Deep Learning Based Salient Object Detection

Huchuan Lu

Dalian University of Technology

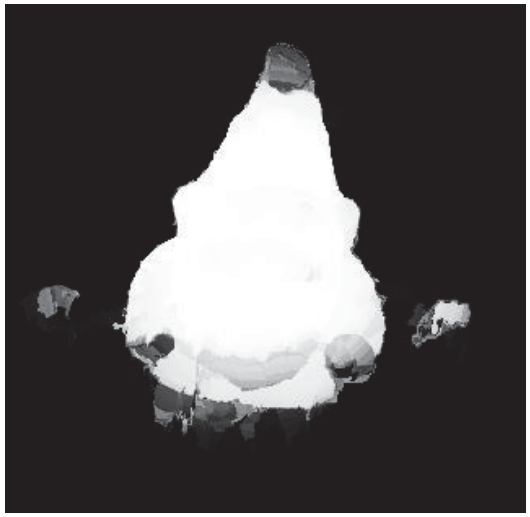
Outline

- Brief Introduction
- Saliency's Application
- Deep Learning Methods for Saliency Detection

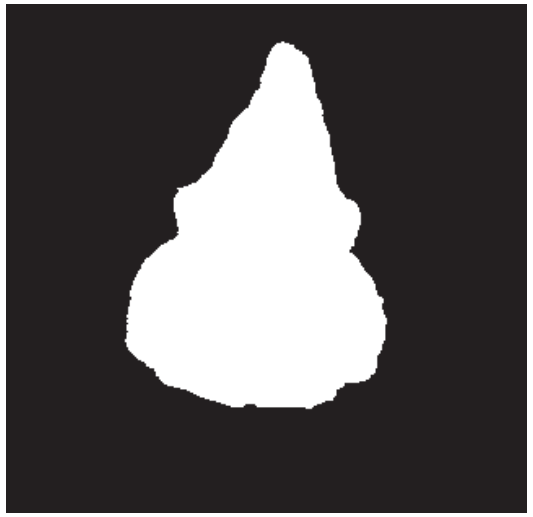
Goal of salient object detection

- **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

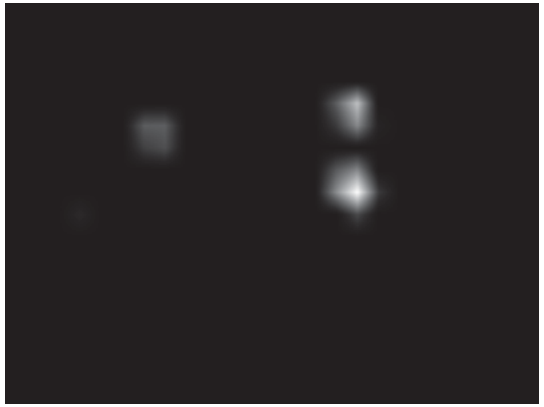
Category

- **Objective**

  — Eye Fixation Prediction

  — Salient Object Detection

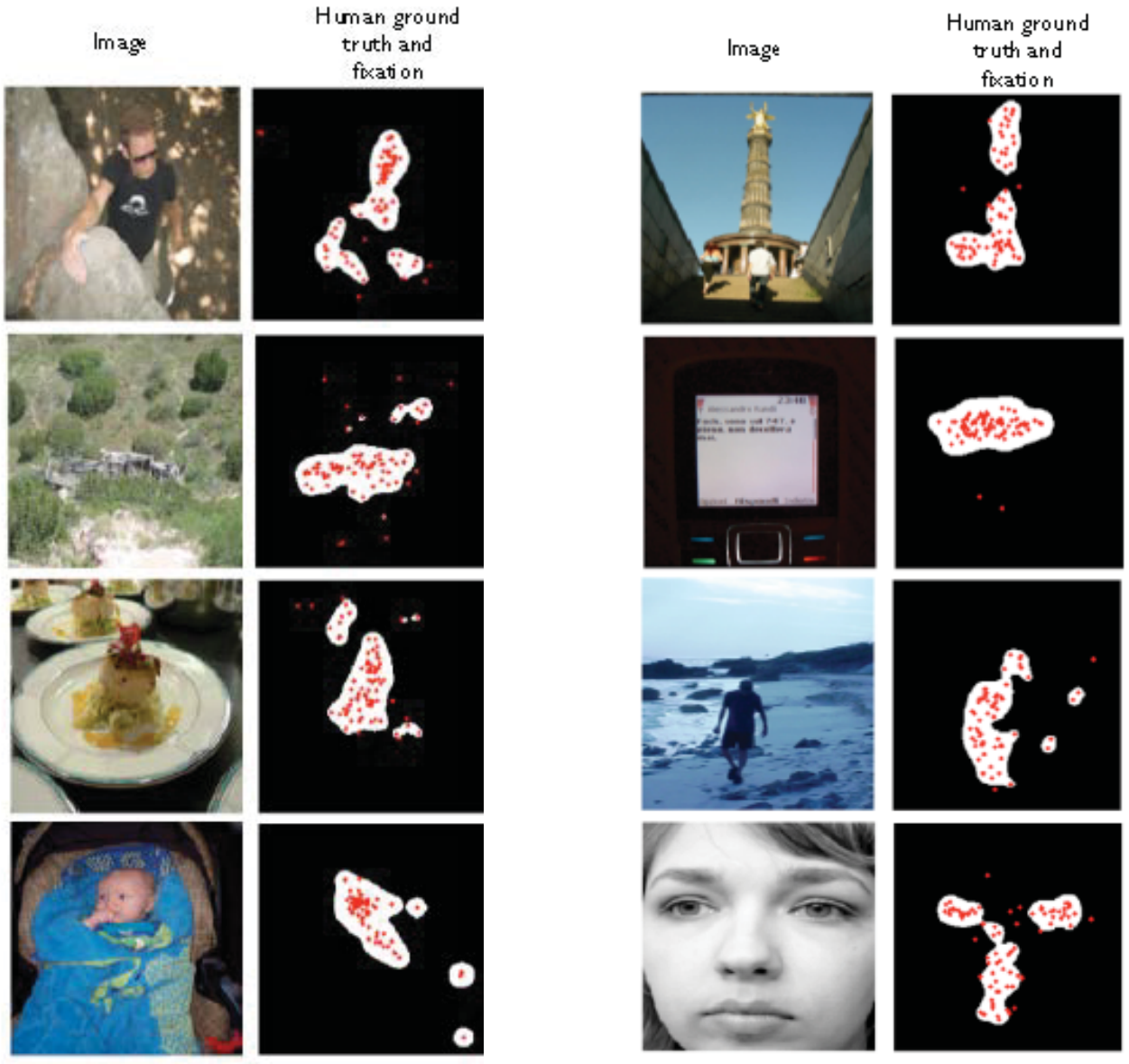
Input Image



Eye Fixation Map

Saliency Map

Eye Fixation Prediction

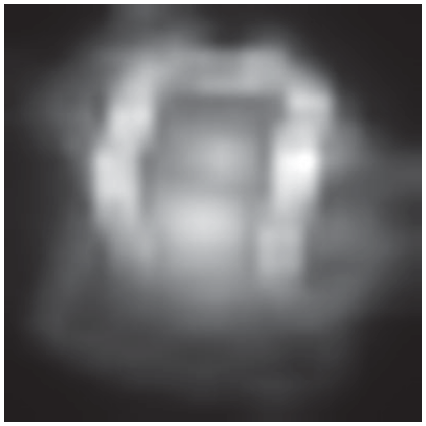


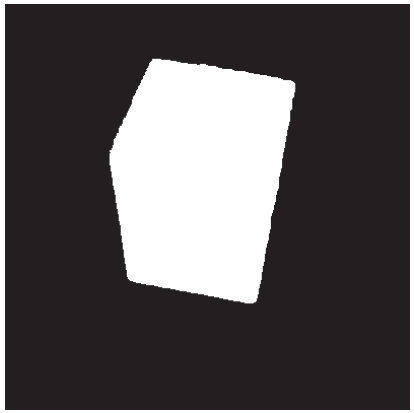
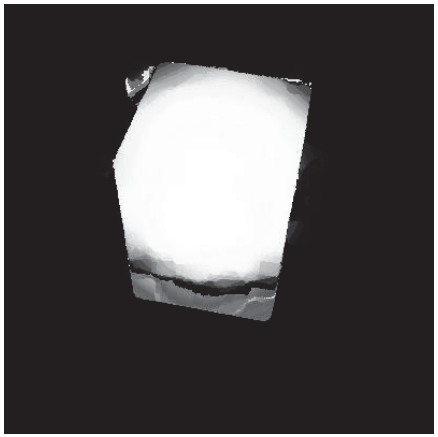
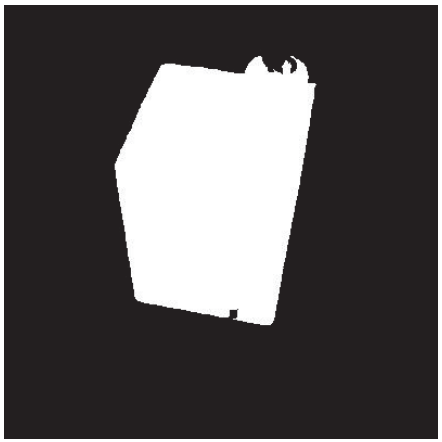
Application

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

Application—Image Segmentation

- Image segmentation is often described as partitioning an image into a set of non-overlapping regions covering the entire image.

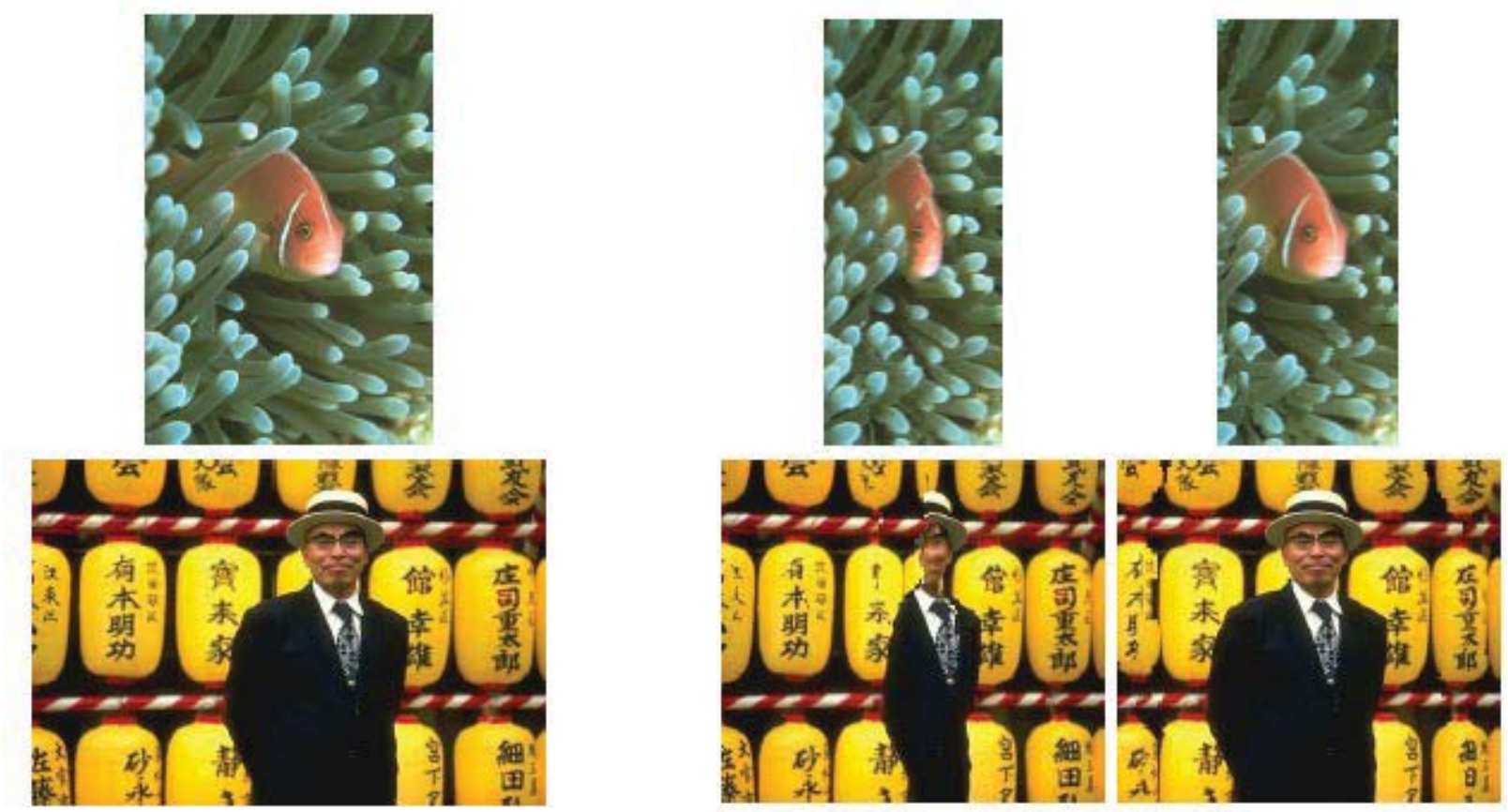
Input Image
Ground Truth

Saliency Map

Segmentation Result

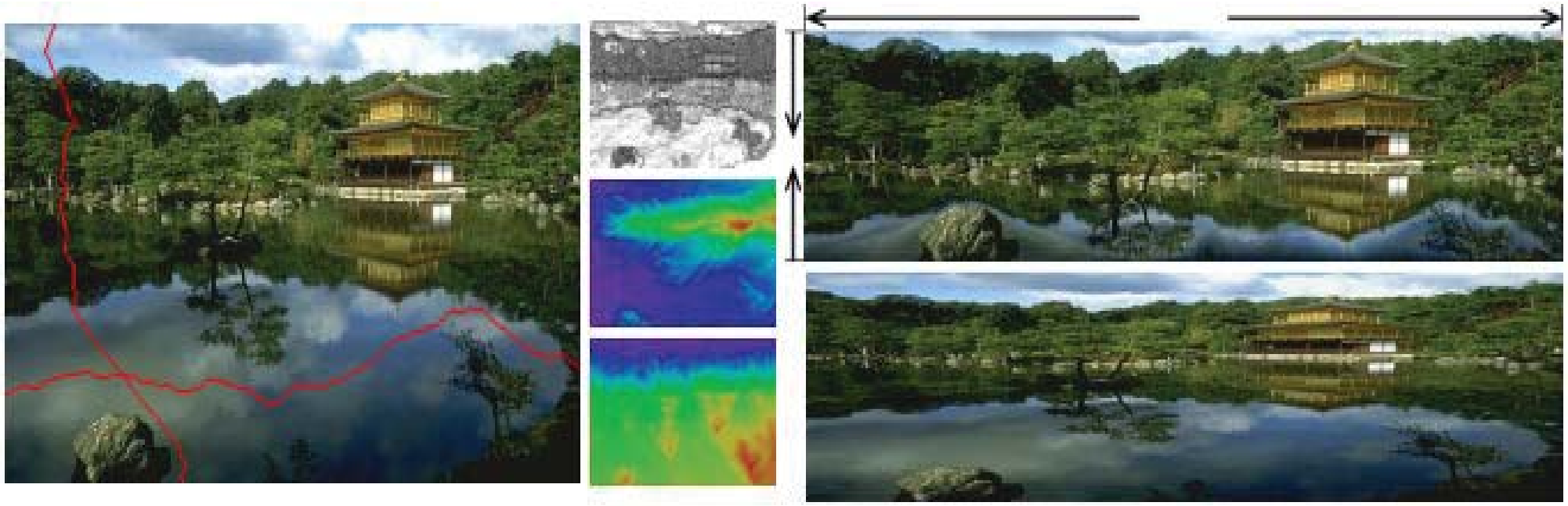
Application—Image Retargeting (Resizing)

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



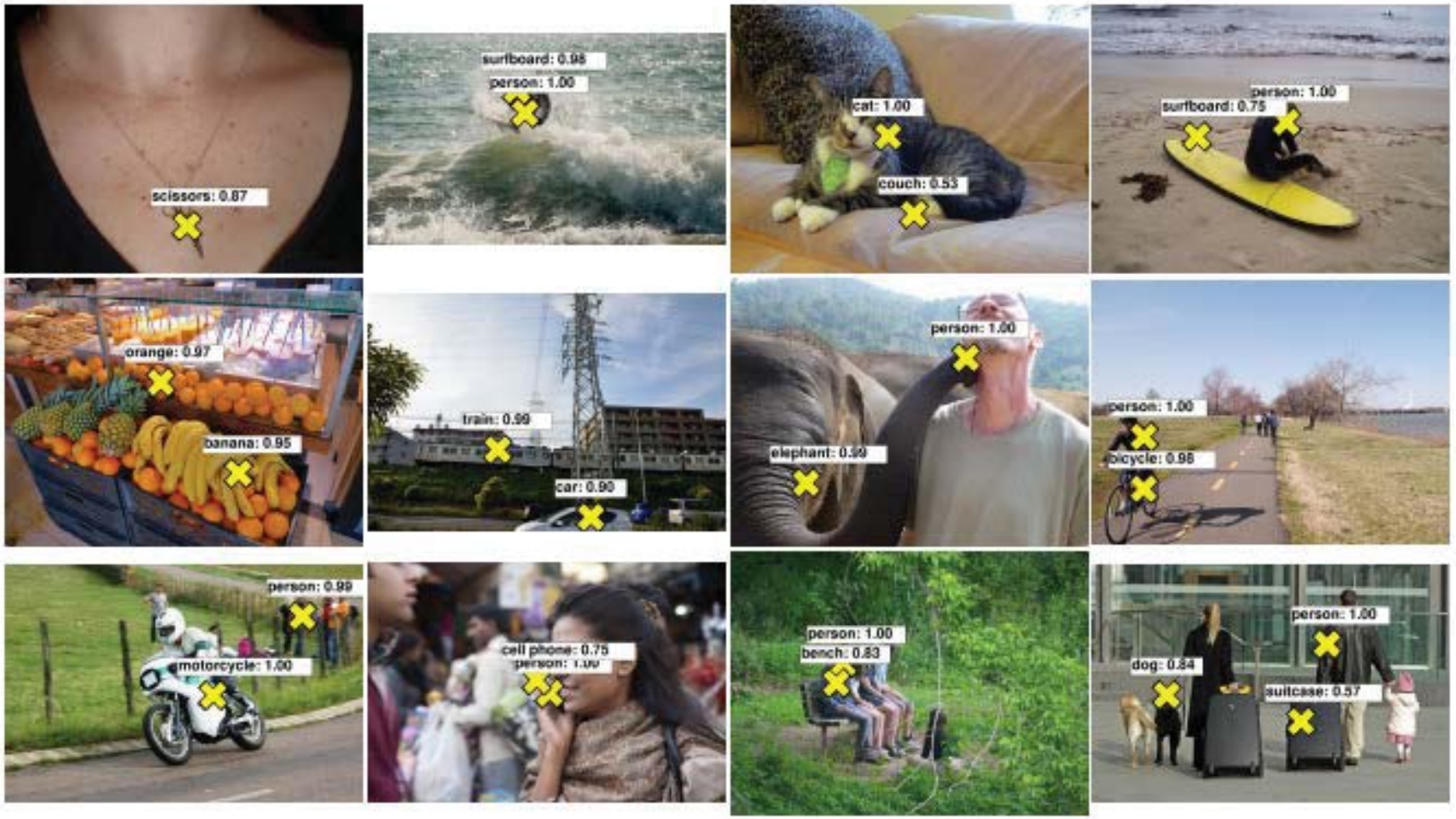
Application—Image Retargeting (Resizing)

- 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.

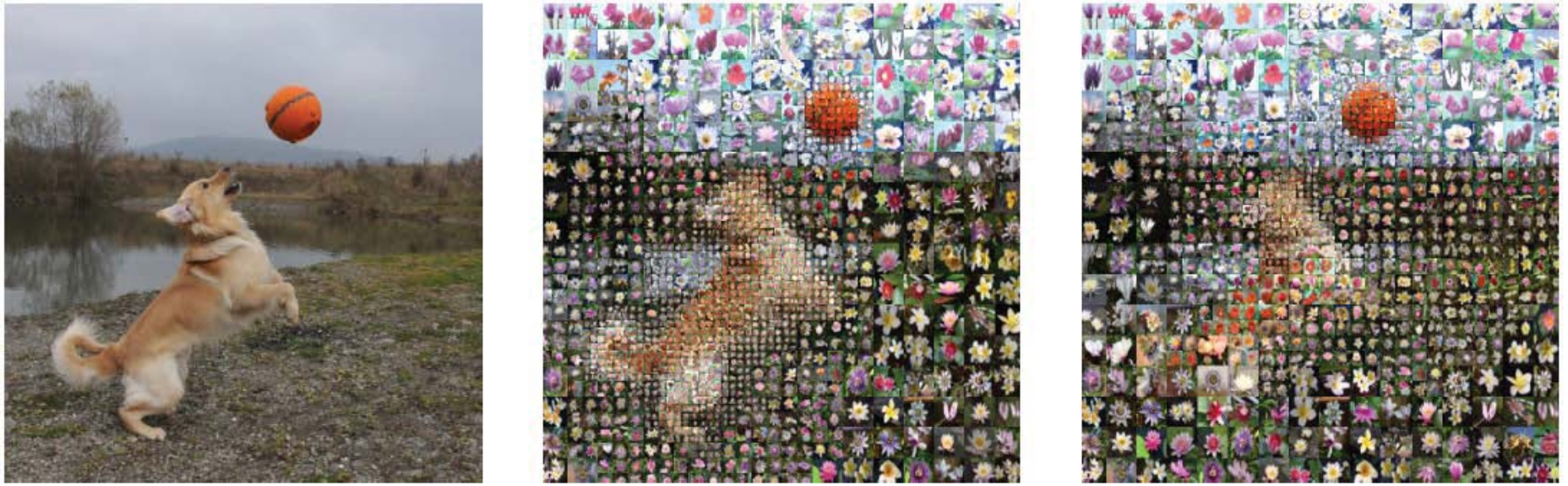
From left to right: query and results.
The correct answers are outlined in green.

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.

Deep Learning for Saliency

- Deep neural networks, e.g. CNNs, have recently achieved great success in various computer vision tasks, such as Image Classification, Object Detection and Semantic Segmentation.

- It has been demonstrated that deep features are highly versatile and have stronger representative power than traditional handcrafted low-level features.

- Training an end-to-end model for single-task or multi-task is more and more popular, and all the deep learning based methods outperform the traditional methods with a large margin.

