A Story from Saliency to Objectness, and Extension by Deep Neural Network with Perspective and Doubt

Shu Kong

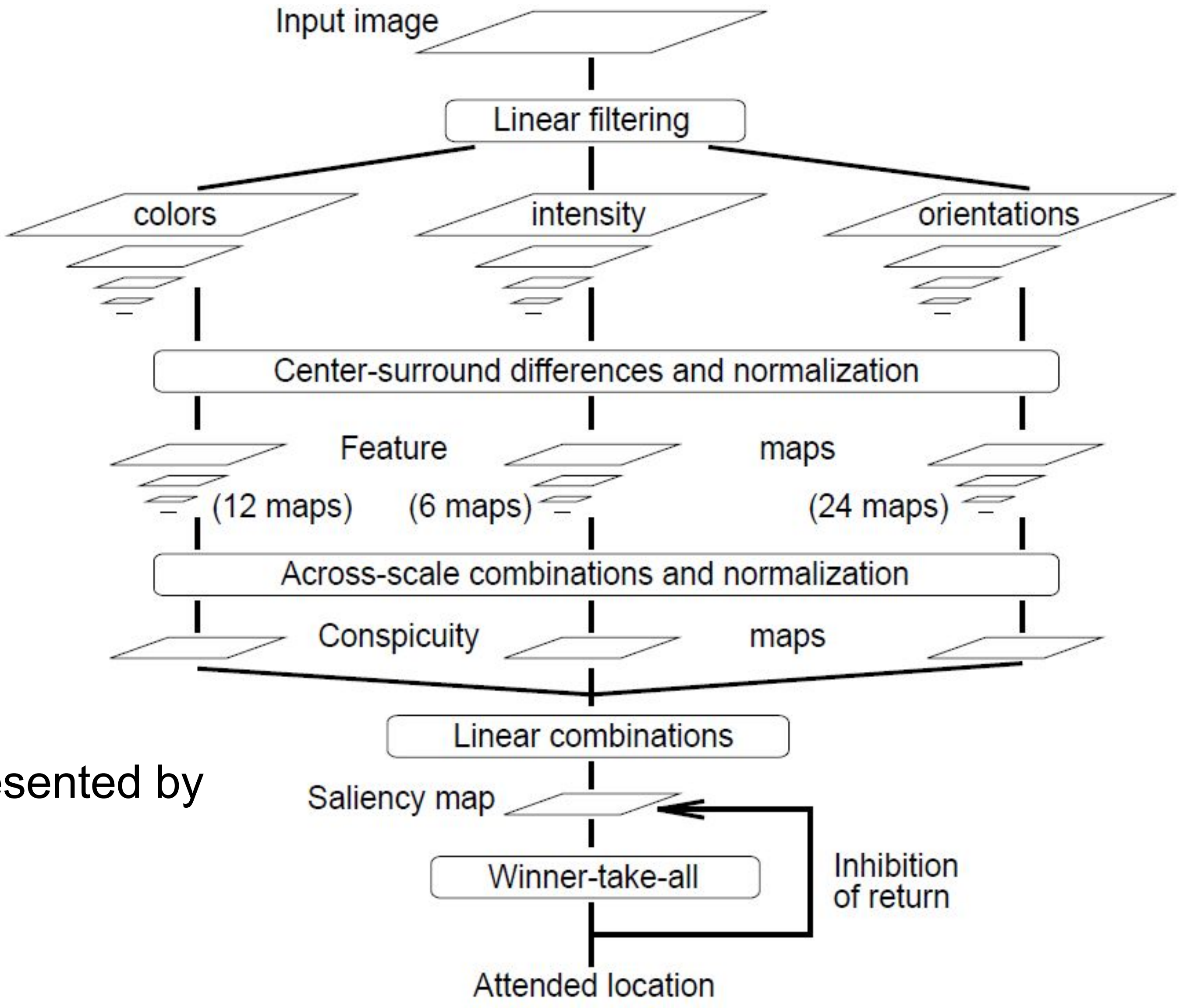
CS, ICS, UCI

Outline

1. Review some papers on saliency detection and objectness detection;
2. Study a paper that uses DNN to do objectness detection by learning features;
3. See some results achieved not by NN;
4. Unleash the mind, and discuss what else the "omnipotent" DNN can do.

1. A Model of Saliency-based Visual Attention for Rapid Scene Analysis
2. Measuring the Objectness of Image Windows
3. BING - Binarized Normed Gradients for Objectness Estimation at 300fps
4. scalable object detection using deep neural networks

Saliency Detection

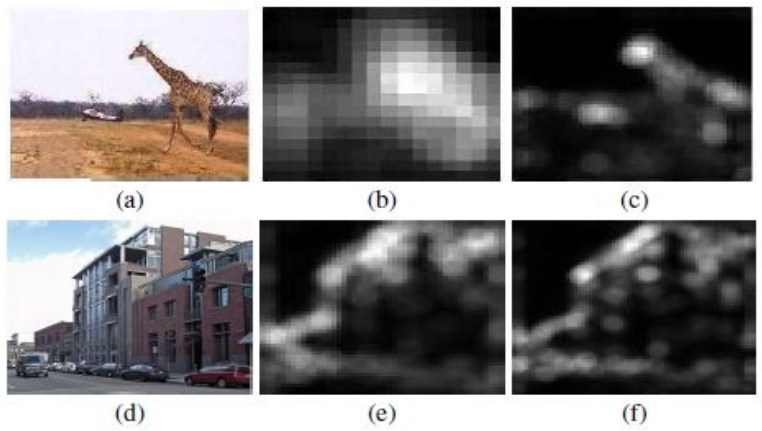


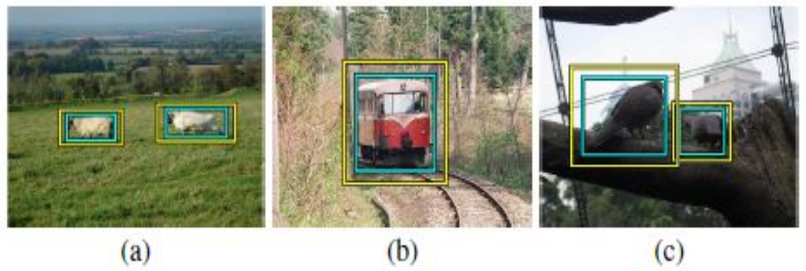
multiscale,
layers,
neural network

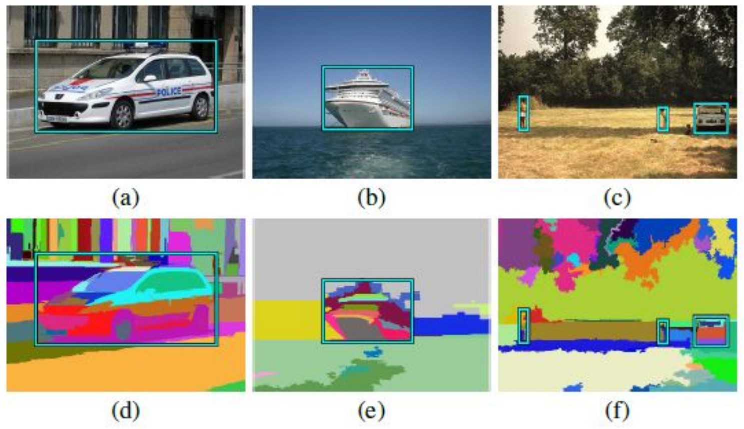
resembling those presented by Peiyun last week

