0, 1]** *is said to measure the degree of the difference of a specific feature distribution between the center (inside the window) and the surround (around the window)*

We define the area covered by superpixels that fall mostly inside a given window (≥ 80% in our experiments) as the **center area**, and the area formed by the neighboring superpixels around the center area as **surround**.

$$\chi^2(\mathbf{h}_{i,c}, \mathbf{h}_{i,s}) = \sum_k \frac{(\mathbf{h}_{i,c}(k) - \mathbf{h}_{i,s}(k))^2}{(\mathbf{h}_{i,c}(k) + \mathbf{h}_{i,s}(k))/2}. \qquad (5)$$

- $\mathbf{h}_{i,c}$ and $\mathbf{h}_{i,s}$ respectively represent the distributions of its center and surround areas
- K-Means (K=20)



- x^2 $0 \rightarrow \infty$; rescale to [0,1]

$$c_i = \frac{1}{1 + \exp\left(-(\chi^2(\mathbf{h}_{i,c}, \mathbf{h}_{i,s}) - \bar{\chi}^2)\right)} \qquad (6)$$

- Altogether of distributions, a topdown view about the saliency of m:

$$\tau_m^s = \frac{\sum_{\{i|m\in w_i\}} c_i x_i^o}{\sum_{\{i'|m\in w_{i'}\}} x_{i'}^o}. \qquad (7)$$

- τ_m^s is a (normalized) sum of object-level saliency values weighted by their respective objectness
- interaction energy: (λ is weight)

$$\Delta(\mathbf{x}^s, \mathbf{x}^o) = \lambda \sum_m (\tau_m^s - x_m^s)^2 \qquad (8)$$

Optimization

$$\|\mathbf{p}^s - \mathbf{x}^s\|^2 + \lambda_s \mathbf{x}^{s\mathrm{T}} L \mathbf{x}^s + \lambda \|\boldsymbol{\tau}^s - \mathbf{x}^s\|^2 \qquad (9)$$

$$\mathbf{x}^s = (\lambda_s L + (1+\lambda) I)^{-1} \cdot (\mathbf{p}^s + \lambda \boldsymbol{\tau}^s) \qquad (10)$$

Laplacian Matrix

$$\lambda_o \|\mathbf{p}^o - \mathbf{x}^o\|^2 + \lambda \|C\mathbf{x}^o - \mathbf{x}^s\|^2 \qquad (11)$$

$$C(m,i) = \frac{c_i}{\sum_{\{i' | m \in w_i\}} \tilde{x}^o_{i'}} \times \delta[m \in w_i] \qquad (12)$$

Experiment

- Objectness dataset = Liu [14] $\mathcal{B}$
- Saliency dataset = MIT set (Judd [12])

- 10.000 windows
- $\lambda\downarrow s$ =1/64 (2)
- $\lambda\downarrow o$ = 1/40 (4)
- λ = **16** (8 - interaction)

- Intel i7, 30 seconds per image

Measure

- Average Precision: the area under the recall-precision curve

$$\left(\sum_i \frac{\sum_{j=1}^{i} \mathbf{g}(\mathbf{r}(j))}{i} \times \mathbf{g}(\mathbf{r}(i))\right) \Big/ \left(\sum_{i'} \mathbf{g}(\mathbf{r}(i'))\right). \tag{15}$$

- mean Average Precision **mAP**

Ours-Rect & Ours-SP

- Ours-Rect and Ours-SP. They differ in how the center-surround areas are decided for computing the window-wise object-level saliency. The former uses a conventional center-surround layout based on two rectangles, while the latter adopts the superpixel-based scheme described in Section 3.3.

Method	Size	mAP-Gaussian	mAP-SP
[11]	200×200	0.2692	0.2481
[2]	Whole image	0.2007	0.1963
[8]	32×32	0.2931	0.2782
[5]	100×100	0.3885	0.3697
[12]	200×200	**0.4536**	0.4176
Ours-Rect	Whole image	0.3934	0.4177
Ours-SP	Whole image	0.4076	**0.4284**

Table 1. Saliency detection results with respect to two ground-truth settings of saliency maps derived by smoothing the fixation maps with a Gaussian filter (mAP-Gaussian) or superpixels (mAP-SP).

T_o	Positive windows	[3]	[3]\Saliency	Objectness $\{x_i^o\}$		Salient objectness $\{c_i \times x_i^o\}$	
				Ours-Rect	Ours-SP	Ours-Rect	Ours-SP
0.5	31.56%	0.5082	0.4938	0.5114	**0.5224**	0.5120	**0.5358**
0.6	15.07%	0.3353	0.3292	0.3435	**0.3567**	0.3420	**0.3934**
0.7	5.11%	0.1797	0.1806	0.1877	**0.1998**	0.1847	**0.2579**
0.8	1.01%	0.0658	0.0685	0.0698	**0.0770**	0.0696	**0.1383**
0.9	0.06%	0.0127	0.0130	0.0131	**0.0149**	0.0148	**0.0442**

Table 2. Results of objectness estimations in mean average precision (mAP).

