Salient Object Detection Via Learning Method

Huchuan Lu

Dalian University of Technology

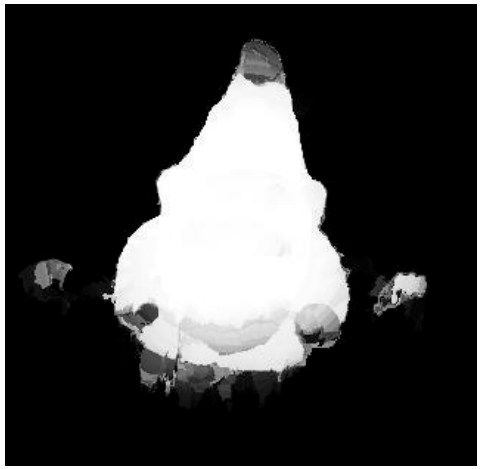
Outline

- Introduction
- Application
- Methodology
- Proposed Methods

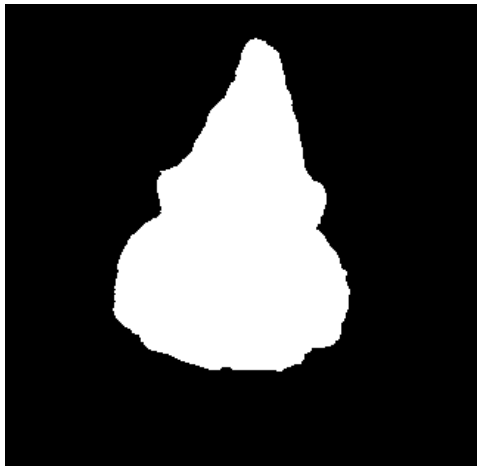
Goal of salient object detection

- **Definition**: Visual saliency is concerned with the distinct perceptual quality of biological systems which makes certain regions of a scene stand out from their neighbors and catch immediate attention.
- **Goal**: Estimate where the interested object may appear in the input image. Identify the most important and informative part of a scene.

Input Image



Saliency Map



Ground Truth

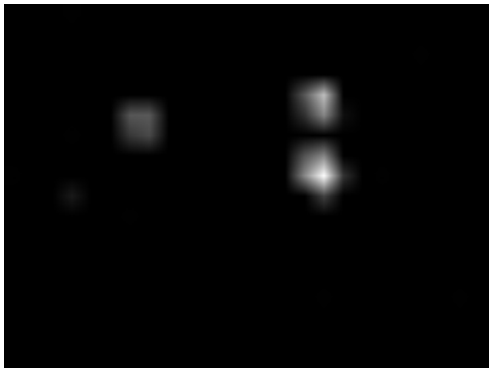
Category

- **Objective**

  — Eye Fixation Prediction

  — Salient Object Detection

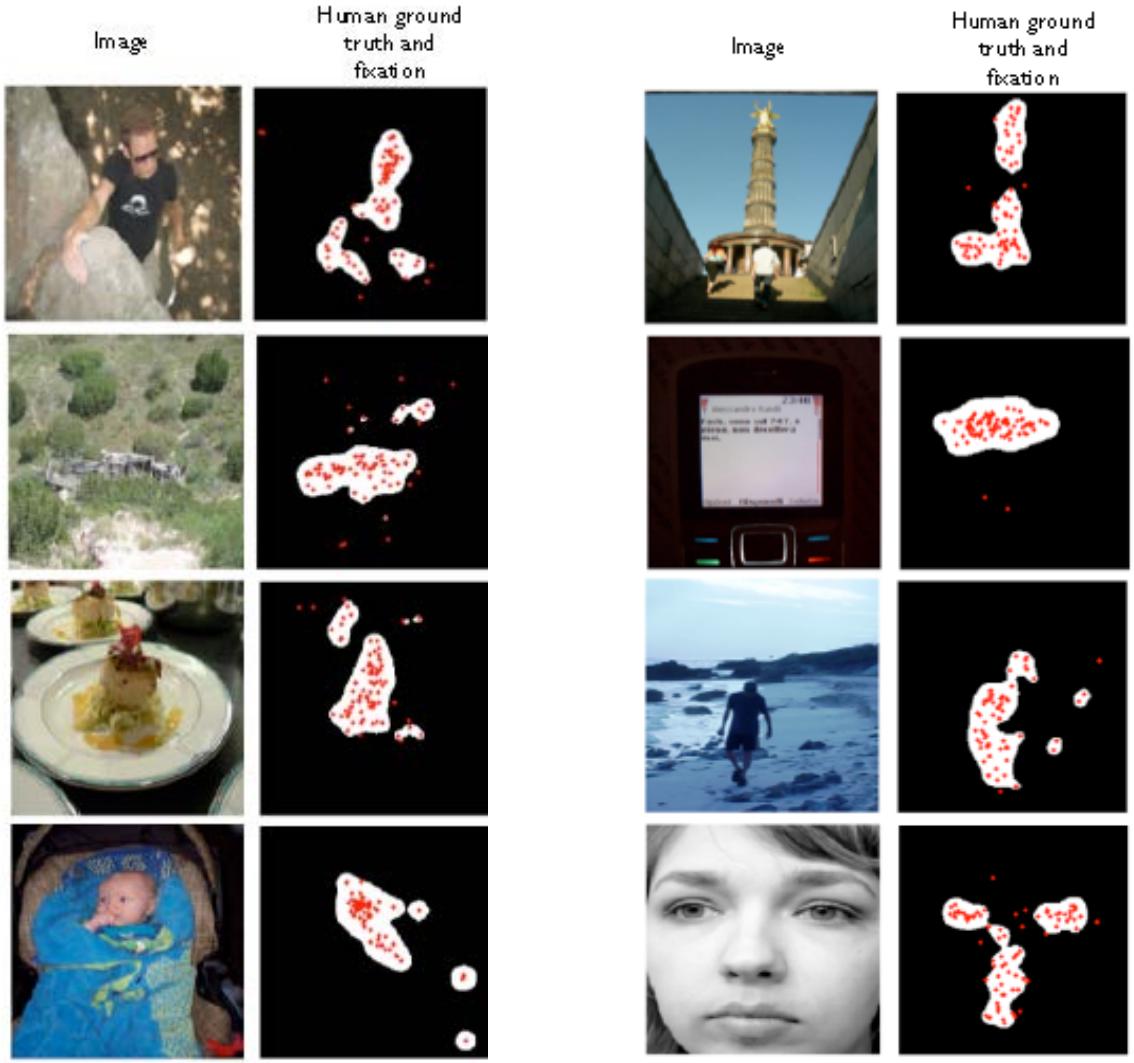
Input Image



Eye Fixation Map

Saliency Map

Eye Fixation Prediction

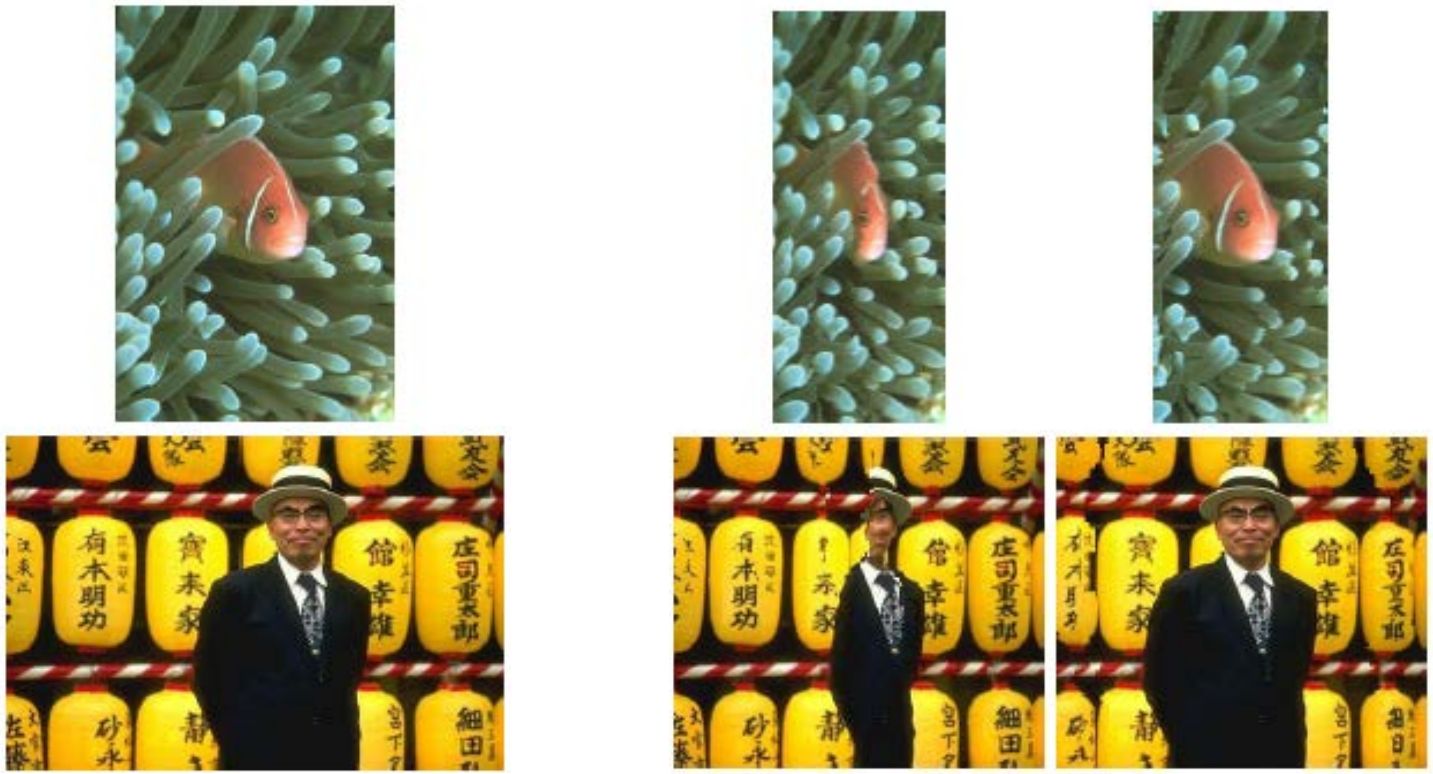


Application

- Image/video segmentation
- Image/video compression
- Object recognition
- Image category
- Interested region detection
- Image retrieval
- Image resizing
- ……

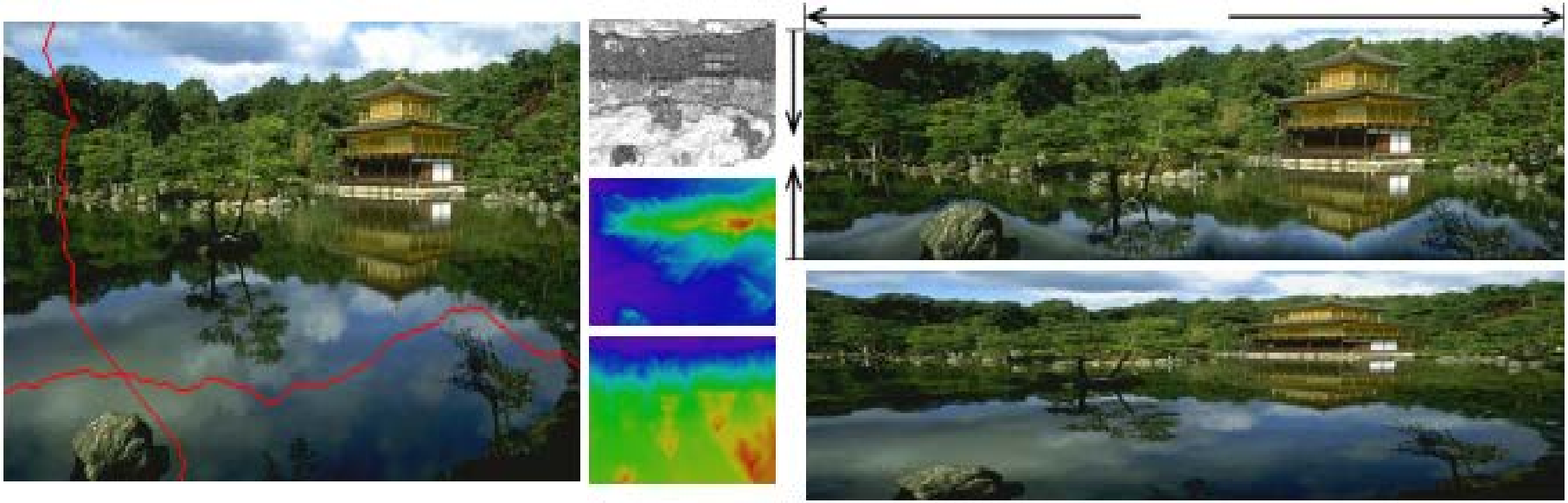
Application—Image Retargeting(Resizing)

- Image retargeting aims at resizing an image by expanding or shrinking the non-informative regions



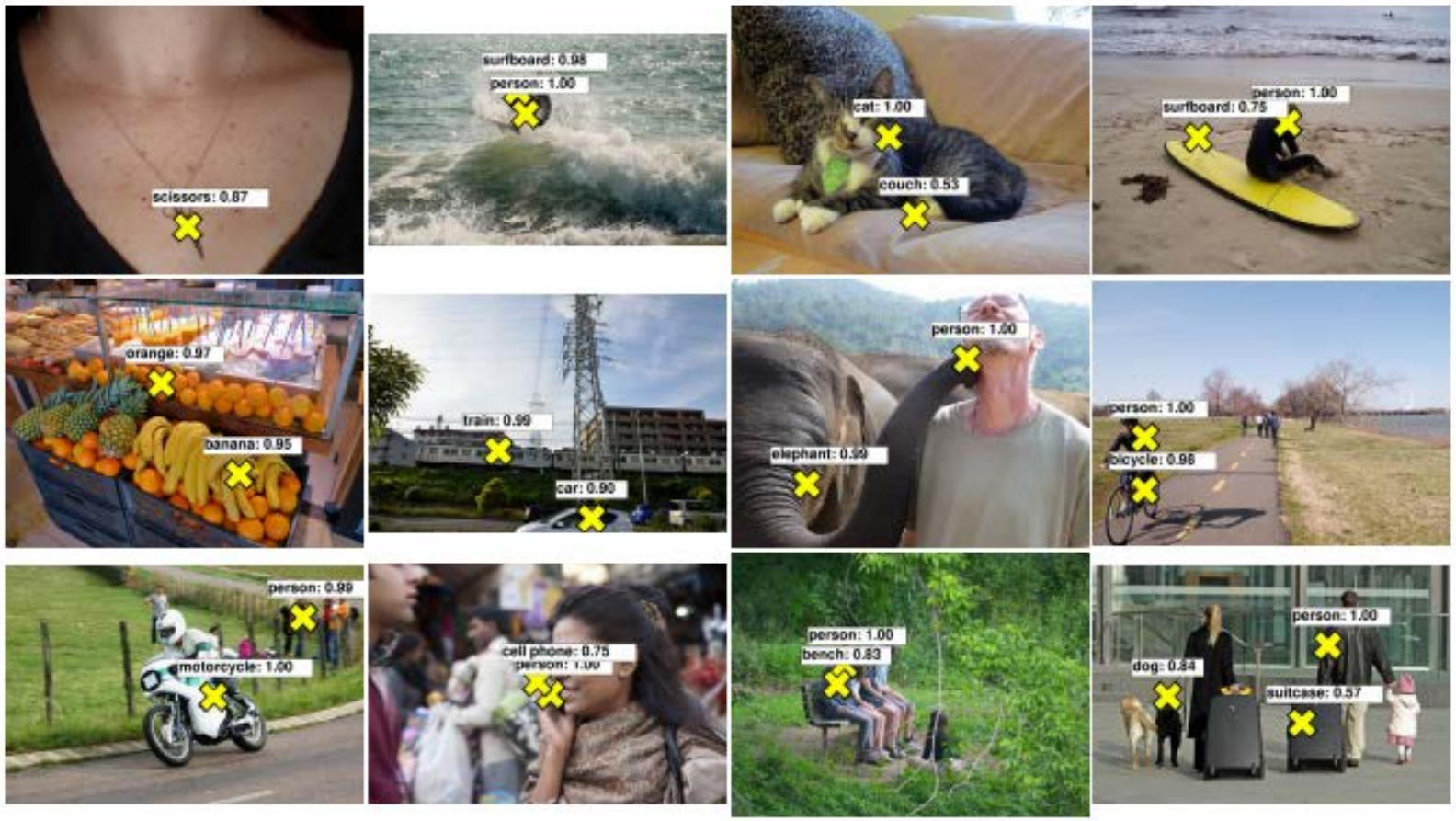
Application—Image Retargeting (Resizing)

- Comparison of different resizing results based on different saliency maps.



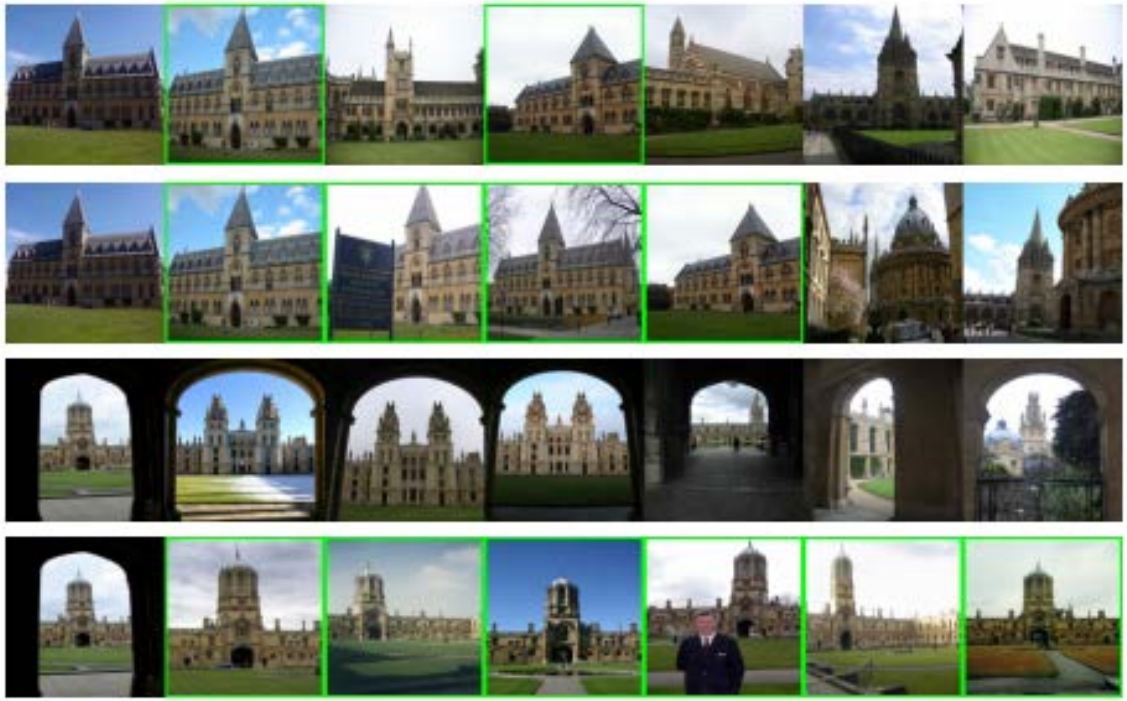
Application—Object Recognition

- task of finding and identifying objects in an image or video sequence.