Deep Learning Methods

Paper	Model	Publication	Year	Author
Deep Networks for Saliency Detection via Local Estimation and Global Search	LEGS	CVPR	2015	Lijun Wang
Visual Saliency Based on Multiscale Deep Features	MDF	CVPR	2015	Guanbin Li
Saliency Detection by Multi-Context Deep Learning	MCDL	CVPR	2015	Rui Zhao
HARF: Hierarchy-associated Rich Features for Salient Object Detection	HARF	ICCV	2015	Wenbin Zou
Deep Saliency with Encoded Low level Distance Map and High Level Features	ELD	CVPR	2016	Gayoung Lee
Saliency Unified: A Deep Architecture for simultaneous Eye Fixation Prediction and Salient Object Segmentation	SU	CVPR	2016	Srinivas Kruthiventi

Deep Learning Methods

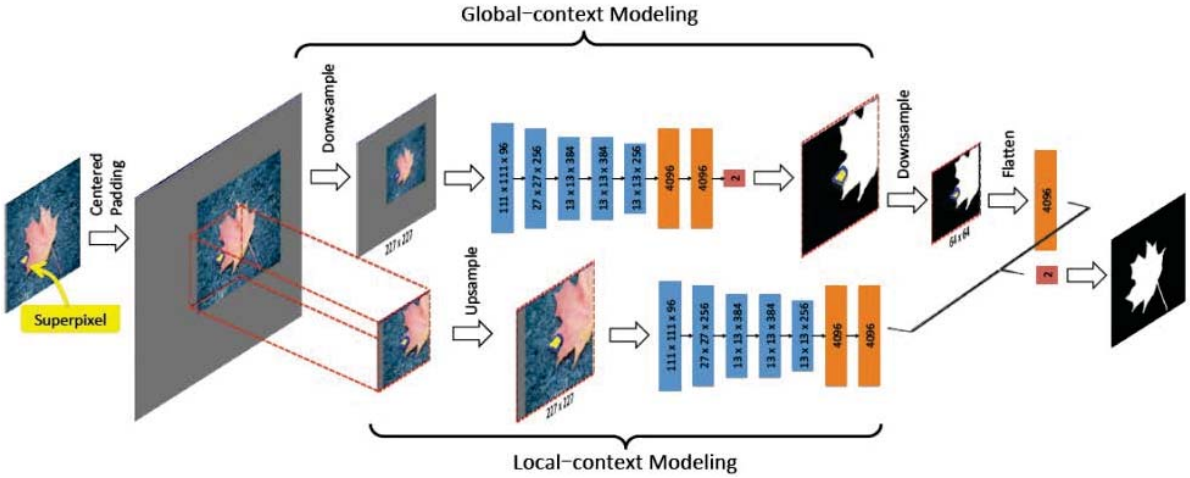
Paper	Model	Publication	Year	Author
DHSNet: Deep Hierarchical Saliency Network for Salient Object Detection	DHSNet	CVPR	2016	Nian Liu
Deep Contrast Learning for Salient Object Detection	DCL	CVPR	2016	Guanbin Li
Recurrent Attentional Networks for Saliency Detection	RACDNN	CVPR	2016	Jason Kuen
Saliency Detection with Recurrent Fully Convolutional Networks	RFCN	ECCV	2016	Lizhao Wang
Saliency Detection via Combining Region-Level and Pixel-Level Predictions with CNNs	CRPSD	ECCV	2016	Youbao Tang
A Shape-based Approach for Salient Object Detection Using Deep Learning	SSD	ECCV	2016	Jongpil Kim
DeepSaliency: Multi-Task Deep Neural Network Model for Salient Object Detection	MTDNN	TIP	2016	Xi Li

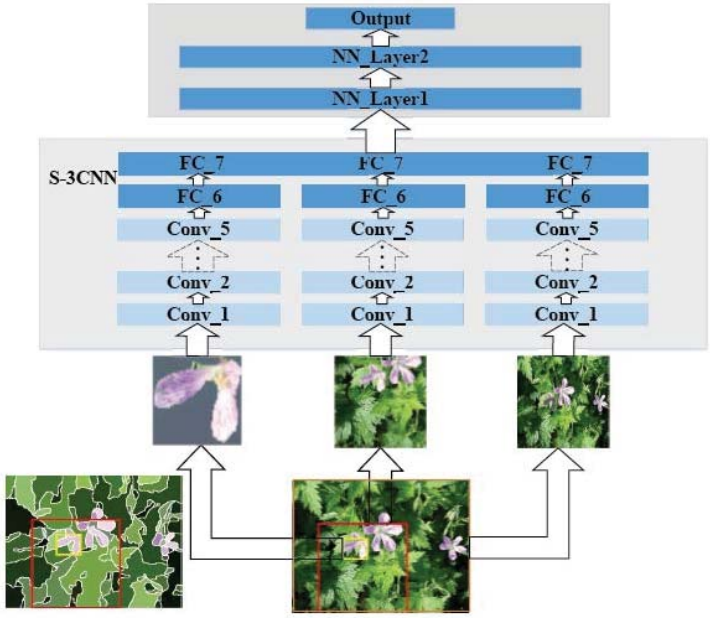
Deep Model => Deep Feature Extraction

◆ Saliency Detection by Multi-Context Deep Learning

Zhao Rui, CVPR2015

- Global context/feature
- Local context/feature





◆ Visual Saliency Based on Multiscale Deep Features

Guanbin Li, CVPR2015

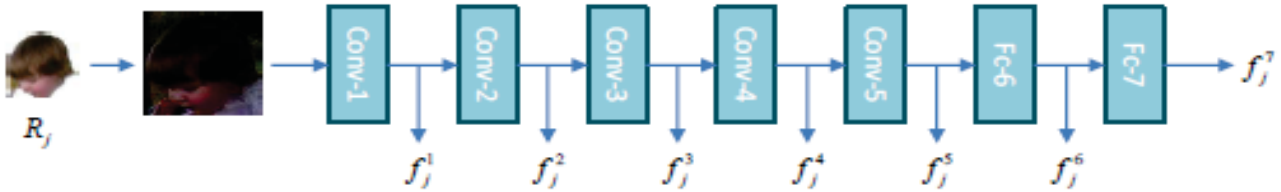
- feature extraction at three different scales

Deep Model => Deep Feature Extraction(VGG)

HARF: Hierarchy-associated Rich Features for Salient Object Detection

Wenbin Zou, ICCV 2015

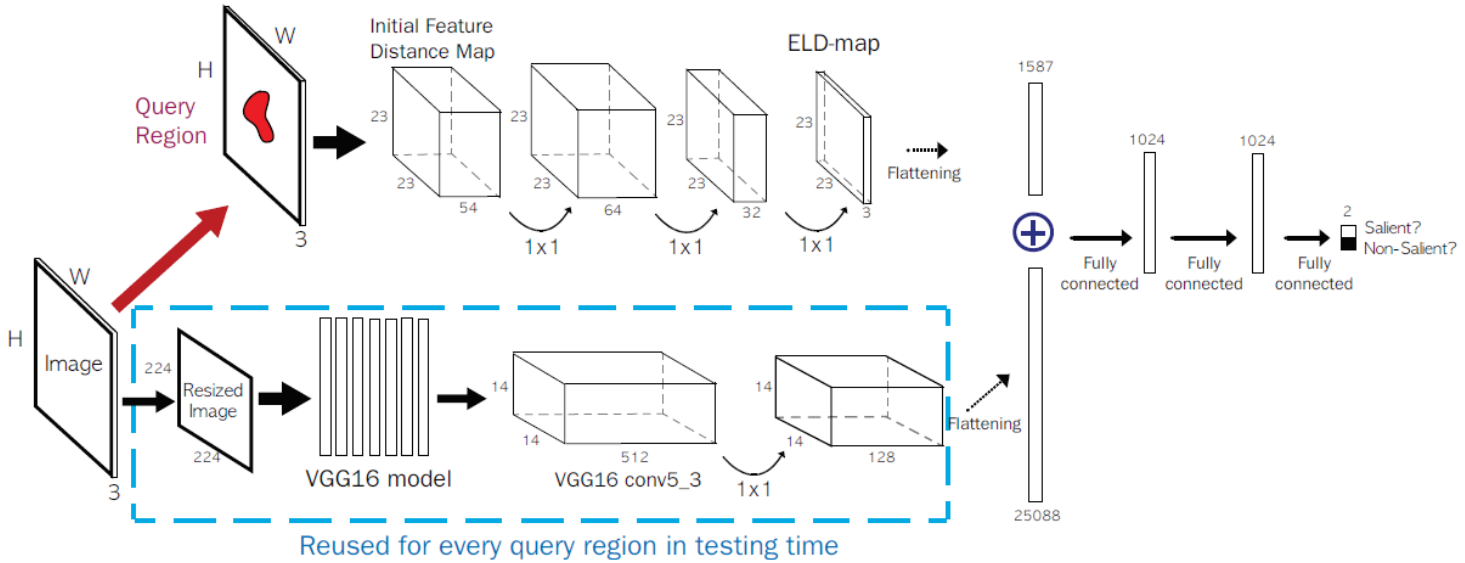
- The outputs of each layer of CNN are used as features



Deep Saliency with Encoded Low Level Distance Map and High Level Features

Gayoung Lee, CVPR2016

- The encoded low level distance map and the high level features are connected to a fully connected neural network classifier to evaluate the saliency of a query region

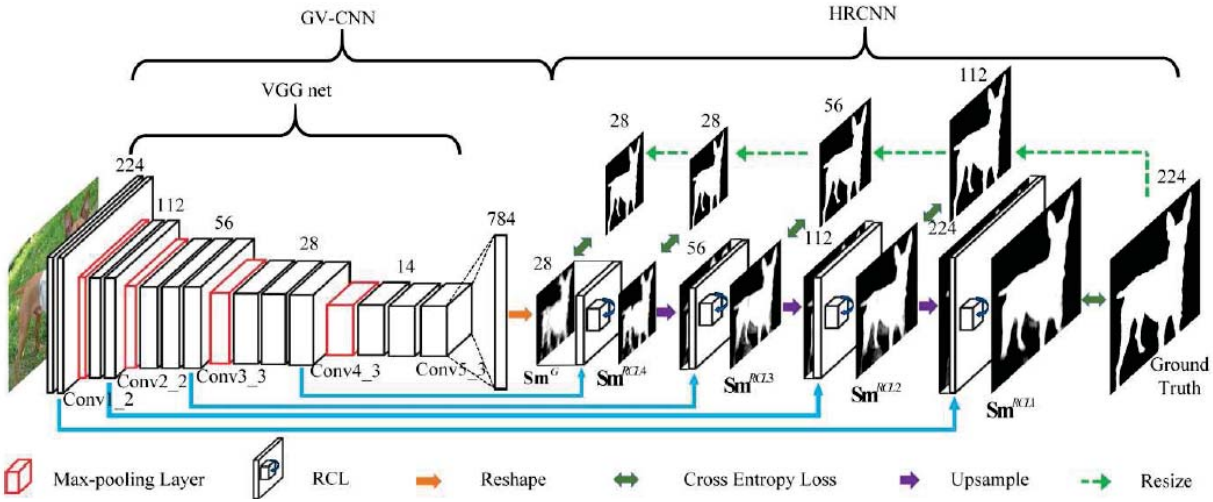


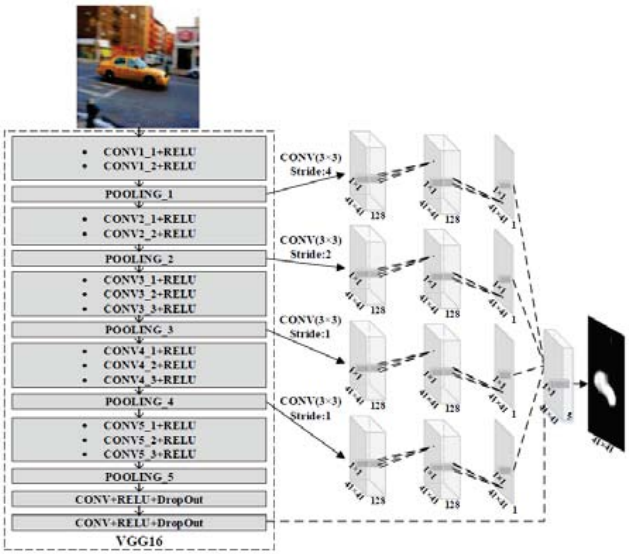
Deep Model => Fully Convolutional Framework

◆ DHSNet: Deep Hierarchical Saliency Network for Salient Object Detection

Nian Liu, CVPR2016

- Deeply supervised learning





◆ Deep Contrast Learning for Salient Object Detection

Guanbin Li, CVPR2016

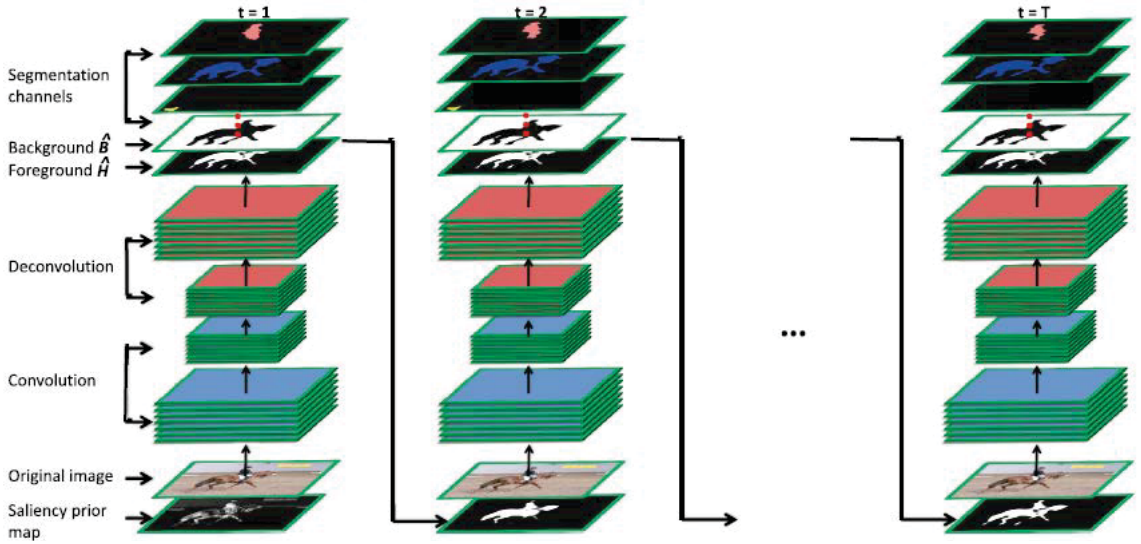
- Multi-scale fully convolutional network

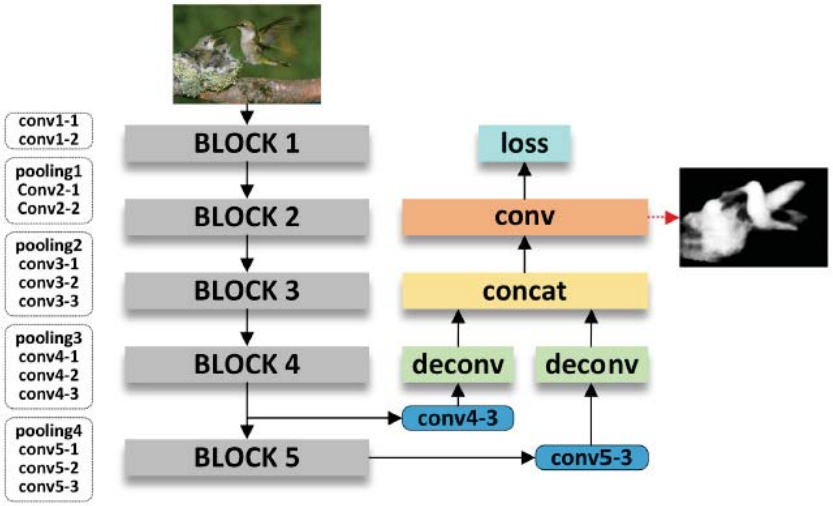
Deep Model => Fully Convolutional Framework

◆ Saliency Detection with Recurrent Fully Convolutional Networks

Lizhao Wang, ECCV2016

- Pre-training strategy using semantic segmentation data.
- Fine-tuning strategy using salient object detection data.





◆ Saliency Detection via Combining Region-Level and Pixel-Level Predictions with CNNs

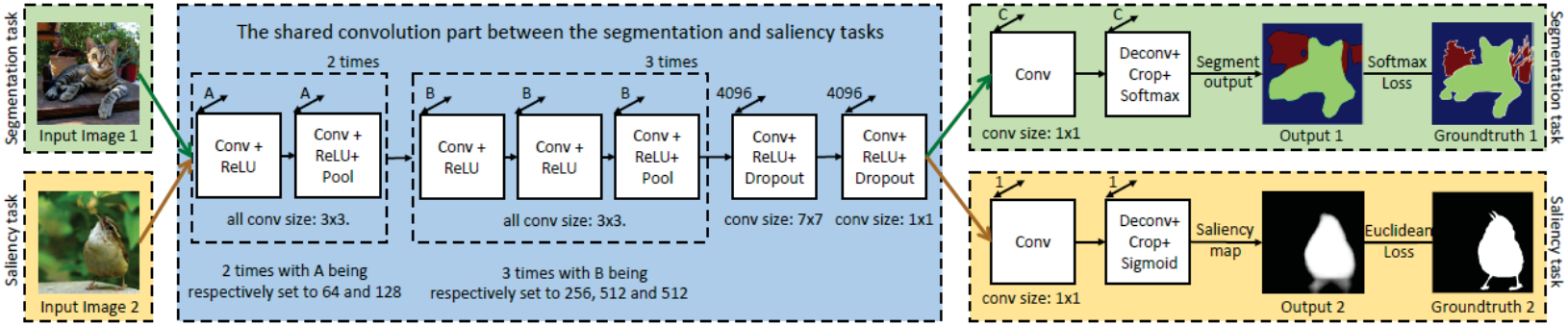
Youbao Tang ECCV2016

- Pixel-level CNN
- Fine-tuning on VGG16
- Last two convolutional layers connected with deconolutional layers

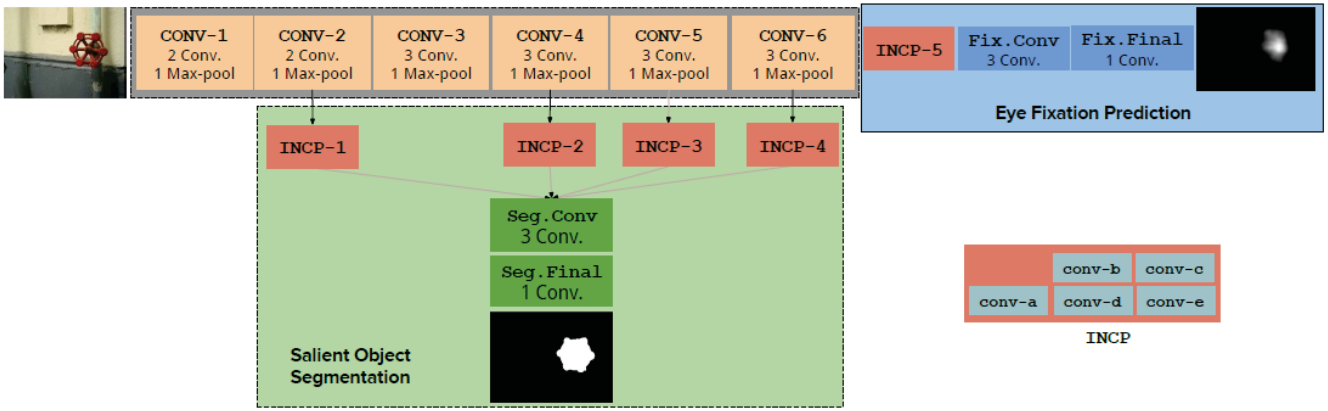
Deep Model => Fully Convolutional Framework

◆ DeepSaliency: Multi-Task Deep Neural Network Model for Salient Object Detection *Xi Li, TIP2016*

- Segmentation
- Saliency Detection



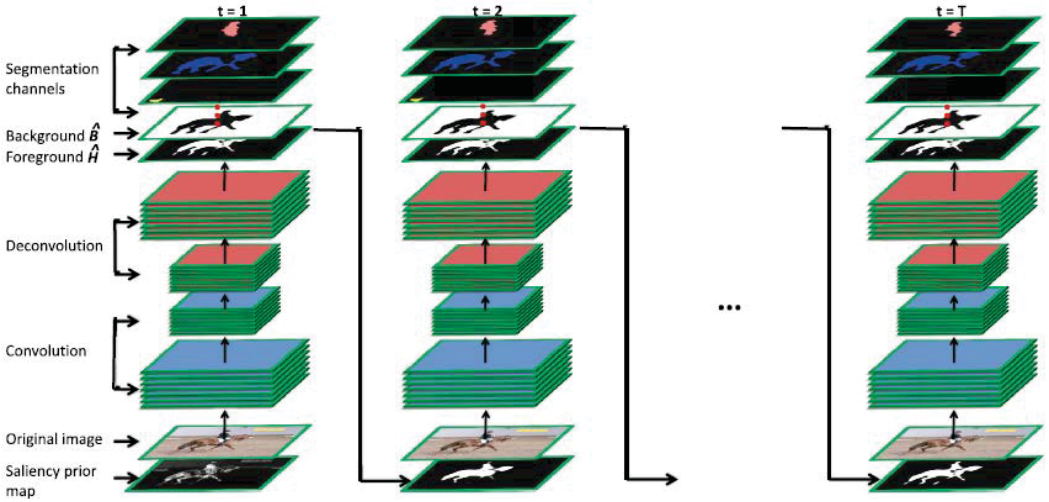
◆ Saliency Unified: A Deep Architecture for simultaneous Eye Fixation Prediction and Salient Object Segmentation *Srinivas S S Kruthiventi, CVPR2016*



- Eye fixation prediction
- Salient object detection

Deep Model => Recurrent Framework

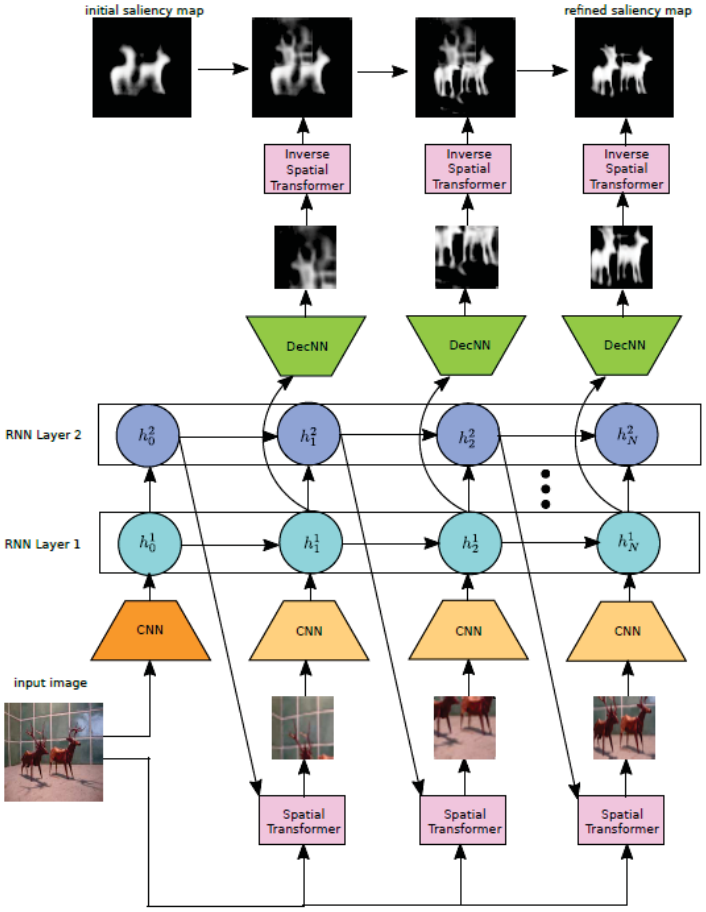
◆ Saliency Detection with Recurrent Fully Convolutional Networks



Lizhao Wang, ECCV2016

◆ Recurrent Attentional Networks for Saliency Detection

Jason Kuen, CVPR2016



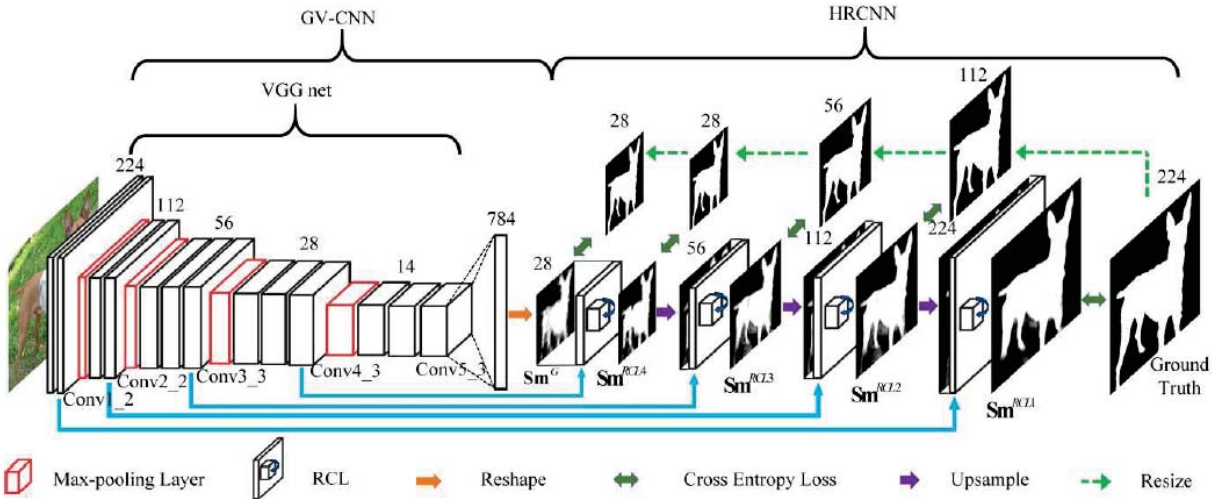
- The recurrent architecture enables to automatically learn to refine the saliency map by correcting its previous errors.

Deep Model => Recurrent Framework

◆ DHSNet: Deep Hierarchical Saliency Network for Salient Object Detection

Nian Liu, CVPR2016

- Deeply supervised learning



Saliency Object Detection Via Learning Method

1. Deep Neural Networks for Saliency Detection, CVPR2015

2. Saliency Detection with Recurrent Fully Convolutional Networks， ECCV2016

Deep Neural Networks for Saliency Detectio

CVPR2015

Lijun Wang, **Huchuan Lu**, Minghsuan Yang

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

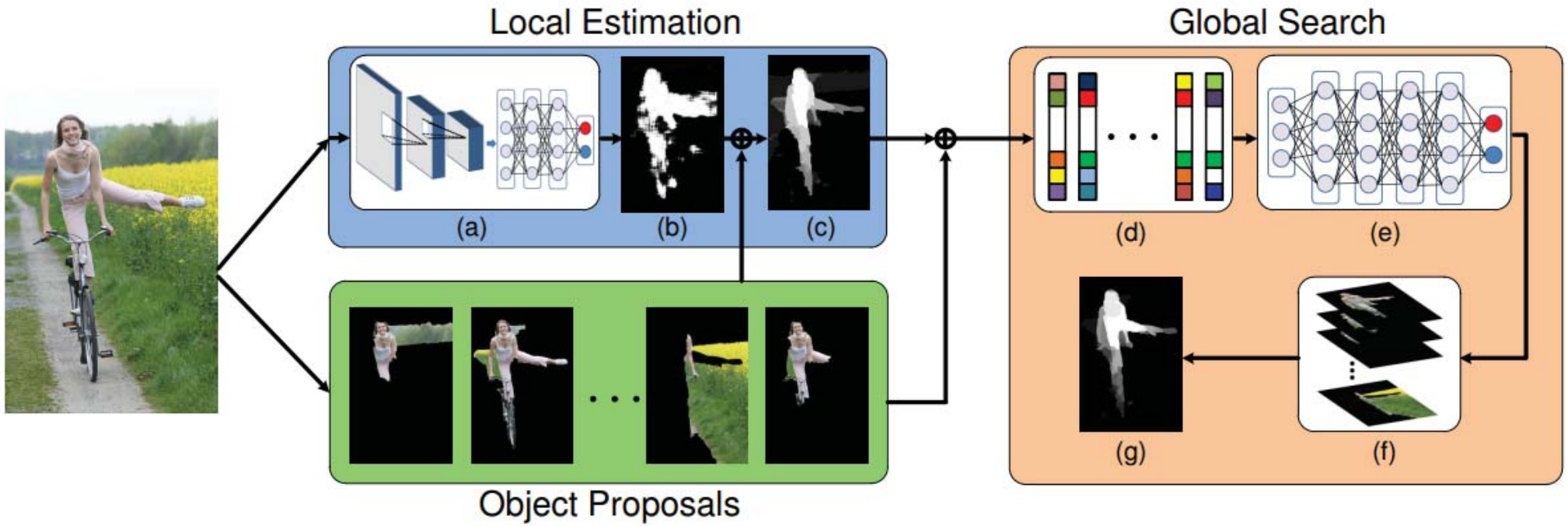


Figure 1. Pipeline of our algorithm. (a) Proposed deep network **DNN-L**. (b) Local saliency map. (c) Local saliency map after refinement. (d) Feature extraction. (e) Proposed deep network **DNN-G**. (f) Sorted object candidate regions. (g) Final saliency map.

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:

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

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

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).