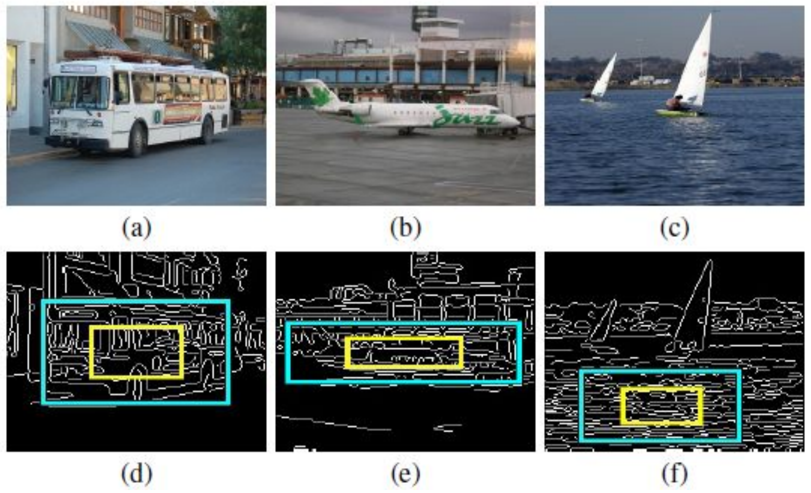
[A Model of Saliency-based Visual Attention for Rapid Scene Analysis]

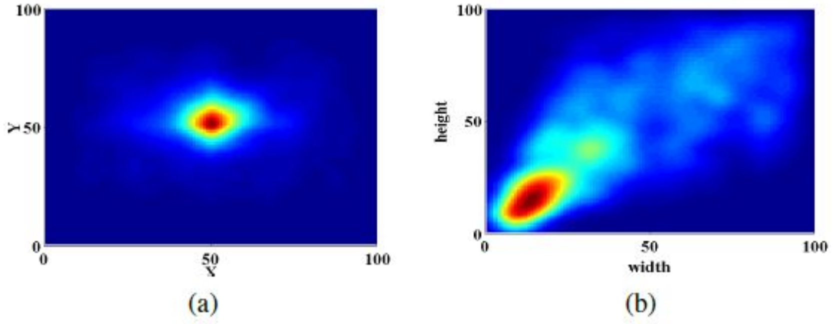
Objectness Detection (What is?)











Multiscale Saliency, Color Contrast
Edge Density, Superpixels Straddling
Local and Size -- these cues are combined by learning weights.

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

Objectness Detection (BING)

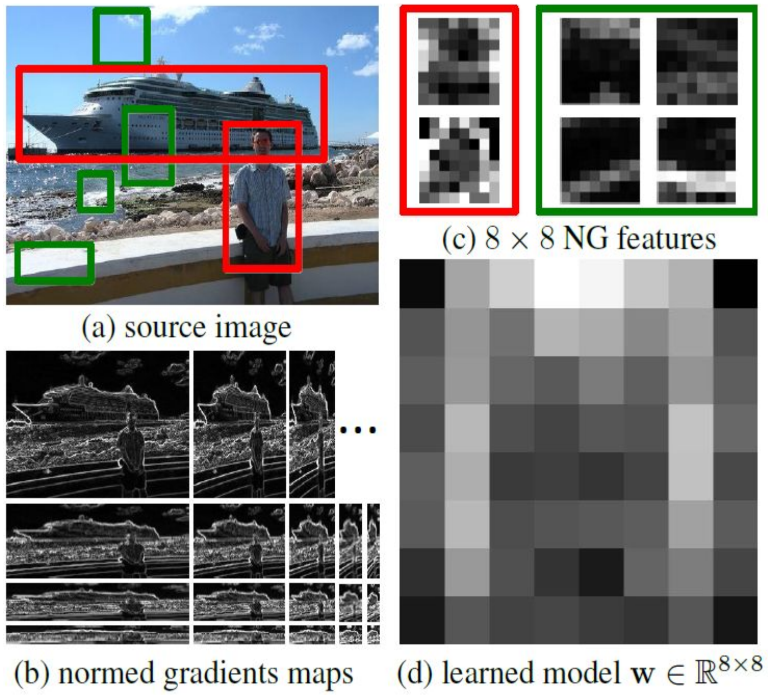
try to find close boundary

objectness score is determined by response to the template and the scaling size, where v_i and t_i are the paramter wrt scaling factor

$$o_l = v_i \cdot s_l + t_i$$

$$s_l = \langle \mathbf{w}, \mathbf{g}_l \rangle$$

$$l = (i, x, y).$$


(a) source image
(b) normed gradients maps
(c) 8×8 NG features
(d) learned model $\mathbf{w} \in \mathbb{R}^{8\times 8}$

w is the template (learned model 8x8)
g is the extracted patch in the feature map
l denotes location, including scaling factor i and position (x,y) of the patch

Remarks
1. there is no reason that left part should be more important than right part
2. scaling factor is pre-defined -- gaps exist among different scales
3. how about snakes, C. elegans, or even athletes

BING - Binarized Normed Gradients for Objectness Estimation at 300fps

Objectness Detection (BING)

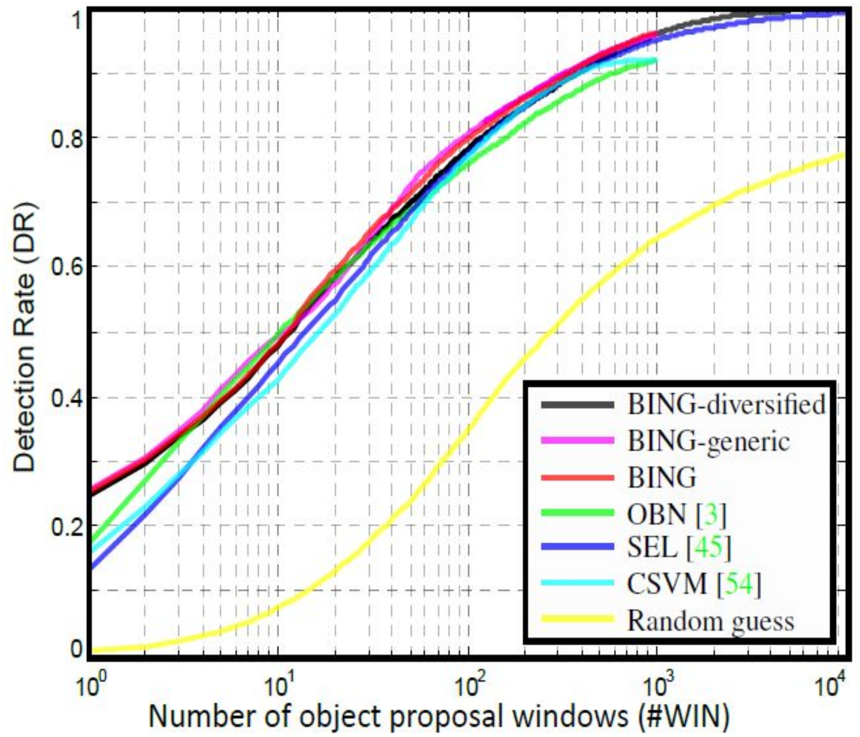


Figure 3. Tradeoff between #WIN and DR (see [3] for more comparisons with other methods [6, 16, 20, 24, 29, 32, 35, 47] on the same benchmark). Our method achieves 96.2% DR using 1,000 proposals, and 99.5% DR using 5,000 proposals. This figure is best viewed in color.

1. good performance on VOC2007;
2. performs fast owing to binary approximation, INT64 and BYTE presentation, and BITWISE computation.

Method	[21]	OBN [3]	CSVM [54]	SEL [45]	Our BING
Time (seconds)	89.2	3.14	1.32	11.2	**0.003**

Table 1. Average computational time on VOC2007.

BING - Binarized Normed Gradients for Objectness Estimation at 300fps

Objectness Detection by DNN

DeepMultiBox -- a saliency-inspired neural network model for detection, which predicts a set of class-agnostic bounding boxes along with a single score for each box, corresponding to its likelihood of containing any object of interest.