Application—Image Retrieval

- An image retrieval system is a computer system for browsing, searching and retrieving images from a large database of digital images.



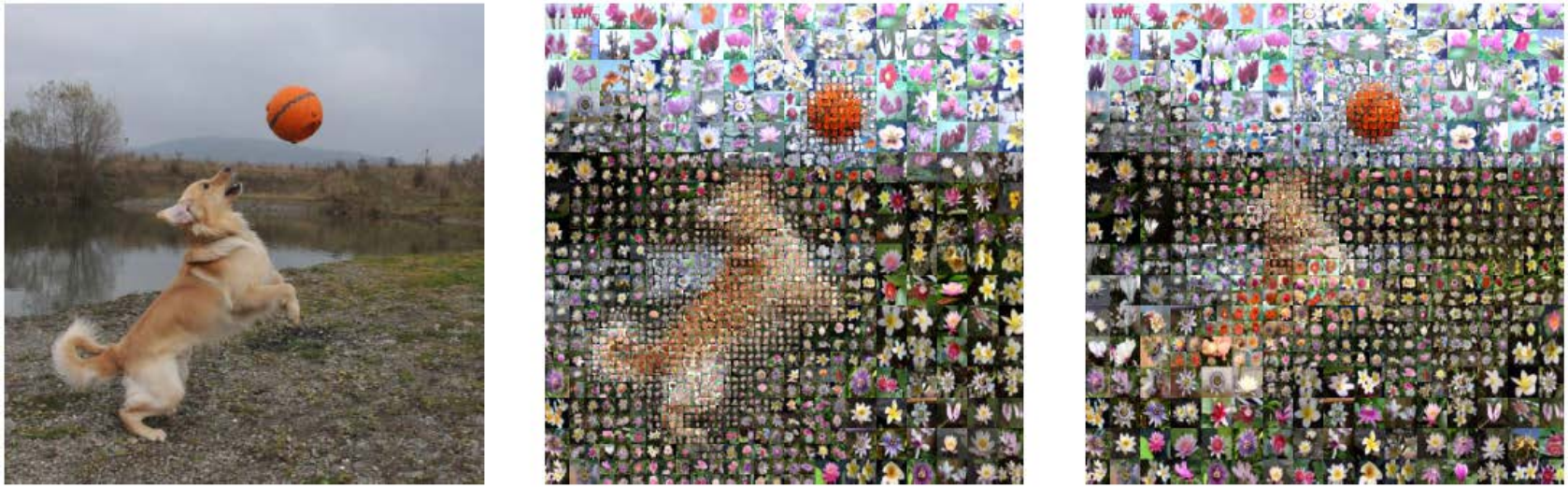
From left to right: query and results.
The correct answers are outlined in green.

Application—Non-photorealistic rendering

- Artists often abstract images and highlight meaningful parts of an image while masking out unimportant regions. Inspired by this observation, a number of non-photorealistic rendering (NPR) efforts use saliency maps to generate interesting effects.

Application—Image mosaicing

- Salient details are preserved with the use of smaller building blocks.



Comparison of different mosaicing results based on different saliency detection models

Application—Webpage Saliency

- How humans deploy their attention when viewing webpages and for the first time

(a) First Fixation

(b) Second Fixation

(c) Third Fixation

Application—Picture Collage

- Picture collage is a kind of visual image summary — to arrange all input images on a given canvas, allowing overlay, to maximize visible visual information.

(a) (b) (c)

(a) the collage of 14 images selected among the first five pages returned by Yahoo's image searching using the keyword "cycling mountain". (b) the collage of 16 "Panda" images from Yahoo's image search. (c) the collage formed by 12 "horse" images from Google's image search.

Application—Video Saliency

- Find out salient objects in video saliency. composite.avi

Application—Anomaly detection

- Anomaly detection means automatically identifying the items or events those are different from the expected pattern or other items in the dataset.

UCSDPed2.avi

Salient Object Detection — Methodology

- **Top-down methods**

  — slow, task-driven, knowledge-driven,

  — entail supervised learning with class labels.

- **Bottom-up methods**

  — fast, stimuli-driven, data-driven and pre-attentive,

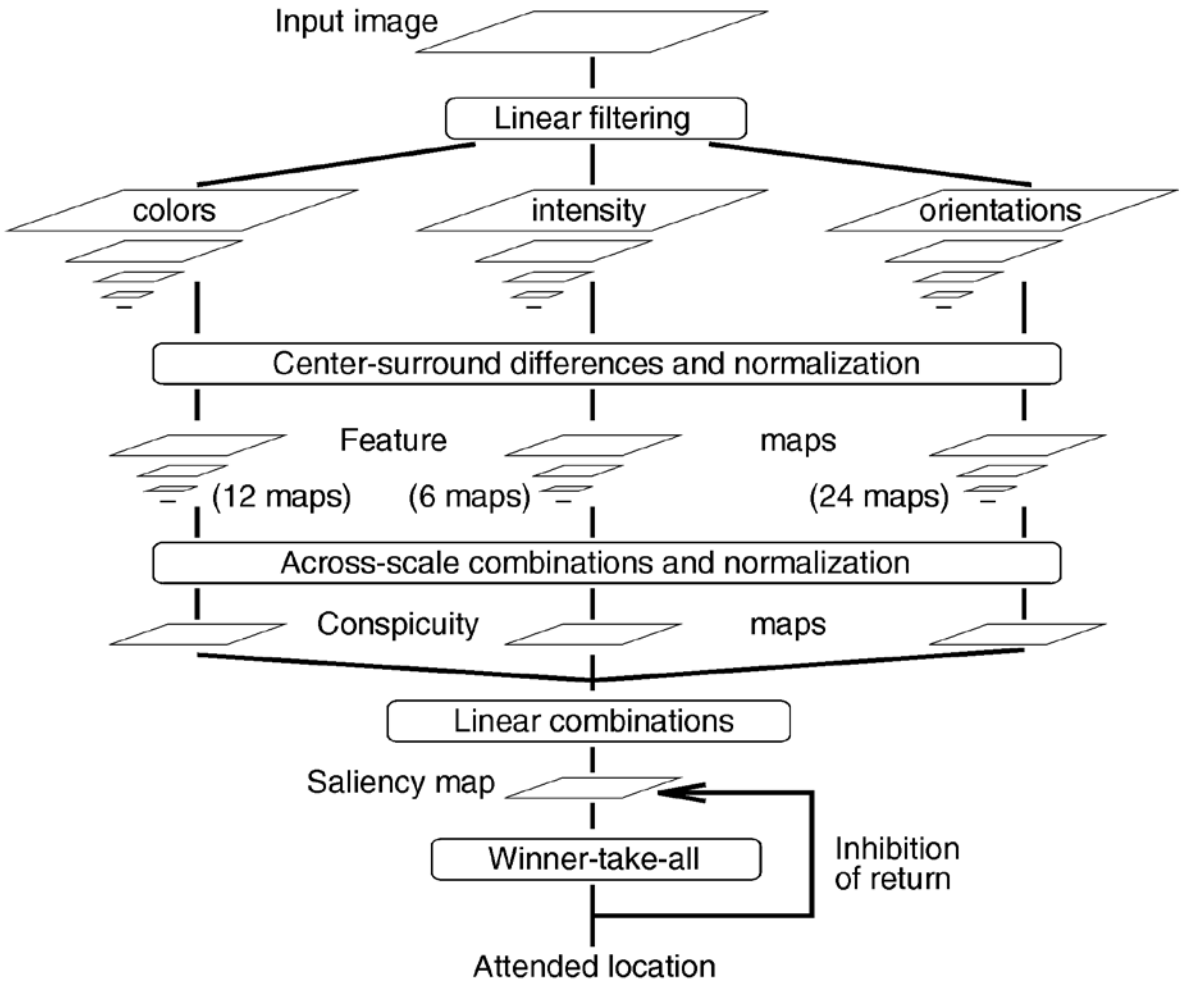
  — rely only on low level features such as color, intensity.

Methodology—Bottom-up methods

A Model of Saliency-Based Visual Attention for Rapid Scene Analysis

Laurent Itti

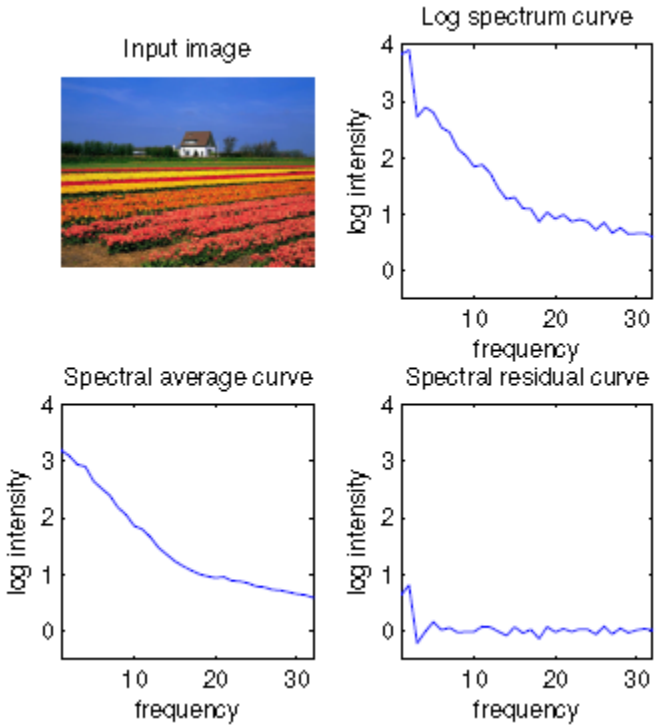
PAMI 1998

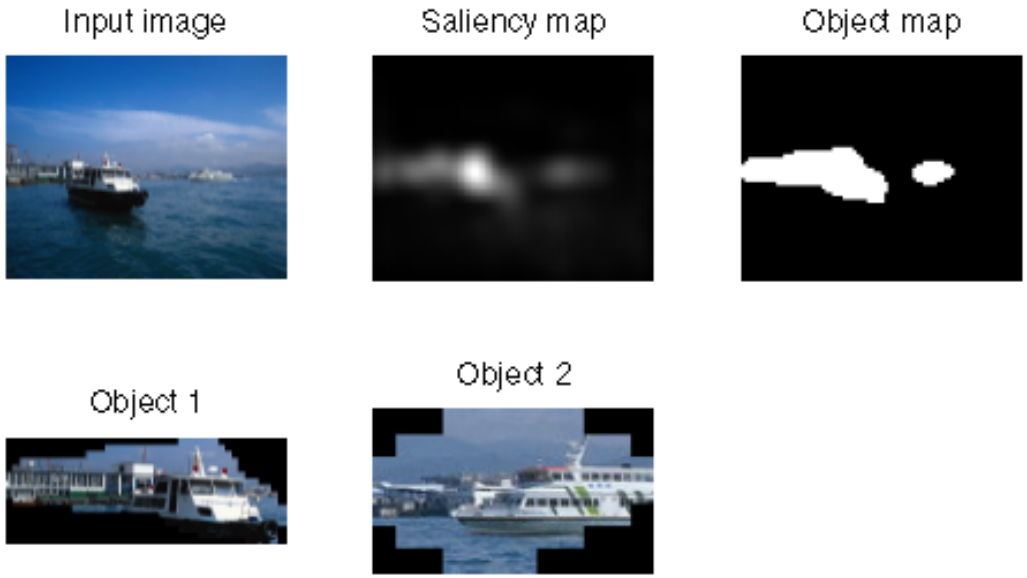


Methodology—Bottom-up methods

Saliency Detection: A Spectral Residual Approach Xiaodi Hou **CVPR 2007**

- By analyzing the log-spectrum of an input image, we extract the spectral residual of an image in spectral domain, and propose a fast method to construct the corresponding saliency map in spatial domain. Outputs full resolution saliency maps with well-defined boundaries of salient objects.





Methodology—Bottom-up methods

Global Contrast based Salient Region Detection Ming-Ming Cheng **CVPR 2011**

- Propose a regional contrast based saliency extraction algorithm, which simultaneously evaluates global contrast differences and spatial coherence.
- Simple, efficient, and yields full resolution saliency maps.

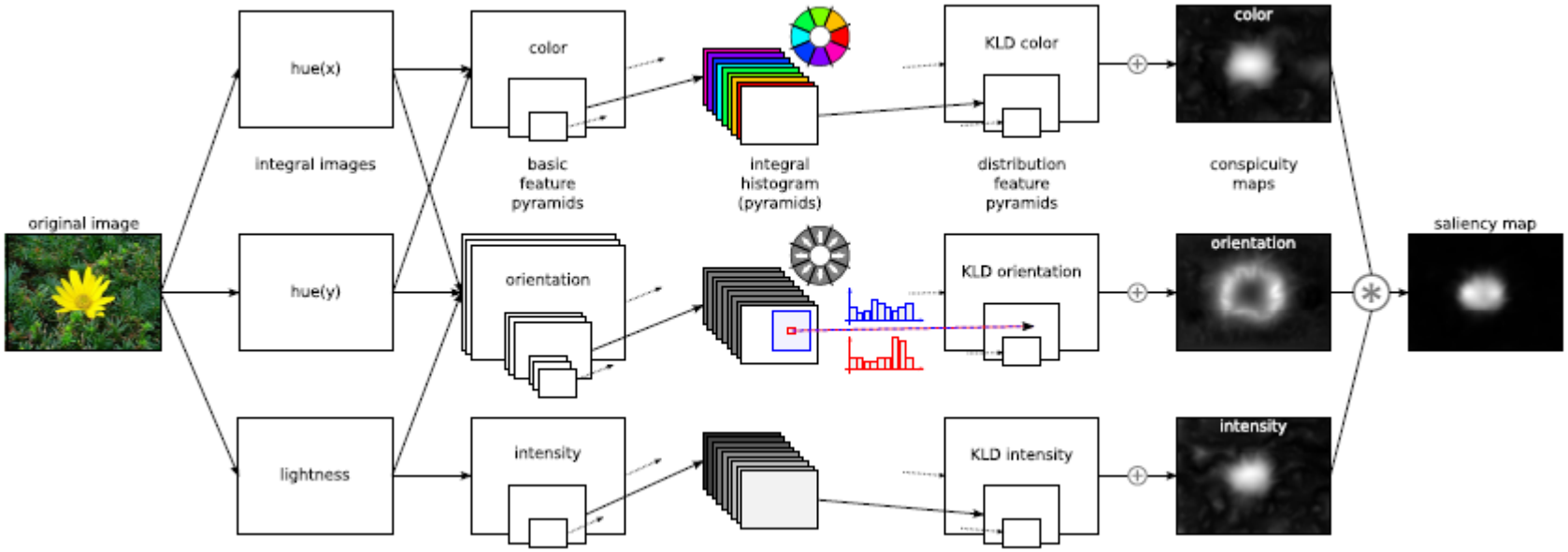
Given input images (top), a global contrast analysis is used to compute high resolution saliency maps (middle), which can be used to produce masks (bottom) around regions of interest.

Methodology—Bottom-up methods

Center-surround Divergence of Feature Statistics for Salient Object Detection Dominik A. Klein

ICCV 2011

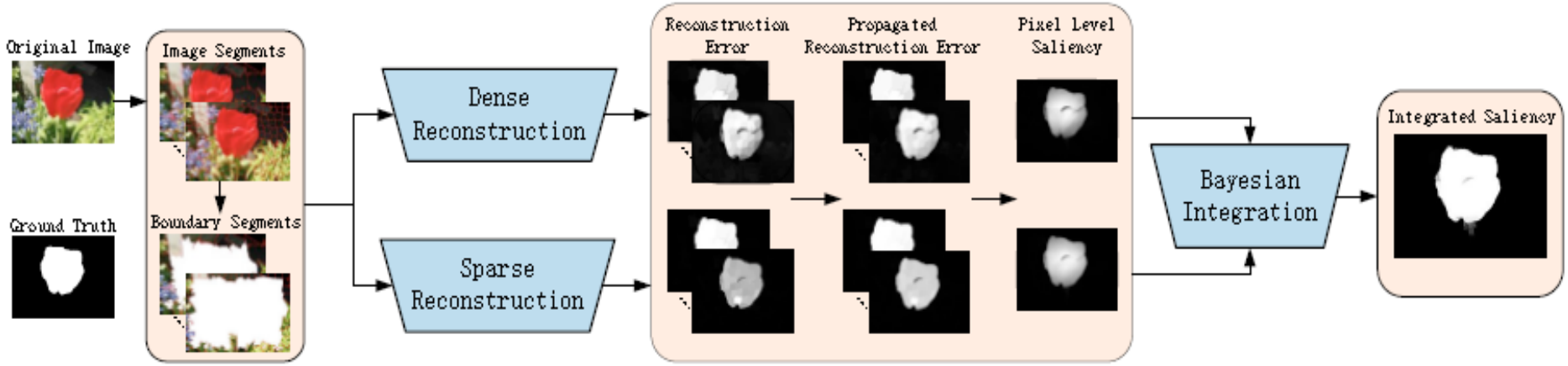
- Center - surround
- Local center - surround pairs
- Define the saliency of an image region in an information-theoretic way by means of the Kullback-Leibler-Divergence (KLD).