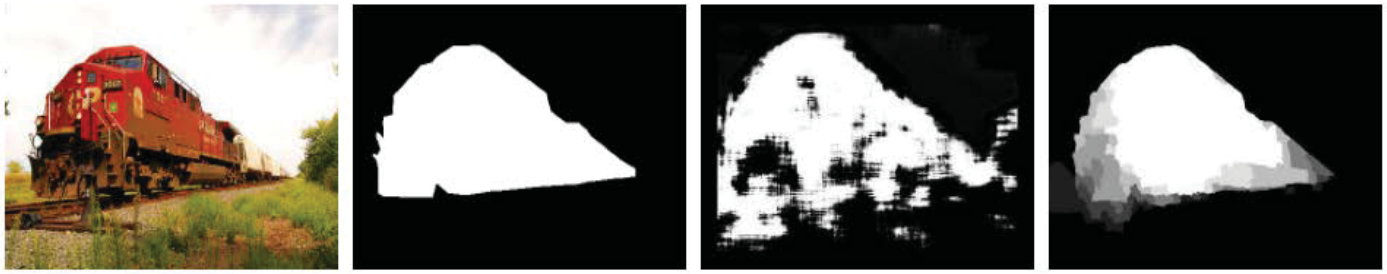


Figure 4. From left to right: source image, ground truth, local saliency map output by DNN-L, local saliency map after refinement.

Pipeline of the proposed approach

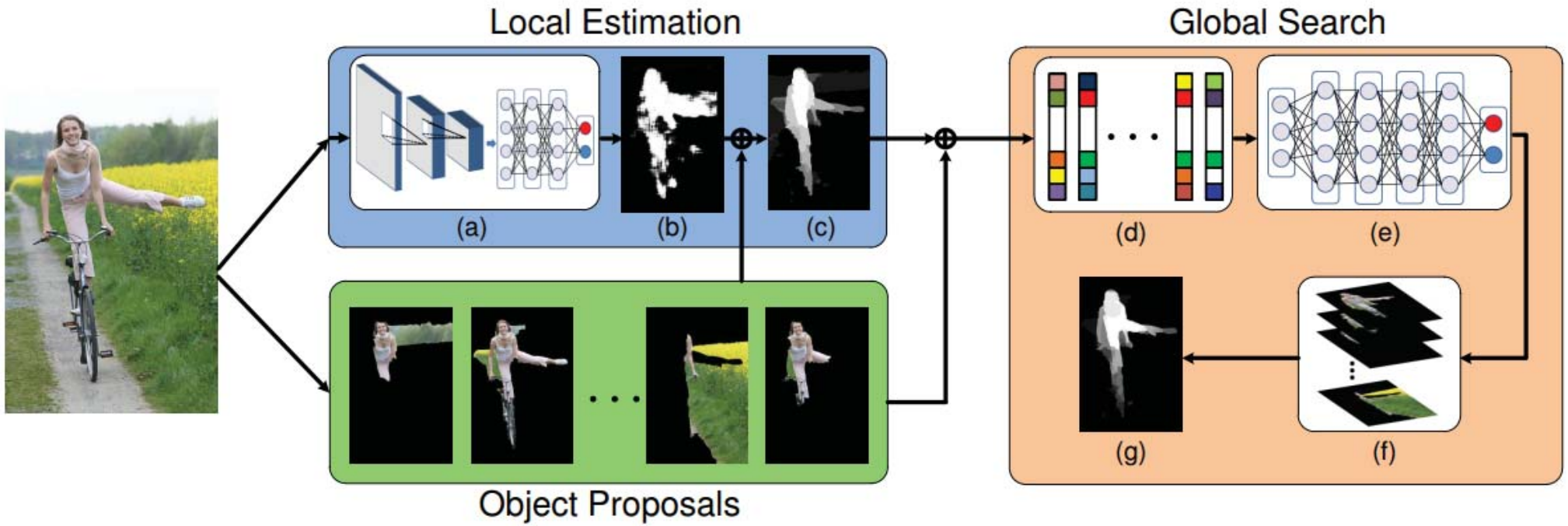


Figure 1. Pipeline of our algorithm. (a) Proposed deep network **DNN-L**. (b) Local saliency map. (c) Local saliency map after refinement. (d) Feature extraction. (e) Proposed deep network **DNN-G**. (f) Sorted object candidate regions. (g) Final saliency map.

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:

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

Where $\phi\left(\mathbf{v}_i\,|\,\boldsymbol{\theta}^G\right)=[\phi_i^1,\phi_i^2]$ 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 $\{\hat{\mathbf{O}}_1, ..., \hat{\mathbf{O}}_N\}$ 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 $\{conf_1^G, ..., conf_N^G\}$. 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

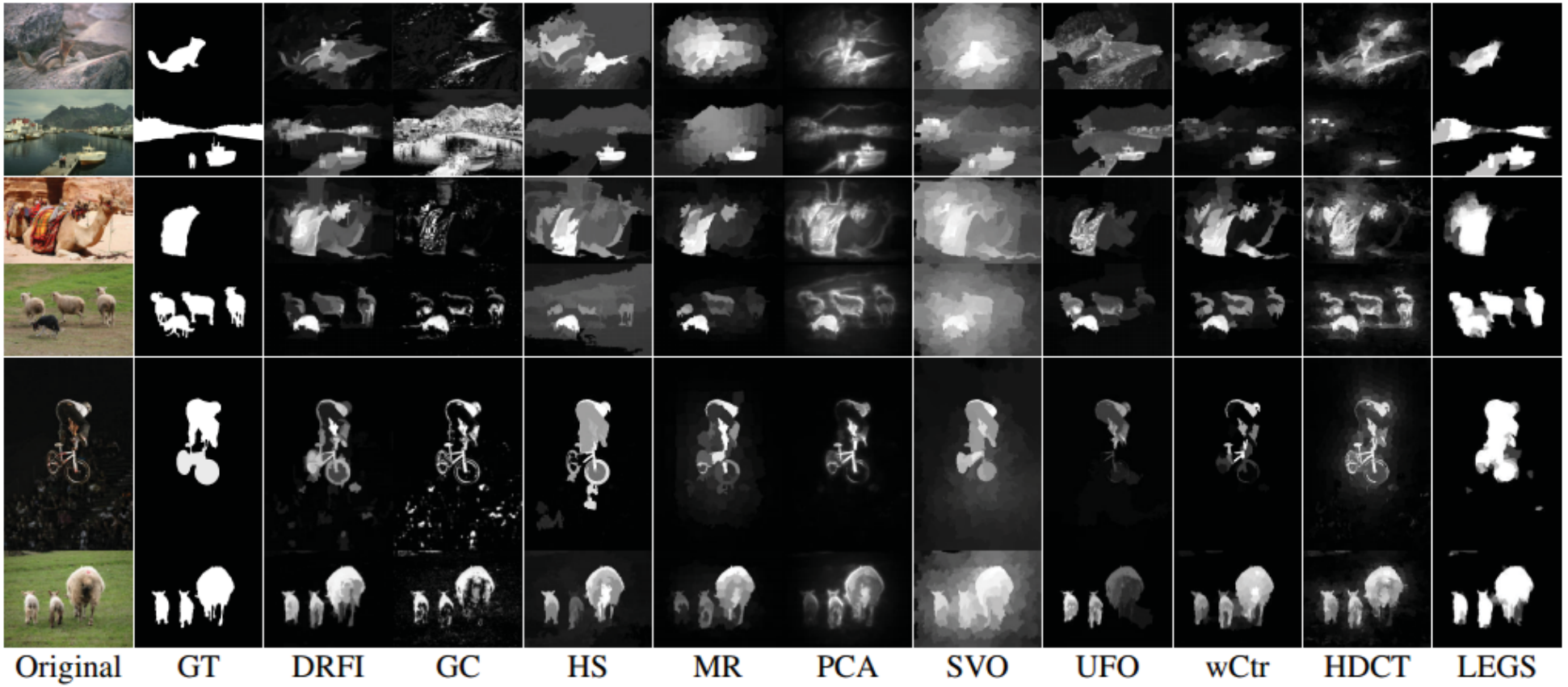


Figure 5. Saliency maps. Top, middle and bottom two rows are images from the SOD, ECCSD and PASCAL-S data sets. GT: ground truth. LEGS: the proposed method.

5. Experimental Results

Quantitative Evaluation

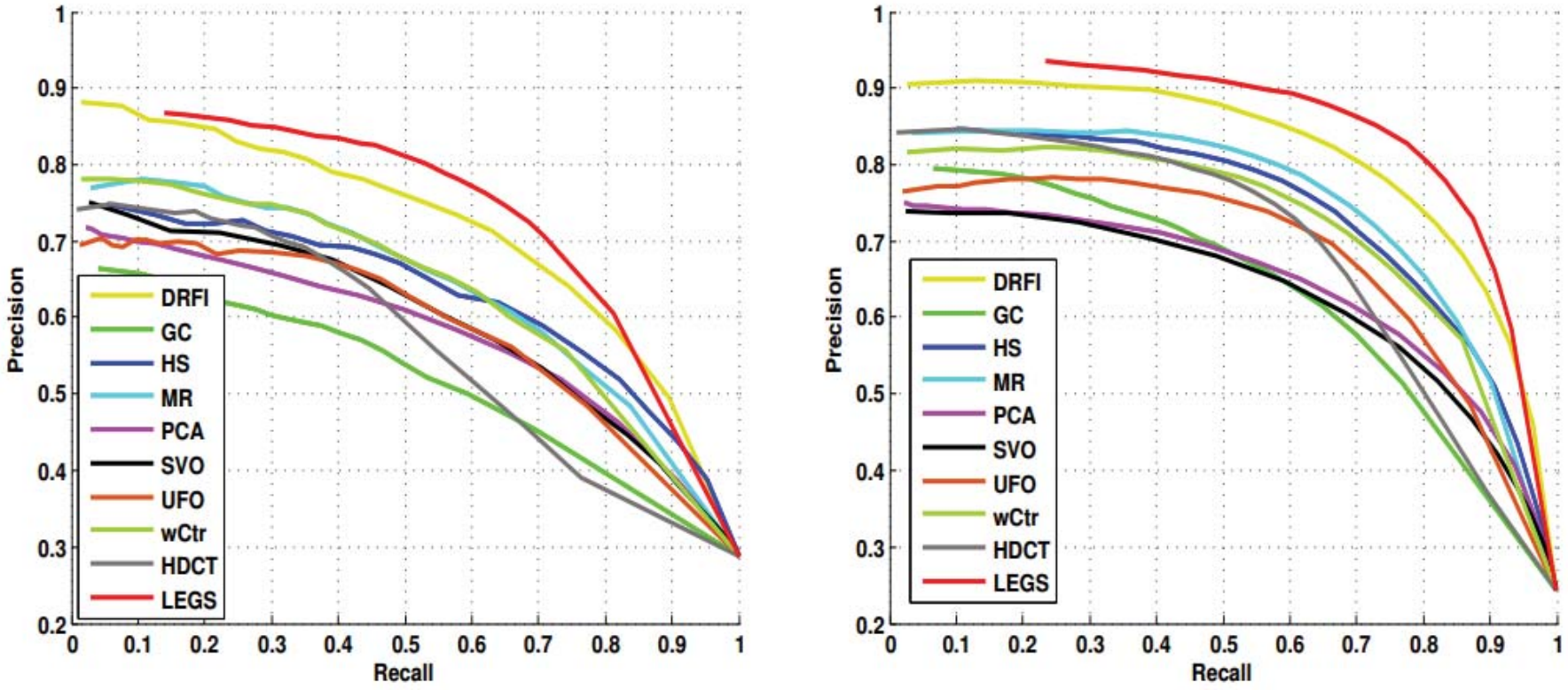


Figure 6. PR curves of saliency detection methods on SOD and ECCSD data set, respectively.

5. Experimental Results

Quantitative Evaluation

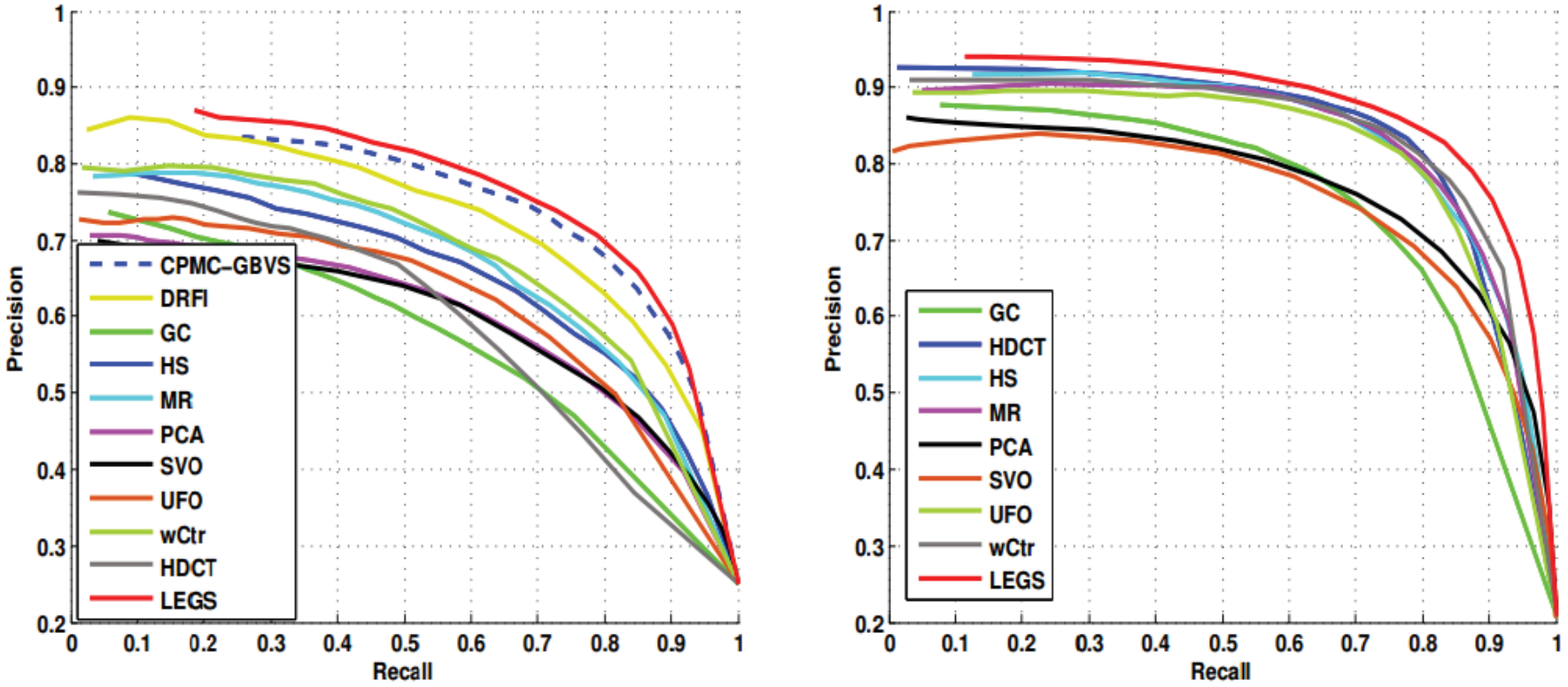


Figure 7. PR curves of saliency detection methods on PASCAL-S and MSRA-5000 data set, respectively.

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.

Saliency Detection with Recurrent Fully Convolutional Networks

ECCV2016

Linzhao Wang, Lijun Wang, **Huchuan Lu**, Pingping Zhang, Xiang Ruan

1. Motivation

• Existing Issues

- **Saliency priors** are completely discarded by most CNN based methods.
- **A limited size** of local image patches is considered by CNNs to predict the saliency label of a pixel.
- **Binary classification**, which represents saliency detection problems, have relatively weak supervision information.

• Our Approach

- **Incorporation** of **saliency priors** into the **network** can facilitate training and inference.
- **Recurrent structure** is utilized to refine the coarse inference from previous time steps.
- **A novel RFCN pre-training** method is designed for saliency detection using semantic segmentation data to leverage strong supervision.

2. Pipeline of the proposed approach

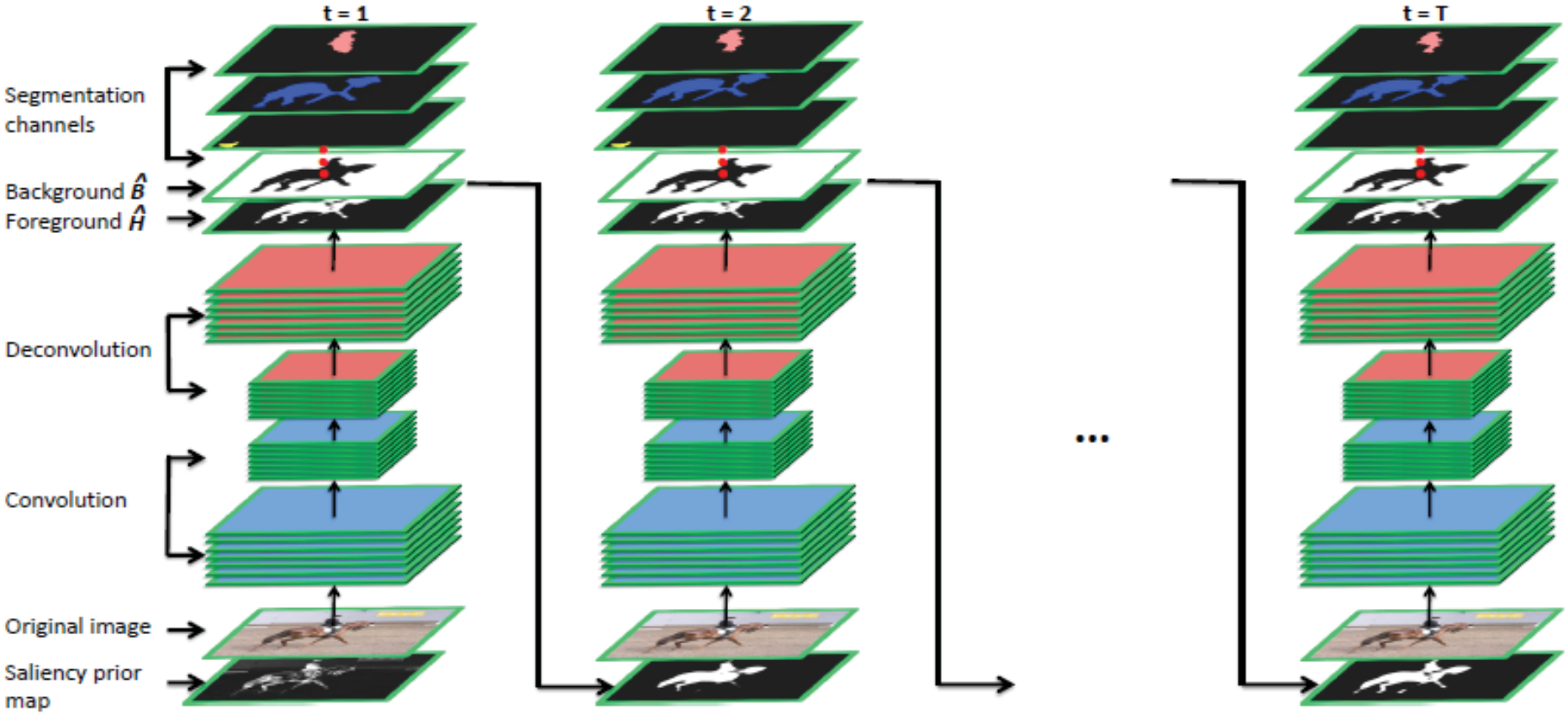


Figure 1. Architecture overview of our RFCN model.

4. Main steps of the proposed approach

1. Saliency Prior Maps
2. Recurrent Architecture
3. Training RFCN
4. Post-Processing

1 Saliency Prior Maps

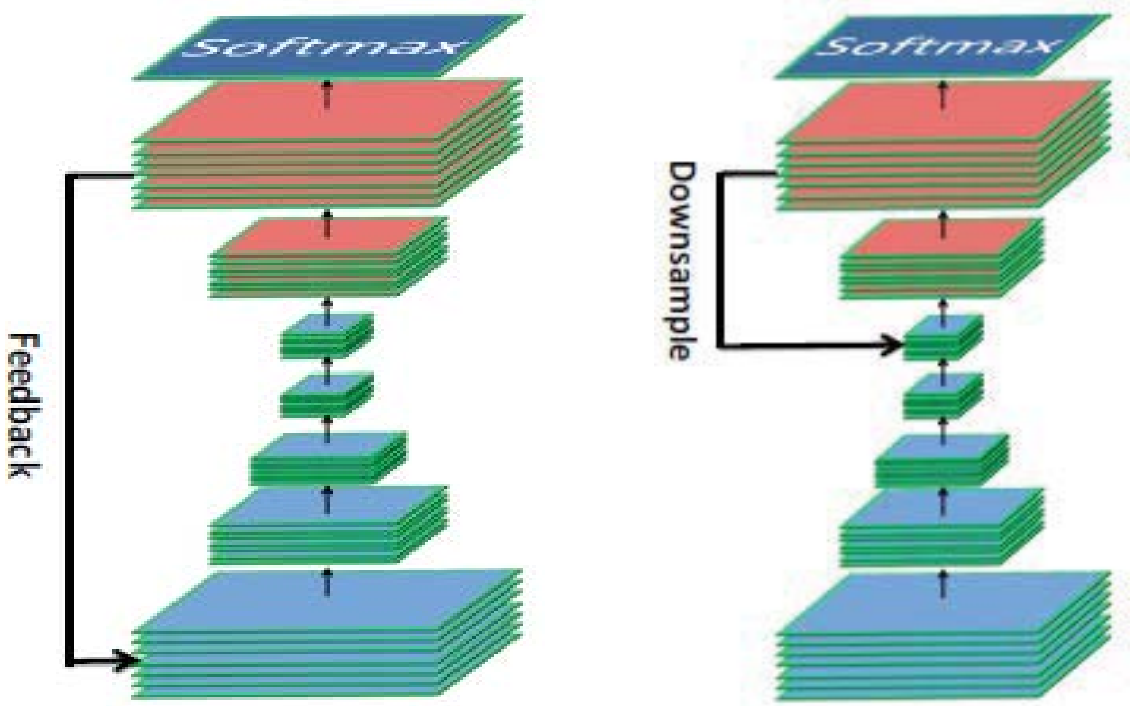
- We encode prior knowledge into a saliency prior map which serves as the input to the network. The priors include color, intensity and orientation feature contrast. Then we integrate these priors together and filter the result with a gaussian function, which proves center prior information.

$$\boldsymbol{P}(s_i) = \mathcal{U}(s_i) \times (\mathcal{G}(s_i) + \mathcal{I}(s_i) + \mathcal{O}(s_i))$$

(a) Image (b) Ground truth (c) Prior map

Figure 2. Saliency prior map.

2 Recurrent Architecture



(a) (b)

Figure 3. Two kind of recurrent architeture.

1. Architecture 1(As in Figure 3(a))

- The saliency prior map is incorporated into the network and the first convolutional layer can be modified by

$$f(\boldsymbol{I}) = \boldsymbol{W_I} * \boldsymbol{I} + \boldsymbol{W_P} * \boldsymbol{P} + \boldsymbol{b}$$

- In the first time step, the prediction is produced with the input image and the saliency prior map. The RFCN then refine the saliency prediction by considering both the input image and the last prediction as

$$\hat{\boldsymbol{Y}}^t = U\left(F(\boldsymbol{I}, \hat{\boldsymbol{H}}^{t-1}; \boldsymbol{\theta}); \boldsymbol{\psi}\right)$$

- For this recurrent architecture, forward propagation of the whole network is conducted in every time step, which is very expensive in terms of both computation and memory. We adopt this architecture for accuracy in our paper.

3 Training RFCN

- Our RFCN training approach consists of two stages: pre-training and fine-tuning.
- Pre-training is conducted on the PASCAL VOC 2010 semantic segmentation data set. Saliency detection and semantic segmentation are highly correlated but essentially different in that saliency detection aims at separating generic salient objects from background, whereas semantic segmentation focuses on distinguishing objects of different categories. The loss function for pre-training across all time steps is defined as

$$L(\boldsymbol{\theta}, \boldsymbol{\psi}) = -\sum_{t=1}^{T} \sum_{\boldsymbol{Z}} \sum_{i,j} \ln p(c_{i,j} = \boldsymbol{S}_{i,j} | \boldsymbol{I}, \hat{\boldsymbol{H}}^{t-1}, \boldsymbol{\theta}, \boldsymbol{\psi}) + \ln p(l_{i,j} = \boldsymbol{G}_{i,j} | \boldsymbol{I}, \hat{\boldsymbol{H}}^{t-1}, \boldsymbol{\theta}, \boldsymbol{\psi})$$

- After pre-training, we modify the RFCN network architecture by removing the first C + 1 channels of the last feature map and only maintaining the last two channels, and then fine-tune it with saliency data set.