Three contributions:

1. define object detection as a regression problem to the coordinates of several bounding boxes, as well as a confidence score of how likely this box contains an object;
2. train the bounding box predictors as part of the network training -- utilizing the excellent representation learning abilities of DNN.
3. scalable way for classification -- post-classifying less than ten boxes to achieve competitive detection results.

Model -- Bndbox and conf values are as values of nodes in last layer of the NN

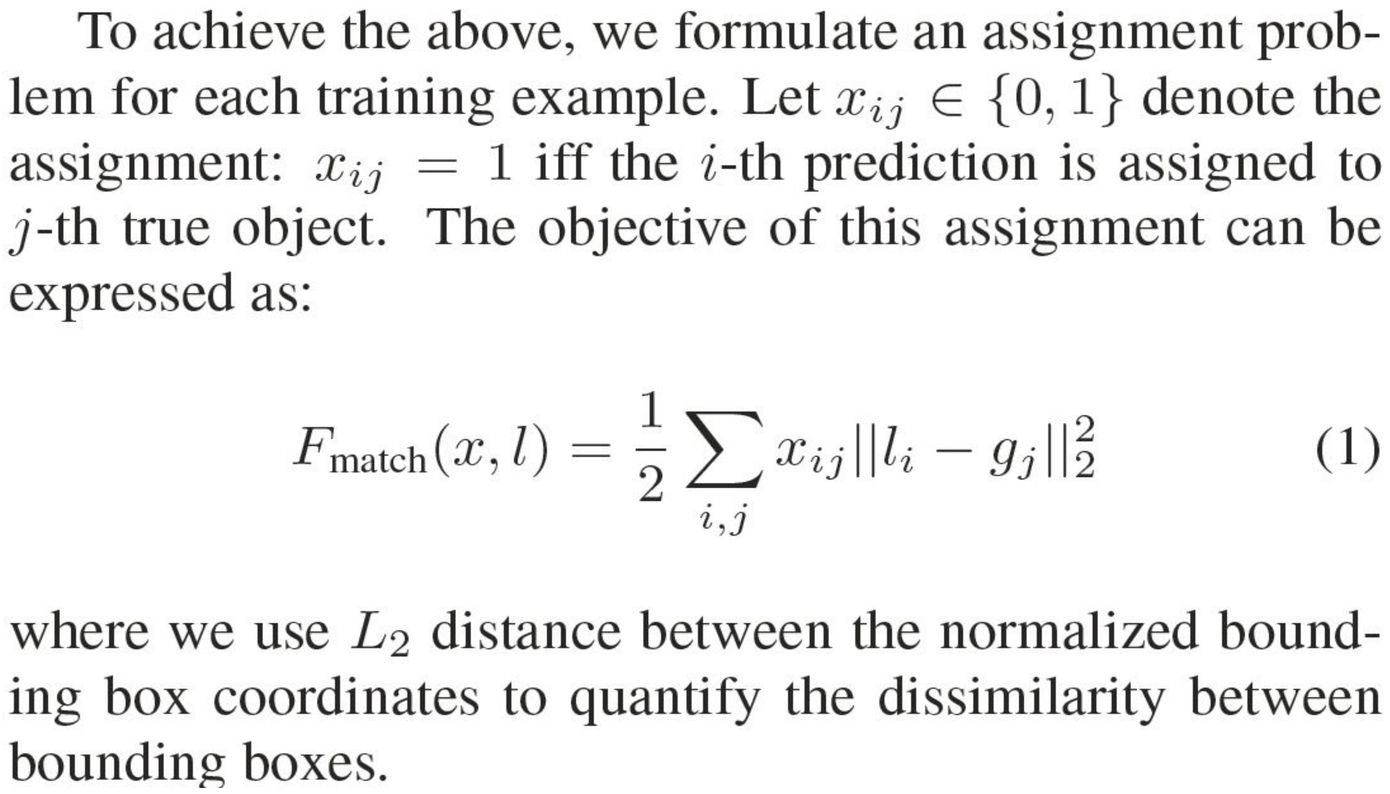
Bndbox -- upper-left and lower-right coordinates as four-length vector, produced by a linear transformation of the last hidden layer

Confidence -- single node value between [0,1], linear transformed from last hidden layer followed by a sigmoid.

scalable object detection using deep neural networks

Objectness Detection by DNN

regression to coordinates -- try to optimize only the subset of predicted boxes which match best the ground truth ones



To achieve the above, we formulate an assignment problem for each training example. Let $x_{ij} \in \{0, 1\}$ denote the assignment: $x_{ij} = 1$ iff the i-th prediction is assigned to j-th true object. The objective of this assignment can be expressed as:

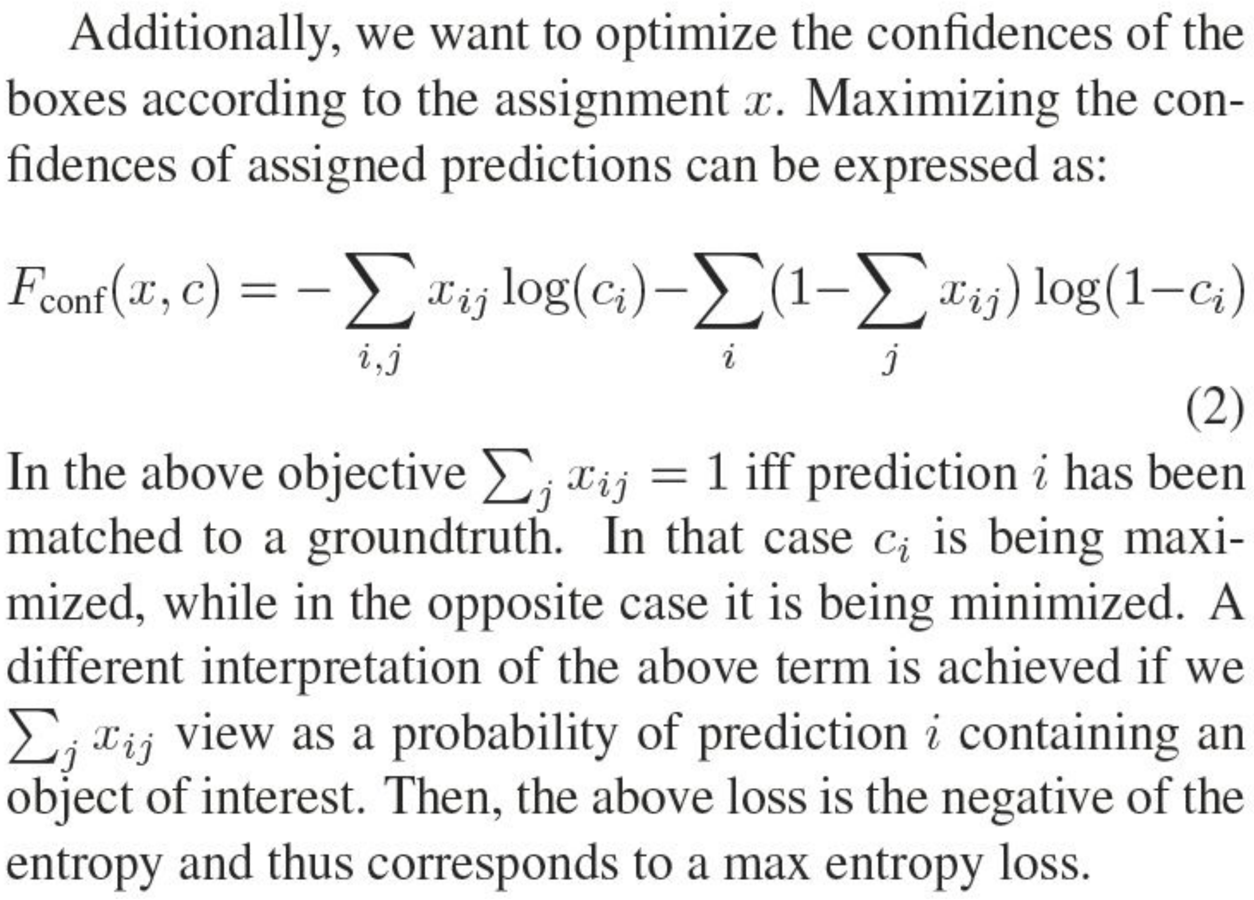
$$F_{\text{match}}(x, l) = \frac{1}{2} \sum_{i,j} x_{ij} ||l_i - g_j||_2^2 \qquad (1)$$

where we use L_2 distance between the normalized bounding box coordinates to quantify the dissimilarity between bounding boxes.

scalable object detection using deep neural networks

Objectness Detection by DNN

At the same time, try to minimize the confidences of the remaining predictions, which are deemed not to localize the true objects well.