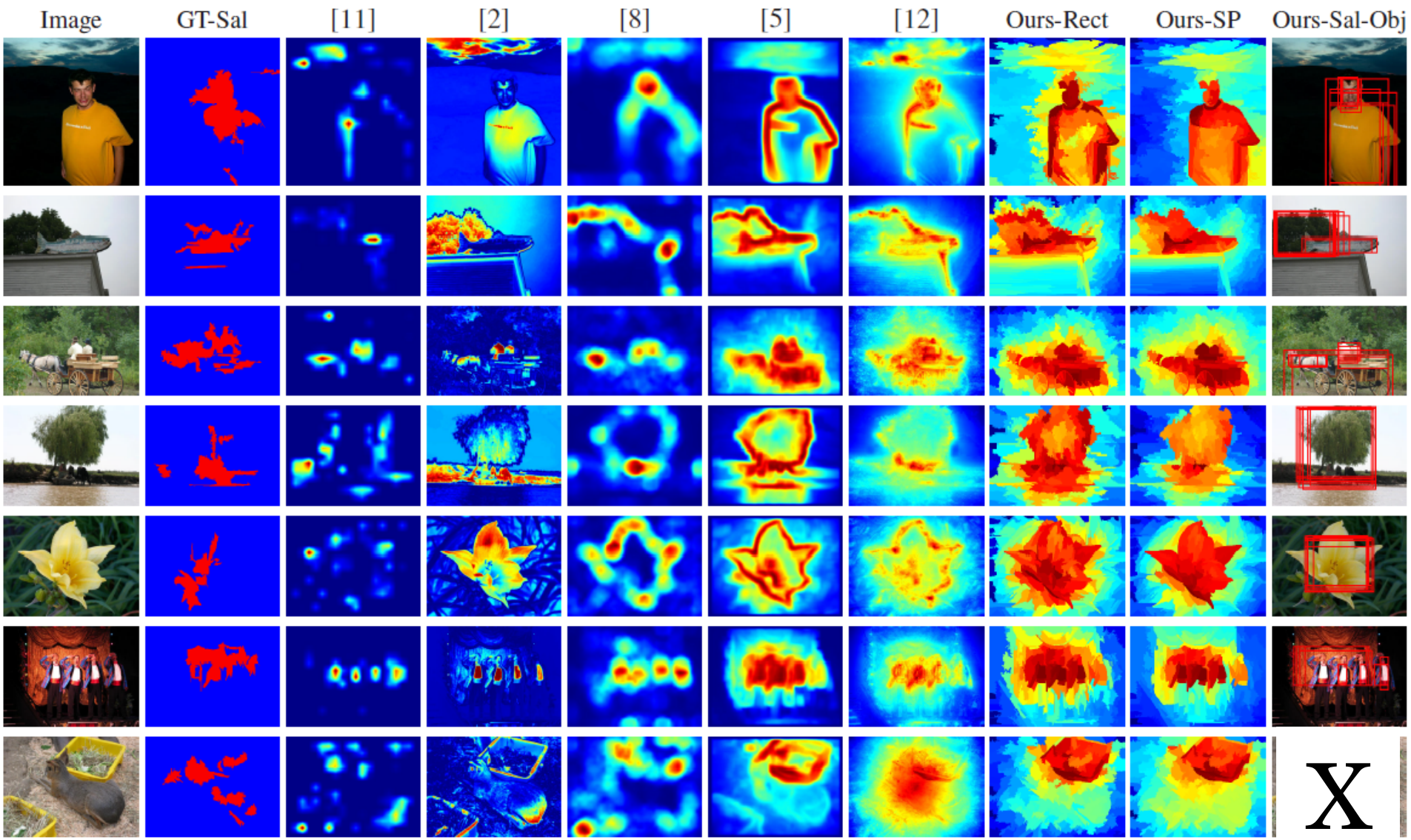


Image
GT-Sal
[11]
[2]
[8]
[5]
[12]
Ours-Rect
Ours-SP
Ours-Sal-Obj

PROS - CONS

- Novel Idea
  - Attacking to Correlation between two know calculations
- Wide Range of Use
- Fast
- Easy to Use
- Object Detection is Better
- Good Comparision, Easy to Understand

- Not ahead of Learning Based Saliency !

CONCLUSION

- Combination of two major aspects
- Would you like to use it ?

- Questions =(
- Thanks =)

Conditional Random Field

- Conditional random fields (CRFs) are a **probabilistic framework for labeling and segmenting structured data**, such as sequences, trees and lattices. The underlying idea is that of defining a conditional probability distribution over label sequences given a particular observation sequence, rather than a joint distribution over both label and observation sequences. The primary advantage of CRFs over hidden Markov models is their conditional nature, resulting in the relaxation of the independence assumptions required by HMMs in order to ensure tractable inference. Additionally, CRFs avoid the label bias problem, a weakness exhibited by maximum entropy Markov models (MEMMs) and other conditional Markov models based on directed graphical models. CRFs outperform both MEMMs and HMMs on a number of real-world tasks in many fields, including bioinformatics, computational linguistics and speech recognition.

http://www.inference.phy.cam.ac.uk/hmw26/crf/