4 Post-Processing

- Given the final saliency score map predicted by the RFCN, we first segment the image into foreground and background regions by thresholding it with its mean saliency score.
- We then compute a spatial confidence and a color confidence score for each pixel. The spatial confidence is defined considering the spatial distance of the pixel to the center of the foreground region and the color confidence is defined to measure the similarity of the pixel to foreground region in RGB color space.

$$\boldsymbol{SC}_{i,j} = \exp(-\frac{\|loc_{i,j} - loc_s\|_2}{\sigma})$$

$$\boldsymbol{CC}_{i,j} = \frac{N_{i,j}}{N_s}$$

- Finally, we weight the predicted saliency scores by spatial and color confidences to dilate the foreground region.

$$\tilde{\boldsymbol{H}}_{i,j} = \boldsymbol{SC}_{i,j} \times \boldsymbol{CC}_{i,j} \times \hat{\boldsymbol{H}}^T$$

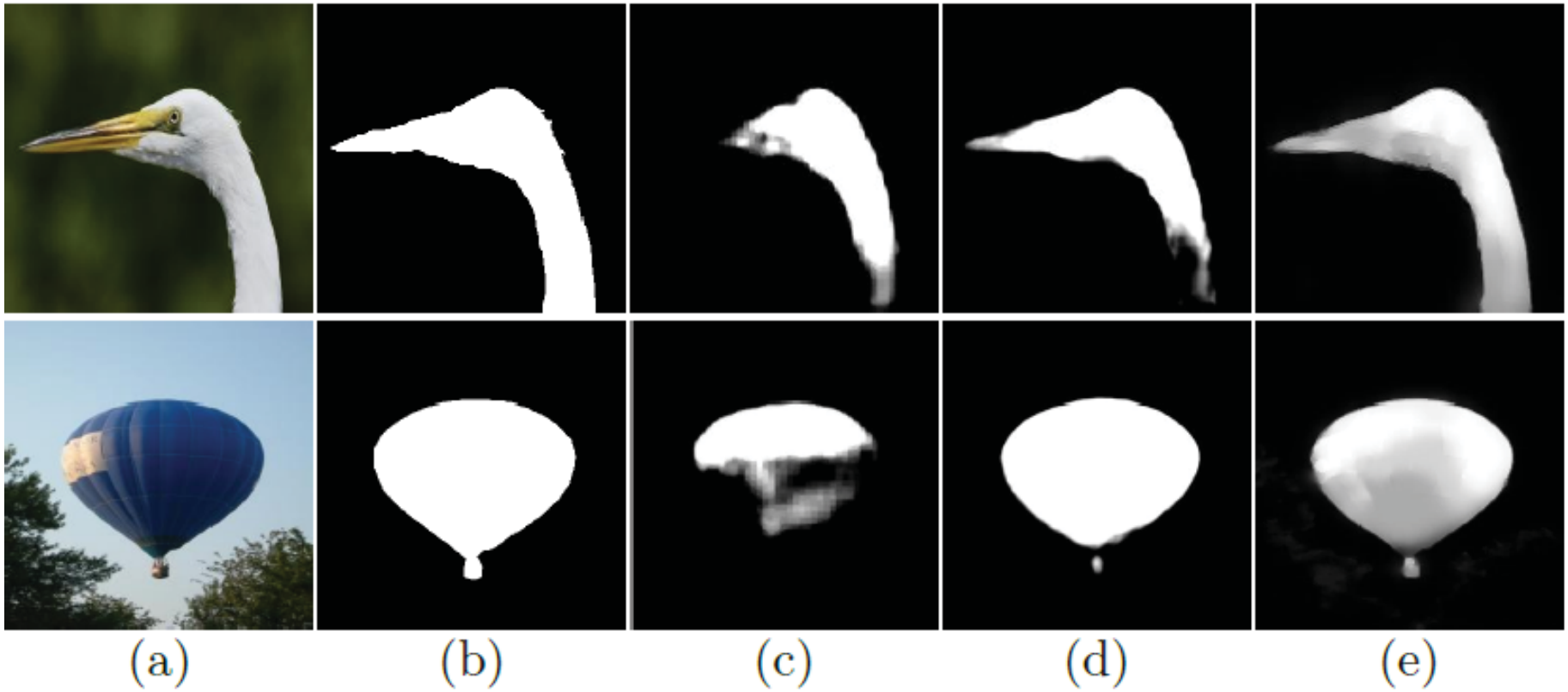


Figure 4. Saliency detection results on different stages. (a)Original images. (b)Ground truth. (c)Results of pre-trained RFCN. (d) Results of ne-tuned RFCN. (e) Result after post-processing.

5. Experimental Results

Setup

- We evaluate the proposed algorithm (RFCN) on five benchmark data sets: SOD [24], ECCSD [35], PASCAL-S [19], SED1 [1] and SED2 [1], against twelve state-of-the-art methods: MTDS [17], LEGS [31], MDF [16], BL [29], DRFI [12], UFO [13], PCA [23], HS [35], wCtr [38], MR [36], DSR [18] and HDCT [14].
- We utilize 10103 training images belonging to 20 object classes from the PASCAL VOC 2010 semantic segmentation data set for pre-training and all the 10k images from THUS10k [2] data set for fine-tuning. We don't adopt any data augmentation.

5. Experimental Results

Quantitative Evaluation

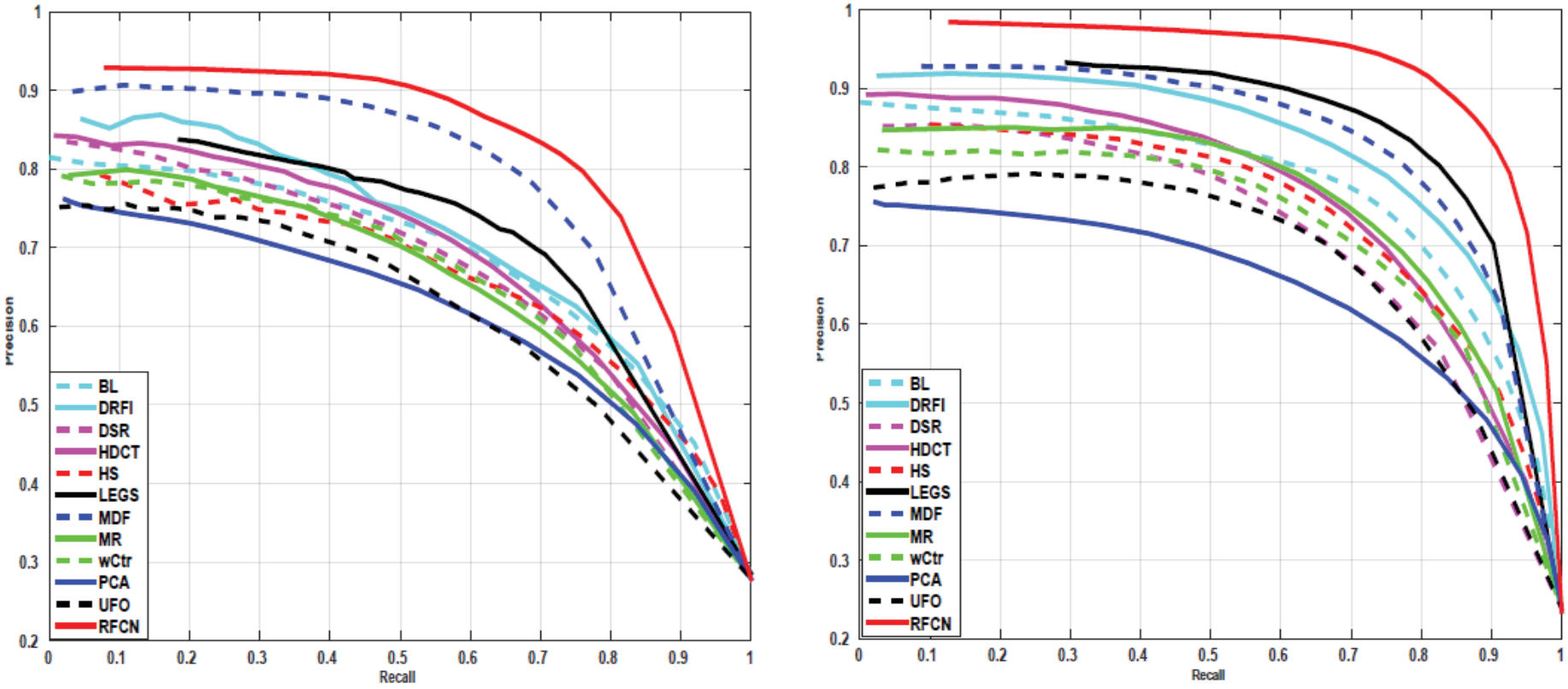


Figure 5. PR curves of saliency detection methods on SOD and ECCSD data set, respectively.

5. Experimental Results

Quantitative Evaluation

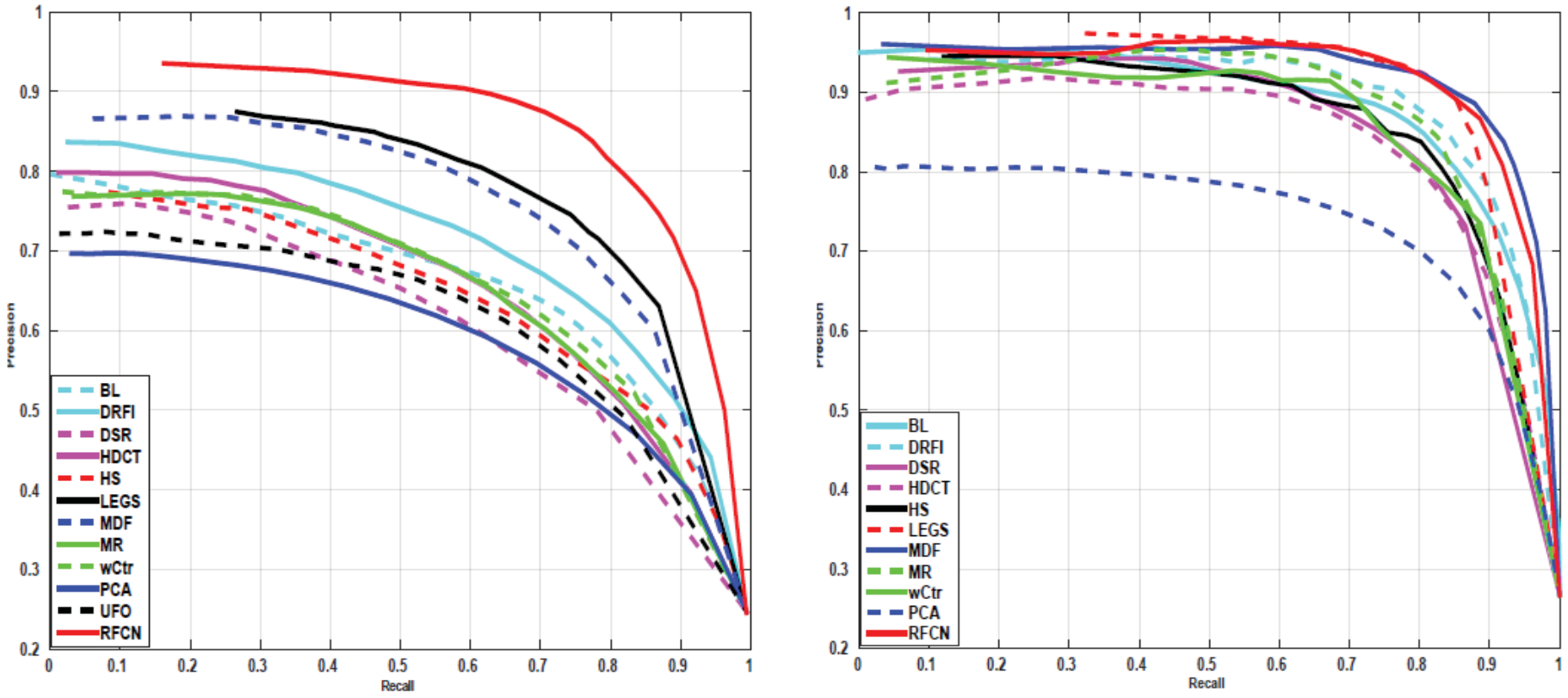


Figure 6. PR curves of saliency detection methods on PASCAL-S and SED1 data set, respectively.

5. Experimental Results

Quantitative Evaluation

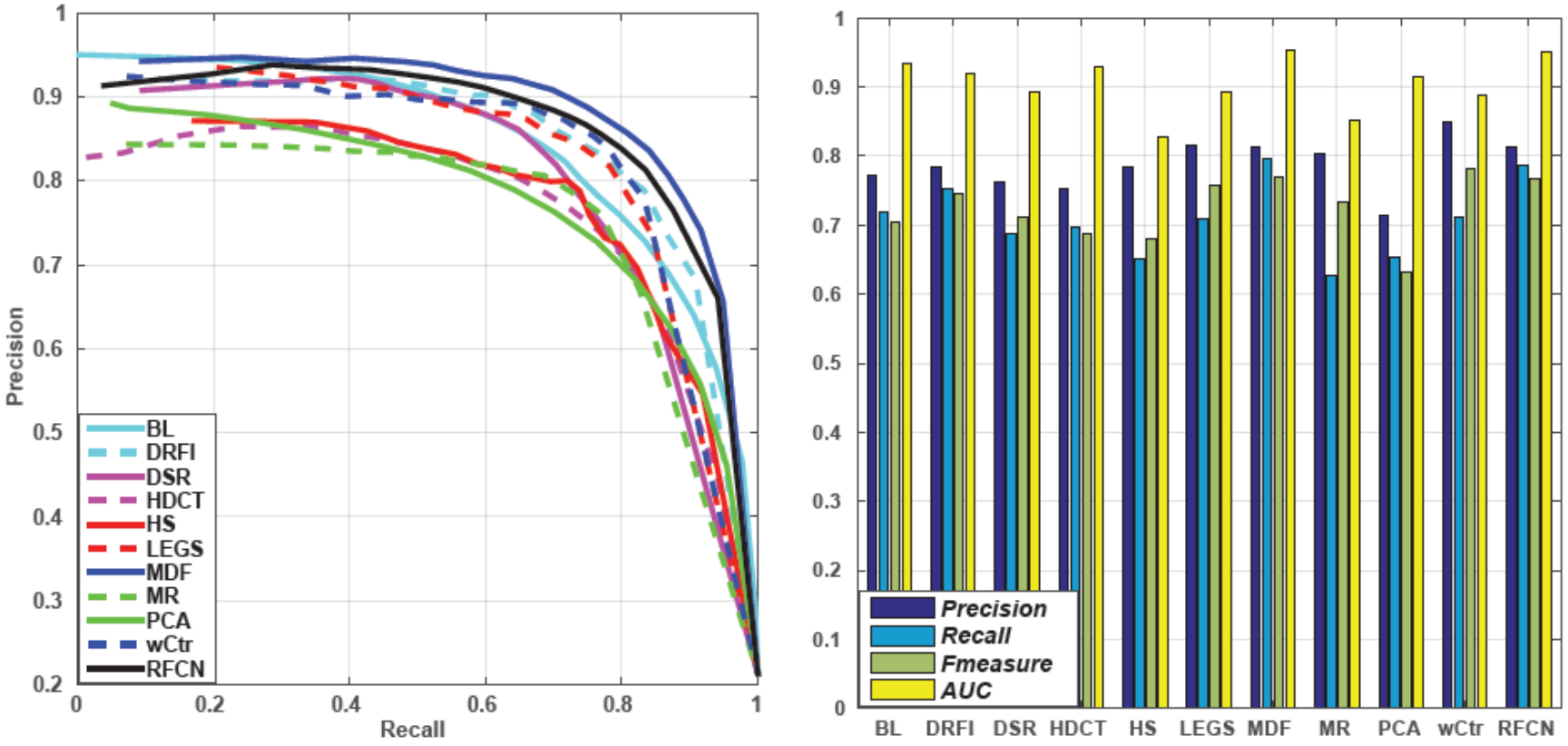


Figure 7. Performance of RFCN on SED2 data set.

5. Experimental Results

Quantitative Evaluation

Table 1. F-measure and AUC (Area Under ROC Curve) on the SOD, ECSSD, PASCAL-S and SED1 data sets. The best two results are shown in red and blue fonts respectively. The proposed methods rank first and second on the four data sets.

*	SOD		ECSSD		PASCAL-S		SED1	
	F-measure	AUC	F-measure	AUC	F-measure	AUC	F-measure	AUC
RFCN	0.7426	0.9053	0.8340	0.9714	0.7468	0.9453	0.8502	0.9640
MTDS	0.6978	0.9233	0.7589	0.9009	0.7310	0.9287	-	-
LEGS	0.6492	0.8117	0.7887	0.9230	0.6951	0.8857	0.8414	0.9328
MDF	0.6966	0.8532	0.7557	0.9180	0.6562	0.8806	0.8194	0.9710
BL	0.5723	0.8503	0.6825	0.9147	0.5668	0.8633	0.7675	0.9528
DRFI	0.6031	0.8464	0.7337	0.9391	0.6159	0.8913	0.8024	0.9528
wCtr	0.5978	0.8014	0.6774	0.8779	0.5972	0.8433	0.7889	0.9159
DSR	0.5968	0.8210	0.6636	0.8604	0.5513	0.8079	0.7877	0.9086
MR	0.5697	0.7899	0.6932	0.8820	0.5881	0.8205	0.8255	0.9223
HS	0.5210	0.8145	0.6363	0.8821	0.5278	0.8330	0.7426	0.9161
PCA	0.5370	0.8212	0.5796	0.8737	0.5298	0.8371	0.6256	0.9030
UFO	0.5480	0.7840	0.6442	0.8587	0.5502	0.8088	-	-

6. Conclution

- ✓ We propose a recurrent fully convolutional network based saliency detection method.
  - Heuristic saliency priors are incorporated into the network to facilitate training and inference.
  - The recurrent architecture enables our method to refine saliency maps based on previous output and yield more accurate predictions.

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

Deep Visual Tracking

Huchuan Lu

lhchuan@dlut.edu.cn

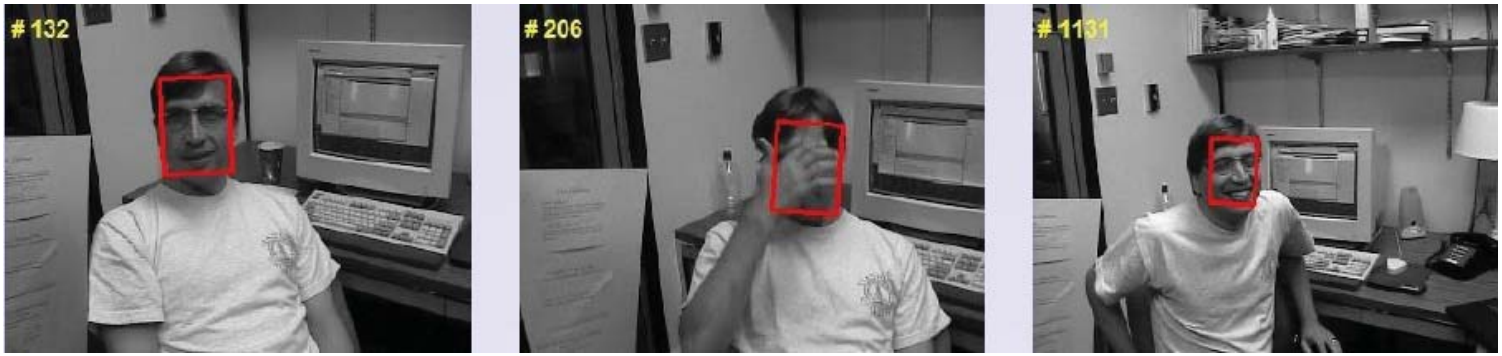
http://ice.dlut.edu.cn/lu/index.html

Outline

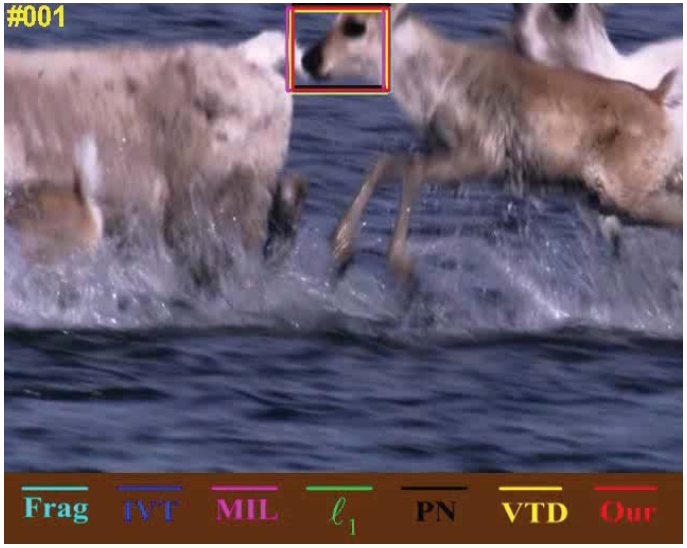
- Visual tracking: a challenging task
- Deep visual tracking: Classification and Review
- FCNT(ICCV15)
- STCT(CVPR16)

Visual Tracking

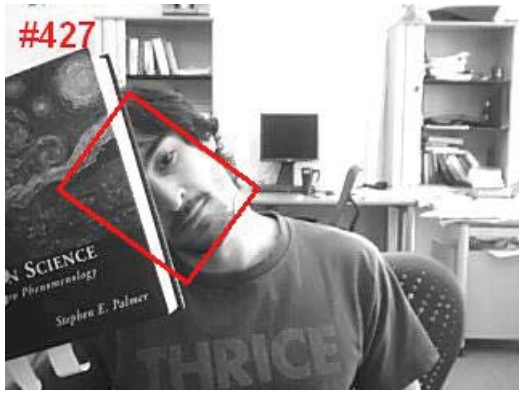
- **Goal**
  - To track an arbitrary object in a video given its initial location.



- **Applications**
  - surveillance
  - motion analysis
  - object recognition
  - human-computer interaction
  - traffic control

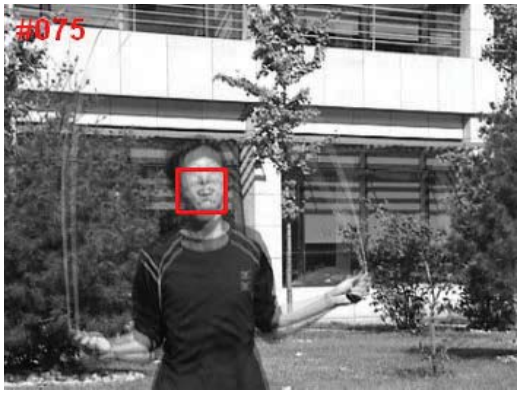


Challenges

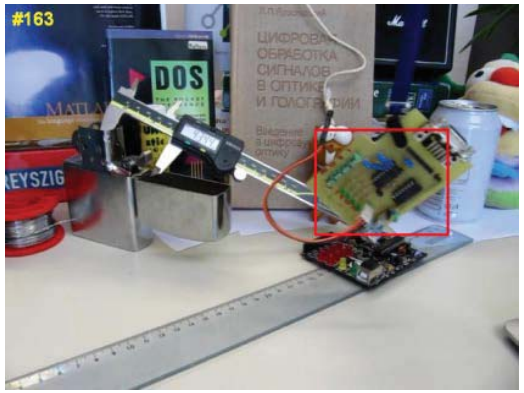


Occlusion

Scale variation



Motion blur



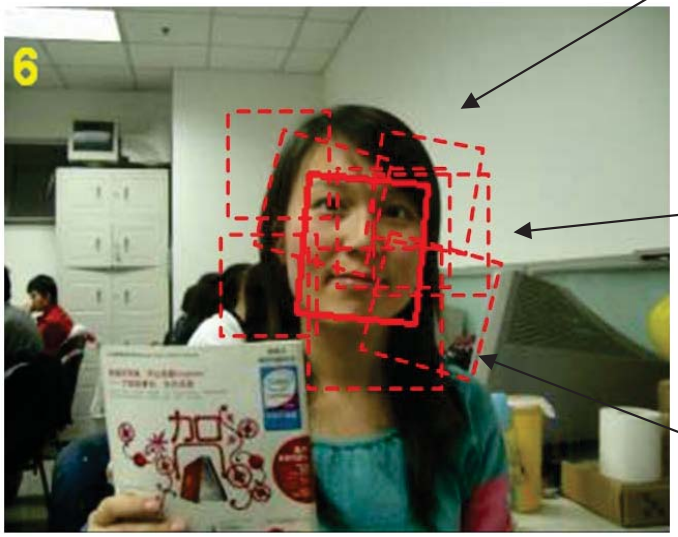
Background clutter

In/out of plane Rotation

Illumination change

Basic Framework:

For each new frame, Which candidate is the best ?(object)



Computing Similarity

Minimising reconstruction error

Classifier

Update scheme:

Online

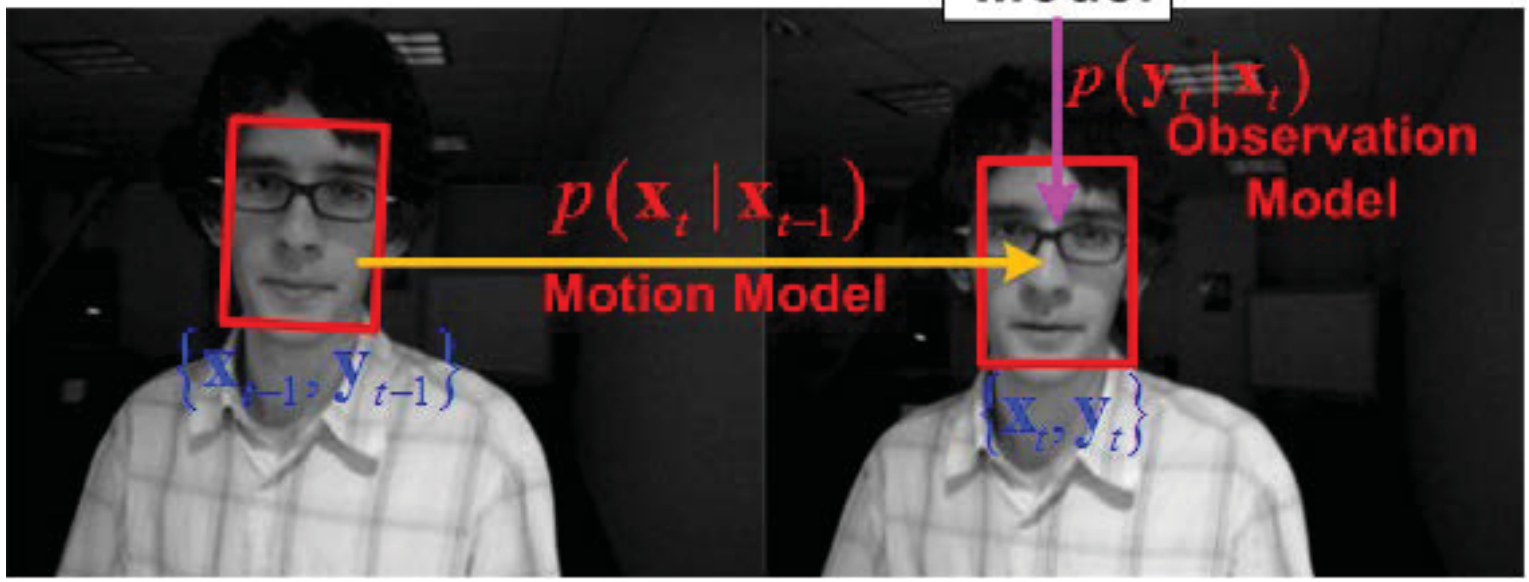
Batch mode method?