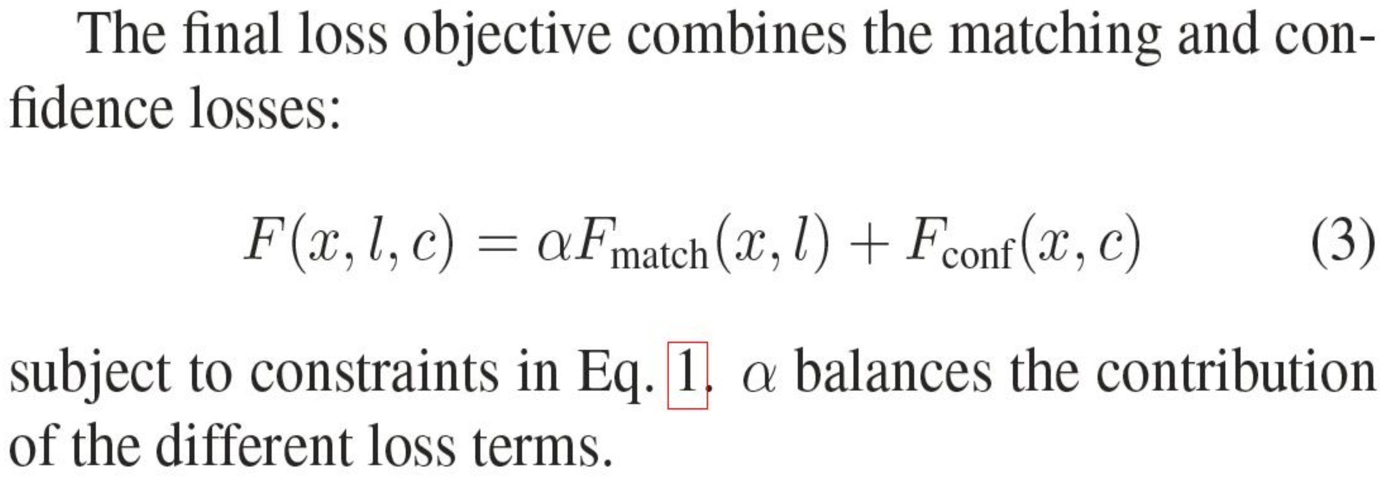
Additionally, we want to optimize the confidences of the boxes according to the assignment x. Maximizing the confidences of assigned predictions can be expressed as:

$$F_{\mathrm{conf}}(x, c) = -\sum_{i,j} x_{ij} \log(c_i) - \sum_{i} \Big(1 - \sum_{j} x_{ij}\Big) \log(1 - c_i) \tag{2}$$

In the above objective $\sum_j x_{ij} = 1$ iff prediction i has been matched to a groundtruth. In that case c_i is being maximized, while in the opposite case it is being minimized. A different interpretation of the above term is achieved if we $\sum_j x_{ij}$ view as a probability of prediction i containing an object of interest. Then, the above loss is the negative of the entropy and thus corresponds to a max entropy loss.

[scalable object detection using deep neural networks]

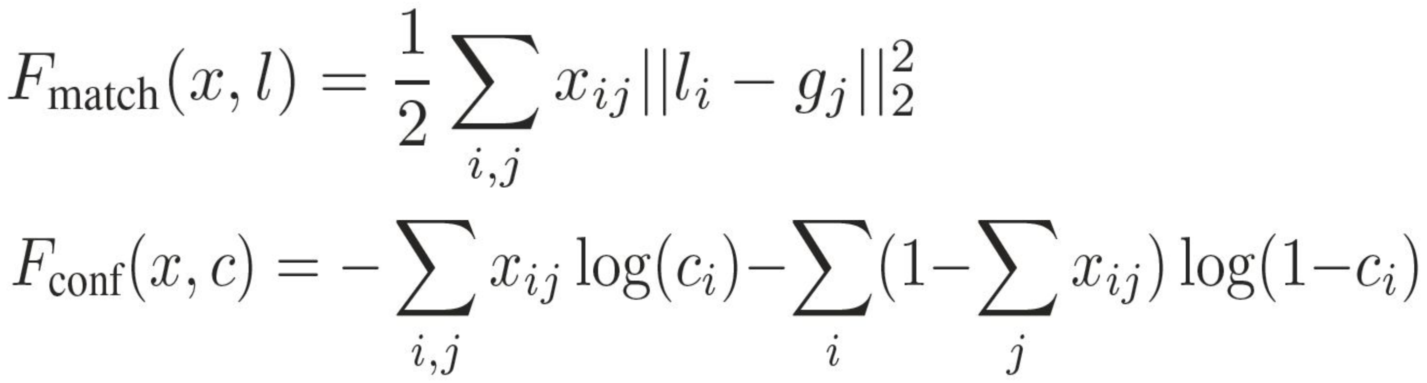
Objectness Detection by DNN



The final loss objective combines the matching and confidence losses:

$$F(x,l,c) = \alpha F_{\text{match}}(x,l) + F_{\text{conf}}(x,c) \tag{3}$$

subject to constraints in Eq. $\boxed{1}$. α balances the contribution of the different loss terms.