Methodology—Bottom-up methods

Saliency Detection via Dense and Sparse Reconstruction Xiaohui Li **ICCV 2013**

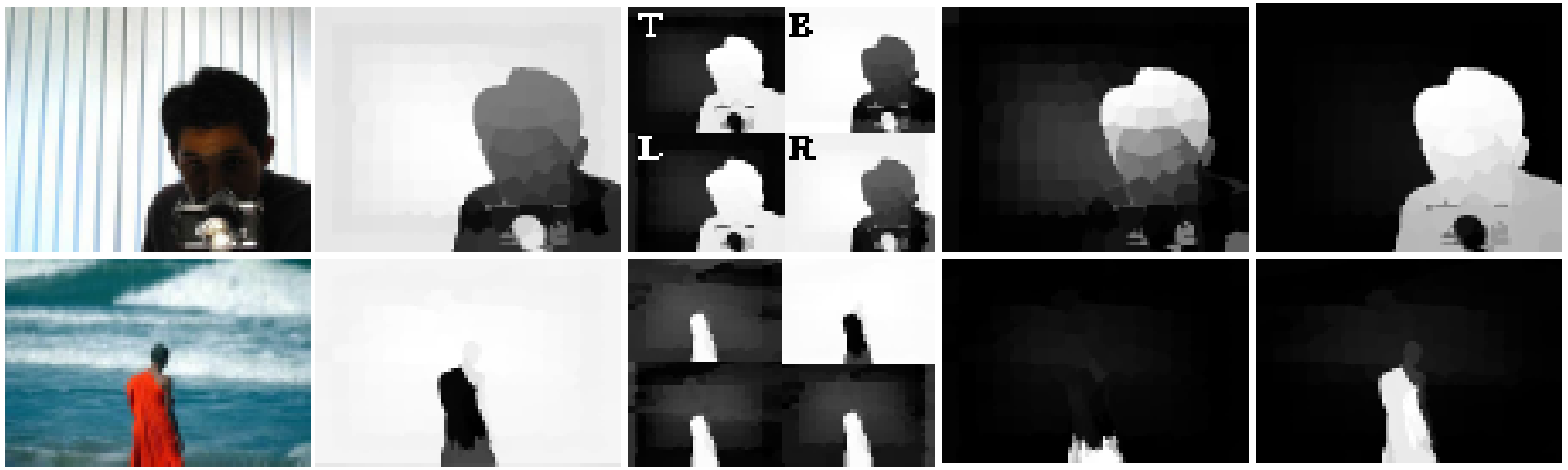
- The image boundaries are extracted via superpixels as likely cues for background templates, from which dense and sparse appearance models are constructed. Compute dense and sparse reconstruction errors.
- Apply the Bayes formula to integrate saliency measures based on dense and sparse reconstruction errors.



Methodology—Bottom-up methods

Saliency Detection via Graph-Based Manifold Ranking Chuan Yang **CVPR 2013**

- Rank the similarity of the image elements (pixels or regions) with foreground cues or background cues via graph-based manifold ranking.
- Represent the image as a close-loop graph with superpixels as nodes.



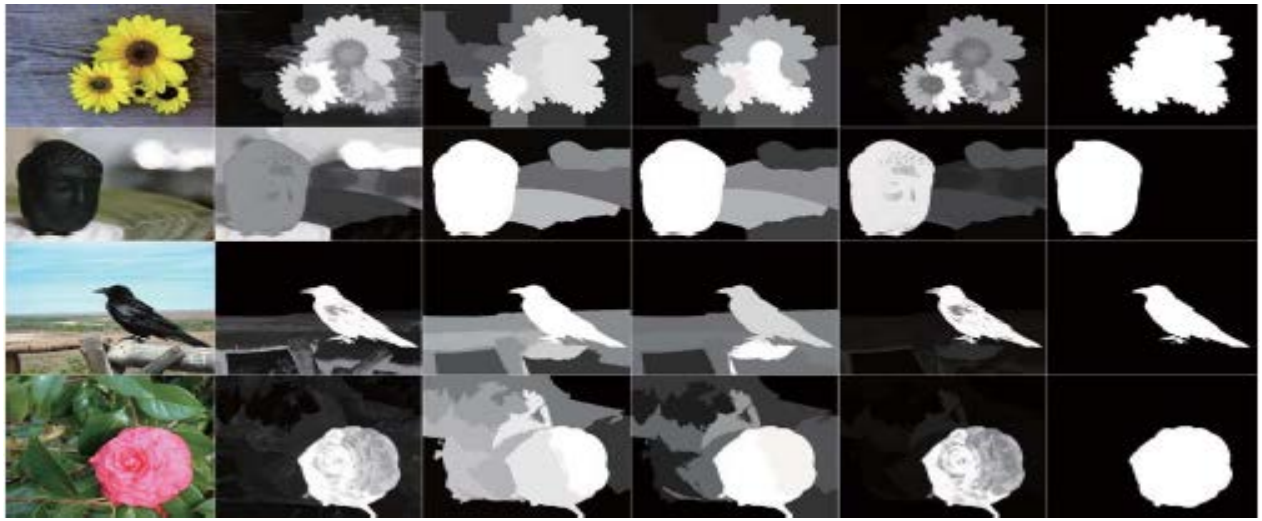
From left to right: input images, saliency maps using all the boundary nodes together as queries, four side-specific maps, integration of four saliency maps, the final saliency map after the second stage.

Methodology—Bottom-up methods

Salient Region Detection by UFO: Uniqueness, Focusness and Objectness

Peng Jiang **ICCV 2013**

- Propose an approach to detect salient region with three visual cues: uniqueness, focusness and objectness.
- Give two level estimation views of proposed three visual cues:pixel-level and region-level.

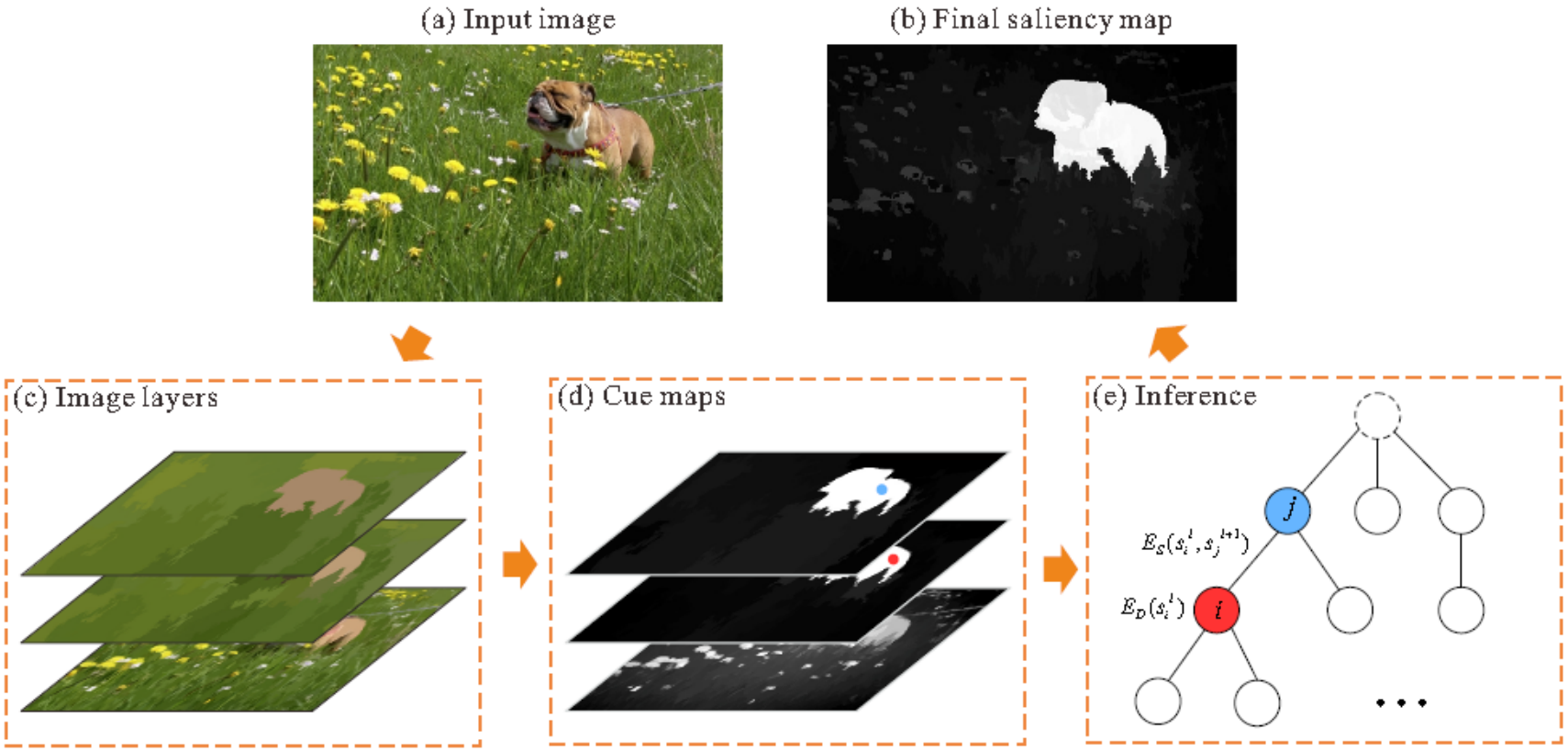


From left to right:source images, uniqueness, focusness, objectness, combined results and ground truth.

Methodology—Bottom-up methods

Hierarchical Saliency Detection Qiong Yan **CVPR 2013**

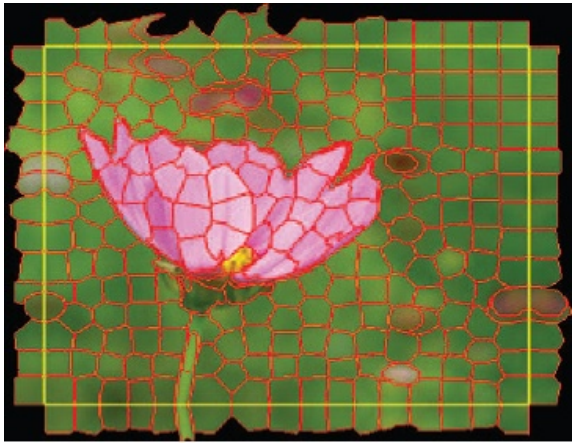
- Propose a multi-layer approach to analyze saliency cues, and the final saliency map is produced in a hierarchical model.



Methodology—Bottom-up methods

Saliency Detection via Absorbing Markov Chain Bowen Jiang **ICCV 2013**

- Formulate saliency detection via absorbing Markov chain on an image graph model.
- The virtual boundary nodes are chosen as the absorbing nodes in a Markov chain and the absorbed time from each transient node to boundary absorbing nodes is computed.



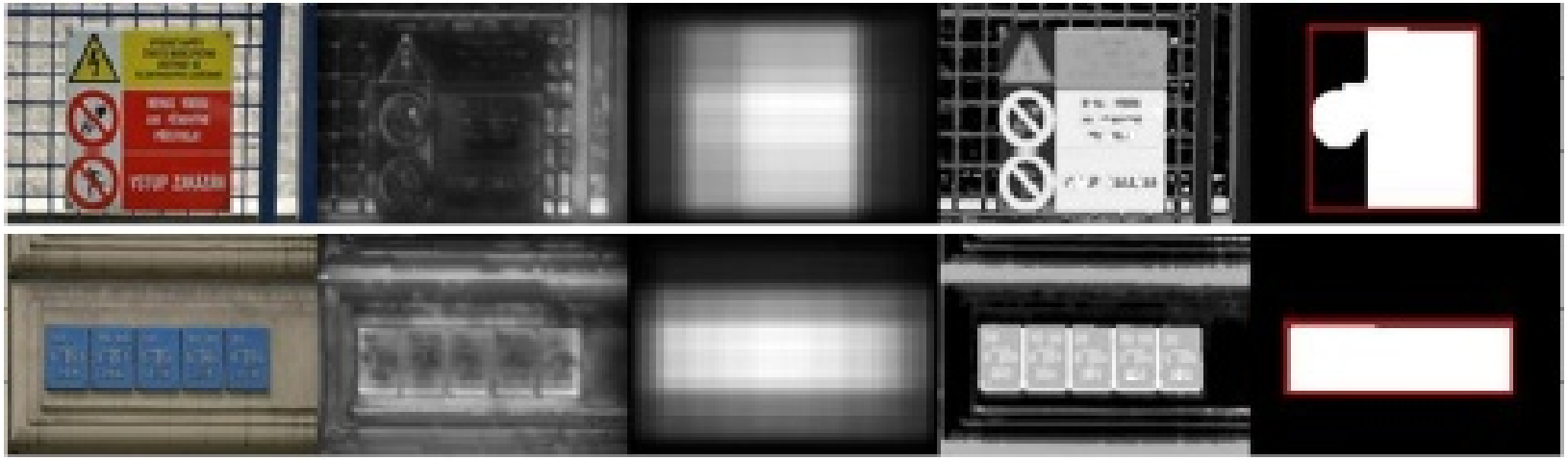
The superpixels outside the yellow bounding box are the duplicated boundary superpixels, which are used as the absorbing nodes.

From top to down: input images, saliency maps

Methodology—Top-down methods

Learning to Detect A Salient Object Tie Liu **CVPR 2007**

- Salient Object features
  - — multi-scale contrast（ locally）
  - — center-surround histogram（ regionally）
  - — color spatial distribution （globally）
- A Conditional Random Field is learned to combine features.



From left to right : Input, multi-scale contrast, center-surround histogram, color spatial distribution, and binary salient mask by CRF.

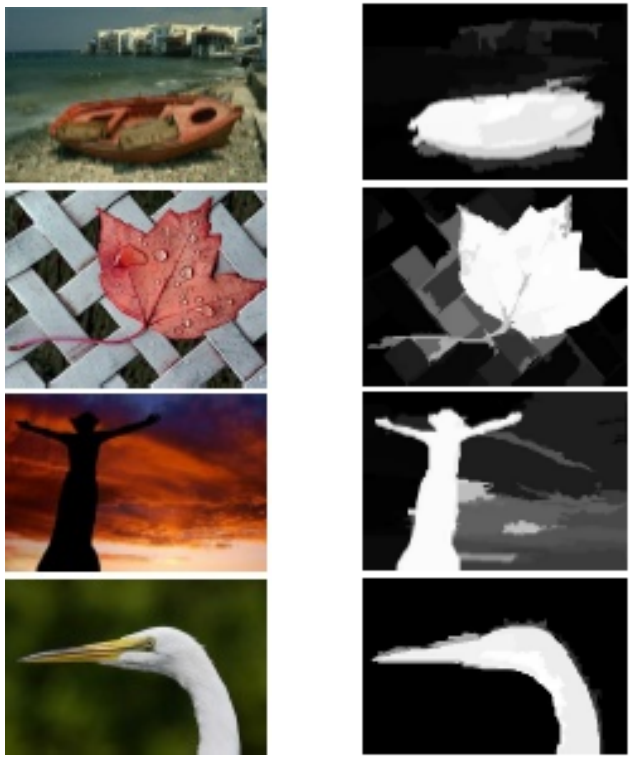
Methodology—Top-down methods

Salient Object Detection: A Discriminative Regional Feature Integration Approach Huaizu Jiang

CVPR 2013

- Regard saliency map computation as a regression problem.

- Regional features
  - — Regional contrast descriptor.
  - — Regional property descriptor.
  - — Regional backgroundness descriptor.



Input DRFI

DEEP LEARNING FOR SALIENCY

- SuperCNN: A Superpixelwise Convolutional Neural Network for Salient Object Detection (IJCV2015)

  -Shengfeng He, Rynson W. H. Lau, Wenxi Liu, Zhe Huang, Qingxiong Yang

- Deep Networks for Saliency Detection via Local Estimation and Global Search (CVPR2015)

  -Lijun Wang, Huchuan Lu, Xiang Ruan and Ming-Hsuan Yang

- Visual Saliency Based on Multiscale Deep Features (CVPR2015)

  -Guanbin Li, Yizhou Yu

- Saliency Detection by Multi-Context Deep Learning (CVPR2015)

  -Rui Zhao, Wanli Ouyang, Hongsheng Li, Xiaogang Wang

- DeepSaliency: Multi-Task Deep Neural Network Model for Salient Object Detection (arXiv)

  -Xi Li, Liming Zhao, Lina Wei, MingHsuan Yang, Fei Wu, Yueting Zhuang, Haibin Ling, and Jingdong Wang

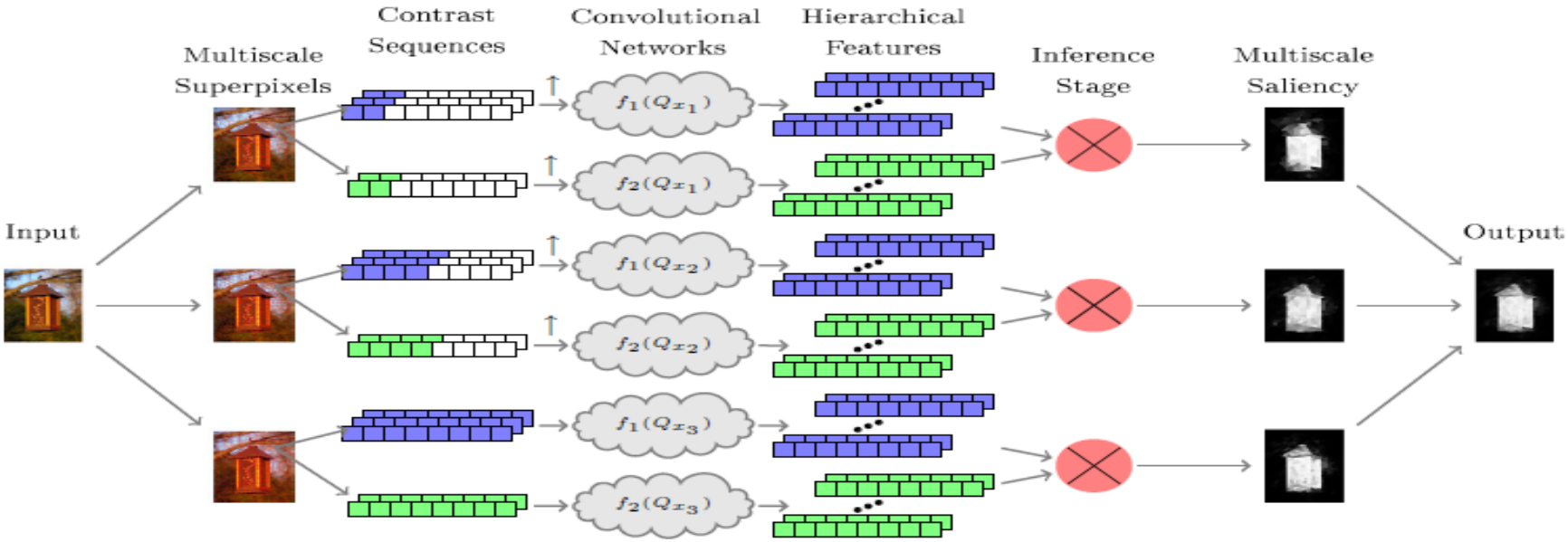
SuperCNN: A Superpixelwise Convolutional Neural Network for Salient Object Detection