Online Method?

Two basic components: motion & appearance

- **At time t-1, how to predict next state of the target**
  - **A Dynamic /Motion Model: like Kalman filter and particle filter**
- **At time t, how to verify predictions using image observations**
  - **An Ap** $p(\mathbf{x}_t \mid Y_t) \propto p(\mathbf{y}_t \mid \mathbf{x}_t) \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \mid Y_{t-1}) d\mathbf{x}_{t-1}$ **ate an obser**

$$Y_t = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_t\}$$



Review

网络名称	时间	Precisions（OTB50）	Overlap(OTB50)	Speed
FCNT	ICCV15	0.856	0.599	3fps
HCFT	ICCV15	0.891	0.605	11fps
DeepSRDCF	ICCV15	0.849	0.641	<10fps(SRDCF)
MDNet	CVPR16	0.948	0.708	1fps
HDT	CVPR16	0.889	0.603	
STCT	CVPR16	0.852	0.640	2.5fps
SINT	CVPR16	0.851	0.635	
RPNT	CVPR16			3.8fps
RTT	CVPR16	0.827	0.588	3-4fps
SiamFC_5s	ECCV16	0.815	0.612	58fps
SiamFC_3s	ECCV16	0.809	0.608	86fps
C-COT	ECCV16	0.899	0.672	
GOTURN	ECCV16			100fps
CNT	TIP16	0.732	0.545	
DLT	NIPS13	0.587	0.436	15fps
CNN-SVM	ICML15	0.852	0.597	
SO-DLT	15			
ROLO	16			20/60fps
TCNN	16	0.937	0.682	1.5fps
SANet	16	0.928（100）	0.692（100）	1fps

注：表格中给出的结果是论文里公布的结果。排名不分先后 。

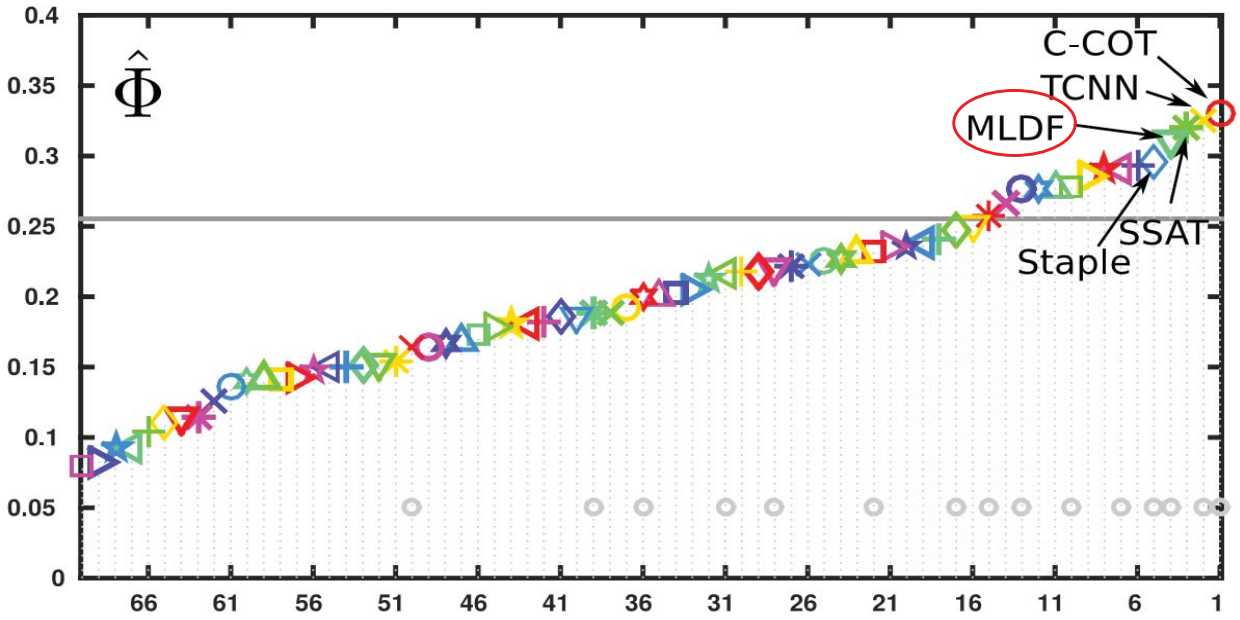
Performance

▲ The MDNet tracker is the winner of VOT 2015 competition.

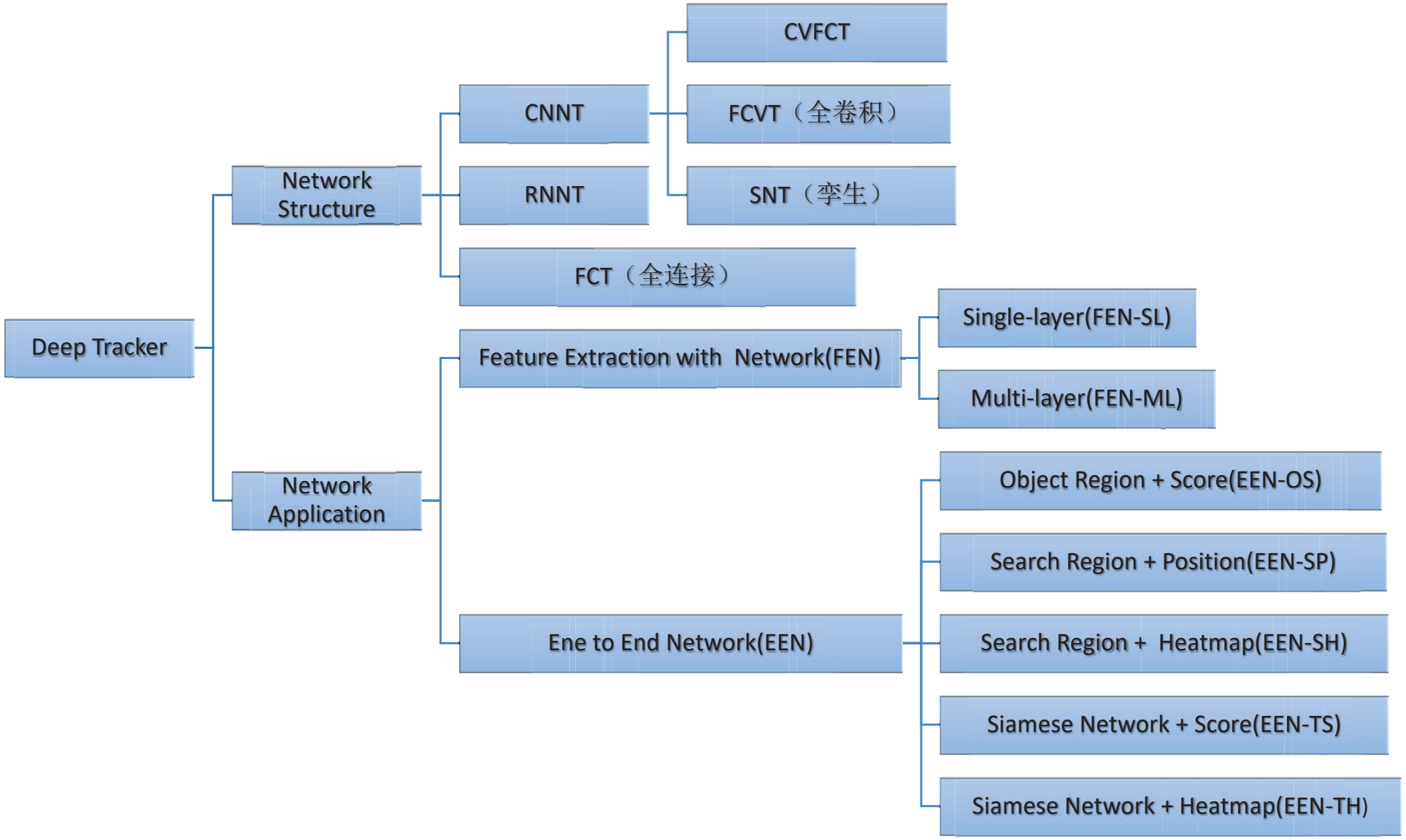
▲ All the top-4 trackers in the VOT2016 competition, including C-COT, TCNN, SSAT^1, MLDF^2, are based on deep neural

^1: SSAT is an extended version of MDNet.

^2: MLDF is designed based on STCT and FCNT.



Classification



Classification Based on Network Structure

▲ **Network Using CNNs:**
General CNNs, Fully Convolutional Networks, Siamese Networks

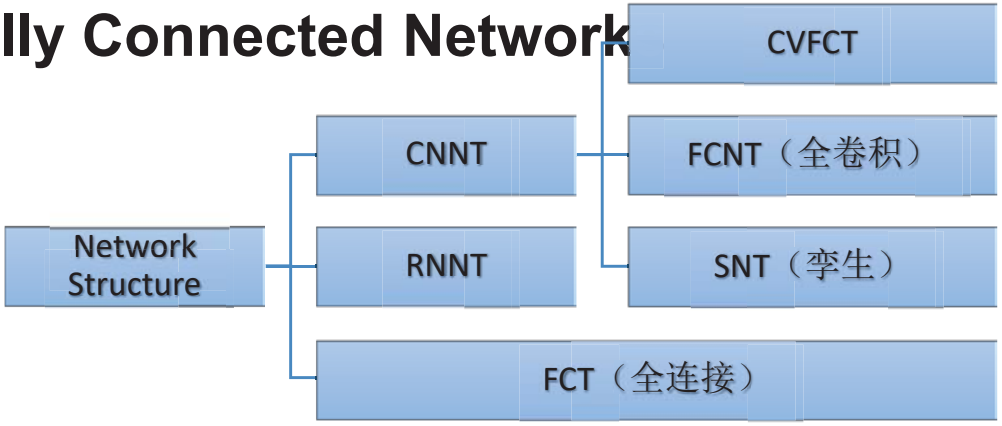
e.t. MDNet, SANet, STCT, FCNT, SINT, Siame_FC,…

▲ **Network Using RNNs**

e.t. RTT, SANet, ROLO,…

▲ **Network Using Fully Connected Network**

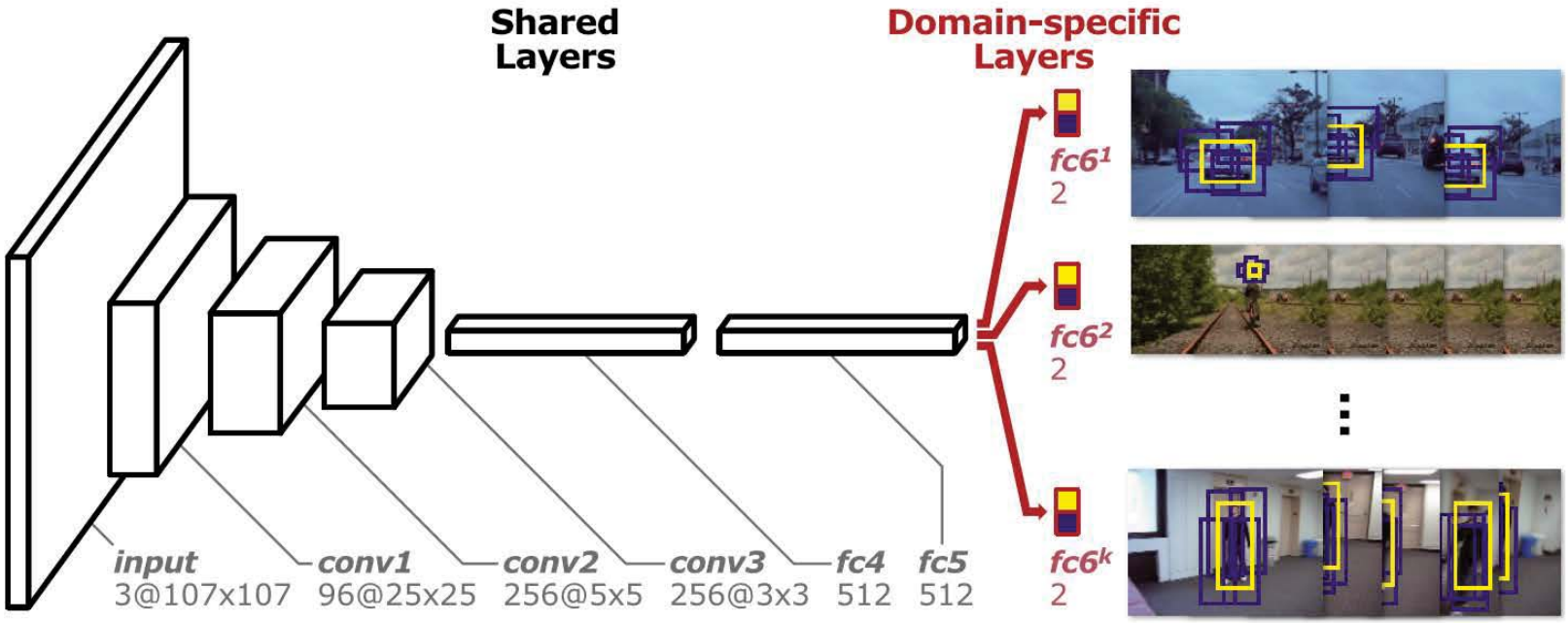
e.t. DLT,…



Network Using CNNs MDNet (CVPR16)

Learning Multi-Domain Convolutional Neural Networks for Visual Tracking

Hyeonseob Nam Bohyung Han
Dept. of Computer Science and Engineering, POSTECH, Korea
{namhs09, bhhan}@postech.ac.kr



Network Using RNNs SANet (16)

SANet: Structure-Aware Network for Visual Tracking

Heng Fan and Haibin Ling
Department of Computer and Information Sciences, Temple University, Philadelphia, USA

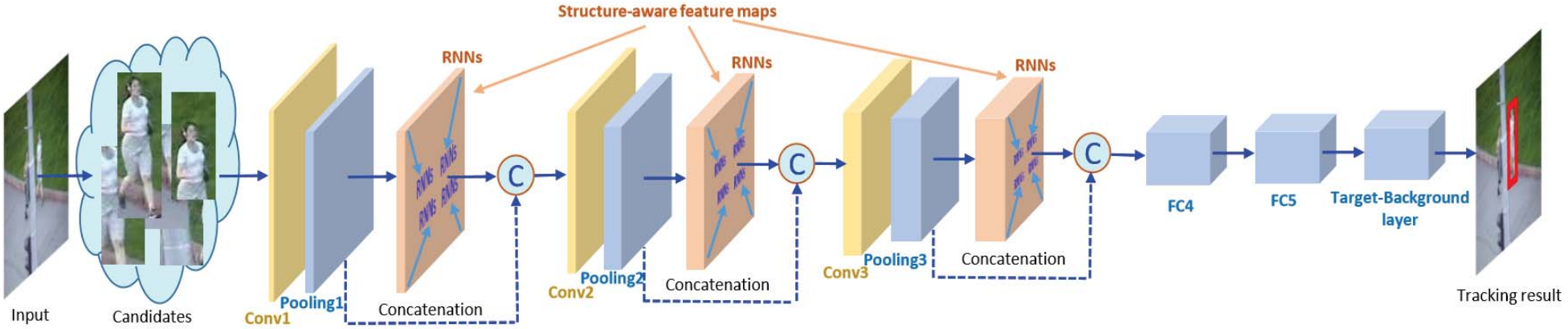
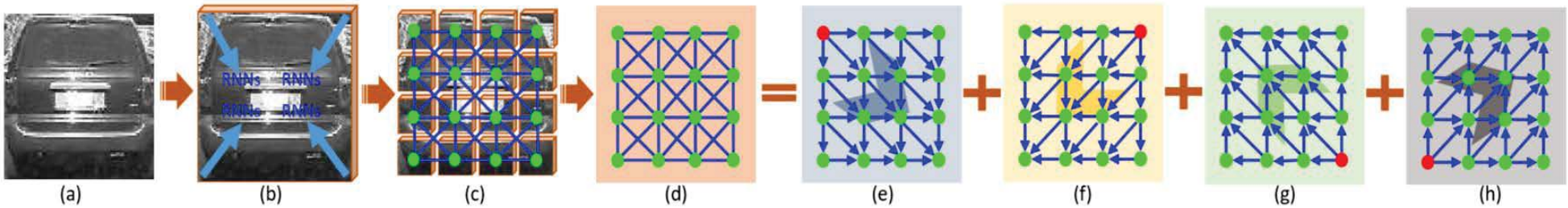


Figure 2. Illustration of the proposed SANet for visual tracking.



Network Using Fully Connected Networks

DLT(NIPS13)

Learning a Deep Compact Image Representation for Visual Tracking

Naiyan Wang **Dit-Yan Yeung**
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
winsty@gmail.com dyyeung@cse.ust.hk

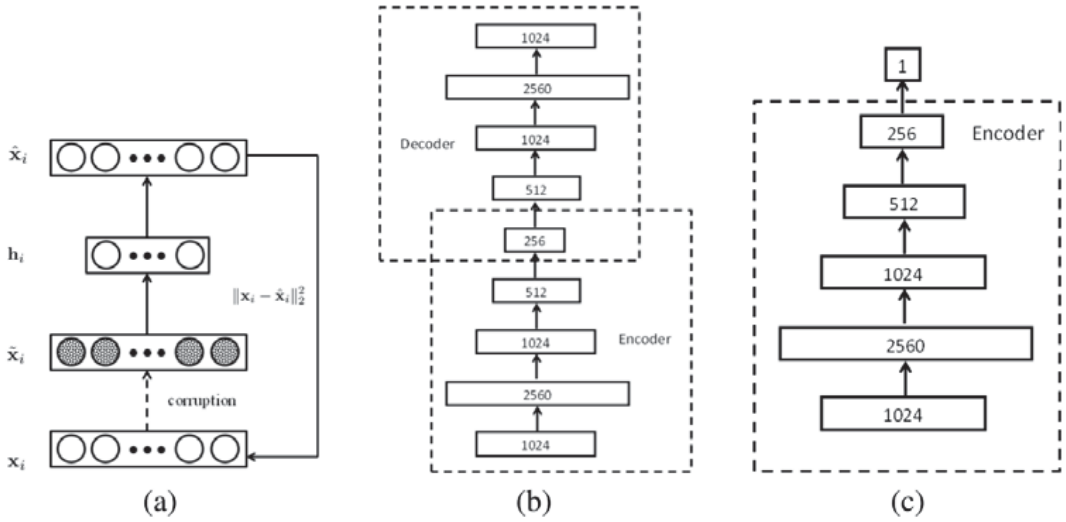
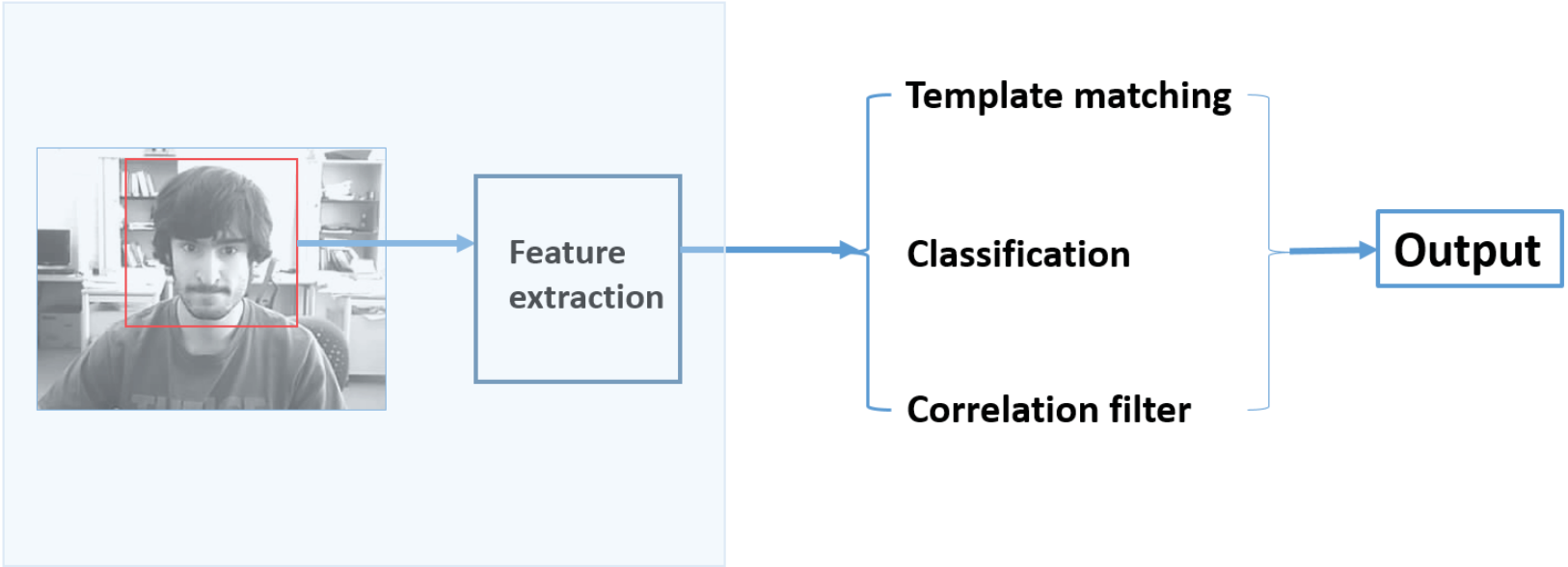


Figure 1: Some key components of the network architecture: (a) denoising autoencoder; (b) stacked denoising autoencoder; (c) network for online tracking.

Classification Based on Network Application

▲ Network for Feature Extraction :
single layer, multiple layers
▲ End to End network:
input object region, search region, …
output particle score, heatmap, position, bounding box



Network for Feature Extraction

▲ **Network for Feature Extraction**

single layer :

DeepSRDCF, CNT, DLT, RPNT, RTT, CNN-SVM,…

Pretrained Network (VGG, AlexNet,…)

multiple layers :

HCFT, HDT, C-COT, …

Mutiple layers usually perform better

HCFT （ICCV15）

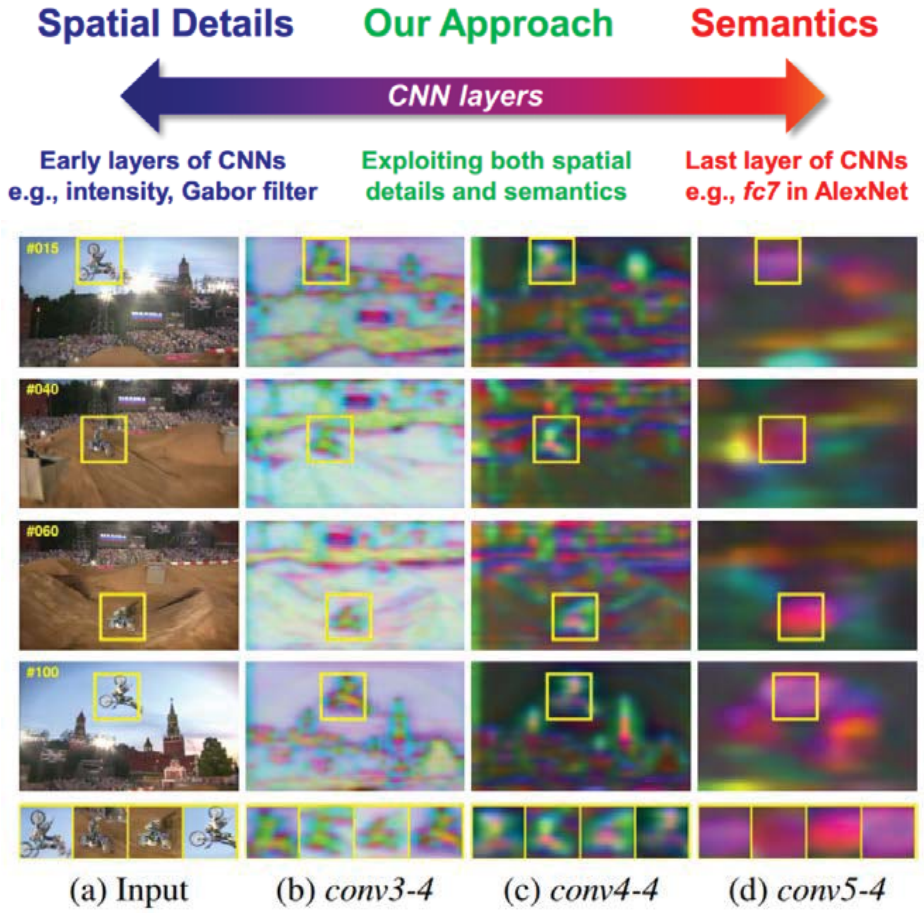
Hierarchical Convolutional Features for Visual Tracking