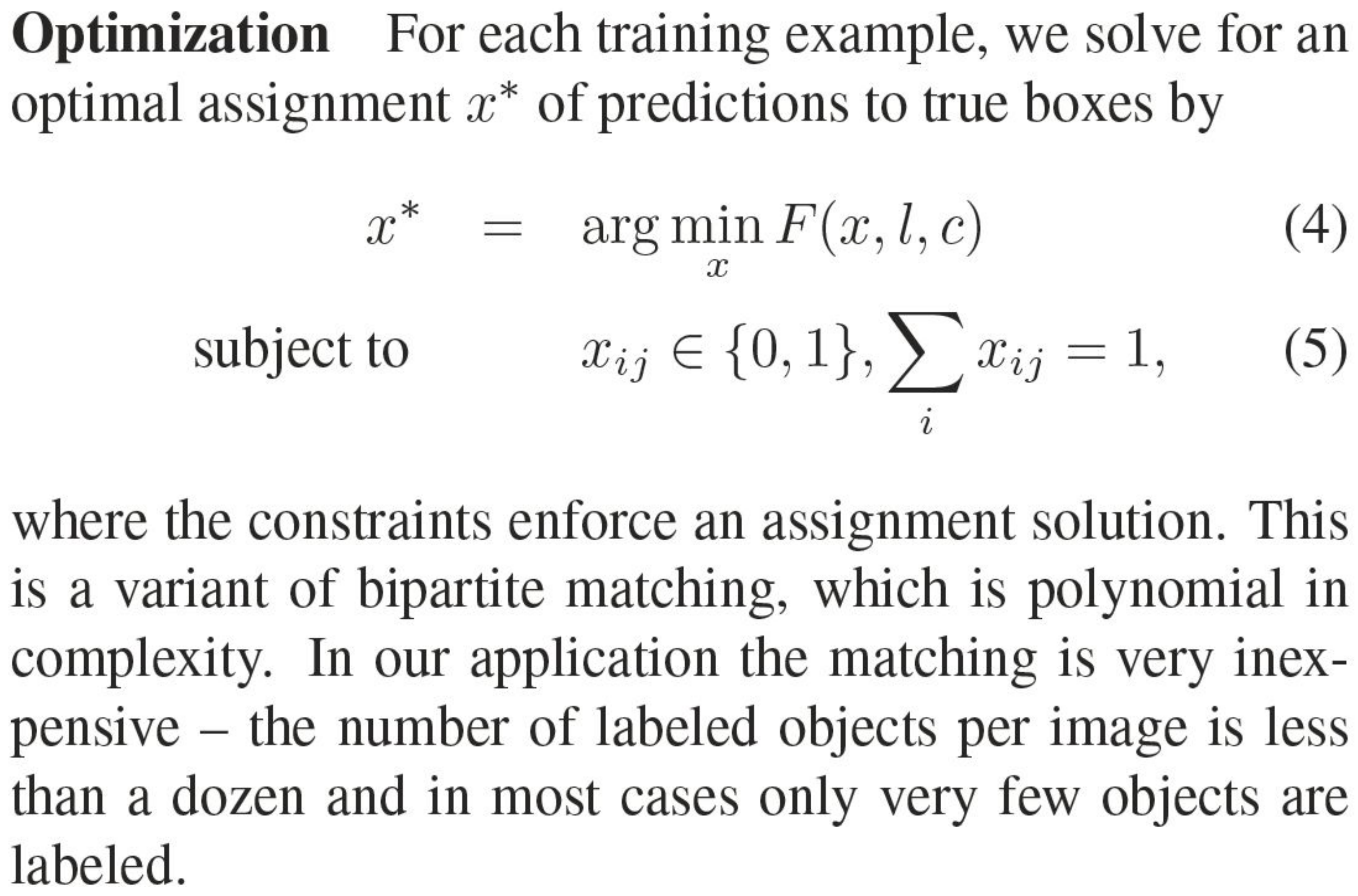


$$F_{\text{match}}(x,l) = \frac{1}{2}\sum_{i,j} x_{ij}||l_i - g_j||_2^2$$

$$F_{\text{conf}}(x,c) = -\sum_{i,j} x_{ij}\log(c_i) - \sum_i \left(1 - \sum_j x_{ij}\right)\log(1-c_i)$$

scalable object detection using deep neural networks

Objectness Detection by DNN



Optimization For each training example, we solve for an optimal assignment x^* of predictions to true boxes by

$$\begin{aligned} x^* &= \arg\min_x F(x, l, c) && (4) \\ \text{subject to} \quad & x_{ij} \in \{0, 1\}, \sum_i x_{ij} = 1, && (5) \end{aligned}$$

where the constraints enforce an assignment solution. This is a variant of bipartite matching, which is polynomial in complexity. In our application the matching is very inexpensive – the number of labeled objects per image is less than a dozen and in most cases only very few objects are labeled.

scalable object detection using deep neural networks

Objectness Detection by DNN

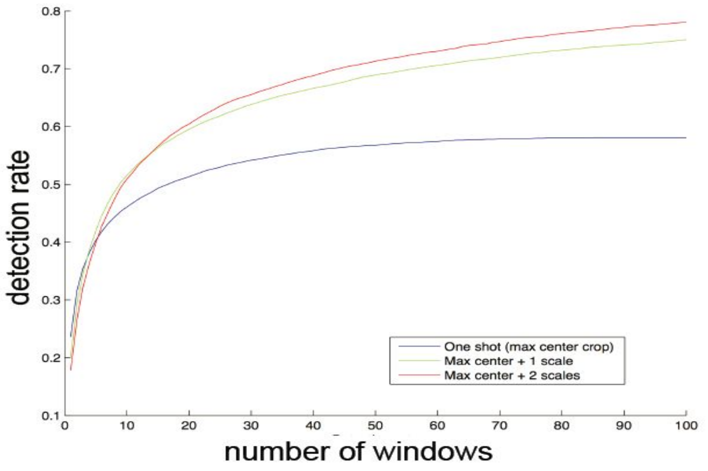


Figure 1. Detection rate of class "object" vs number of bounding boxes per image. The model, used for these results, was trained on VOC 2012.

Table 2. Performance of Multibox (the proposed method) vs. clas sifying ground-truth boxes directly and predicting one box pe class

Method	det@5	class@5
One-box-per-class	61.00%	79.40%
Classify GT directly	82.81%	82.81%
DeepMultiBox, top 1 window	56.65%	73.03%
DeepMultiBox, top 3 windows	58.71%	77.56%
DeepMultiBox, top 5 windows	58.94%	78.41%
DeepMultiBox, top 10 windows	59.06%	78.70%
DeepMultiBox, top 25 windows	59.04%	78.76%

1. It has to generate data (10-30M).
2. If the bndbox predicted is outside the crop area, it has to map and truncate it to the final image area at the end. It uses NMS to prune the boxes.
3. It also uses second model to classify each bndbox as objects of interest or "background".
4. From about 100 candidates, it carefully select 10 highest scoring detections after NMS (with overlap 0.5) for subsequent classification in a separate passes through another network -- two-step process for detection.
5. It is hard to imagine that it can predicts coordinates, much different from classification and detection (the output domain is fixed in a span).

scalable object detection using deep neural networks

Objectness Detection (Comparison)

BING

1. can be a hard-wired feature, train and test are very fast
2. natural to be capitalized with bitwise operations
3. need to pre-define scaling patterns, and produce relatively more candidates

DeepMultiBox

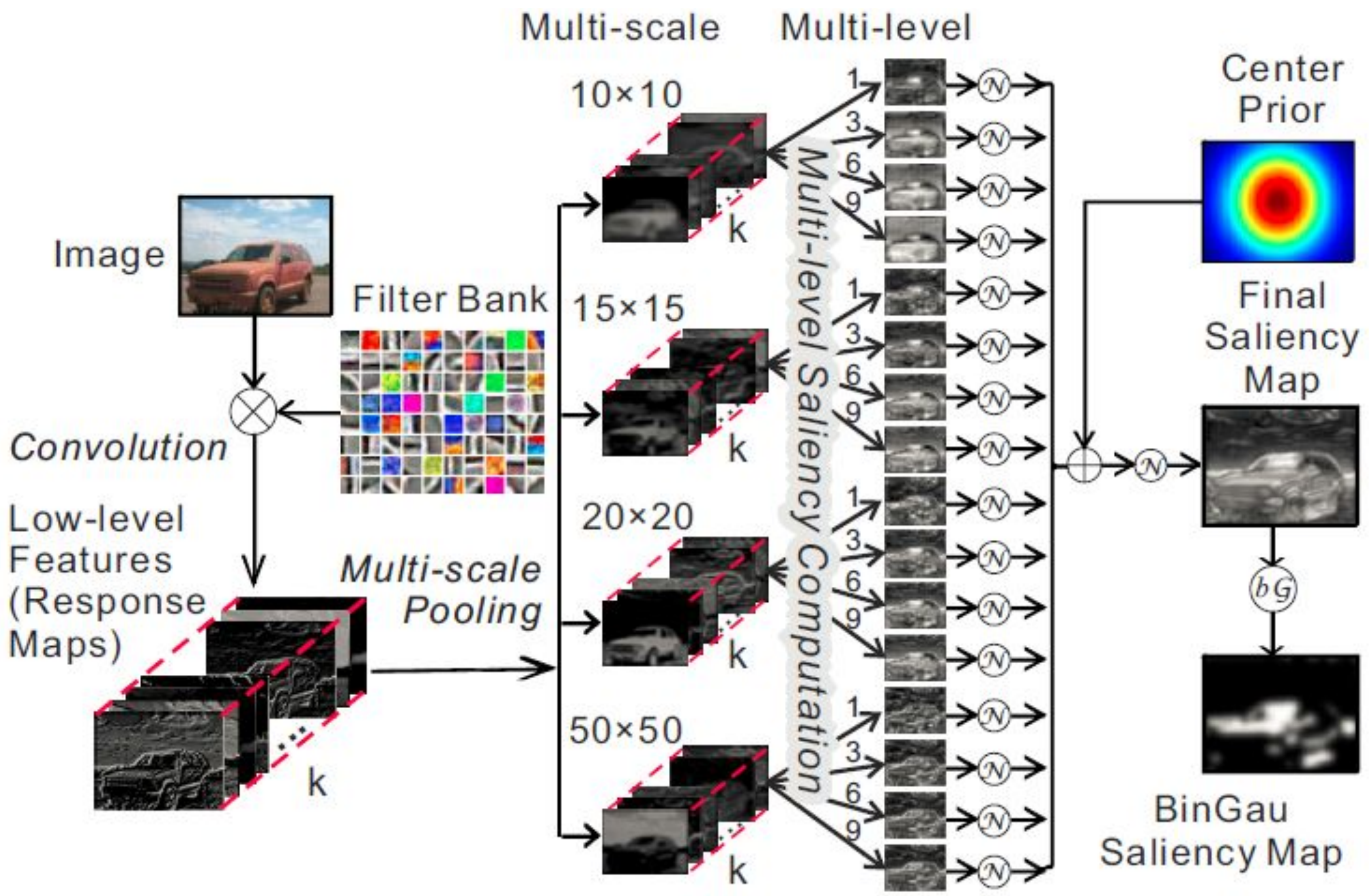
1. learned feature, need tons of data to train
2. appealing to predict coordinates of less candidates
3. need to deal with output of bndbox, postprocessing and selection