Two input sequences(superpixel-based):
-Color Uniqueness Sequence(3 channels) → - f_1
-Color Distribution Sequence(2 channels) → - f_2

Two network colomns:

Detection framework:

1. SLIC
2. Feature extraction : two feature sequences
3. Saliency inference: $S(r_x) = \prod_{u\in(1,2)} v_c^u \cdot S_u(r_x)$ Classification function: Softmax
4. Multiscale saliency fusion



Deep Networks for Saliency Detection via Local Estimation and Global Search

Local Estimation:

1. Patch-based classification
2. GOP refinement:

Accuracy score: $A_i = \frac{\sum_{x,y} O_i(x,y) \times S^L(x,y)}{\sum_{x,y} O_i(x,y)}$

Coverage score: $C_i = \frac{\sum_{x,y} O_i(x,y) \times S^L(x,y)}{\sum_{x,y} S^L(x,y)}$

Confidence: $conf_i^L = \frac{(1+\beta) \times A_i \times C_i}{A_i \times C_i}$

Global Search:

1. Input x

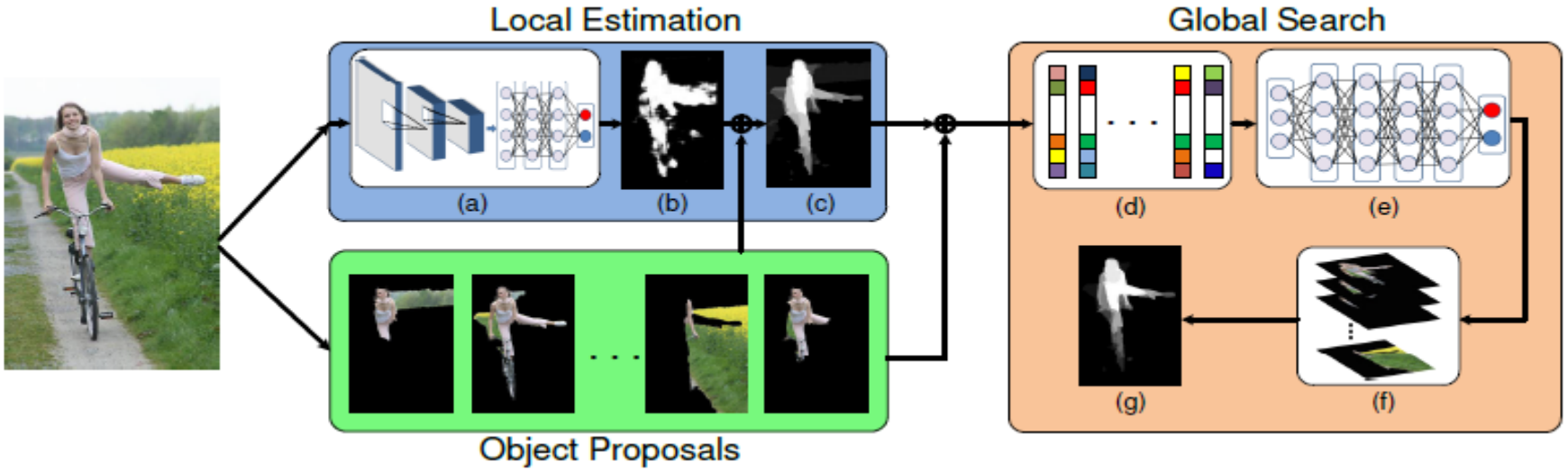
-Global contrast features

-Geometric information

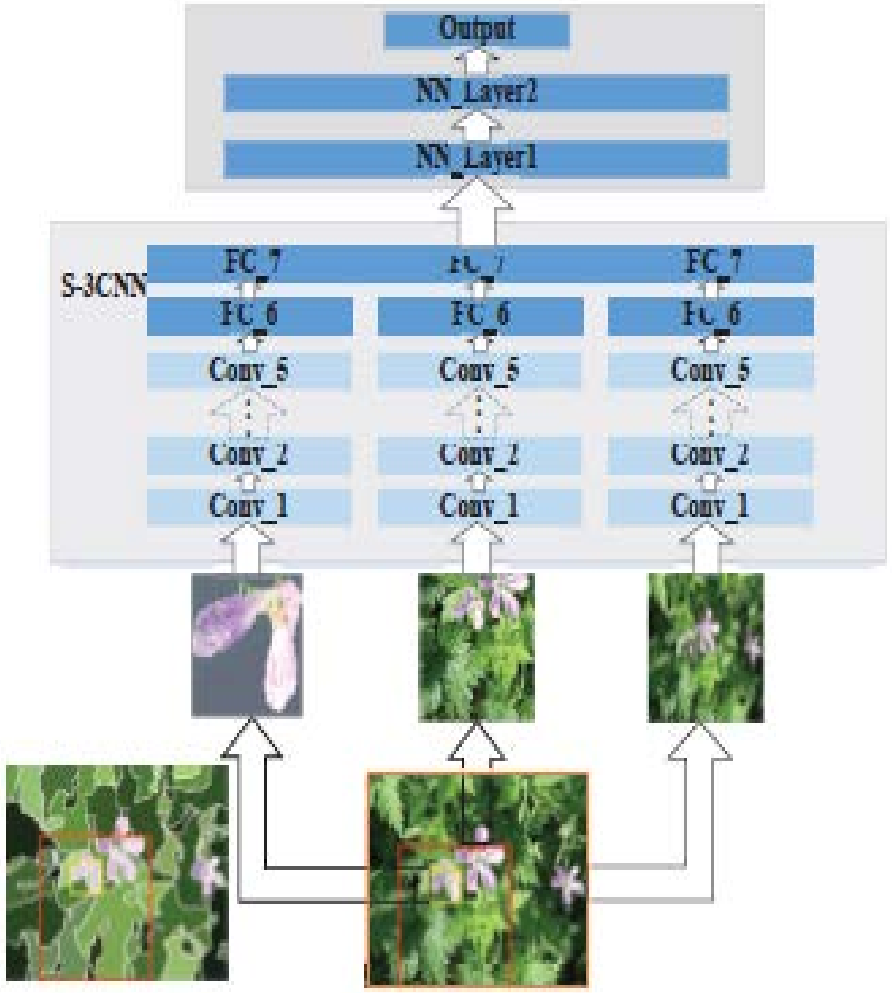
-Local saliency measurements

2. Output y

$y_i = [p_i, o_i], p_i : precision, o_i : overlap\ rete$



Visual Saliency Based on Multiscale Deep Features



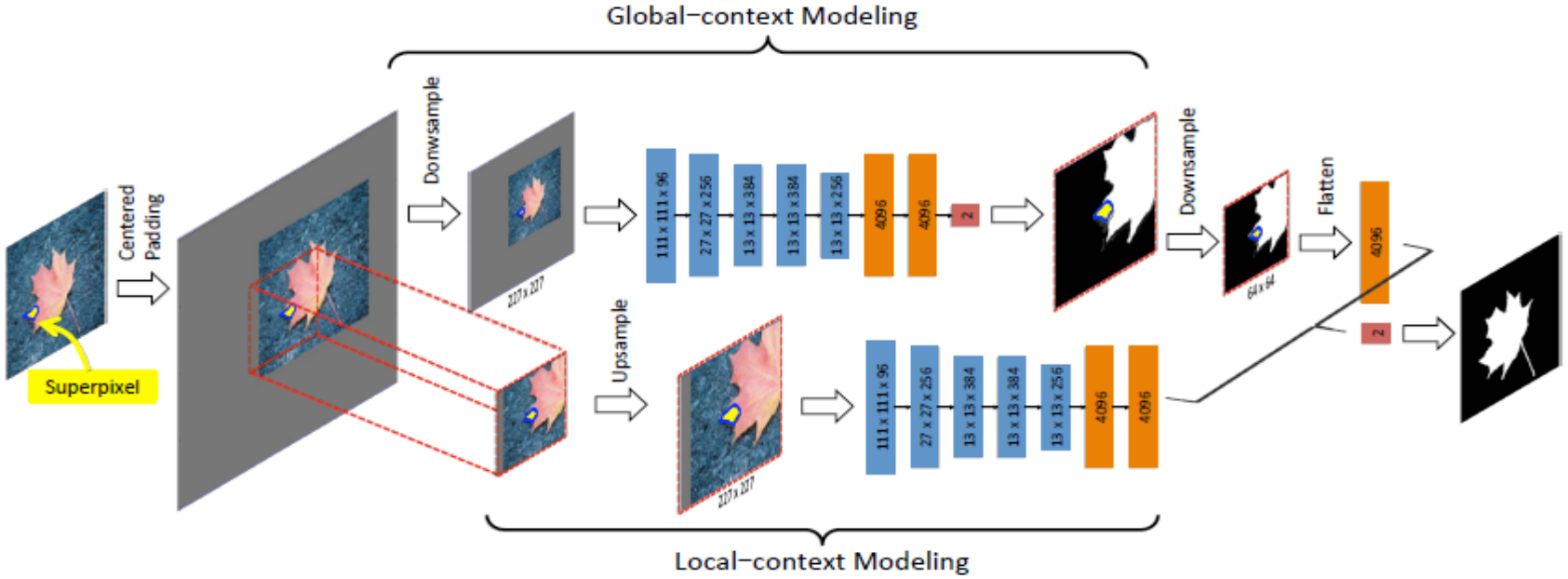
Network input:
-A: A bounding box of a region and the value outside the region is filled with the mean pixel value.
-B: A bounding box of the considered region and its immediate neighboring regions.
-C: The entire image where the onsidered region is masked with mean pixel values for indicating the position of the region.

Complete algorithm:
-Multilevel region decomposition
-Spatial coherence
-Saliency map fusion

Saliency Detection by Multi-Context Deep Learning

Task-specific pre-training:
-superpixel-level annotation: thresholding the overlap ratio between the centered superpixel and corresponding groundtruth salient object mask.
-object-level annotation: Object bounding box

Network structure:
-Global-context model: robustly model saliency with few large errors
-Local-context model: refine the saliency prediction of the centered superpixel.



DeepSaliency: Multi-Task Deep Neural Network Model for Salient Object Detection

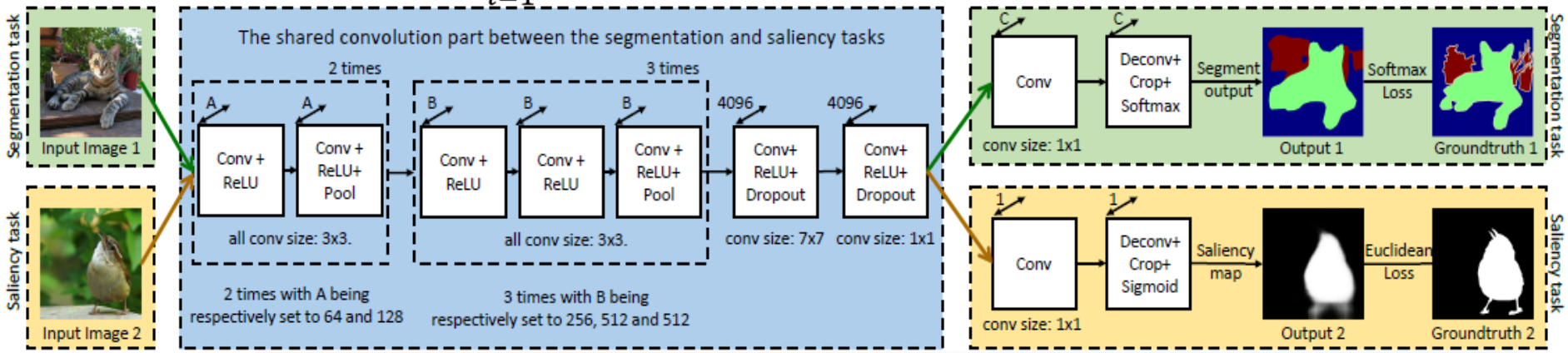
Multitask:
-Object segmentation task
Cost function:

$$J_1(X;\theta_s,\theta_h) = -\frac{1}{N_1}\sum_{i=1}^{N_1}\sum_{c=1}^{C}\sum_{j=1}^{P}\sum_{k=1}^{Q} 1\{Y_{ijk}=c\}\log(h_{cjk}(X_i;\theta_s,\theta_h))$$

-Saliency detection task
Cost function:

$$J_2(Z;\theta_s,\theta_f) = \frac{1}{N_2}\sum_{i=1}^{N_2} ||M_i - f(Z_i;\theta_s,\theta_f)||_F^2$$

Training strategy:
-Sharing parts of the parameters
-In an alternating manner



Saliency Object Detection Via Learning Method

1. Salient Object Detection via Bootstrap Learning,CVPR2015
2. Deep Neural Networks for Saliency Detection, CVPR2015

Salient Object Detection via Bootstrap Learning

Motivation

Difficulties

- Most previous methods only construct their models based on low-level cues via **contrast**-based algorithm.
- Previous methods use **single** feature or **same** features for all the images.
- Previous classifier-based methods need **sample images** with **ground truth** for learning.

Solution

- Our approach formulate a classifier-based (**SVM**-based) method to better exploit the low-level cues in **one image**.
- Our approach utilizes the optimal combination of **multiple features** via multiple SVM-kernels, different features for different images.
- Our approach restrains the learning progress within **one image**, **and no ground truth** is needed.

Na tong, Huchuan Lu. et al. Salient object detection via bootstrap learning, CVPR2015

Framework of the proposed approach

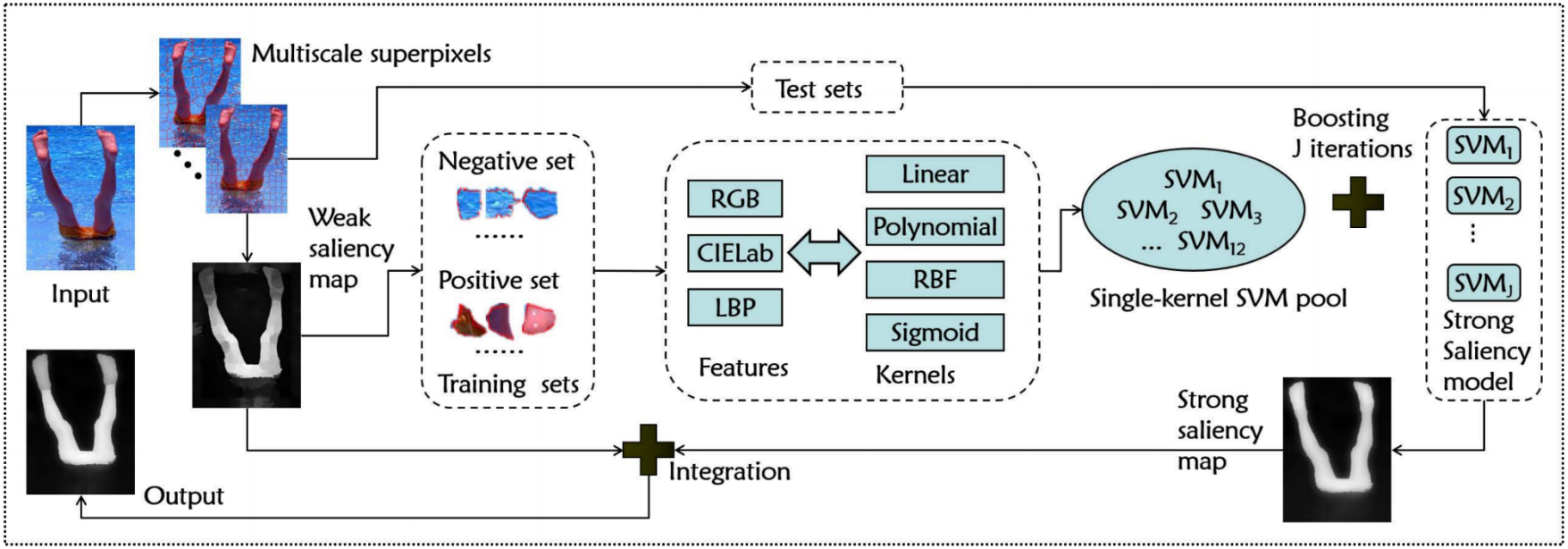
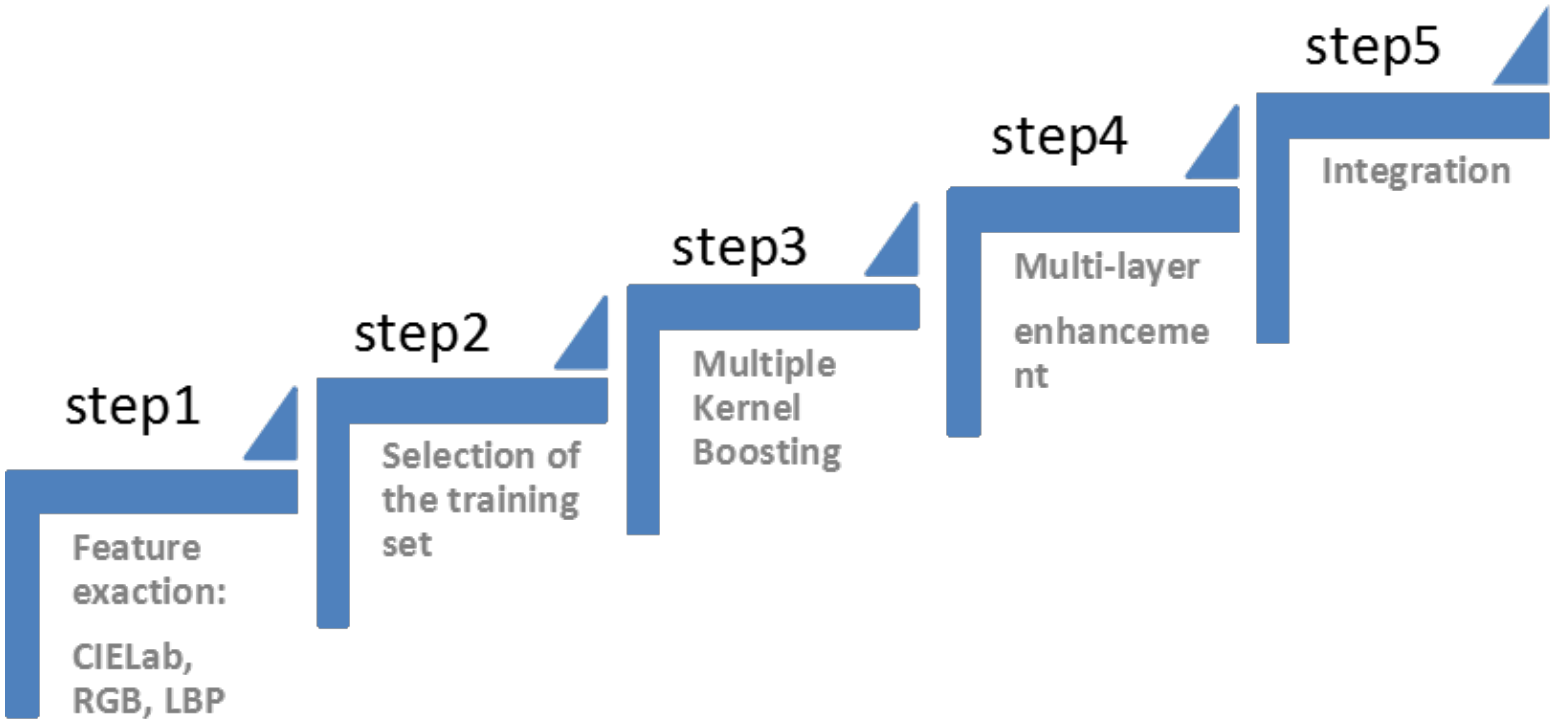


Figure 2. Bootstrap learning for salient object detection. A weak saliency map is constructed based on image priors to generate training samples for a strong model. A strong classifier based on multiple kernel boosting is learned to measure saliency where three feature descriptors are extracted and four kernels are used to exploit rich feature representations. Detection results at multiple scales are The weak and strong saliency maps are weighted combined to generate the final saliency map.