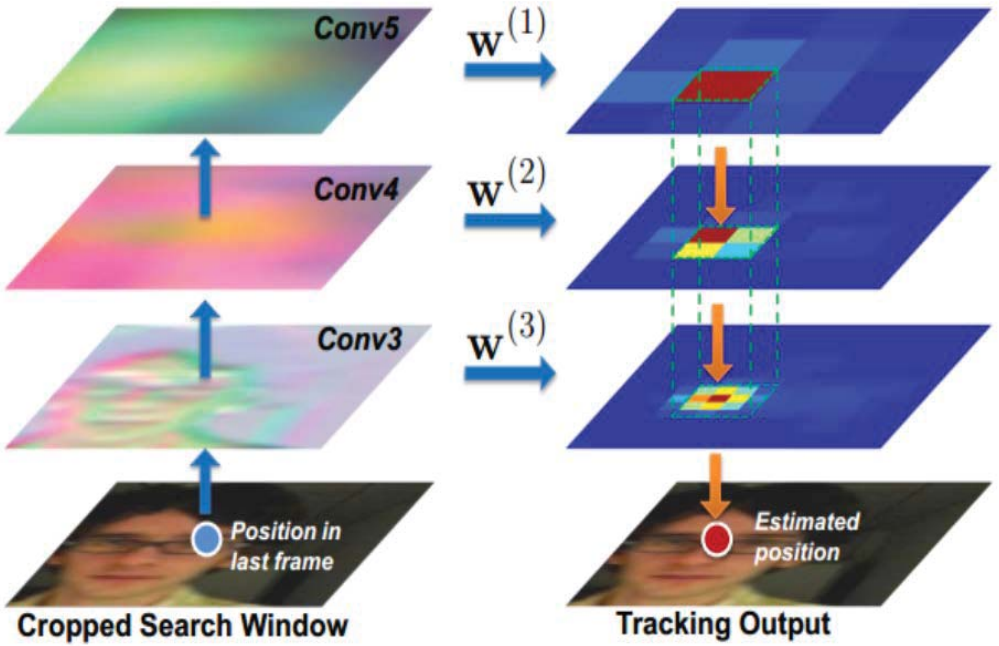
Chao Ma
SJTU
chaoma@sjtu.edu.cn

Jia-Bin Huang
UIUC
jbhuang1@illinois.edu

Xiaokang Yang
SJTU
xkyang@sjtu.edu.cn

Ming-Hsuan Yang
UC Merced
mhyang@ucmerced.edu





Deep Features + KCF

End to End Network

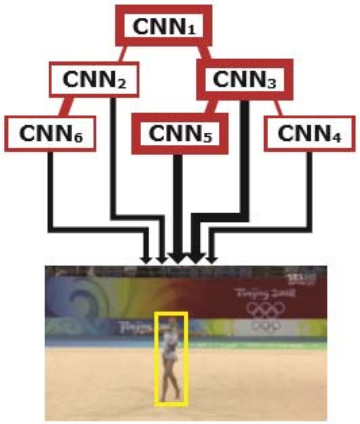
▲ **End to End network:**
input object region, search region, …
output particle score, heatmap, position, bounding box,…

	Input Region	Output Region
MDNet	Object region	Particle score
FCNT	Search region	Heatmap
STCT	Search region	Heatmap
SINT	Two object region	Particle score
GOTURN	Object region & Search region	Bounding box

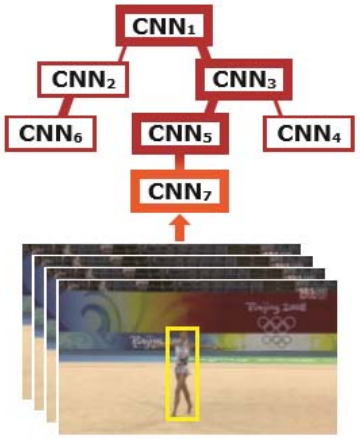
TCNN （VOT16）

Modeling and Propagating CNNs in a Tree Structure for Visual Tracking

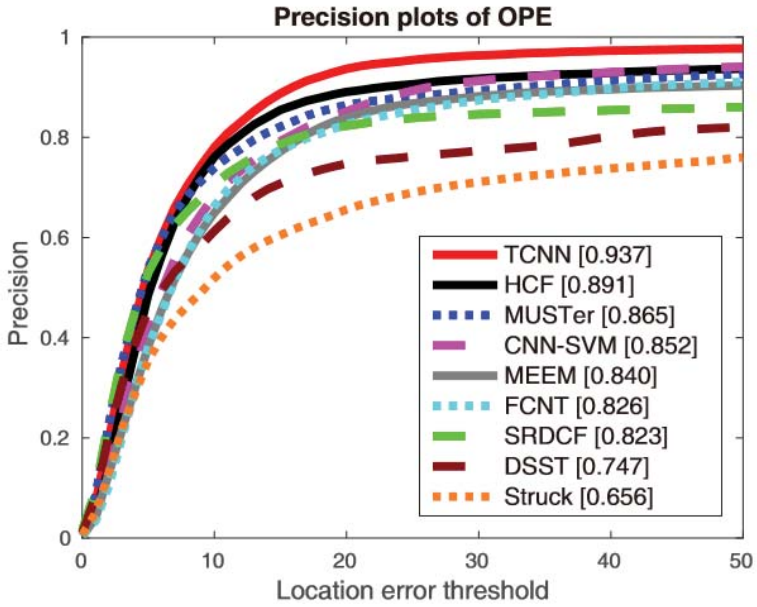
Hyeonseob Nam* Mooyeol Baek* Bohyung Han
Department of Computer Science and Engineering, POSTECH, Korea
{namhs09, mooyeol, bhhan}@postech.ac.kr



(a) State estimation



(b) Model update

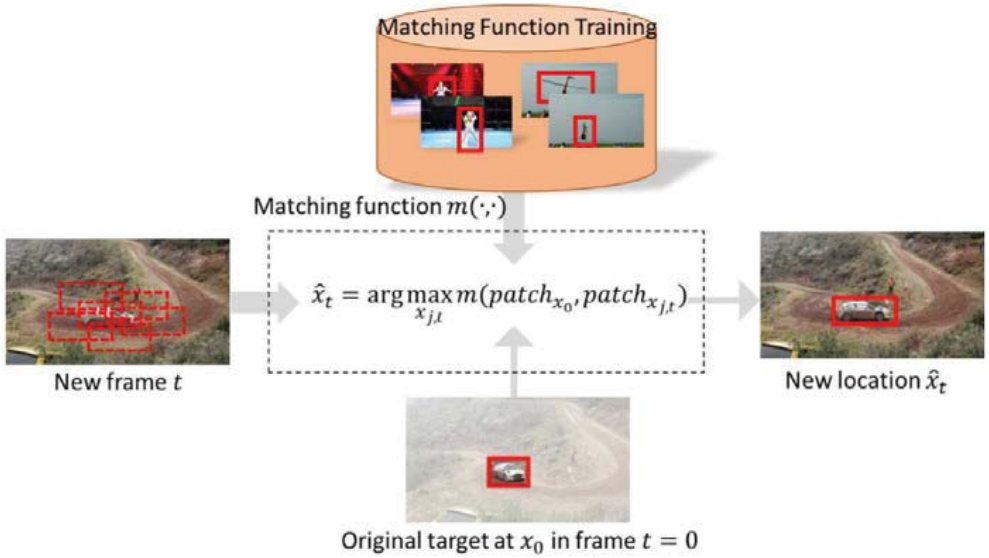


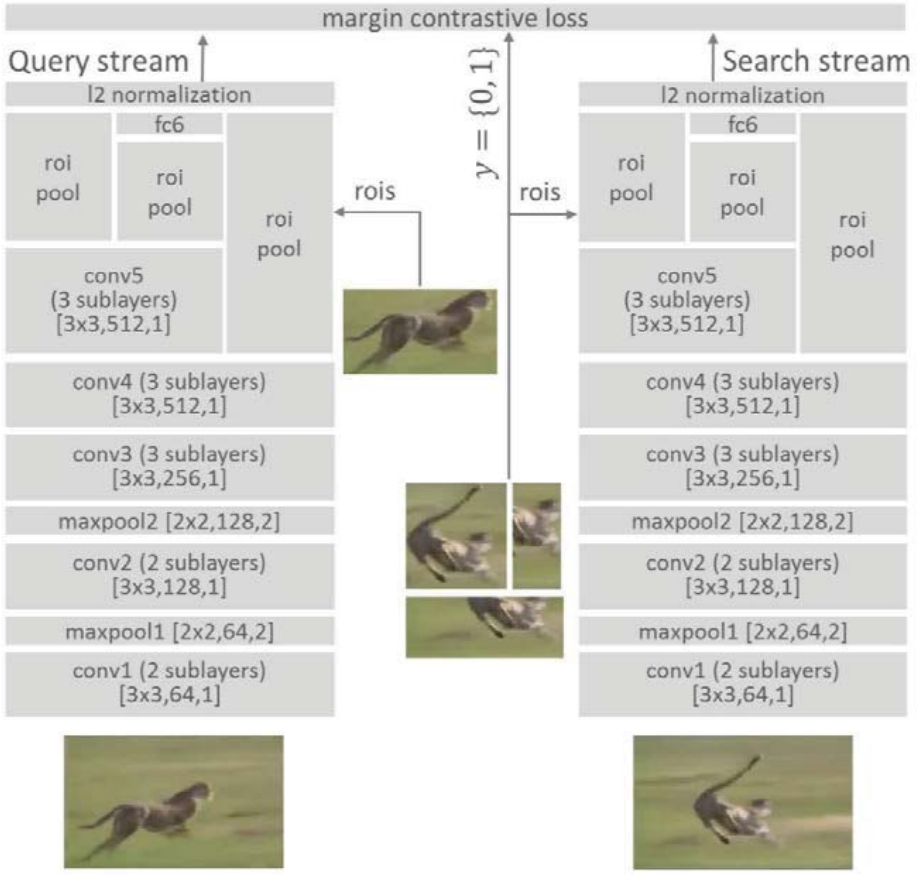
Results for OTB50 sequences

SINT （CVPR16）

Siamese Instance Search for Tracking

Ran Tao, Efstratios Gavves, Arnold W.M. Smeulders
QUVA Lab, Science Park 904, 1098 XH Amsterdam
{r.tao,egavves,a.w.m.smeulders}@uva.nl





Our Work

FCNT (ICCV15)

Visual Tracking with fully Convolutional Networks

Lijun Wang[1,2], Wanli Ouyang[2], Xiaogang Wang[2], and Huchuan Lu[1]
[1] Dalian University of Technology, Dalian, China
[2] The Chinese University of Hong Kong, Hong Kong, China

STCT (CVPR16)

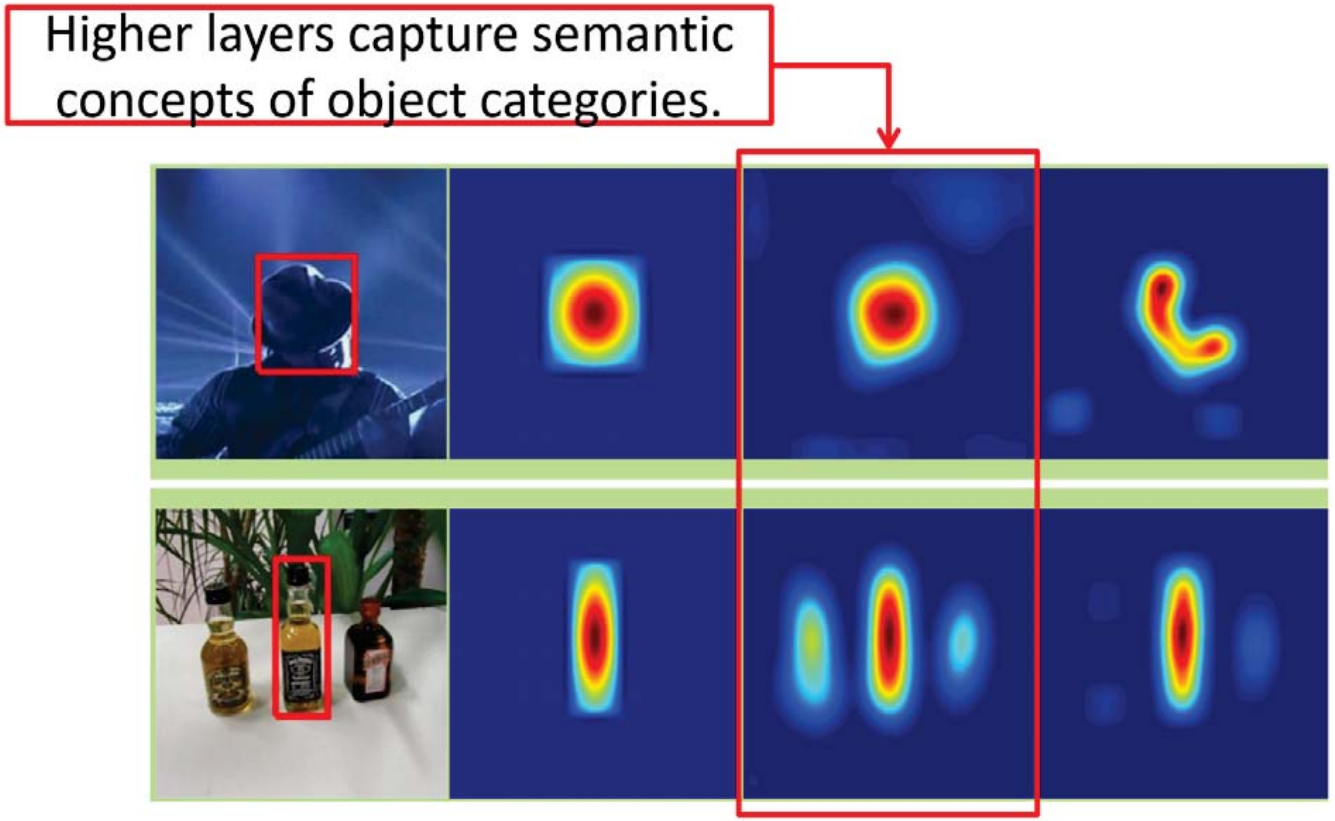
STCT: Sequentially Training Convolutional Networks for Visual Tracking

Lijun Wang[1,2], Wanli Ouyang[2], Xiaogang Wang[2], and Huchuan Lu[1]
[1]Dalian University of Technology, China
[2]The Chinese University of Hong Kong, Hong Kong, China

FCNT （ICCV15）

Motivations

- ***Observation 1: Different layer encode different types of features.***

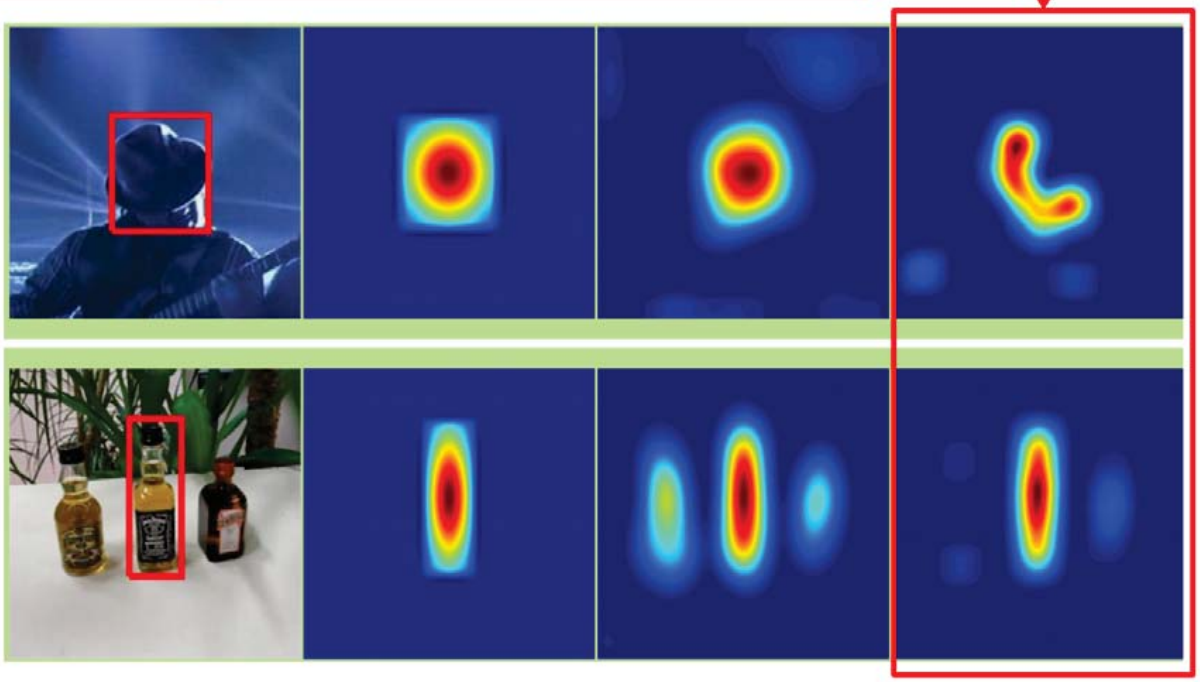


FCNT （ICCV15）

Motivations

- ***Observation 1: Different layer encode different types of features.***

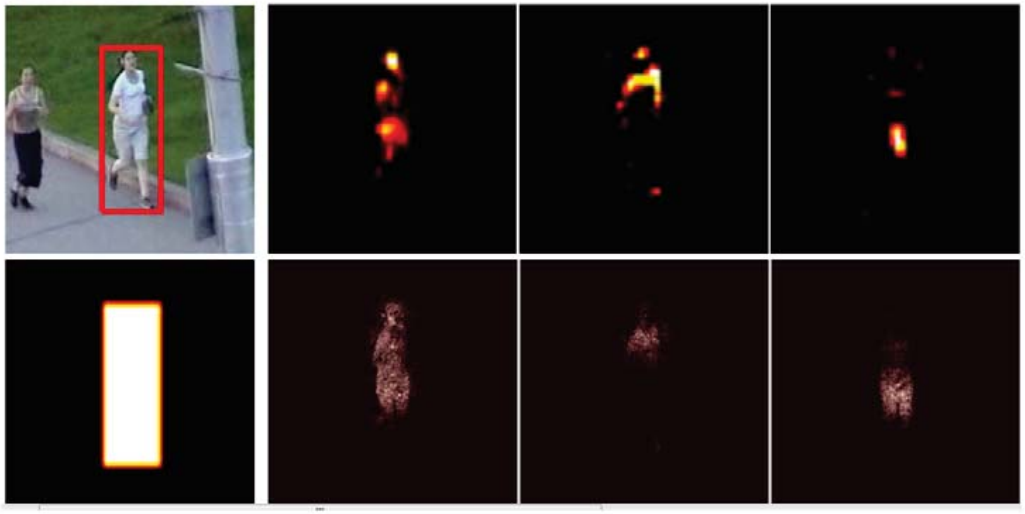
Lower layers encode more discriminative features to capture intra class variations



FCNT （ICCV15）

Motivations

- ***Observation 2: Although the receptive field of CNN feature maps is large, activated feature maps are sparse and localized. Activated regions are highly correlated to the regions of semantic objects***



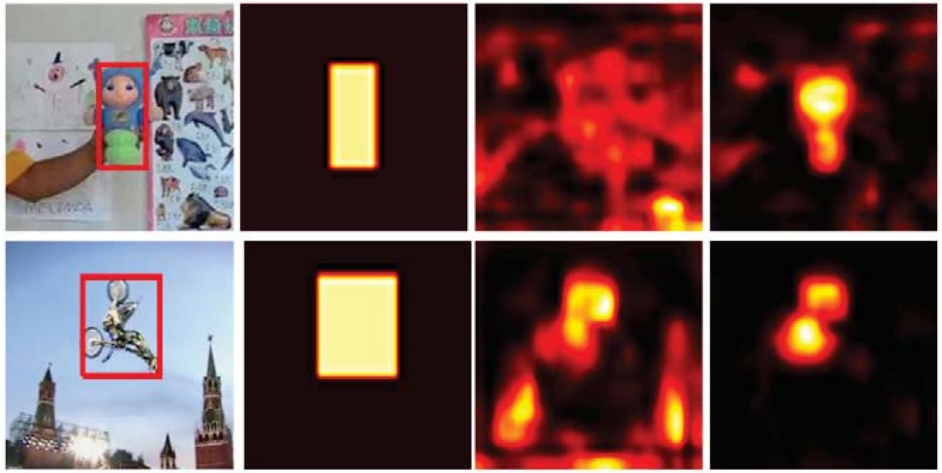
(a) (b)

(a) Input image (top) and ground truth foreground mask (bottom); (b) Activated feature maps (top) and corresponding saliency maps (bottom).

FCNT （ICCV15）

Motivations

- ***Observation 3: Many CNN feature maps are noisy or unrelated for the task of discriminating a particular target from background.***

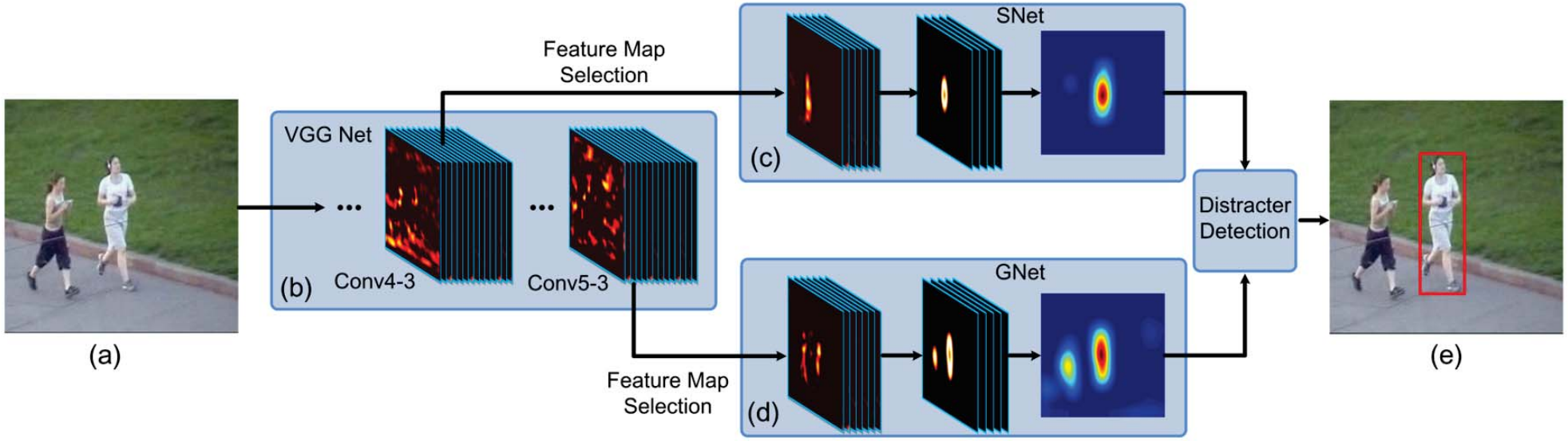


(a) (b) (c)

(a) Ground truth foreground mask; (b) average feature maps of convolutional layers; (c) average selected feature maps.

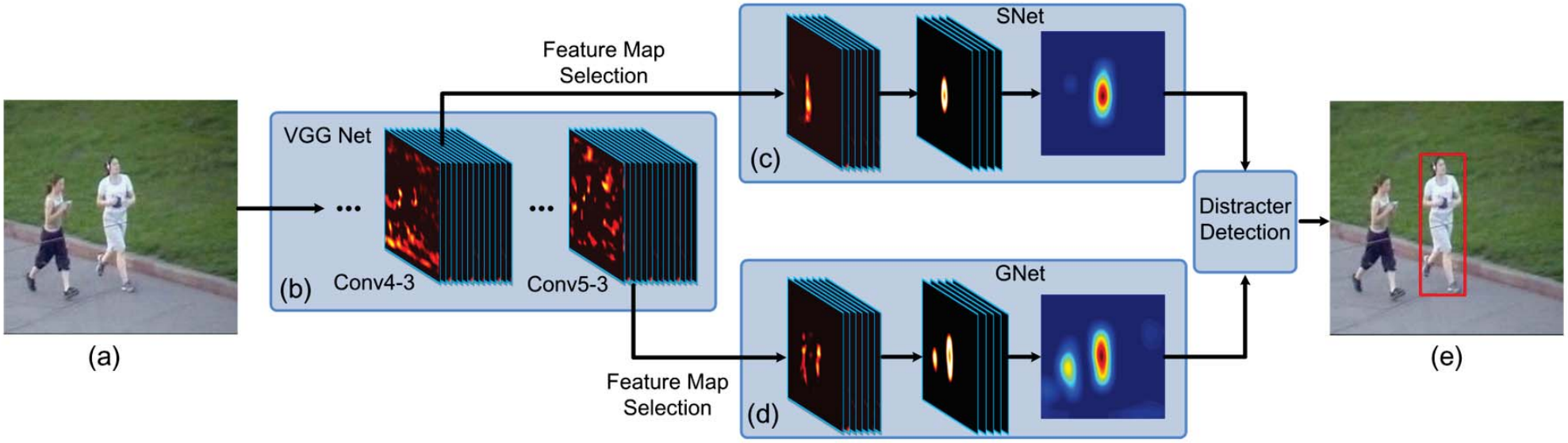
FCNT （ICCV15）

- For a new frame, a region of interest (a) centered at the last target location containing both target and background context is cropped and propagated through the fully convolutional network.



FCNT （ICCV15）

- GNet captures the category information of the target and is built on top layer of VGG.
- SNet discriminates the target from background with similar appearance and is built on the lower layer of VGG.