Which one do you think is better?

Do we really need DNN?

Baseline Method Without DNN (1)

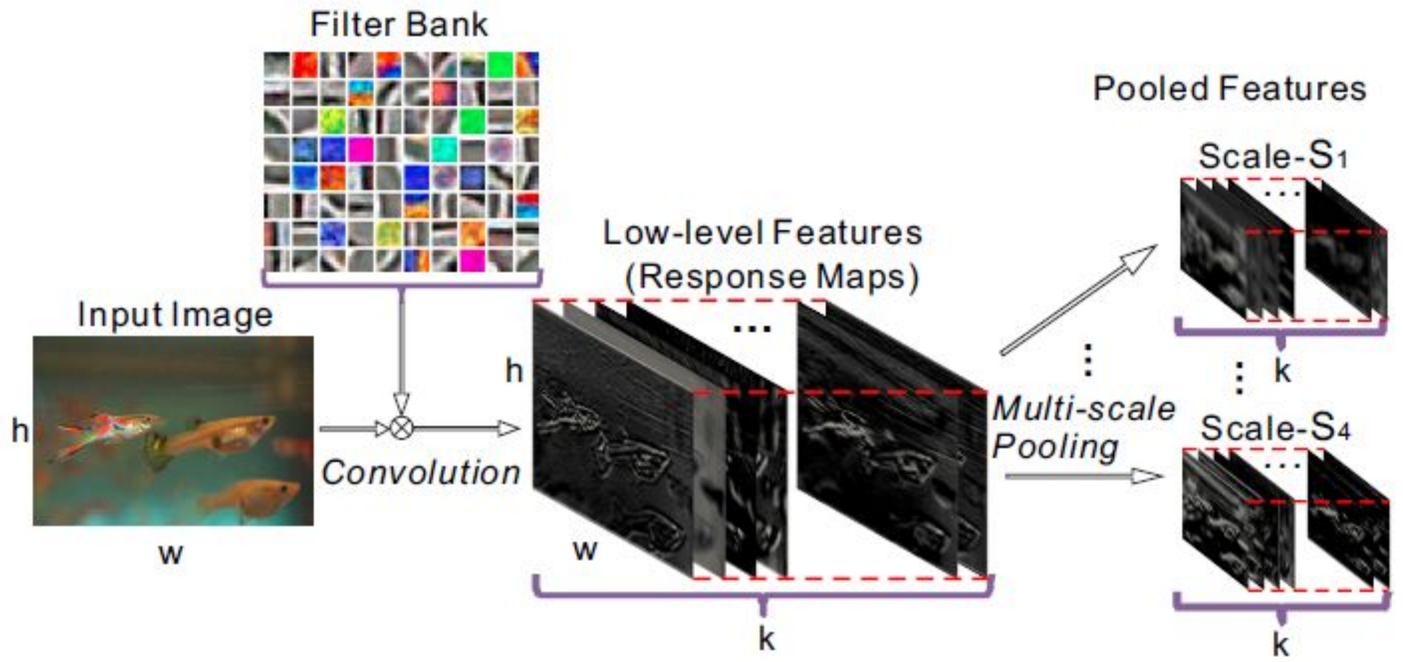
Saliency Detection with Layers, but NO Training



saliency detection within a deep convolutional architecture

Baseline Method Without DNN (1)

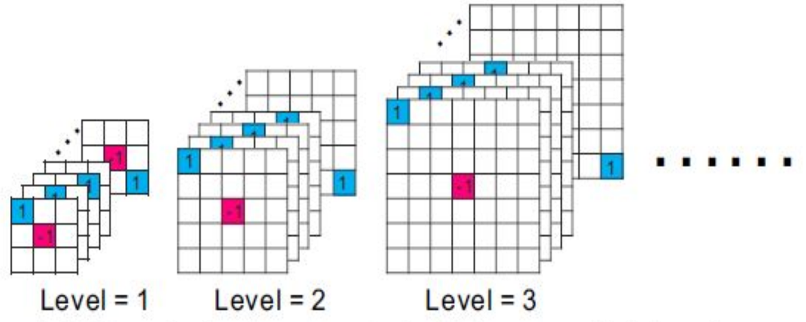
Saliency Detection with Layers, but NO Training



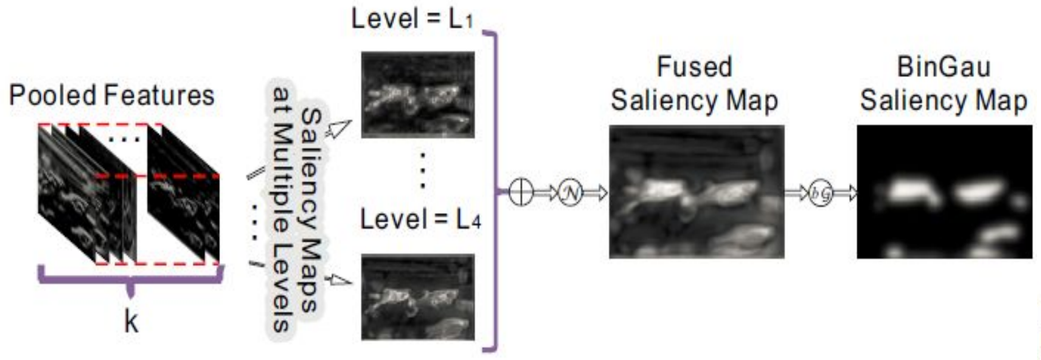
saliency detection within a deep convolutional architecture

Baseline Method Without DNN (1)