Main steps of the proposed approach



Step1. Feature extraction

- First, we construct superpixels with fixed number on the input image using the SLIC method [1].

- Then three descriptors are extracted based on **three feature spaces**:

➢ For the **RGB** and **CIELAB** color space, we formulate two feature space of 3 dimensions, which represent the average components of the superpixel in the three color channel.
➢ We employ the original **LBP** [2] to the whole image by considering an eight-neighborhood for each pixel. Then each pixel is assigned to a value between [0, 58] from the uniform pattern. We construct a LBP histogram for every superpixel according to the LBP values of the pixels within the region.

[1] R. Achanta, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk., "Slic superpixels." EPFL, Tech.Rep. 149300, Tech. Rep., 2010.
[2] M. Topi, O. Timo, P. Matti, and S. Maricor, "Robust texture classification by subsets of local binary patterns," in Pattern Recognition, 2000. Proceedings. 15th International Conference on, vol. 3. IEEE, 2000, pp. 935–938.

Step2. Selection of training set – Weak model

- First, we compute a rough saliency map based on contrast with the image border regions and two priors, center-prior and **dark-channel prior**:

$$f_0(c_i) = g(c_i) \times S_d(c_i) \times \sum_{\kappa \in \{F_1, F_2, F_3\}} \left(\frac{1}{N_B} \sum_{j=1}^{N_B} d_\kappa(c_i, n_j) \right)$$

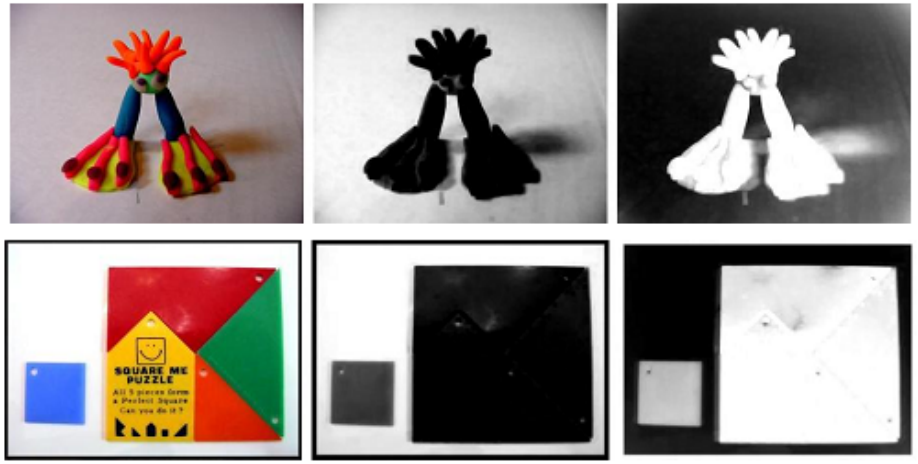


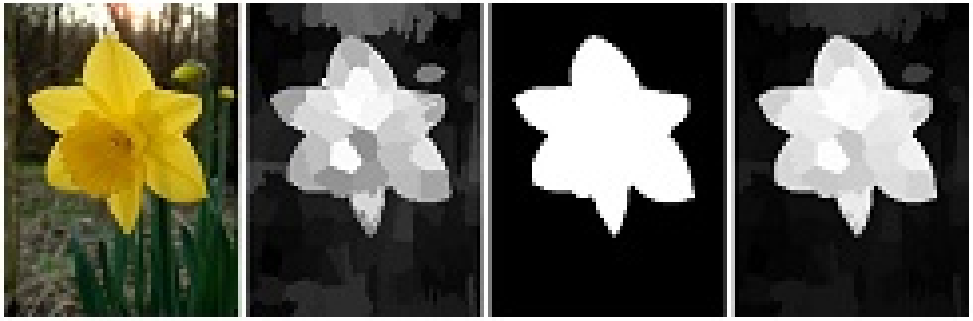
Figure 3. Examples of dark channel prior. Left to right: input, dark channel map and dark channel prior (the opposite of dark channel map and brighter pixels indicate higher saliency values).

Step2. Selection of training set – Weak model

- First, we compute a rough saliency map based on contrast with the image border regions and two priors, center-prior and **dark-channel prior**:

$$f_0(c_i) = g(c_i) \times S_d(c_i) \times \sum_{\kappa \in \{F_1, F_2, F_3\}} \left(\frac{1}{N_B} \sum_{j=1}^{N_B} d_\kappa(c_i, n_j) \right)$$

- Then we use the **Graph Cut** measure to fulfill the foreground and smooth the background in order to get a weak map:

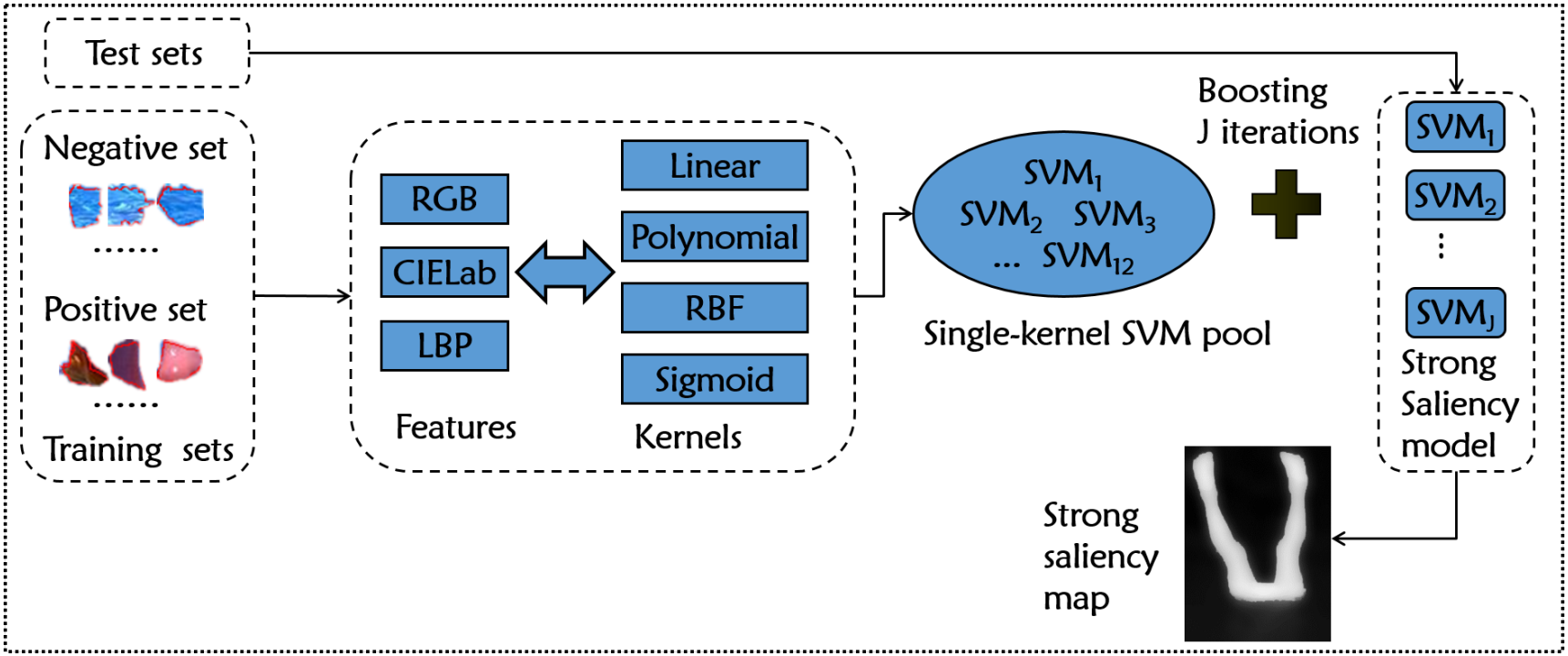


- Last, **construct the training set**.

we compute the average saliency value for each region in the map and set two thresholds for decision of the training set. Note that we set an adaptive value for the high threshold, **T** times of the average value over the whole image.

Step3. Multiple Kernel Boosting – Strong model

- The Multiple Kernel Boosting measure aims to **search for an optimal combination of multiple features and kernels**. We treat each single-SVM with different kernels and features as a weak classifier and then learn a strong classifier model using the boosting method:



Step3. Multiple Kernel Boosting – Strong model

- The Multiple Kernel Boosting measure aims to **search for an optimal combination of multiple features and kernels**. We treat each single-SVM with different kernels and features as a weak classifier and then learn a strong classifier model using the boosting method:

$$Y(r) = \sum_{j=1}^{J} \beta_j z_j(r).$$

where J denotes the iteration times of the Boosting process. We consider the single SVM as weak classifiers and the final strong classifier $Y(r)$ is the weighted vote of all the weak classifiers.

- Again, To improve the quality of the maps, we use the Graph cut get a optimized map.

- Furthermore, we formulate the strong map $\check{\mathcal{M}}_s$ by enhancing the saliency map with the **guided filter** [3], which performs well as an edge-preserving smoothing operator.

[3] K. He, J. Sun, and X. Tang, "Guided image filtering," in Proceedings of European Conference on Computer Vision, 2010.

Step4. Multi-layer enhancement

- The train set we selected according to the above steps are **few in number**, not enough for a robust classifier.

- We generate **four layers of superpixels** simply by setting different region numbers for the SLIC method.

- Then the positive set and negative set are selected according to Step2.

- Next, all the train sets from all the four layers are sent to the MKB model at the same time and all the test sets are tested by the learned classifier simultaneously.

- At last, we sum up the weak saliency maps from the four layers, defined as:

$$\mathcal{M}_s = \frac{1}{4}\sum_{i=1}^{4}\check{\mathcal{M}}_{s_i}$$

Step5. Integration

The weak map is likely to detect fine details and to capture local structural information due to the contrast-based measure. In contrast, the strong map works well by focusing on global shapes for most images except the case when there are large difference between the test samples and the train samples. Both of them have complementary strengths and flaws.

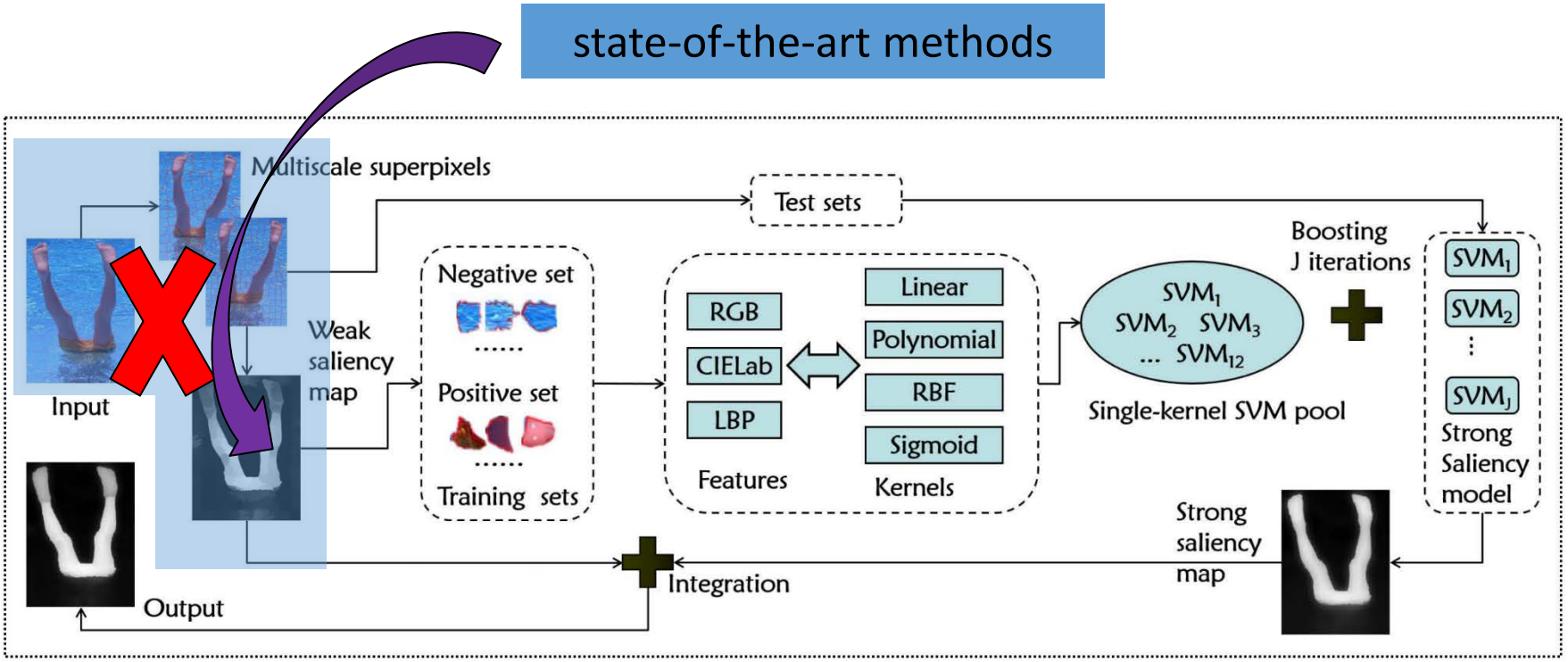
Therefore we integrate the two maps by weighted combination,

$$\mathcal{M} = \sigma \mathcal{M}_s + (1 - \sigma)\mathcal{M}_w,$$

where σ is the weight for the combination, σ = 0.7 in our method.

Application: bootstrapping other methods

Using other state-of-the-art methods instead of the proposed weak model.
The performances of other methods after bootstrapped are largely improved.



Application: bootstrapping other methods

Using other state-of-the-art methods instead of the proposed weak model.
The performances of other methods after bootstrapped are largely improved.

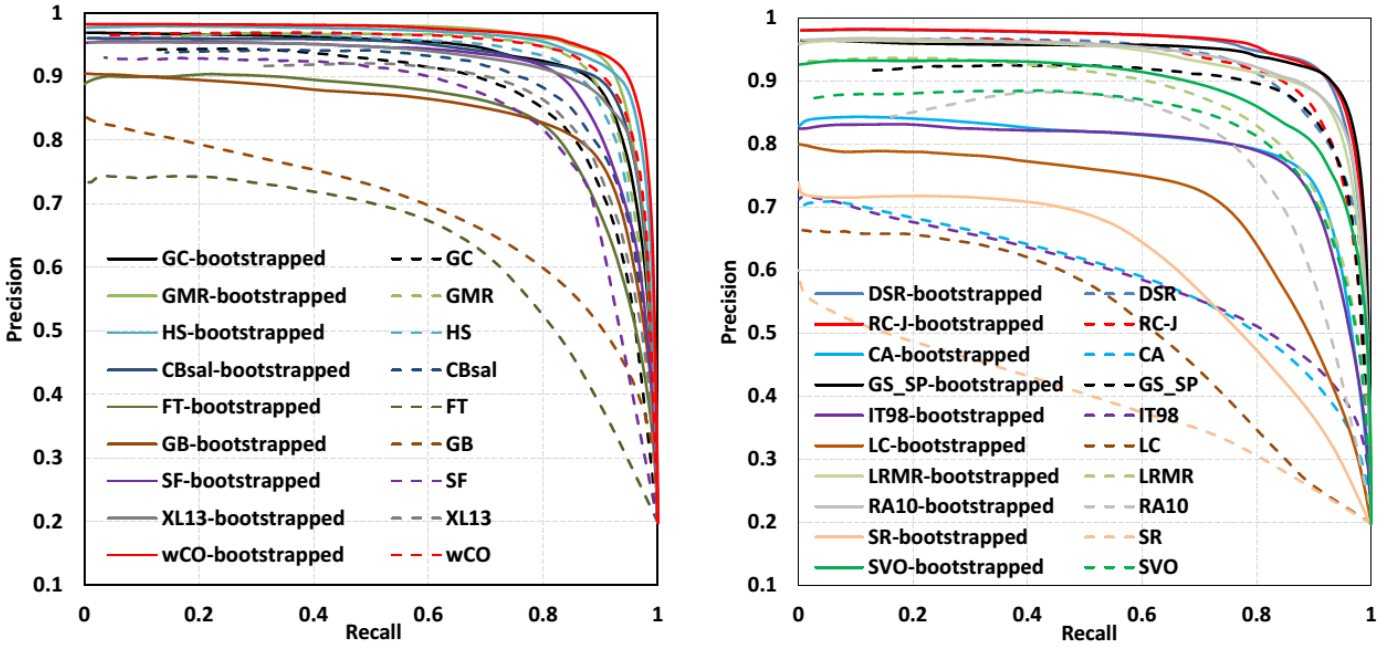


Figure 9. P-R curve results show improvement of state-of-the-art methods by the bootstrap learning approach on the ASD dataset.