FCNT （ICCV15）

Selection of Feature Maps

- The change of the loss function L_{sel} caused by the perturbation of the feature maps $\delta\mathbf{F}$:

$$\delta L_{sel} = \sum_i g_i \delta f_i + \frac{1}{2}\sum_i h_{ii}\left(\delta f_i\right)^2 + \frac{1}{2}\sum_{i\neq j} h_{ij}\delta f_i \delta f_j$$

- The significance of the element f_i is defined as the change of the objective function after setting f_i to zero:

$$s_i = -g_i f_i + \frac{1}{2} h_{ii} f_i^2$$

FCNT （ICCV15）

Selection of Feature Maps

- The significance value of the $k-th$ feature map is defined as

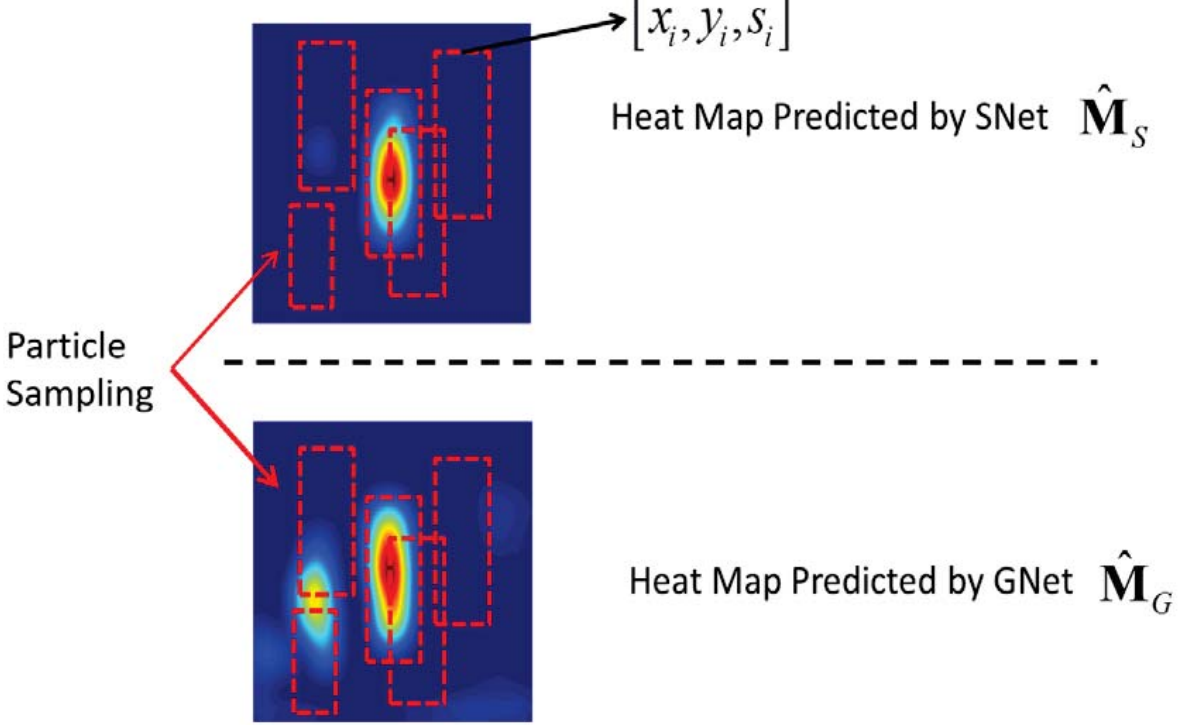
$$S_k = \sum_{x,y} s(\mathrm{x}, \mathrm{y}, \mathrm{k})$$

- The significance value measures the impact of the feature map on the objective function.
- 384/512 for best performance
- 64/512 for state-of-the-art performance

FCNT （ICCV15）

Target Localization

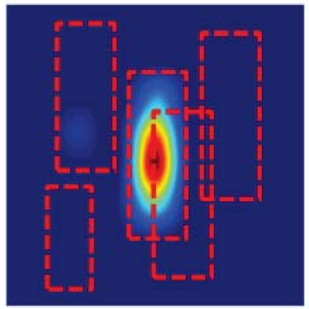
- Target candidates are sampled according to a Gaussian distribution $p\left(X^t \mid \hat{X}^{t-1}\right) = N\left(X^t; \hat{X}^{t-1}, \Sigma\right)$



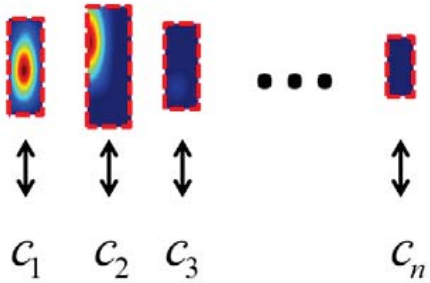
FCNT （ICCV15）

Target Localization

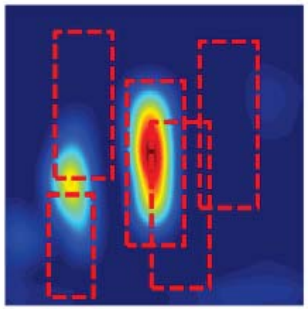
- The confidence of the a candidate is the summation of all the heat map values within the candidate region.



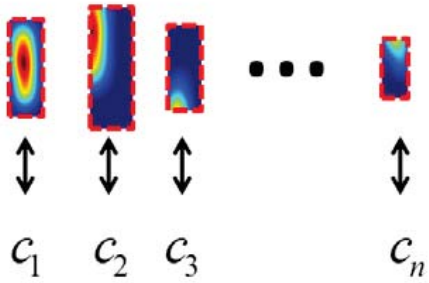
$\hat{\mathbf{M}}_S$



$$c_i = \sum_{j \in \mathbf{R}_i} \hat{\mathbf{M}}_S(j)$$



$\hat{\mathbf{M}}_G$

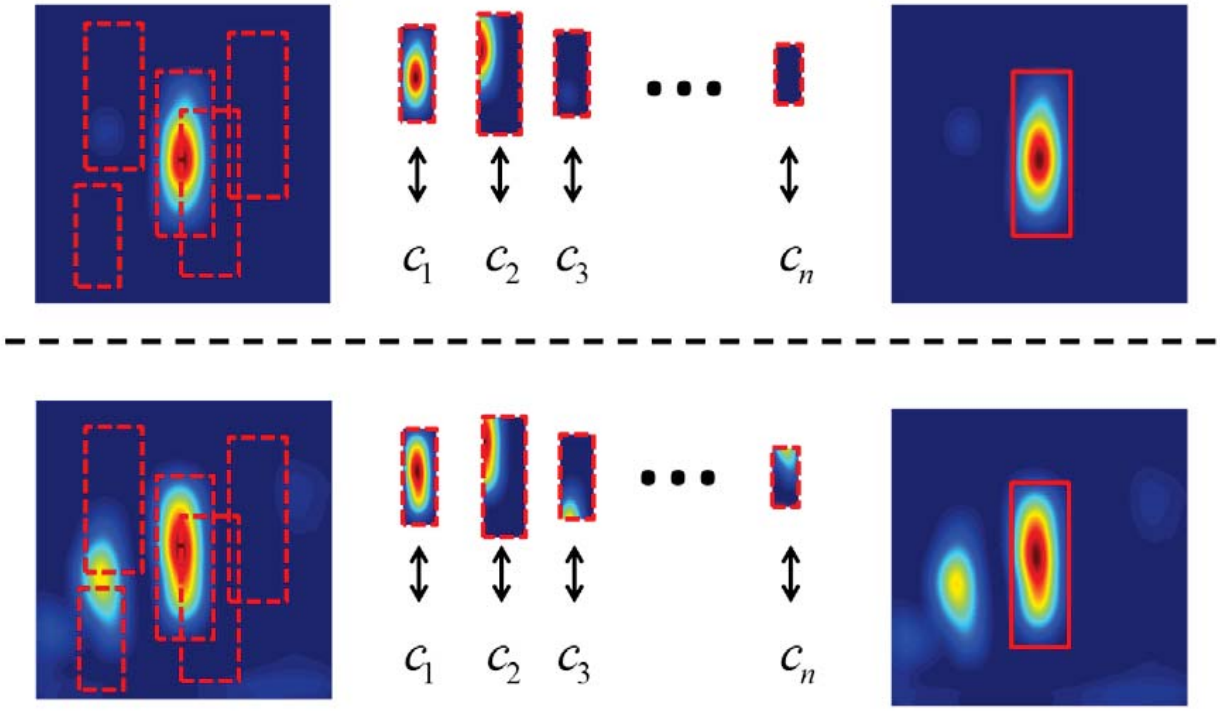


$$c_i = \sum_{j \in \mathbf{R}_i} \hat{\mathbf{M}}_G(j)$$

FCNT （ICCV15）

Target Localization

- The candidate with the highest confidence is selected as the target location.



FCNT （ICCV15）

Online Update

- To avoid the background noise, GNet is fixed after initialization.
- SNet is updated following two update rules:
  - Adaptation rule: adapt SNet to target appearance variation.
  - Discrimination rule: improve the discriminative power for foreground and background.

FCNT （ICCV15）

Online Update

- Adaptation rule: fine-tune SNet every 20 frames using the most confident tracking result.

$$\min \beta \left\| \mathbf{W}_S \right\|_F^2 + \sum_{x,y} \left\{ \left[\hat{\mathbf{M}}_S^t \left(x,y \right) - \mathbf{M}^t \left(x,y \right) \right]^2 \right\}$$

FCNT （ICCV15）

Online Update

- Discrimination rule: fine-tune SNet when distracter detected

$$\min \beta \left\| \mathbf{W}_S \right\|_F^2 + \sum_{x,y} \left\{ \left[\hat{\mathbf{M}}_S^1(x,y) - \mathbf{M}^1(x,y) \right]^2 + \left[1 - \mathbf{\Phi}^t(x,y) \right] \left[\hat{\mathbf{M}}_S^t(x,y) - \mathbf{M}^t(x,y) \right]^2 \right\}$$

The first frame provides target appearance information.

FCNT （ICCV15）

Online Update

- Discrimination rule: fine-tune SNet when distracter detected

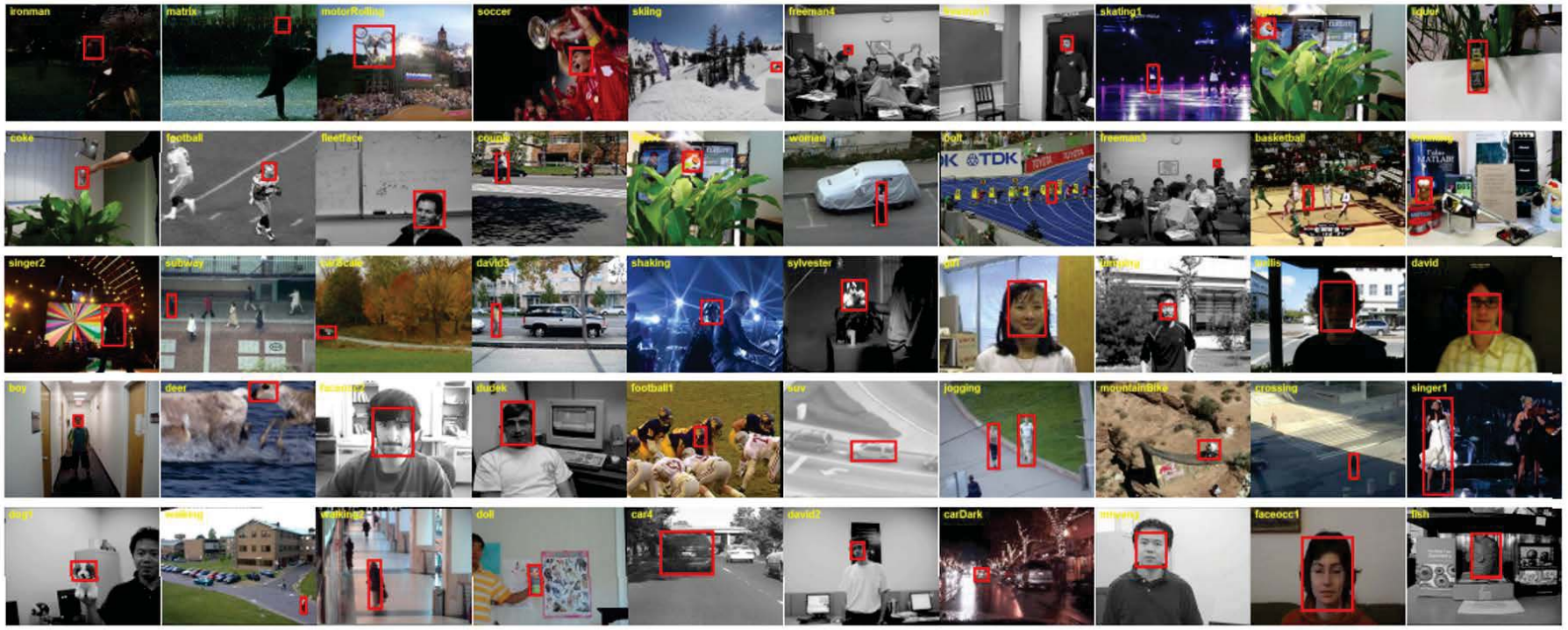
$$\min \beta \left\| \mathbf{W}_S \right\|_F^2 + \sum_{x,y} \left\{ \left[\hat{\mathbf{M}}_S^1(x,y) - \mathbf{M}^1(x,y) \right]^2 + \left[1 - \mathbf{\Phi}^t(x,y) \right] \left[\hat{\mathbf{M}}_S^t(x,y) - \mathbf{M}^t(x,y) \right]^2 \right\}$$

The background region in current frame is used to suppress distractors.

FCNT （ICCV15）

Experimental Results

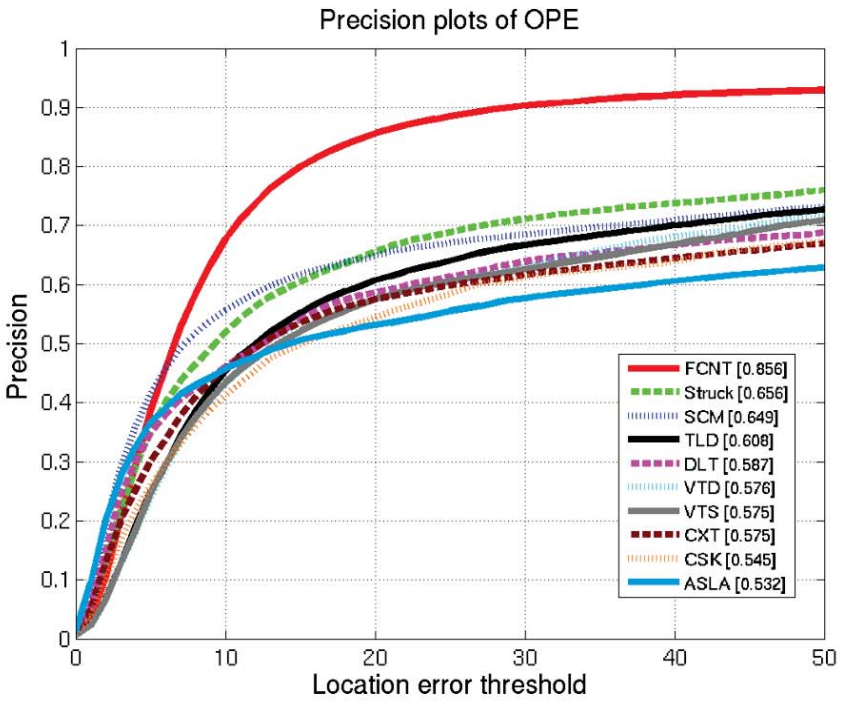
- 50 sequences with 11 challenging attributes

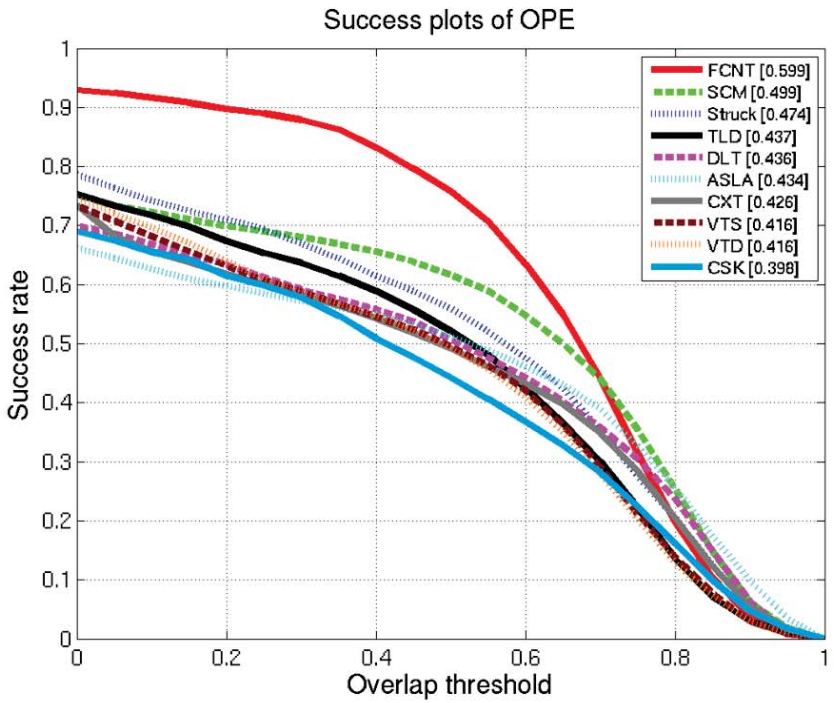


FCNT （ICCV15）

Experimental Results

- Comparison against state-of-the-arts





FCNT （ICCV15）

Experimental Results

- Results on 11 attributes

	SCM	Struck	TLD	DLT	ASLA	CXT	MEEM	KCF	TGPR	FCNT
IV	0.594	0.558	0.537	0.534	0.517	0.501	0.778	0.728	0.687	**0.830**
OPR	0.618	0.597	0.596	0.561	0.518	0.574	**0.838**	0.729	0.741	**0.831**
SV	0.672	0.639	0.606	0.590	0.552	0.550	0.787	0.679	0.703	**0.830**
OCC	0.640	0.564	0.563	0.574	0.460	0.491	**0.801**	0.749	0.708	**0.797**
DEF	0.586	0.521	0.512	0.563	0.445	0.422	0.859	0.740	0.768	**0.917**
MB	0.339	0.551	0.518	0.453	0.278	0.509	0.740	0.650	0.578	**0.789**
FM	0.333	0.604	0.551	0.446	0.253	0.515	**0.757**	0.602	0.575	**0.767**
IPR	0.597	0.617	0.584	0.548	0.511	0.610	0.790	0.725	0.705	**0.811**
OV	0.429	0.539	0.576	0.444	0.333	0.510	**0.730**	0.650	0.576	**0.741**
BC	0.578	0.585	0.428	0.495	0.496	0.443	**0.807**	0.753	0.761	**0.799**
LR	0.305	0.545	0.349	0.396	0.156	0.371	0.494	0.381	0.539	**0.765**
Overall	0.649	0.656	0.608	0.587	0.532	0.575	**0.828**	0.740	0.766	**0.856**

STCT (CVPR16)

STCT: Sequentially Training Convolutional Networks for Visual Tracking

Lijun Wang[1,2], Wanli Ouyang[2], Xiaogang Wang[2], and Huchuan Lu[1]
[1]Dalian University of Technology, China
[2]The Chinese University of Hong Kong, Hong Kong, China

Motivation

- **Problems**
  - The limited amount of training samples is prone to overfitting.
  - Features learned from conventional method for training CNNs is highly correlated to each other.

- **Sequentially training CNN as learning ensembles of base learners:**
  - Each base learner is trained using different loss criterions to reduce feature correlation and avoid over-training.

Motivation

- **Problems**
  - Significant target appearance may changes caused by abrupt occlusion.

- **Convolution with random binary mask:**
  - Enforce the learned convolution kernels to focus on different part of the input feature map
  - Further reduce the correlation between the learned features and prevent over-training.

CNN Training as Ensemble Learning

- Sequential Sampling for Ensemble Learning
  - Friedman and Popescu formulate the prediction function as :

$$F(\boldsymbol{x}) \simeq a_0 + \sum_{m=1}^{M} a_m f(\boldsymbol{x}; \boldsymbol{\gamma}_m)$$

  - For a base learner $f(x;\ \gamma_{\mathrm{m}})$,its irrelevance to the current problem is defined as:

$$Q(\boldsymbol{\gamma}) = \min_{\alpha_0, \alpha} \frac{1}{N} \sum_{i=1}^{N} L(y_i, \alpha_0 + \alpha f(\boldsymbol{x}; \boldsymbol{\gamma}))$$

CNN Training as Ensemble Learning

- Sequential Sampling for Ensemble Learning
  - The optimal single point can be obtained by minimizing the irrelevance :

$$\gamma^* = \arg\min Q(\gamma)$$

  - For an ensemble of base learners $\{f(x;\ \gamma_{\mathrm{m}})\}_1^M$,the characteristic scale can be employed to measure its quality:

$$\sigma = \frac{1}{M}\sum_{m=1}^{M}[Q(\gamma_m) - Q(\gamma^*)]$$

A small value of σ imply that many base learners in the ensemble are vary similar to the optimal base learner $f(x;\ \gamma^*)$. So they are highly correlated with each other.

If σ is too large ,most of the base learners are irrelevant to the problem.

CNN Training as Ensemble Learning

- Sequential Sampling for Ensemble Learning

  The irrelevance measure of each successive point γ_m depends on the previous sampled points $\{\gamma_l\}_1^{m-1}$ is the ensemble learning.

$$Q_m(\boldsymbol{\gamma}|\{\boldsymbol{\gamma}_l\}_1^{m-1}) = \min_{\alpha_0,\alpha_m} \frac{1}{N}\sum_{i=1}^{N} L\Big(y_i, \alpha_0 + \alpha_m f(\boldsymbol{x}_i;\boldsymbol{\gamma}) + \eta\sum_{l=1}^{m-1}\alpha_l f(\boldsymbol{x}_i;\boldsymbol{\gamma}_l)\Big),$$

- Then each sequentially selected parameter point γ_m is determined by

$$\boldsymbol{\gamma}_m = \arg\min_{\boldsymbol{\gamma}\in\Gamma} Q_m(\boldsymbol{\gamma}|\{\boldsymbol{\gamma}_l\}_1^{m-1})$$

CNN Training as Ensemble Learning

- Online Training CNNs as Sequential Ensemble learning
  - The feature map in the second layer is obtained by convolving the kernel with the feature map in the first layer as

$$\boldsymbol{F}_2^c(\boldsymbol{X}) = \sum_{k=1}^{C_1} \boldsymbol{w}_k^c * \boldsymbol{F}_1^k(\boldsymbol{X}) + b_c$$

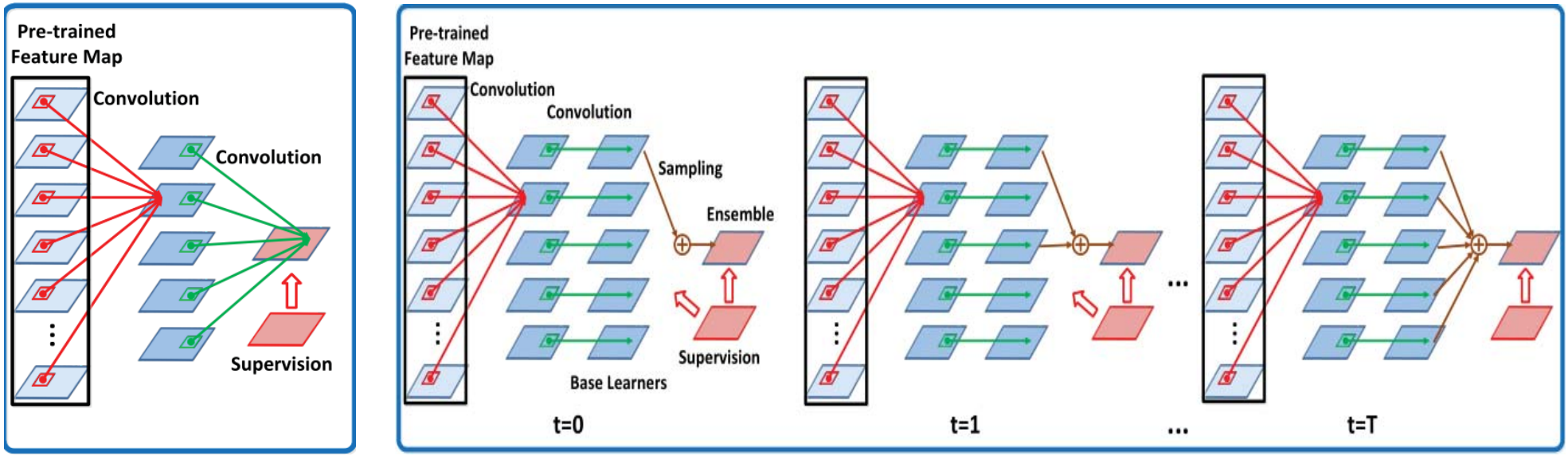
  - We regard the feature map as an ensemble of base learners

$$\boldsymbol{F}_2^c(\boldsymbol{X}) = \sum_{k=1}^{C_1} \boldsymbol{f}(\boldsymbol{X}; \boldsymbol{\gamma}_k^c)$$

  - The base learner is defined as

$$\boldsymbol{f}(\boldsymbol{X}; \boldsymbol{\gamma}_k^c) = \boldsymbol{w}_k^c * \boldsymbol{F}_1^k(\boldsymbol{X}) + b_c^k$$

CNN Training as Ensemble Learning



(a) (b)

(a)Conventional method for training a two-layer CNN online to transfer pre-trained deep feature.
(b)The proposed method trains the CNN method trains the CNN model via sequentially sampling optimal base learners into an ensemble.