Saliency Detection with Layers, but NO Training



(a) Filters for local contrast calculation at multiple levels



(b) Saliency Map Calculation at Multiple Local Contrast Levels

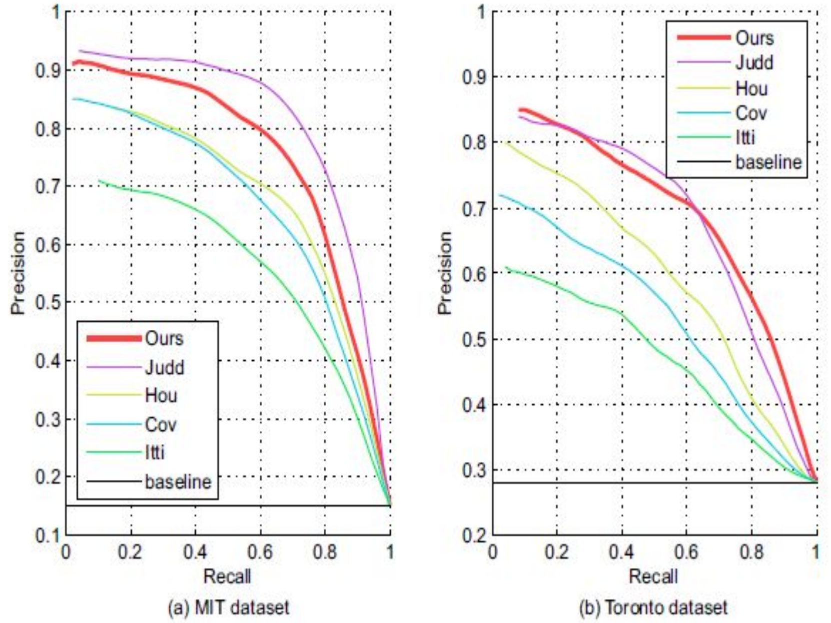
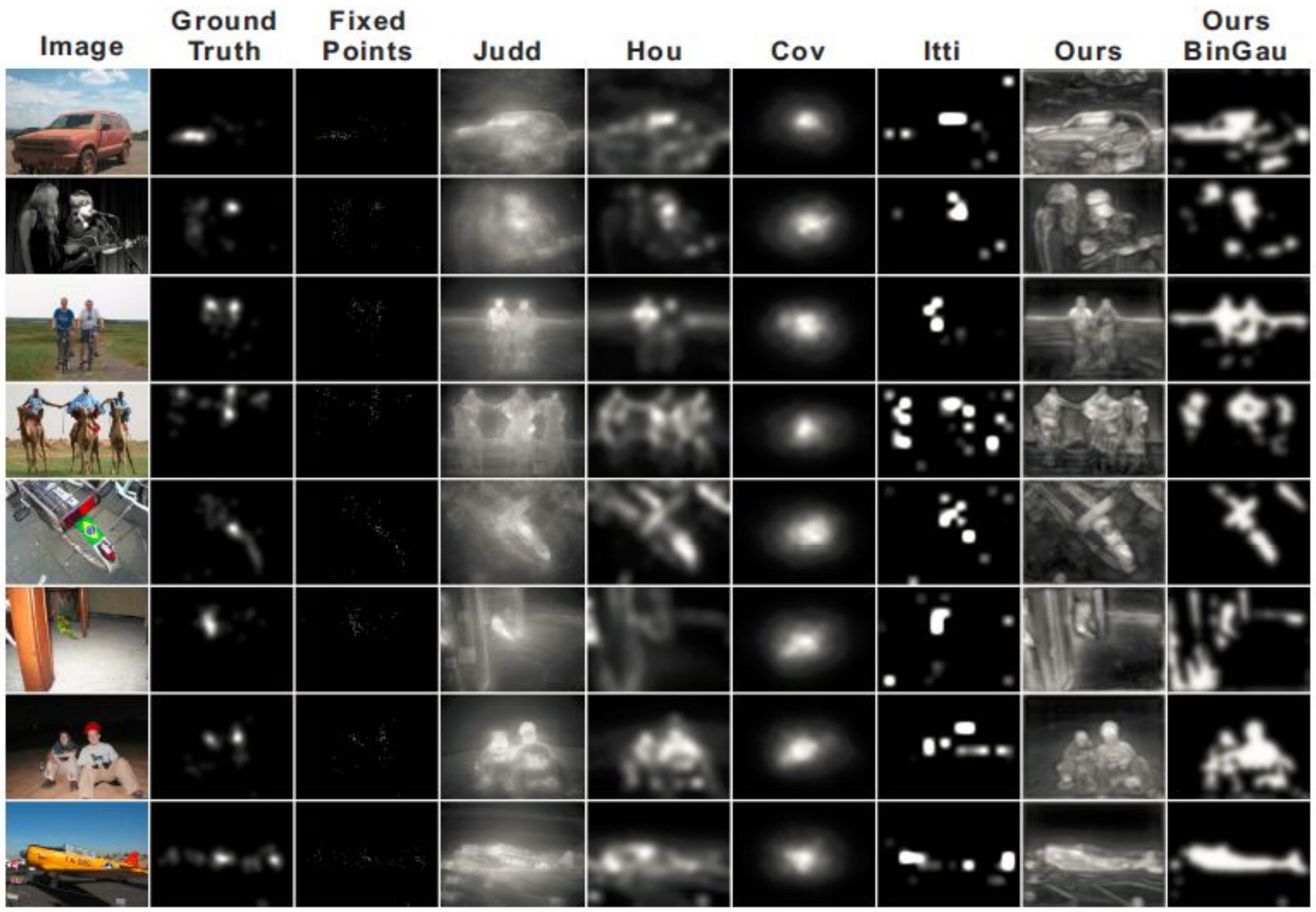


Figure 5: Smoothed precision-recall curve on MIT (left) and Toronto (right) datasets.

[saliency detection within a deep convolutional architecture]

Baseline Method Without DNN (1)

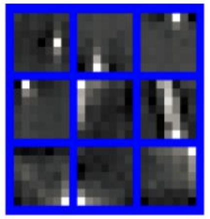
Saliency Detection with Layers, but NO Training



[saliency detection within a deep convolutional architecture]

Baseline Method Without DNN (2)

Off-the-Shelf Tricks for Face Recognition



k-means
convolution
normalization
soft-threshold
3D pooling

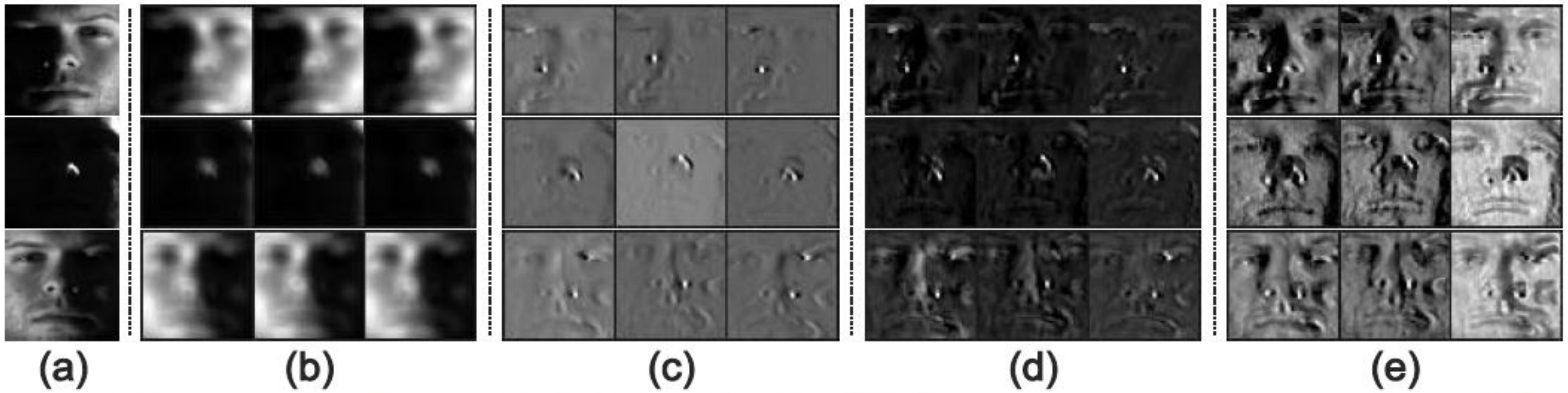
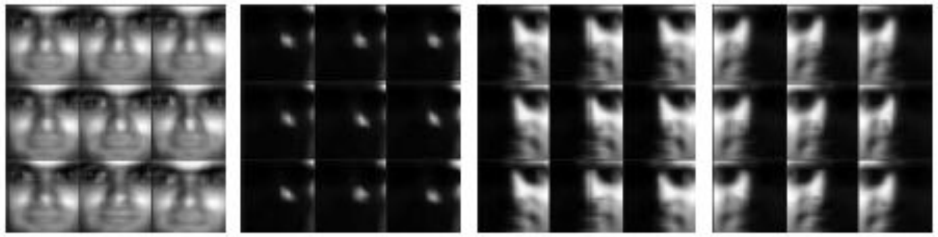