Experimental results

- Experimental setup

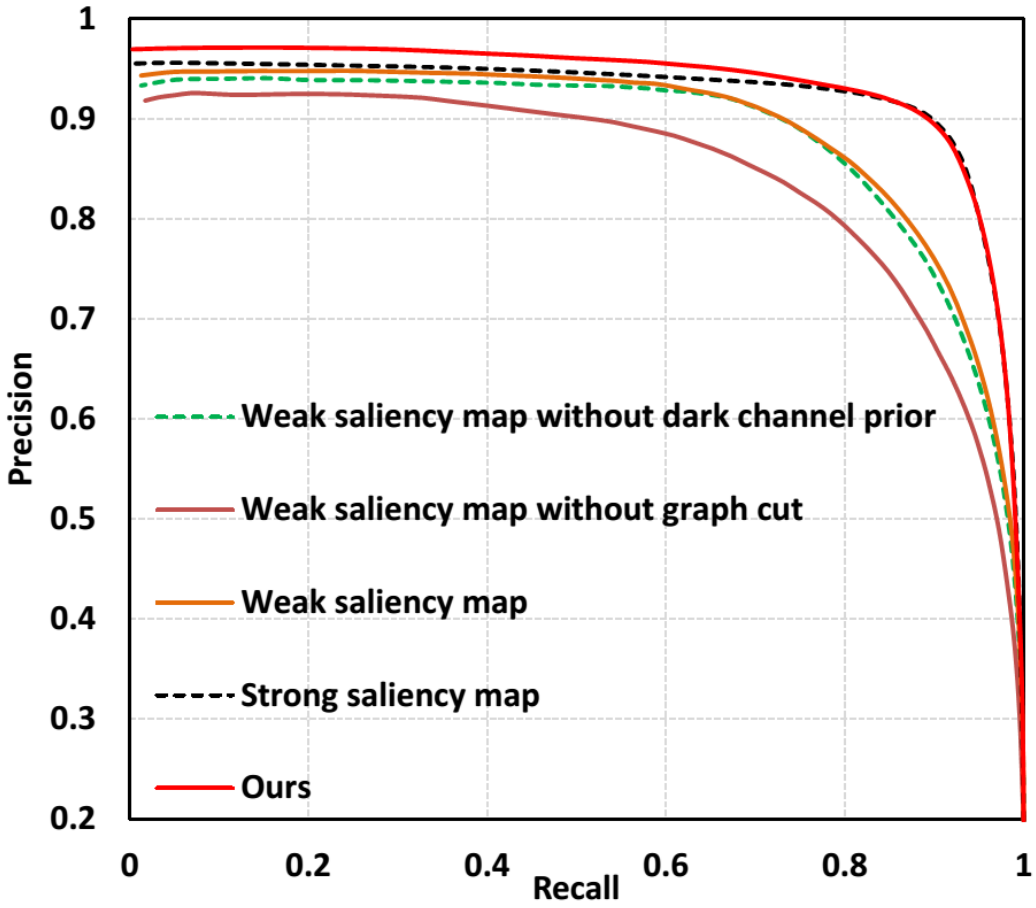
  - Component Analysis of the proposed method
  - Comparative results with **19** saliency detection methods on **six** databases (i.e., **ASD, Pascal-S, SOD, SED2, THUS, DUT-OMRON**). Both qualitative and quantitative results are presented.

- Evaluation protocol

  - Precision and Recall (P-R) curve
  - AUC value
  - F-measure value.

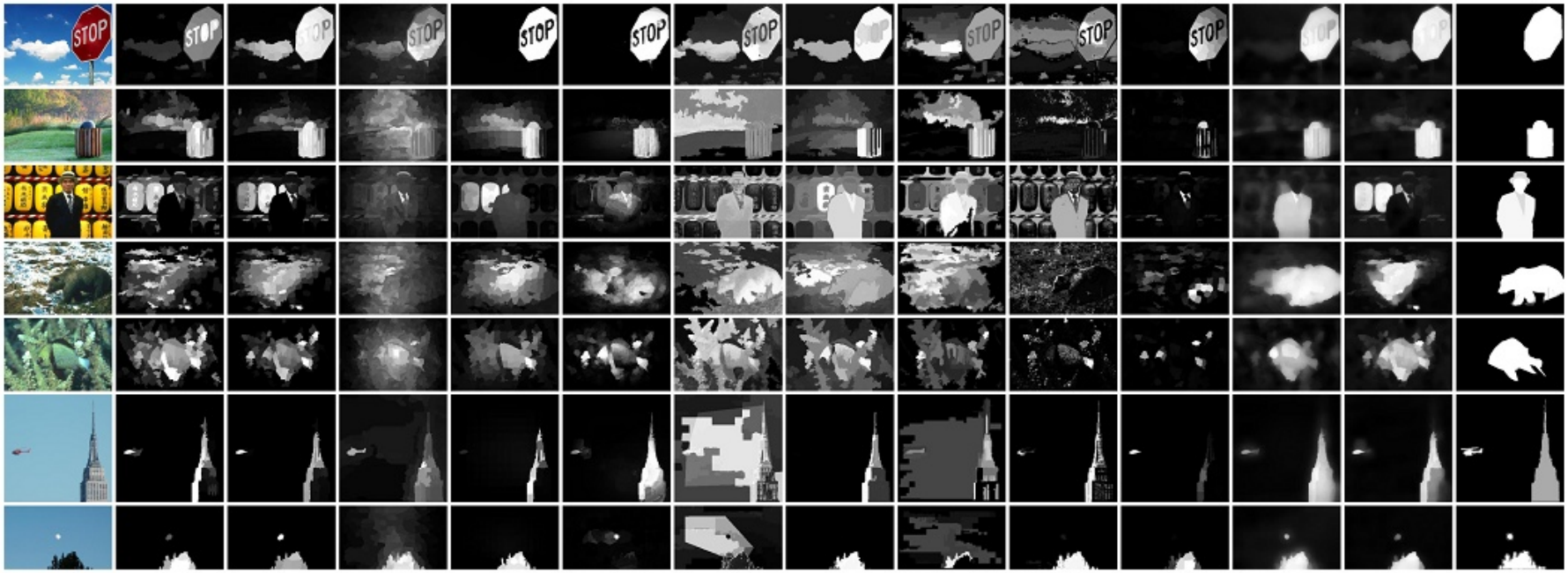
Experimental results – Component Analysis

Every component in the proposed algorithm contributes to the final saliency map.



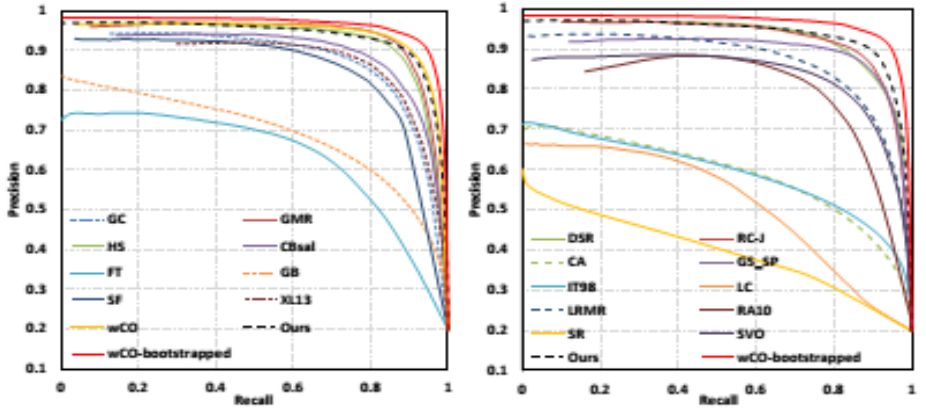
Experimental results

- Qualitative comparison: the following figure shows that the proposed algorithms highlight the salient objects well with fewer noisy results.
- Quantitative comparison: we further compare the proposed method with 19 saliency detection methods on six databases quantitatively.

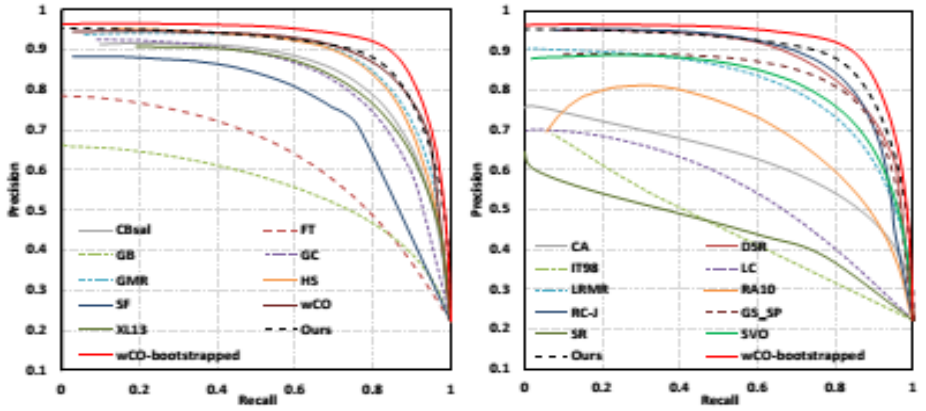


(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l) (m) (n)

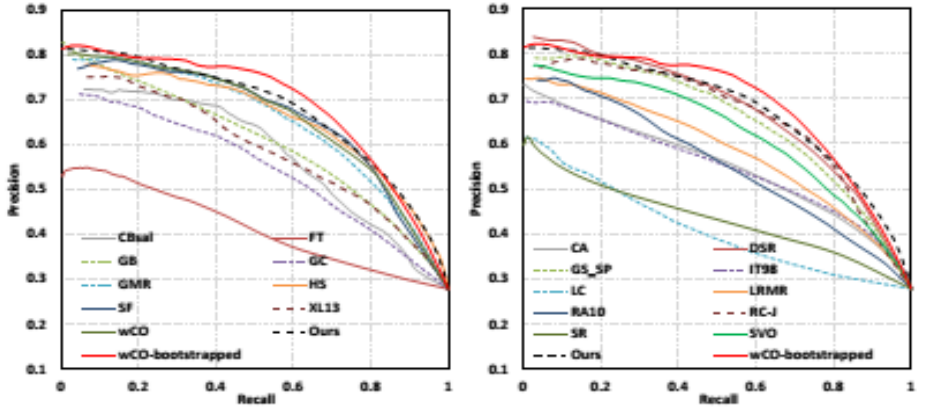
Figure 5. Comparison of our saliency maps with ten state-of-the-art methods. Left to right: (a) input (b) GS_SP [34] (c) wCO [43] (d) LRMR [30] (e) GMR [38] (f) DSR [23] (g) XL13 [36] (h) HS [37] (i) RC-J [9] (j) GC [10] (k) SF [28] (l) Ours (m) wCO-bootstrapped (n) ground truth. Our model is able to detect both the foreground and background uniformly.



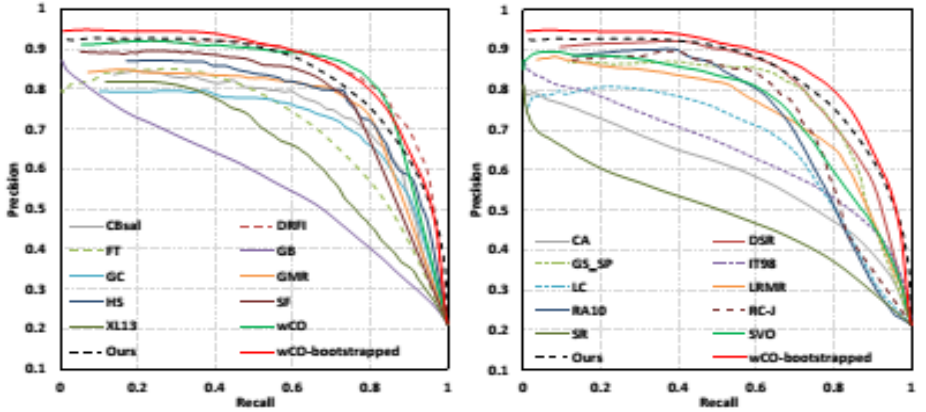
(a) ASD dataset



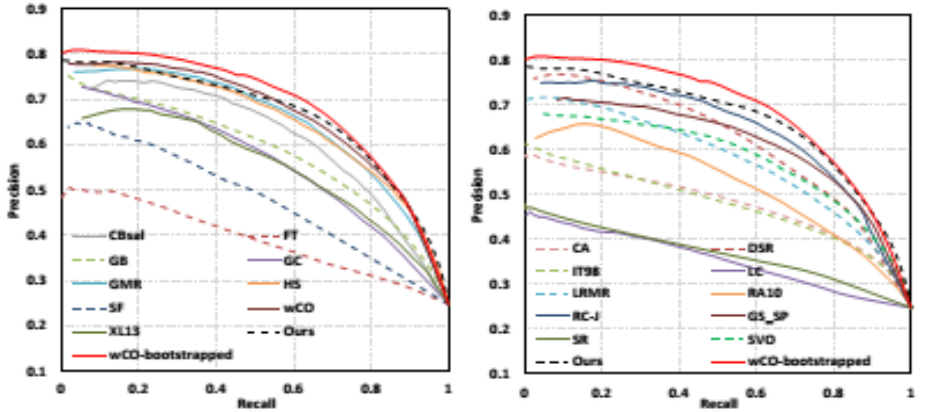
(b) THUS dataset



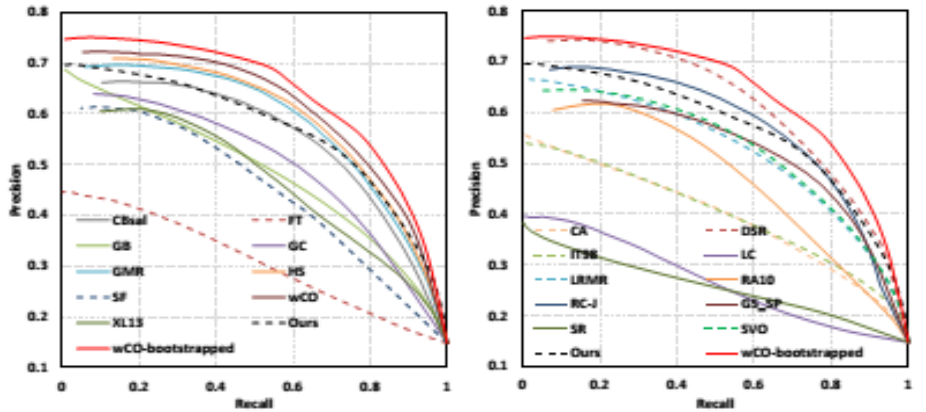
(c) SOD dataset



(d) SED2 dataset



(e) Pascal-S dataset



(f) DUT-OMRON dataset

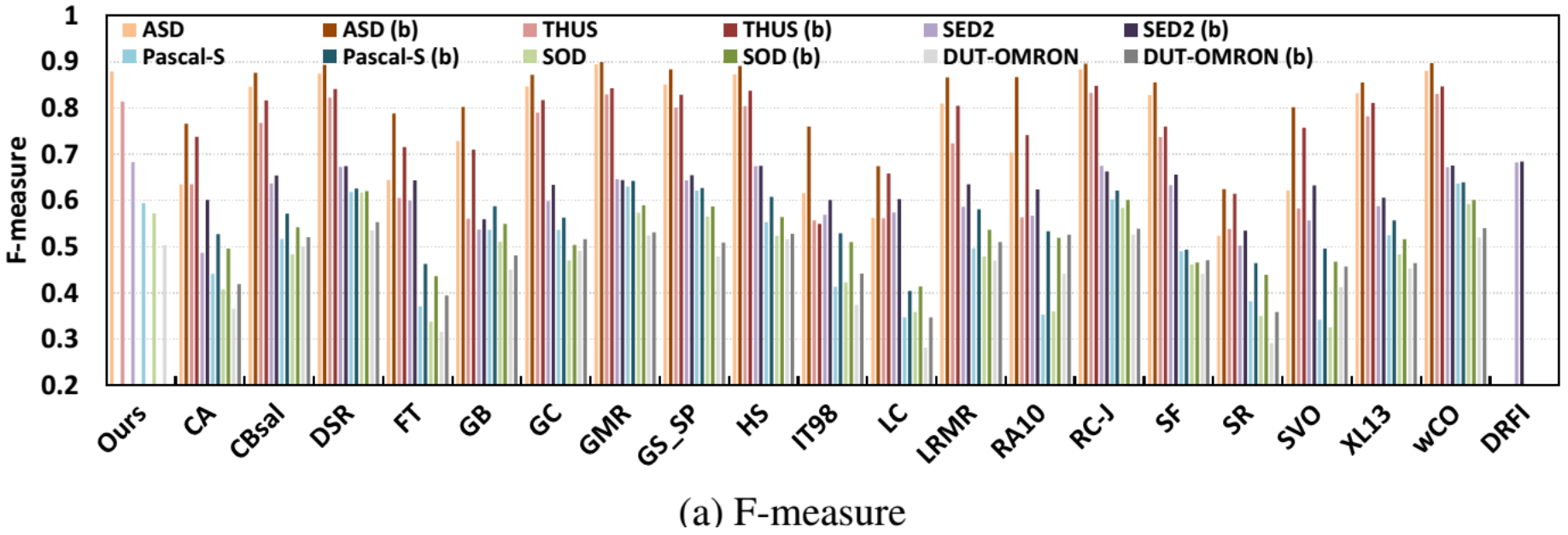
Figure 6. P-R curve results on six datasets.