CNN Training as Ensemble Learning

- Online Training CNNs as Sequential Ensemble learning
  - Regard a CNN as an ensemble with channels as base learners.
  - Each base learner is trained using a different loss criterion.
  - All the base learners are sequentially sampled into the ensemble

(a) CNN

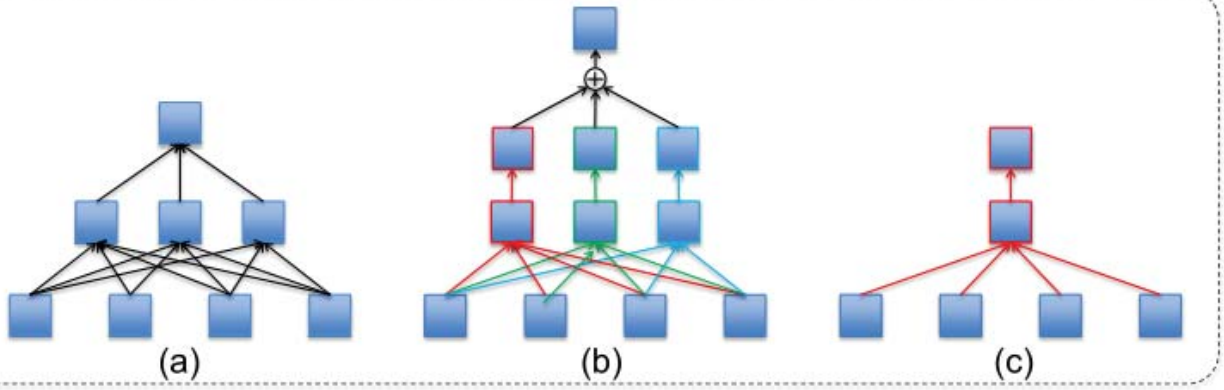
$$F_2^c(X) = \sum_{k=1}^{C_1} w_k^c * F_1^k(X) + b_c$$

(b) CNN as Ensemble

$$F_2^c(X) = \sum_{k=1}^{C_1} f_k(X)$$

(c) Base Learner

$$f_k(X) = w_k^c * F_1^k(X) + b_c^k$$



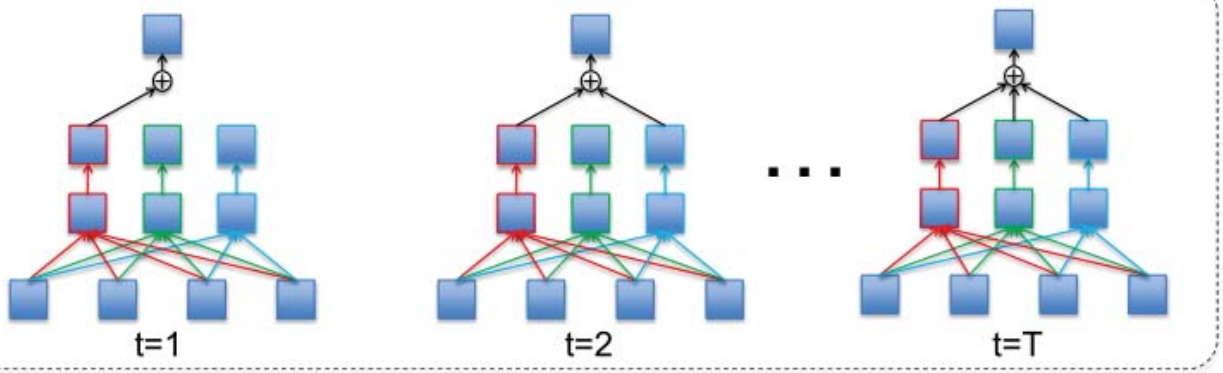
Sequential Training

(a) Ensemble:

$$L_\varepsilon = L(Y_t, F(X_t; \varepsilon))$$

(b) Candidate BL:

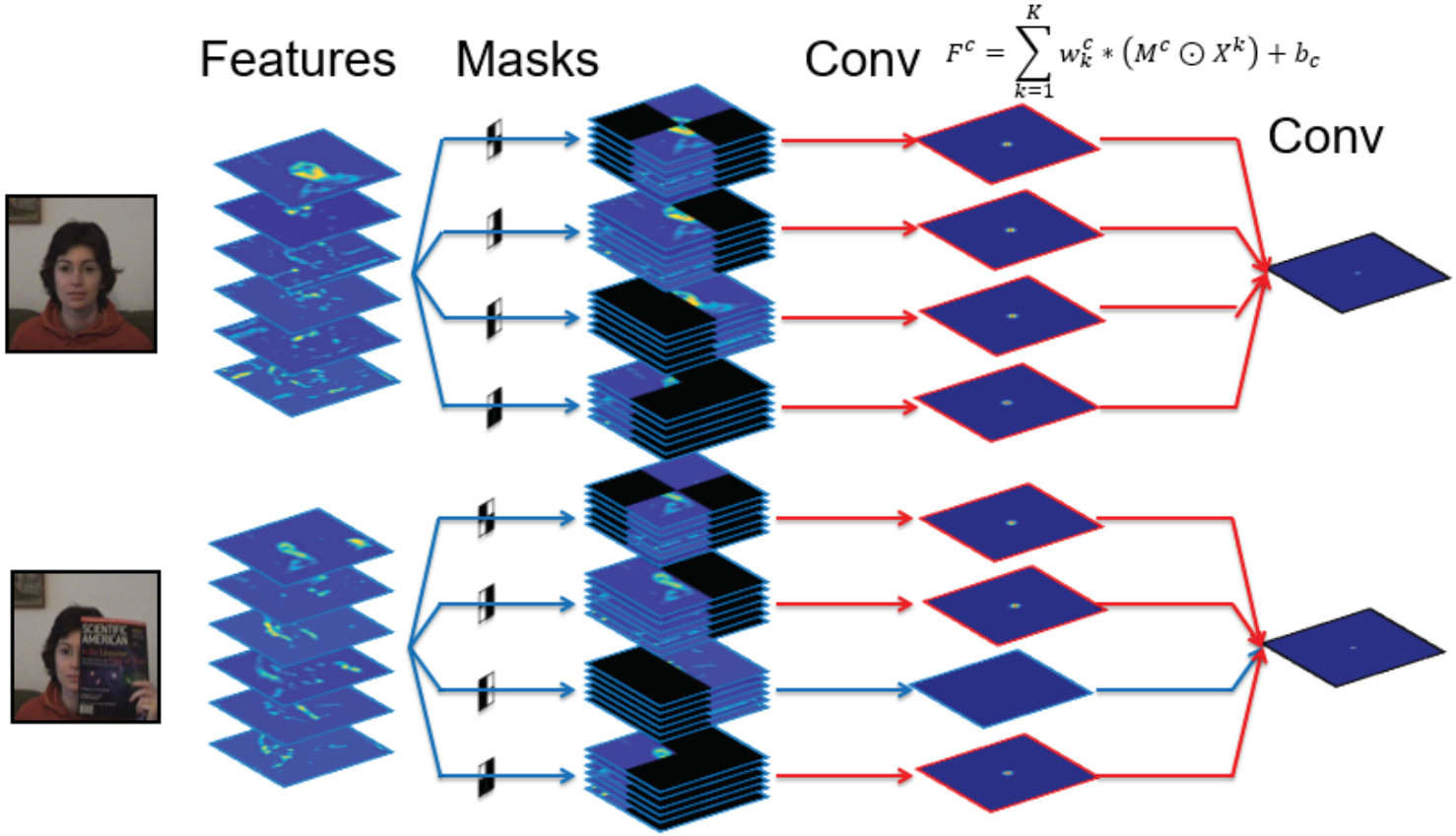
$$L_c\left(Y_t, f(X_t; \gamma_j)\right) = L\left(Y_t, f(X_t; \gamma_j) + \eta F(X_t; \varepsilon)\right)$$



CNN Training as Ensemble Learning

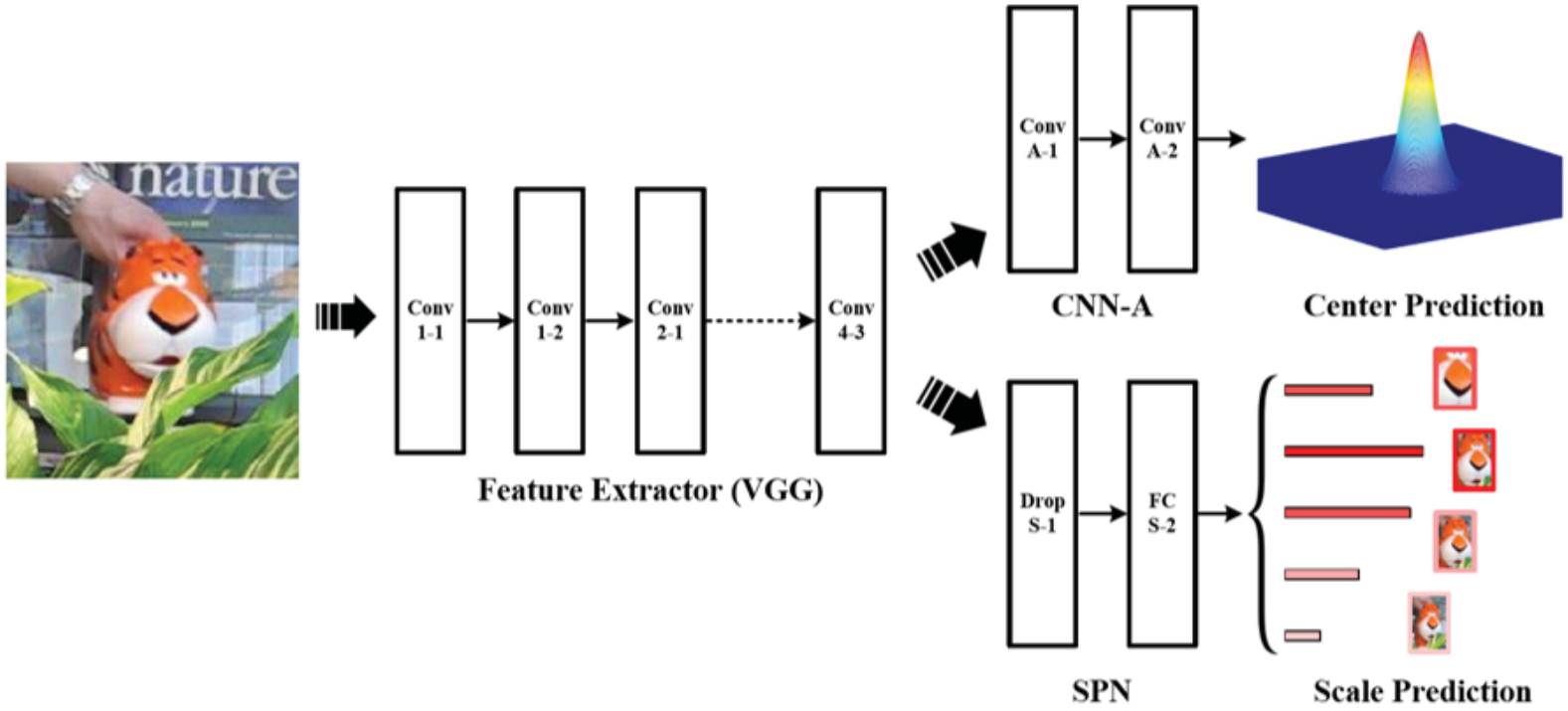
- Convolutional with Mask Layer

  We divide each mask into a grid of 2*2 blocks . All the values within each block are initialized by one random variable which is drawn from a Bernoulli distribution.



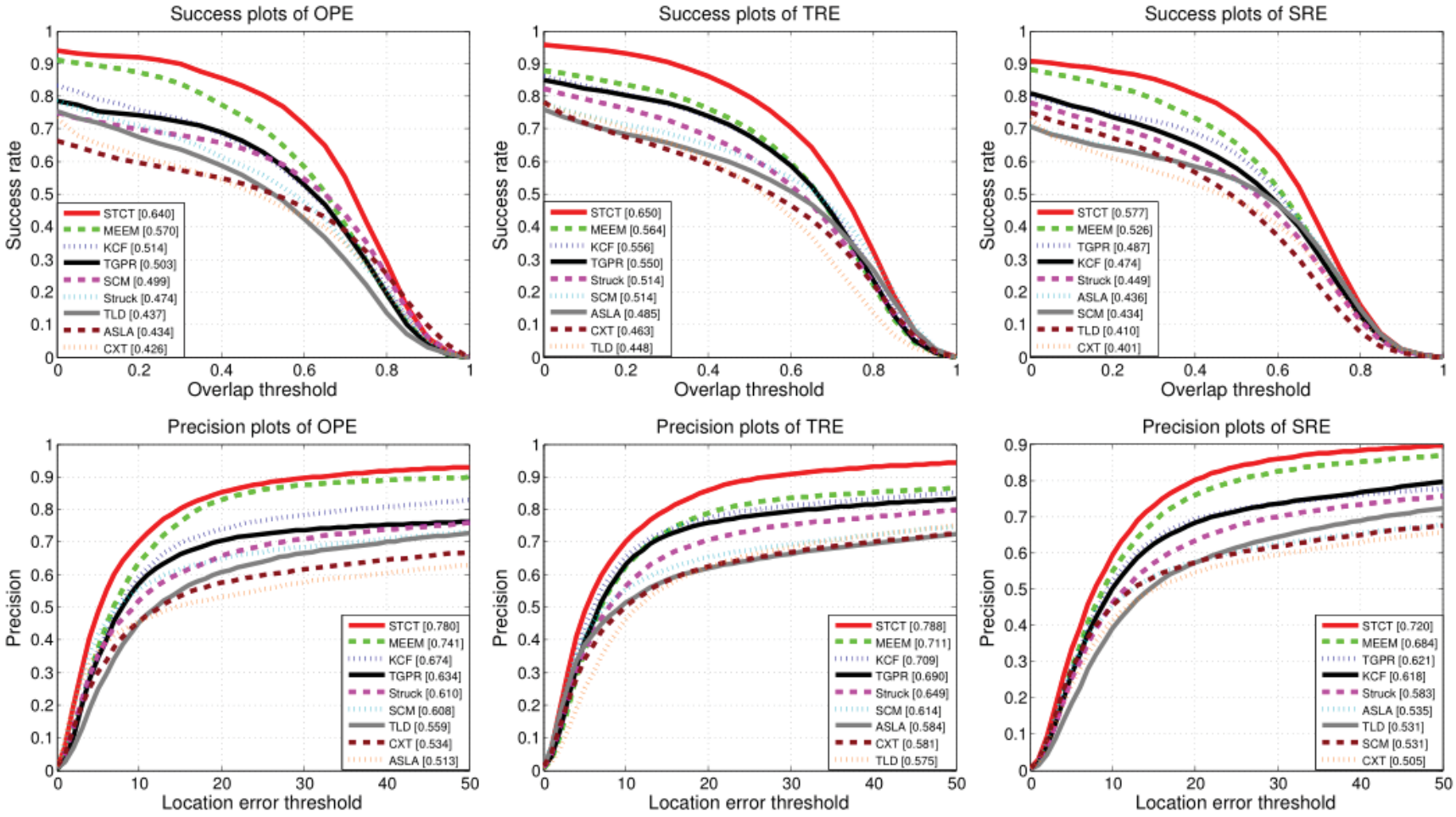
Tracking Algorithm

- Overview
  - Employs pre-trained VGG as the feature extractor.
  - Sequentially train CNN-A network to predict target center.
  - Learning SPN network to handle scale variations.



Experimental Results

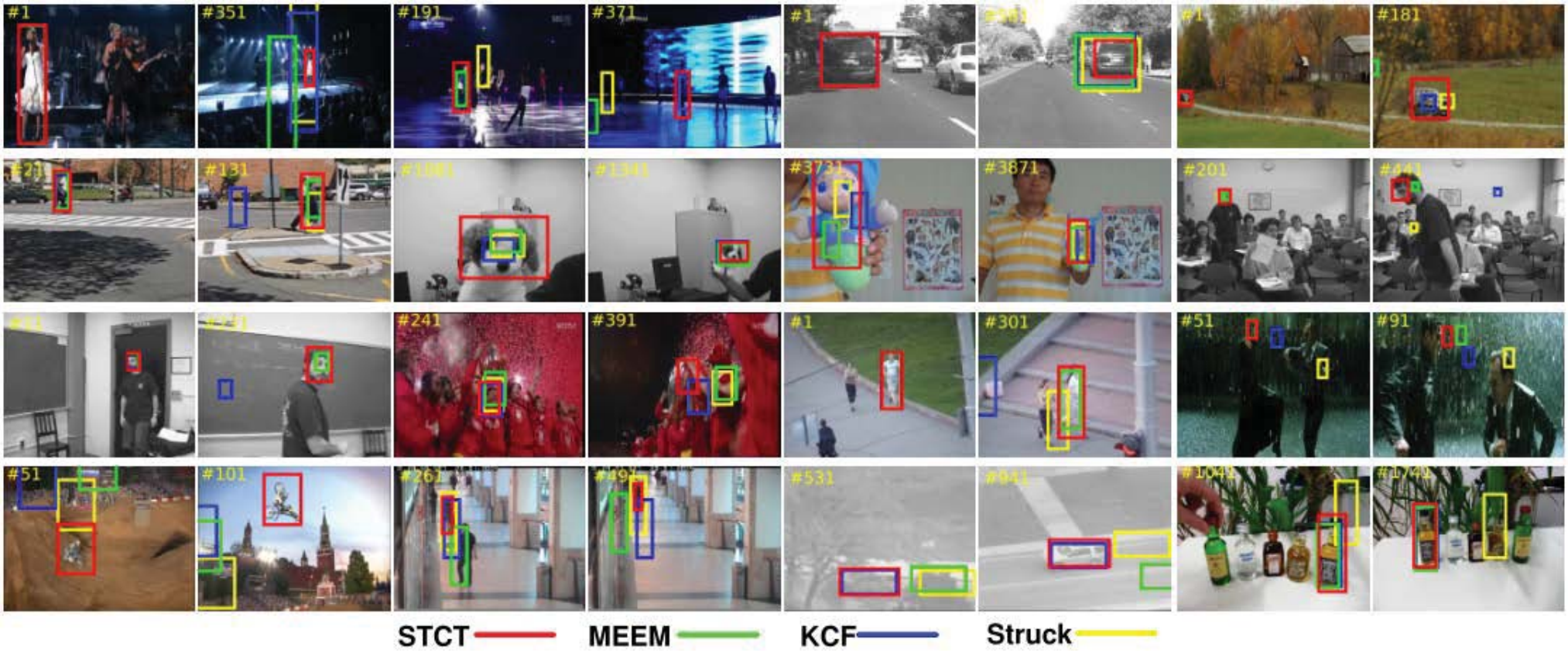
- Evaluation on OTB Date Set



Average success plots and precision plots of nine method in OTB data set for OPE,TRE and SRE evaluations. Trackers are ranked according to the Area Under

Experimental Results

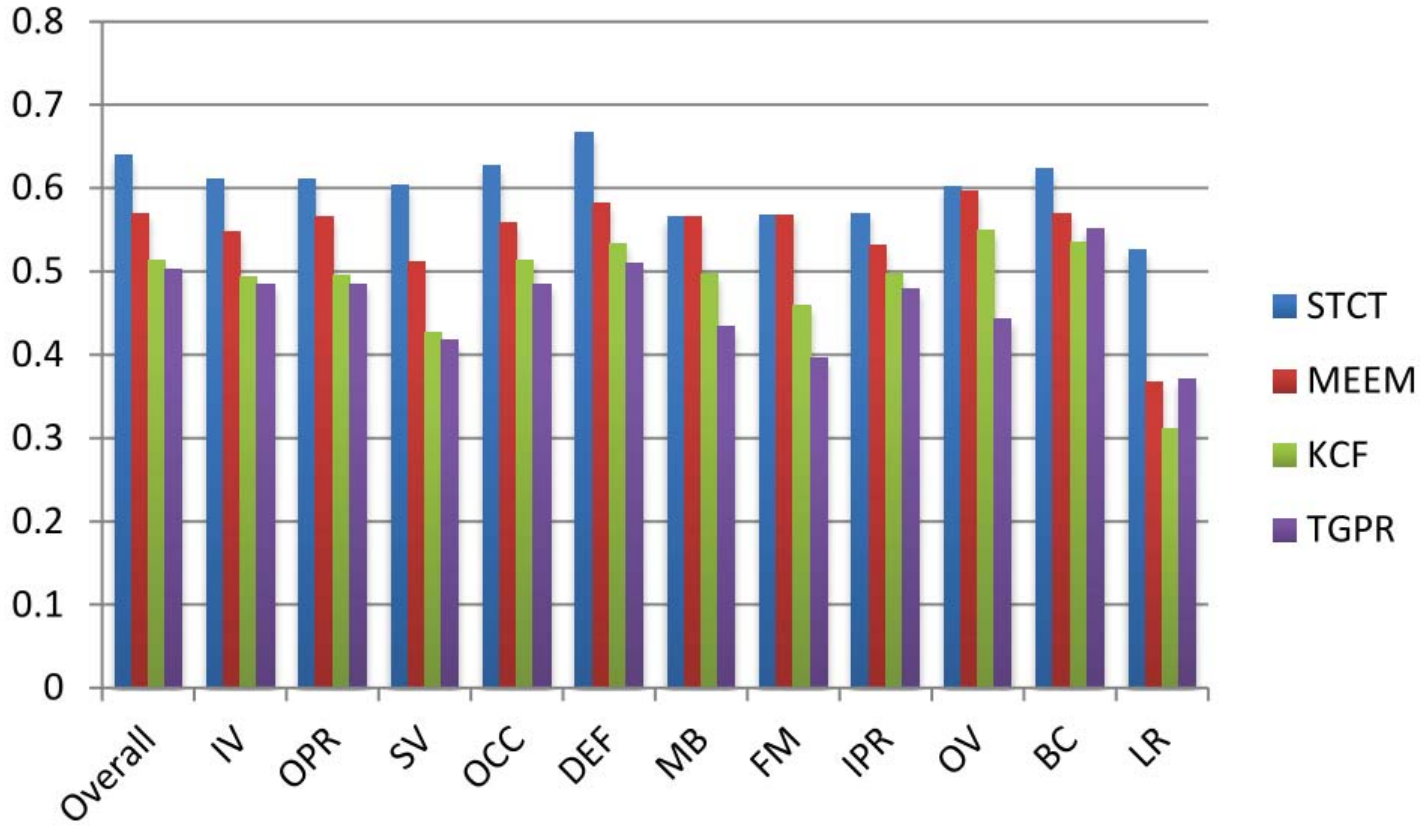
- Evaluation on OTB Date Set



Qualitative results of the proposed STCT tracker on a subset of challenging sequences: Singer1, Skating1, Car4, CarScale,Couple, Dog1, Doll, Freeman3, Freeman1, Soccer, Jogging-2, ,Matrix, MotorRolling , Walking2, Suv and Liquor.

Experimental Results

- Evaluation on OTB Date Set



Average AUC scores of the success plots of the four leading trackers under different attributes of test sequences in OPE, including: illumination variation (IV), out-of-plane rotation (OPR),scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-view (OV), background cluttered (BC) and low resolution (LR).

Experimental Results

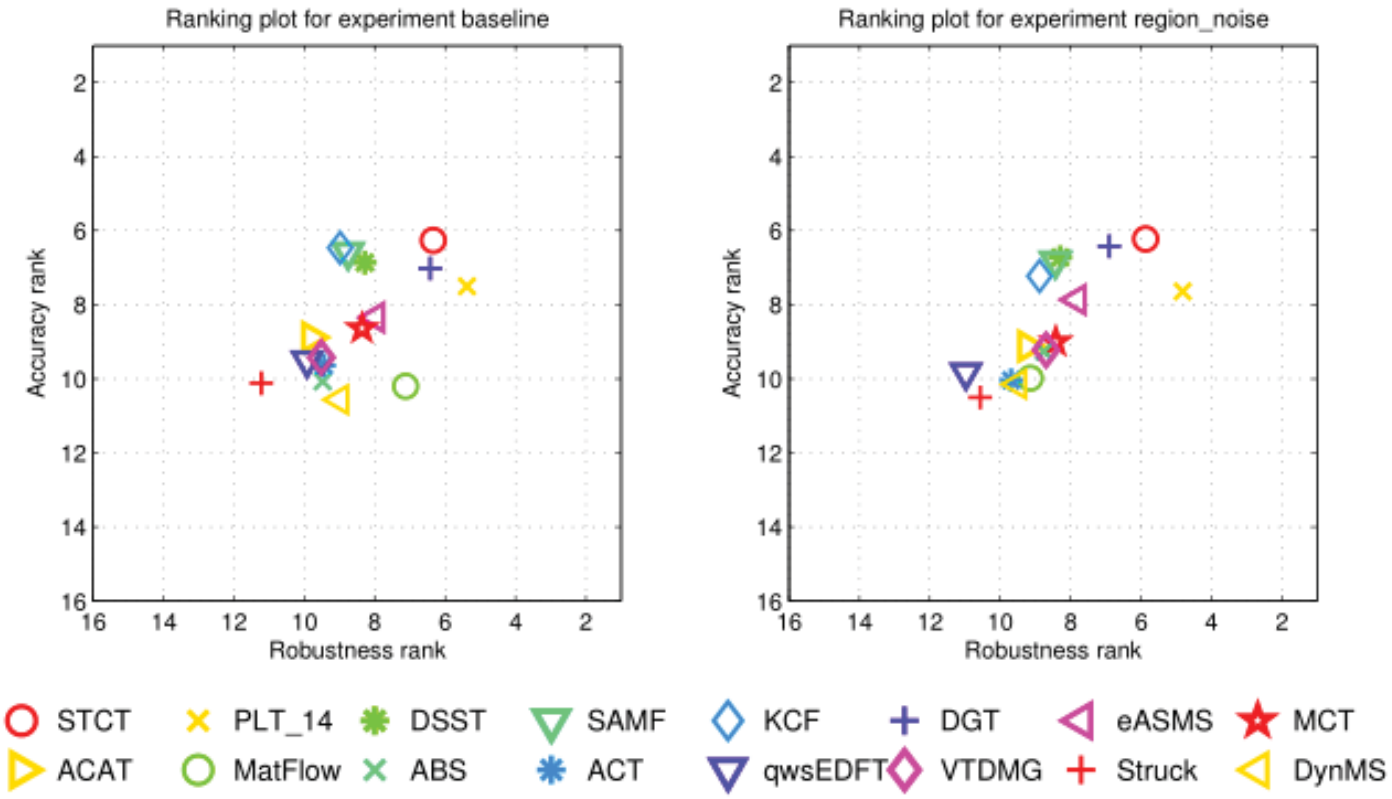
- Evaluation on VOT2014 Date Set

Trackers	baseline			region_noise			Overall Rank
	Acc. Rank	Rob. Rank	Average	Acc. Rank	Rob. Rank	Average	
STCT	6.26	6.34	6.30	6.22	5.87	6.05	6.17
PLT_14	7.50	5.38	6.44	7.64	4.81	6.23	6.33
DGT	7.02	6.42	6.72	6.42	6.90	6.66	6.69
DSST	6.86	8.28	7.57	6.72	8.29	7.51	7.54
SAMF	6.58	7.67	8.76	6.82	8.43	7.63	7.65
KCF	6.46	8.98	7.72	7.22	8.88	8.05	7.89
eASMS	8.34	7.98	8.16	7.86	7.83	7.85	8.00
MCT	8.64	8.36	8.50	9.00	8.42	8.71	8.61
MatFlow	10.20	7.12	8.66	9.98	9.15	9.57	9.11
VTDMG	9.42	9.52	8.47	9.22	8.70	8.96	9.21

The average ranks of accuracy and robustness under baseline and region noise experiments in VOT2014. The first, second and third best methods are highlighted in red, blue and green colors, respectively

Experimental Results

- Evaluation on VOT2014 Date Set



The robustness-accuracy ranking plots of 16 leading tracking methods under baseline and region noise experiments in VOT2014 data set. The better trackers are located at the upper-right corner.

Thank You

Thank You!!

http://ice.dlut.edu.cn/lu/index.html