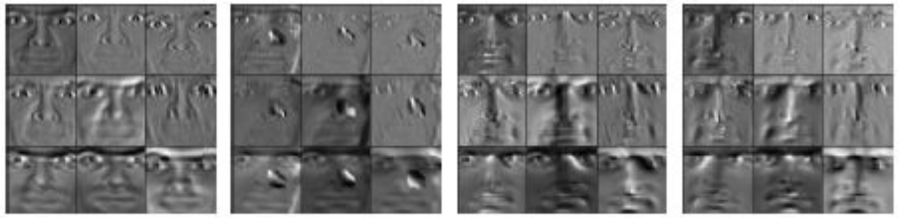
Fig. 2. Demonstration of soft convolution: (a) nine low-level filters learned by k-means over the face dataset [14]; (b) three images of the same person under different illumination conditions; (c) three convolutional feature maps of each image displayed in each row with three different filters; (d) normalized maps over (c); (e) thresholded maps over (d); and (f) normalized maps over (e).

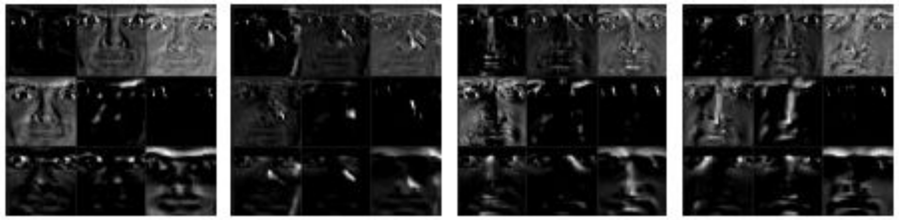
Modeling Neuron Selectivity over Simple Mid-Level Features for Image Classification

Off-the-Shelf Tricks for Face Classification

(a)

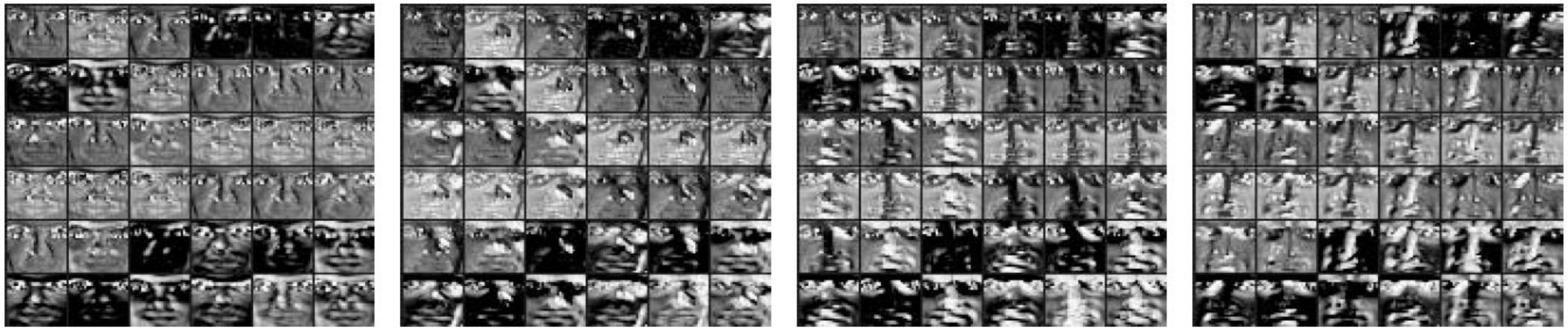

(b)


(c)


(d)

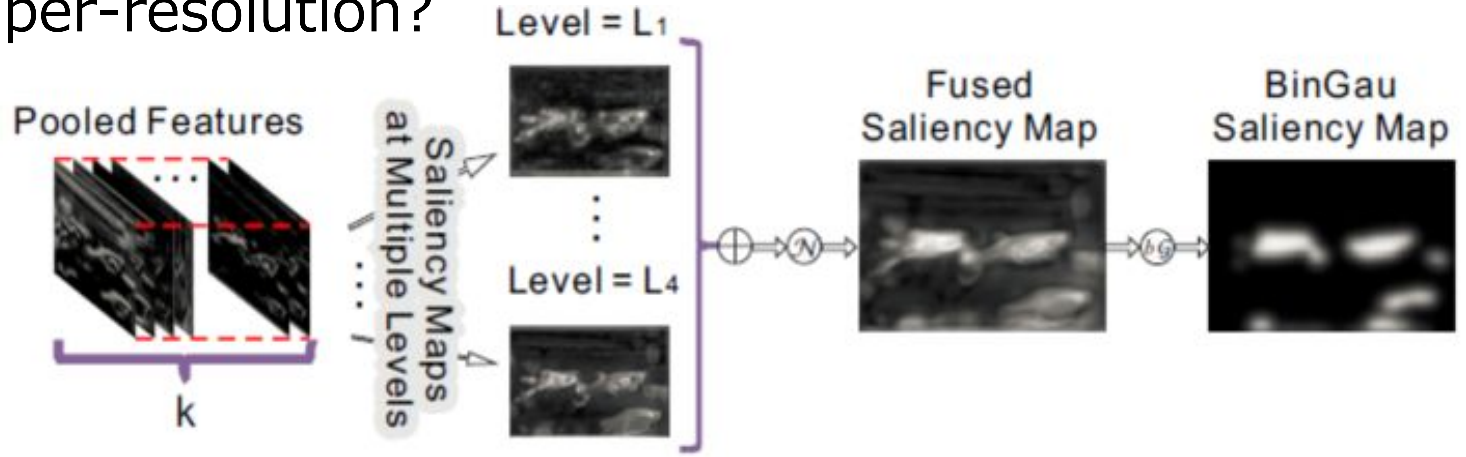
(e)

99.6% accuracy!


(f)

What else can DNN do?

1. If it is possible, can DNN be used to evaluate image quality and image blurry degree, or applied to image deblur and super-resolution?



2. Can DNN be specifically trained as rotation invariant mechanism on simple database? Or, scale-invariant, affine-invariant mechanism? (due to big data?)

3. Can DNN be trained to deal with shape information with distortions, such as snake and C. elegans?

Thank you

Abstract

We will first briefly review some papers from saliency detection to objectness detection, and then carefully study a paper that uses DNN to do objectness detection. It predicts coordinates of multiple objectness in an image.

With the idea that solving the two problems helps image interpretation, regardless of whether they mimic neural science, we will see how recent methods work on objectness detection so well, from hard-wired feature (BING) to learned feature by DNN. It seems DNN is so omnipotent that it can predict the coordinates.

Then, we will see some interesting and amazing results on saliency detection and face recognition. These results are achieved by myself as a baseline study with simple off-the-shelf tricks, e.g. convolution, thresholding, 3D pooling and normalization, other than learning a complicated DNN.

But, is the success owing to DNN's unthinkable trained nonlinearility? Is it possible that other workarounds other than DNN work better on specific problems? What else can it do if we unleash our mind?