Experimental results

Table 1. AUC (Area Under ROC Curve) on the ASD, SED2, SOD, THUS, Pascal-S and DUT-OMRON datasets. The best two results are shown in **red** and **blue** fonts respectively. The colomn named "wCO-b" denotes the wCO model after bootstrapping using the proposed approach. The proposed methods rank first and second on the six datasets. The two rows named "*ASD (b)*" show the AUC of the saliency results by taking other state-of-the-art saliency maps as the weak saliency maps in the proposed approach on the ASD dataset. All the evaluation results of the state-of-the-art methods are largely improved over the original results as shown in the two rows named "*ASD*".

	wCO-b	Ours	HS	RC-J	GC	DSR	GS_SP	GMR	SF	XL13	CBsal
ASD	**.9904**	**.9828**	.9683	.9735	.9456	.9774	.9754	.9700	.9233	.9609	.9628
SED2	**.9399**	**.9363**	.8387	.8606	.8618	.9136	.8999	.8620	.8501	.8470	.8728
SOD	**.8525**	**.8477**	.8169	.8238	.7181	.8380	.7982	.7982	.8238	.7868	.7409
THUS	**.9723**	**.9635**	.9322	.9364	.9032	.9504	.9462	.9390	.8510	.9353	.9270
Pascal-S	**.8774**	**.8682**	.8368	.8379	.7479	.8299	.8553	.8315	.6830	.7983	.8087
DUT-OMRON	**.9063**	.8794	.8604	.8592	.7931	.8922	.8786	.8500	.7628	.8160	.8419
ASD (b)	-	-	.9876	.9869	.9773	.9872	**.9888**	.9844	.9723	.9791	.9811
	DRFI	wCO	LRMR	RA10	SVO	GB	FT	CA	SR	LC	IT98
ASD	-	.9805	.9593	.9326	.9530	.9146	.8375	.8736	.6973	.7772	.8738
SED2	.9349	.9062	.8886	.8500	.8773	.8448	.8185	.8585	.7593	.8366	.8904
SOD	-	.8217	.7810	.7710	.8043	.8191	.6006	.7868	.6695	.6168	.7862
THUS	-	.9525	.9199	.8810	.9280	.8132	.7890	.8712	.7149	.7673	.6282
Pascal-S	-	.8597	.8121	.7836	.8226	.8380	.6220	.7829	.6585	.6191	.7797
DUT-OMRON	-	**.8927**	.8566	.8264	.8662	.8565	.6758	.8137	.6799	.6549	.8218
ASD (b)	-	**.9904**	.9825	.9817	.9722	.9619	.9506	.9531	.8530	.8988	.9479

Experimental results



(a) F-measure

Figure 7. (a) is the F-measure values of 21 methods on six datasets. Note that " * (b)" shows improvement of state-of-the-art methods by the bootstrap learning approach on the corresponding dataset. All the methods are largely improved.

Conclusion

- We propose a bootstrap learning model for salient object detection in which both weak and strong models are exploited.

- We utilize the Multiple Kernel Boosting method (a boosted version of the MKL method).

- Our learning process is restrained within one image and unsupervised since we select our train sets automatically by constructing a weak saliency map, where center-surround contrast, center-prior and dark channel prior are considered.

- We evaluate the proposed algorithm on six publicly available databases compared with 19 state-of-the-art methods. The experimental results demonstrate our approach outperforms other methods especially on P-R curve and AUC values and our model is able to detect both the foreground and background uniformly.

Deep Neural Networks for Saliency Detection

1. Motivation

Existing Issues

- Previous methods mainly rely on hand-crafted features which fail to describe complex image scenarios.
- The adopted saliency priors are combined based on heuristics and it is not clear how these features can be better integrated.
- DNNs fail to capture the global relationship of image regions and maintain local label consistency.

Our Approach

- We propose an approach to apply DNNs to saliency detection from both local and global views.
- We use a DNN to learn local patch features to estimate the saliency value of each pixel from a local view.
- We train another DNN to predict region saliency by investigating the complex relationships among saliency cues through a supervised learning scheme.

2. Pipeline of the proposed approach

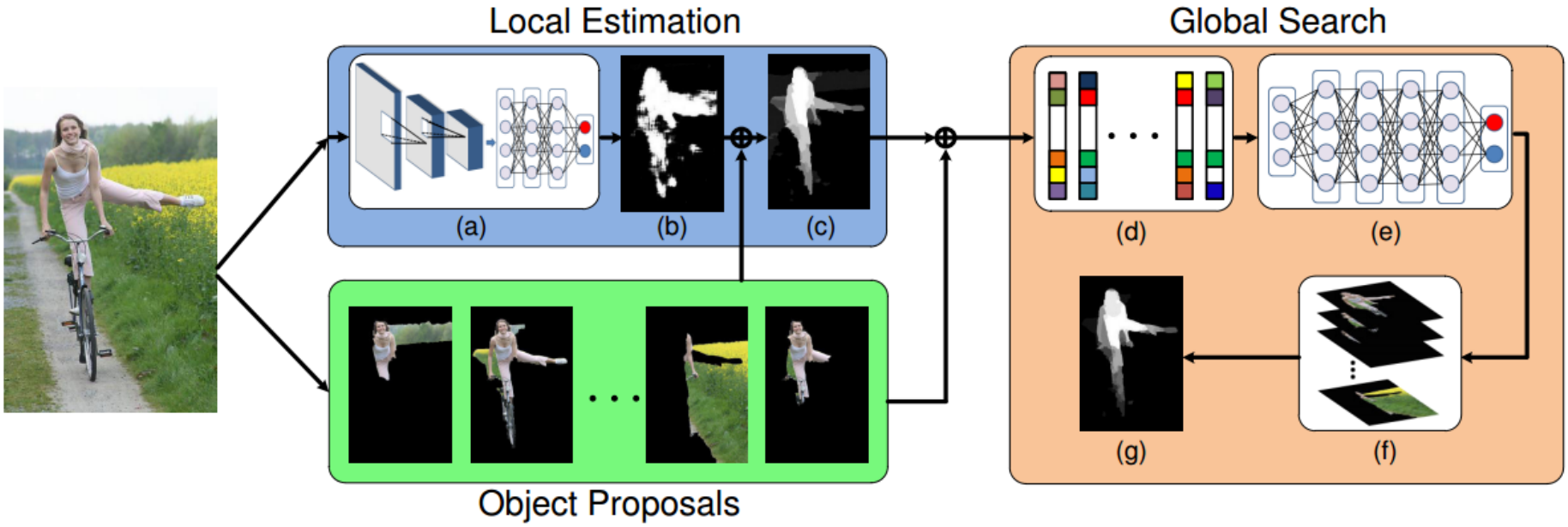


Figure 1. Pipeline of our algorithm. (a) Proposed deep network **DNN-L**. (b) Local saliency map. (c) Local saliency map after refinement. (d) Feature extraction. (e) Proposed deep network **DNN-G**. (f) Sorted object candidate regions. (g) Final saliency map.

3. Related Work

- **Saliency detection methods** , which conduct saliency detection from either local or global views.
- **Generic object detection** (also known as object proposal) , which aim at generating the locations of all category independent objects in an image.
- **Deep Neural Networks** used for scene labeling

4. Local Estimation

- We formulate a binary classification problem to determine whether each pixel is salient (1) or non-salient (0) based on its surrounding.
- We use a deep network, namely **DNN-L**, to conduct classification since DNNs do not rely on hand-crafted features.
- By incorporating object level concepts into local estimation, we present a refinement method to enhance the spatial consistency of local saliency maps.

4. Local Estimation

Architecture Details of DNN-L

Table 1. The architecture of the proposed **DNN-L**. C: convolutional layer; F: fully connected layer; R: ReLUs; L: local response normalization; D: dropout; S: softmax layer; Channels: the number of output feature maps; Input size: the spatial size of input feature maps

Layer	1	2	3	4	5	6 (Output)
Type	C+R+L	C+R	C+R	F+R+D	F+R+D	F+S
Channels	96	256	384	2048	2048	2
Filter size	11x11	5x5	3x3	–	–	–
Pooling size	3x3	2x2	3x3	–	–	–
Polling stride	2x2	2x2	3x3	–	–	–
Input size	51x51	20x20	8x8	2x2	1x1	1x1

4. Local Estimation

Training Data of DNN-L

- For each image in the training set, we collect samples by cropping 51×51 RGB image patches in a sliding window fashion with a stride of 10 pixels.
- The patch $\mathbf{B}$ is labeled as a positive training example if i). the central pixel is salient, and ii). it sufficiently overlaps with the ground truth salient region $\mathbf{G}$: $|\mathbf{B} \cap \mathbf{G}| \geq 0.7 \times \min(|\mathbf{B}|, |\mathbf{G}|)$
- The patch $\mathbf{B}$ is labeled as a negative training example if i). the central pixel is located within the background, and ii). its overlap with the ground truth salient region is less than a predefined threshold:
  $|\mathbf{B} \cap \mathbf{G}| < 0.3 \times \min(|\mathbf{B}|, |\mathbf{G}|)$
- We do not pre-process the training samples, except for subtracting the mean values over the training set from each pixel.

4. Local Estimation

Training DNN-L

Given the training patch set $\{\mathbf{B}_i\}_{N^L}$ and the corresponding label set $\{l_i\}_{N^L}$, we use the softmax loss with weight decay as the cost function:

$$L(\boldsymbol{\theta}\hat{\boldsymbol{\theta}}) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=0}^{1} 1\{l_i = j\}\log P\left(l_i = j \mid {}^{L}\right) + \lambda\sum_{k=1}^{6}\left\|\mathrm{W}_k^L\right\|_2^2,$$

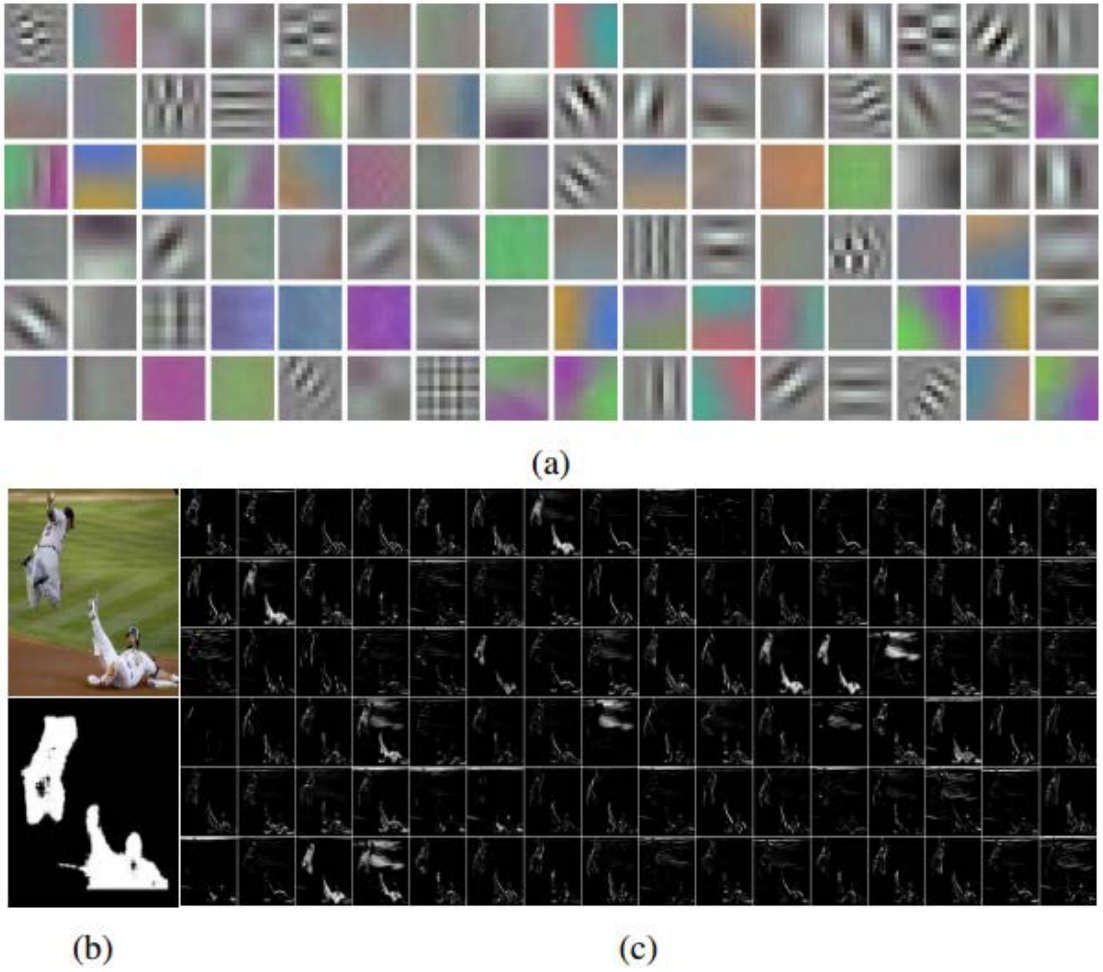
where $\boldsymbol{\theta}^L$ is the learnable parameter set of DNN-L including the weights and bias of all layers; $1\{\square\}$ is the indicator function; $P\left(l_i = j \mid \boldsymbol{\theta}^L\right)$ is the label probability of the i-th training samples predicted by **DNN-L**; W_k^L is the weight of the k-th layer.

DNN-L is trained using stochastic gradient descent with a batch size of m = 256, momentum of 0.9, and weight decay of 0.0005.

4. Local Estimation

Visualization of DNN-L

Figure 2. (a) 96 convolutional filters in the first layer. (b) Input image (top) and the local saliency map (bottom) generated by **DNN-L**. (c) Output feature maps of the first layer by applying **DNN-L** to the input image in a sliding window manner.



4. Local Estimation

Refinement

- Given an input image, we first generate a set of object candidate masks $\{O_i\}_{N_O}$ using the GOP method [19] and a local saliency map S^L using our local estimation method.
- We compute the accuracy score A and the coverage score C of each object candidate as follows:

$$A_i = \frac{\sum_{x,y} O_i(x,y) \times S^L(x,y)}{\sum_{x,y} O_i(x,y)}$$

$$C_i = \frac{\sum_{x,y} O_i(x,y) \times S^L(x,y)}{\sum_{x,y} S^L(x,y)}$$

Figure 3. Different object candidate regions with their corresponding accuracy scores A and coverage scores C.

4. Local Estimation

Refinement

We define the confidence for the i-th candidate by considering both the accuracy score and the coverage score as

$$conf_i^L = \frac{(1+\beta) \times A_i \times C_i}{\beta A_i + C_i}$$

To find a subset of optimal object candidates, we sort all the candidates by their confidences in a descending order. The refined local saliency map is generated by averaging the top K candidate regions (K is set to 20 in all the experiments).

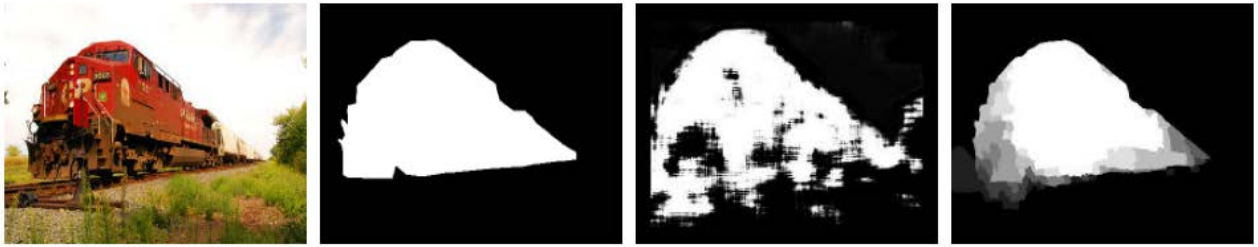


Figure 4. From left to right: source image, ground truth, local saliency map output by DNN-L, local saliency map after refinement.

Pipeline of the proposed approach

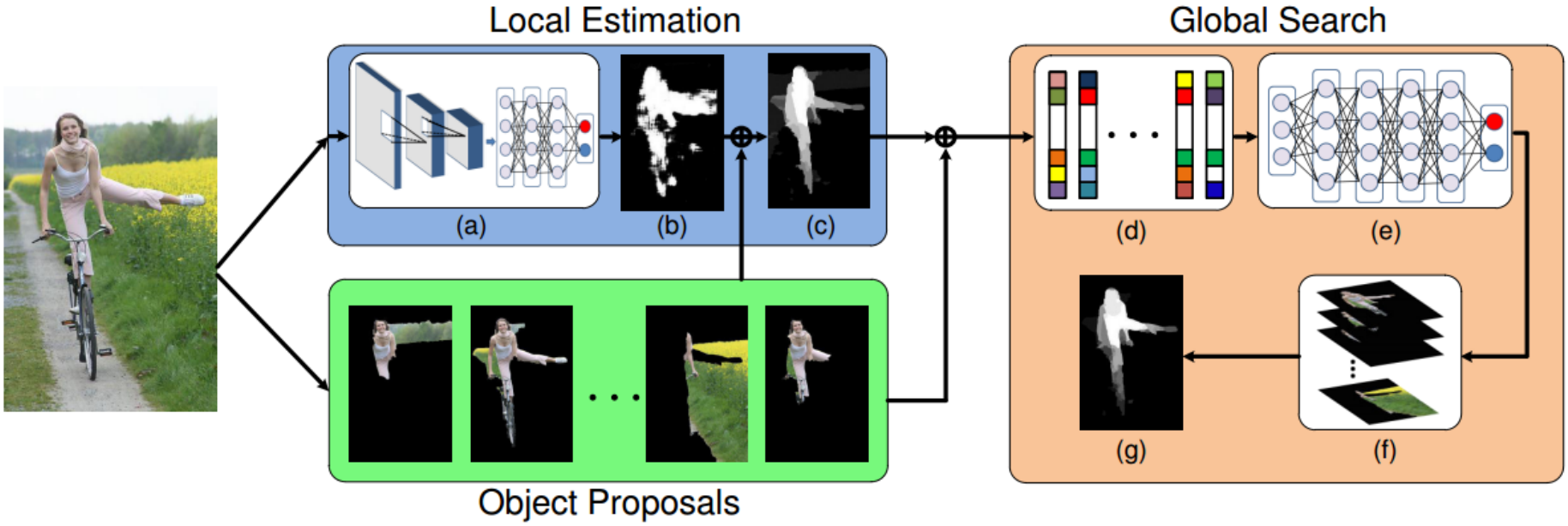


Figure 1. Pipeline of our algorithm. (a) Proposed deep network **DNN-L**. (b) Local saliency map. (c) Local saliency map after refinement. (d) Feature extraction. (e) Proposed deep network **DNN-G**. (f) Sorted object candidate regions. (g) Final saliency map.

4. Global Search

- We formulate a DNN-based regression method for saliency detection, where various saliency cues are considered simultaneously and their complex dependencies are learned automatically through a supervised learning scheme.

- For each input image, we first detect local saliency using the proposed local estimation method. A 72-dimensional feature vector is extracted to describe each object candidate generated by the GOP method from a global view.

- The proposed deep network **DNN-G** takes the extracted features as inputs and predicts the saliency values of the candidate regions through regression.

4. Global Search

Global Features

The proposed 72-dimensional feature vector covers global contrast features, geometric information, and local saliency measurements of object candidate regions.

Table 2. Global contrast features of object candidate regions.

Feature	Definition	Feature	Definition
$c_1 - c_4$	$\chi^2(h_{\mathbf{O}}^{RGB}, h_{\mathbf{B}}^{RGB})$	c_{49}	$\chi^2(h_{\mathbf{O}}^{RGB}, h_{\mathbf{I}}^{RGB})$
$c_5 - c_8$	$\chi^2(h_{\mathbf{O}}^{Lab}, h_{\mathbf{B}}^{Lab})$	c_{50}	$\chi^2(h_{\mathbf{O}}^{Lab}, h_{\mathbf{I}}^{Lab})$
$c_9 - c_{12}$	$\chi^2(h_{\mathbf{O}}^{HSV}, h_{\mathbf{B}}^{HSV})$	c_{51}	$\chi^2(h_{\mathbf{O}}^{HSV}, h_{\mathbf{I}}^{HSV})$
$c_{13} - c_{24}$	$d(m_{\mathbf{O}}^{RGB}, m_{\mathbf{B}}^{RGB})$	$c_{52} - c_{54}$	$var_{\mathbf{O}}^{RGB}$
$c_{25} - c_{36}$	$d(m_{\mathbf{O}}^{HSV}, m_{\mathbf{B}}^{HSV})$	$c_{55} - c_{57}$	$var_{\mathbf{O}}^{Lab}$
$c_{37} - c_{48}$	$d(m_{\mathbf{O}}^{Lab}, m_{\mathbf{B}}^{Lab})$	$c_{58} - c_{60}$	$var_{\mathbf{O}}^{HSV}$

4. Global Search

Global Features

The proposed 72-dimensional feature vector covers global contrast features, geometric information, and local saliency measurements of object candidate regions.

Table 3. Geometric information and local saliency measurements of object regions.

Geometric Information				Local Saliency Measurement	
Feature	Definition	Feature	Definition	Feature	Definition
g_1	Bounding box aspect ratio	g_6	Major axis length	s_1	Accuracy score A
g_2	Bounding box height	g_7	Minor axis length	s_2	Coverage score C
g_3	Bounding box width	g_8	Euler number	s_3	$A \times C$
$g_4 - g_5$	Centroid coordinates			s_4	Overlap rate

4. Global Search

Architecture of DNN-G

Table 4. The architecture of the proposed **DNN-G**. F: fully connected layer; R: ReLUs; D: dropout; Channels: the number of output feature maps; Input size: the spatial size of input feature maps

Layer	1	2	3	4	5	6 (Output)
Type	F+R+D	F+R+D	F+R+D	F+R+D	F+R+D	F
Channels	1024	2048	2048	1024	1024	2
Filter size	–	–	–	–	–	–
Pooling size	–	–	–	–	–	–
Polling stride	–	–	–	–	–	–
Input size	1x1	1x1	1x1	1x1	1x1	1x1

4. Global Search

Training of DNN-G

- For each image in the training data set, around 1200 object regions are generated as training samples using the GOP method.
- The proposed 72-dimensional global feature vector **v** is extracted from each candidate region and then preprocessed by subtracting the mean and dividing the standard deviation of the elements.
- A label vector of precision p_i and overlap rate o_i, $\mathbf{y}_i = [\mathrm{p}_i, o_i]$ is assigned to each object region $\mathbf{O}_i$.

4. Global Search

Training of DNN-G

The network parameters $\boldsymbol{\theta}^G$ of DNN-G are learned by solving the following optimization problem:

$$\arg\min_{\boldsymbol{\theta}^G} \frac{1}{m}\sum_{i=1}^{m}\left\|\mathbf{y}_i - \boldsymbol{\theta}\mathbf{W}\left(\mathbf{v}_i \mid {}^{G}\right)\right\|_2^2 + \eta\sum_{k=1}^{6}\left\| {}^{G}_{k}\right\|_2^2$$

Where $\phi\left(\mathbf{v}_i\boldsymbol{\theta}\right| {}^{G}\right) = [\phi_i^1, \phi_i^2]$ is the output of **DNN-G** for the i-th training sample; $\mathbf{W}_k^G$ is the weight of the k-th layers. The optimization is conducted by stochastic gradient descent with a batch size m of 1000 and momentum of 0.9.

4. Global Search

Saliency Prediction via DNN-G Regression

- The global confidence score of the i-th candidate region is defined by

$$conf_i^G = \phi_i^1 \times \phi_i^2$$

- Denote $\left\{\hat{\hat{\mathbf{O}}}_1, ..., \mathbf{O}_N\right\}$ as the mask set of all the candidate regions in the input image sorted by the global confidence scores in a descending order. The corresponding global confidence scores are represented by $\left\{conf_1^G, ..., conf_N^G\right\}$. The final saliency map is computed by a weighted sum of the top K candidate masks,

$$\mathbf{S}^G = \frac{\sum_{k=1}^{K} conf_k^G \times \hat{\mathbf{O}}_k}{\sum_{k=1}^{K} conf_k^G}$$

5. Experimental Results

Setup

- We evaluate the proposed algorithm (LEGS) on four benchmark data sets: MSRA-5000 [22], SOD [25], ECCSD [32] and PASCAL-S [21], against ten state-of-the-art methods: SVO [4], PCA[24], DRFI [15], GC [6], HS [32], MR [33], UFO [16], wCtr [34], CPMCGBVS [21] and HDCT [17].

- We randomly sample 3000 images from the MSRA-5000 data set and 340 images from the PASCAL-S data set to train the proposed two networks. The remaining images are used for tests. Both horizontal refection and rescaling ($\pm$5%) are applied to all the training images to augment the training data set.

5. Experimental Results

Qualitative Evaluation

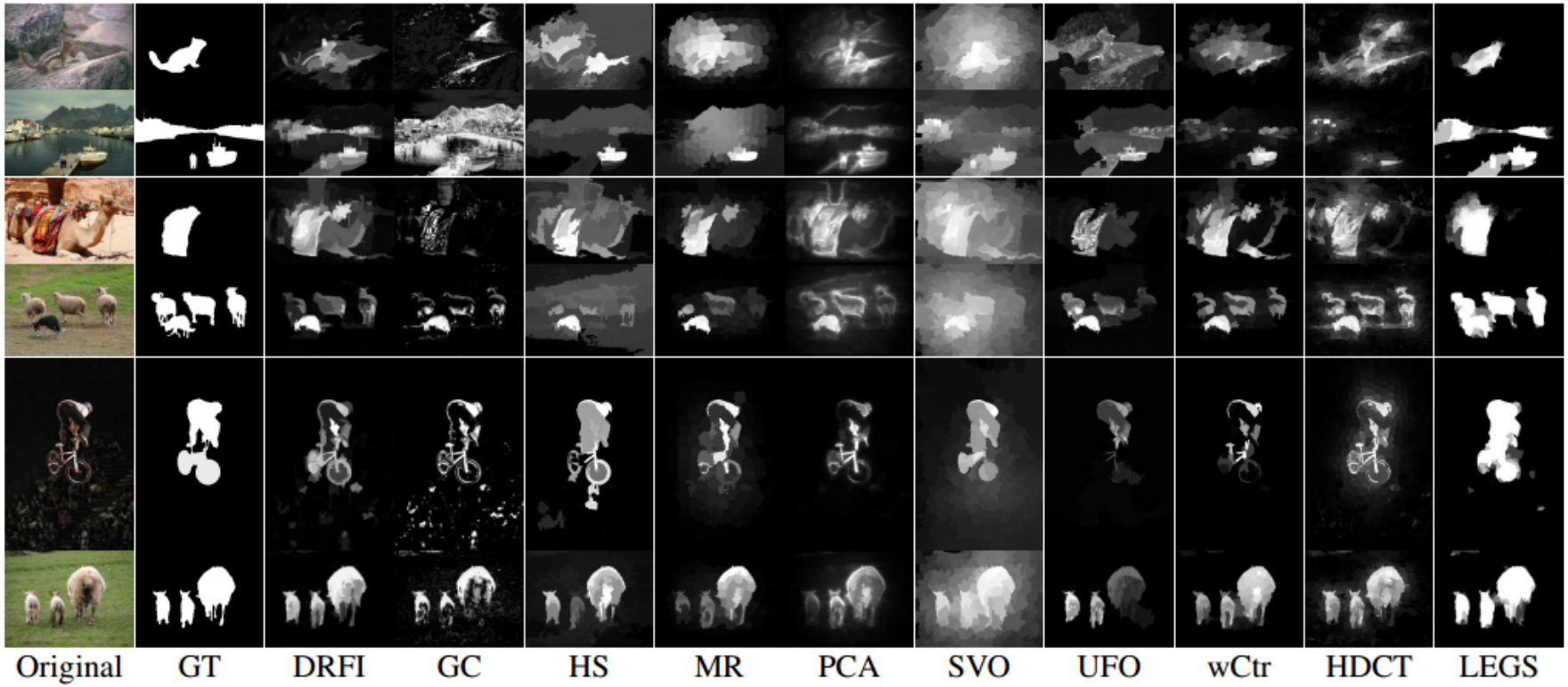


Figure 5. Saliency maps. Top, middle and bottom two rows are images from the SOD, ECCSD and PASCAL-S data sets. GT: ground truth. LEGS: the proposed method.

5. Experimental Results

Quantitative Evaluation

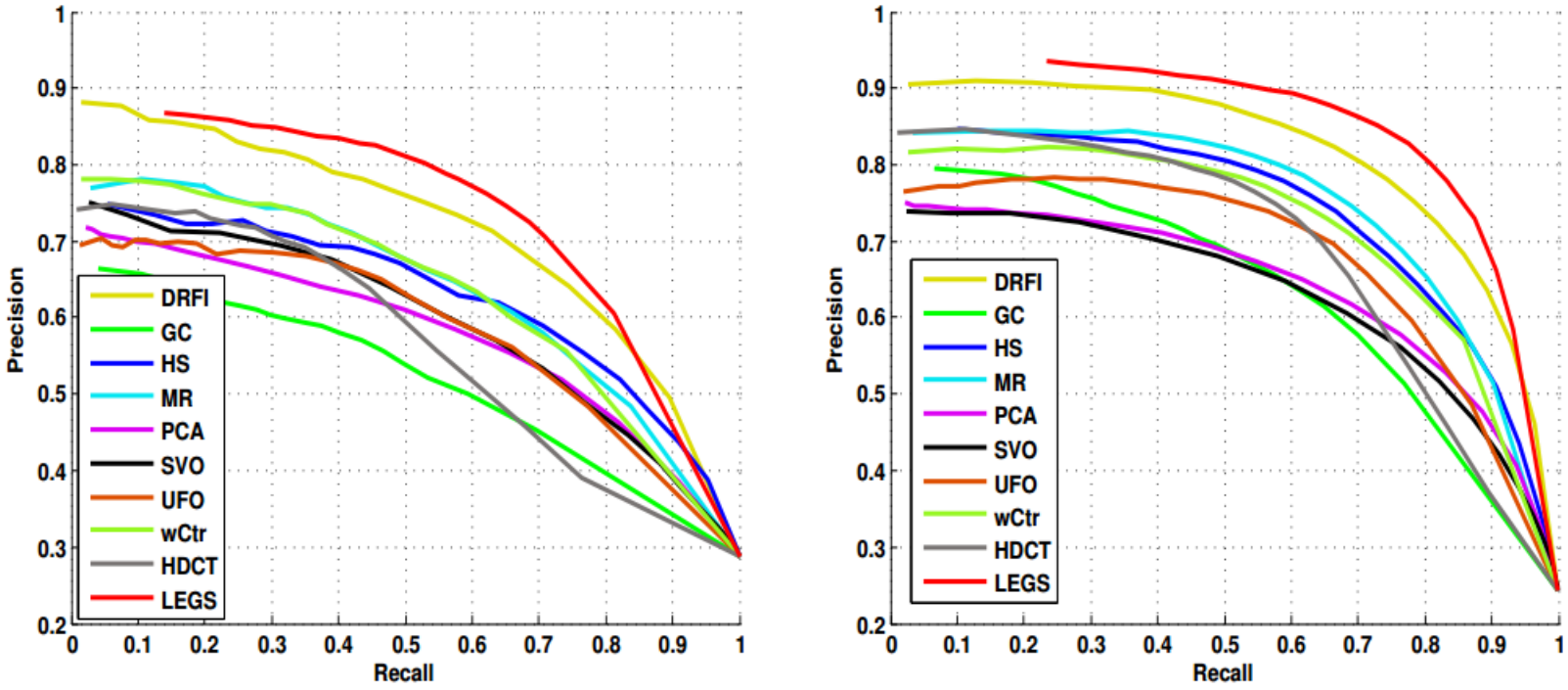


Figure 6. PR curves of saliency detection methods on SOD and ECCSD data set, respectively.

5. Experimental Results

Quantitative Evaluation

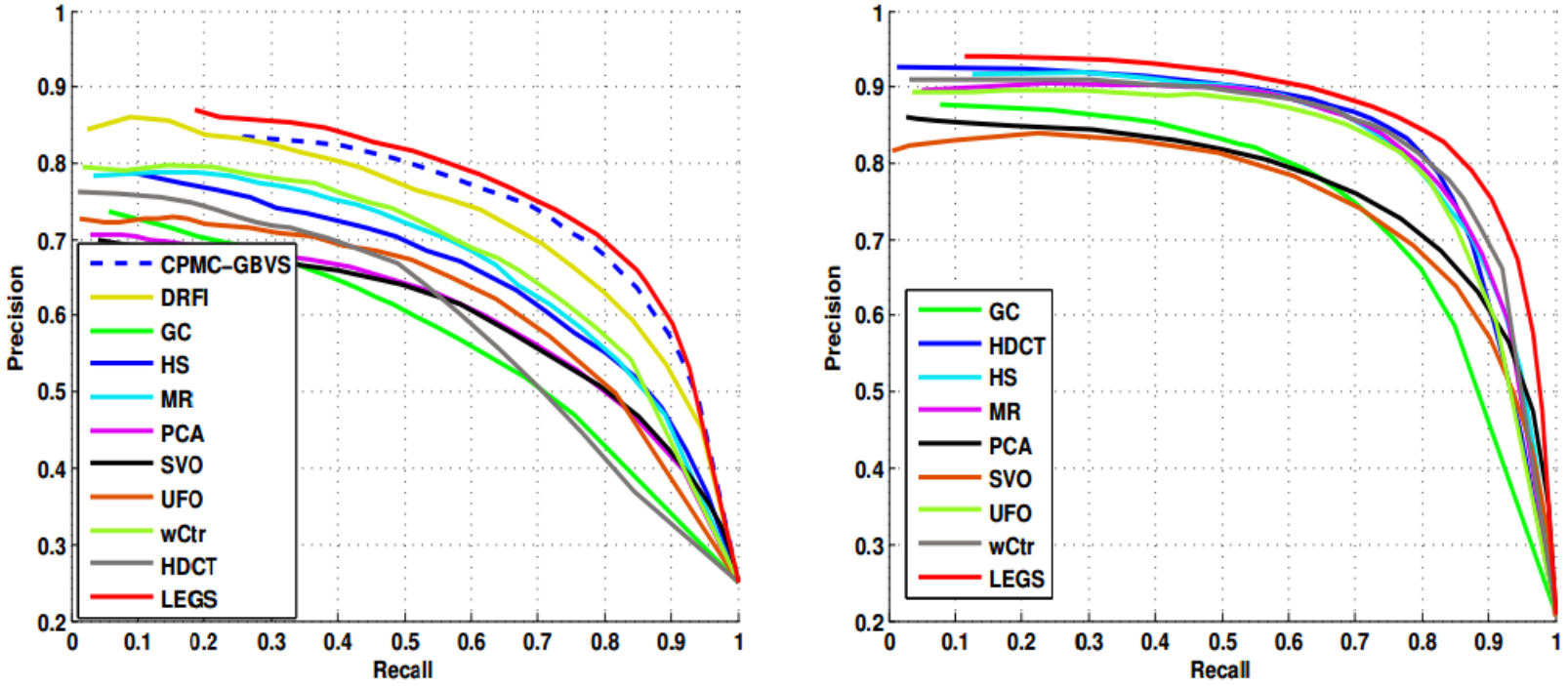


Figure 7. PR curves of saliency detection methods on PASCAL-S and MSRA-5000 data set, respectively.

5. Experimental Results

Quantitative Evaluation

Table 5. Quantitative results using F-measure and MAE. The best and second best results are shown in **red** color and **blue** color.

Data Set	Metric	DRFI	GC	HS	MR	PCA	SVO	UFO	wCtr	CPMC-GBVS	HDCT	LEGS
SOD	F-Measure	**0.617**	0.433	0.480	0.542	0.498	0.217	0.521	0.567	–	0.511	**0.630**
	MAE	**0.230**	0.288	0.301	0.274	0.290	0.414	0.272	0.245	–	0.260	**0.205**
ECCSD	F-Measure	**0.726**	0.568	0.631	0.689	0.575	0.237	0.638	0.672	–	0.641	**0.775**
	MAE	**0.172**	0.218	0.232	0.192	0.252	0.406	0.210	0.178	–	0.204	**0.137**
PASCAL-S	F-Measure	0.619	0.496	0.536	0.600	0.531	0.266	0.552	0.611	**0.654**	0.536	**0.669**
	MAE	0.195	0.245	0.249	0.219	0.239	0.373	0.227	0.193	**0.178**	0.226	**0.170**
MSRA-5000	F-Measure	–	0.704	0.765	**0.789**	0.707	0.302	0.774	0.788	–	0.773	**0.803**
	MAE	–	0.149	0.160	0.130	0.189	0.364	0.145	**0.110**	–	0.141	**0.128**

6. Conclution

- ✓ We propose DNNs for saliency detection by combining local estimation and global search. In the local estimation stage, the proposed DNN-L estimates local saliency by learning rich image patch features from local contrast, texture and shape information. In the global search stage, the proposed DNN-G effectively exploits the complex relationships among global saliency cues and predicts the saliency value for each object region.

- ✓ Our method integrates low level saliency and high level objectness through a supervised DNN-based learning schemes.

- ✓ Experimental results on four benchmark data sets show that the proposed algorithm achieves favorable results against the state-of-theart methods.

Thanks for your attention