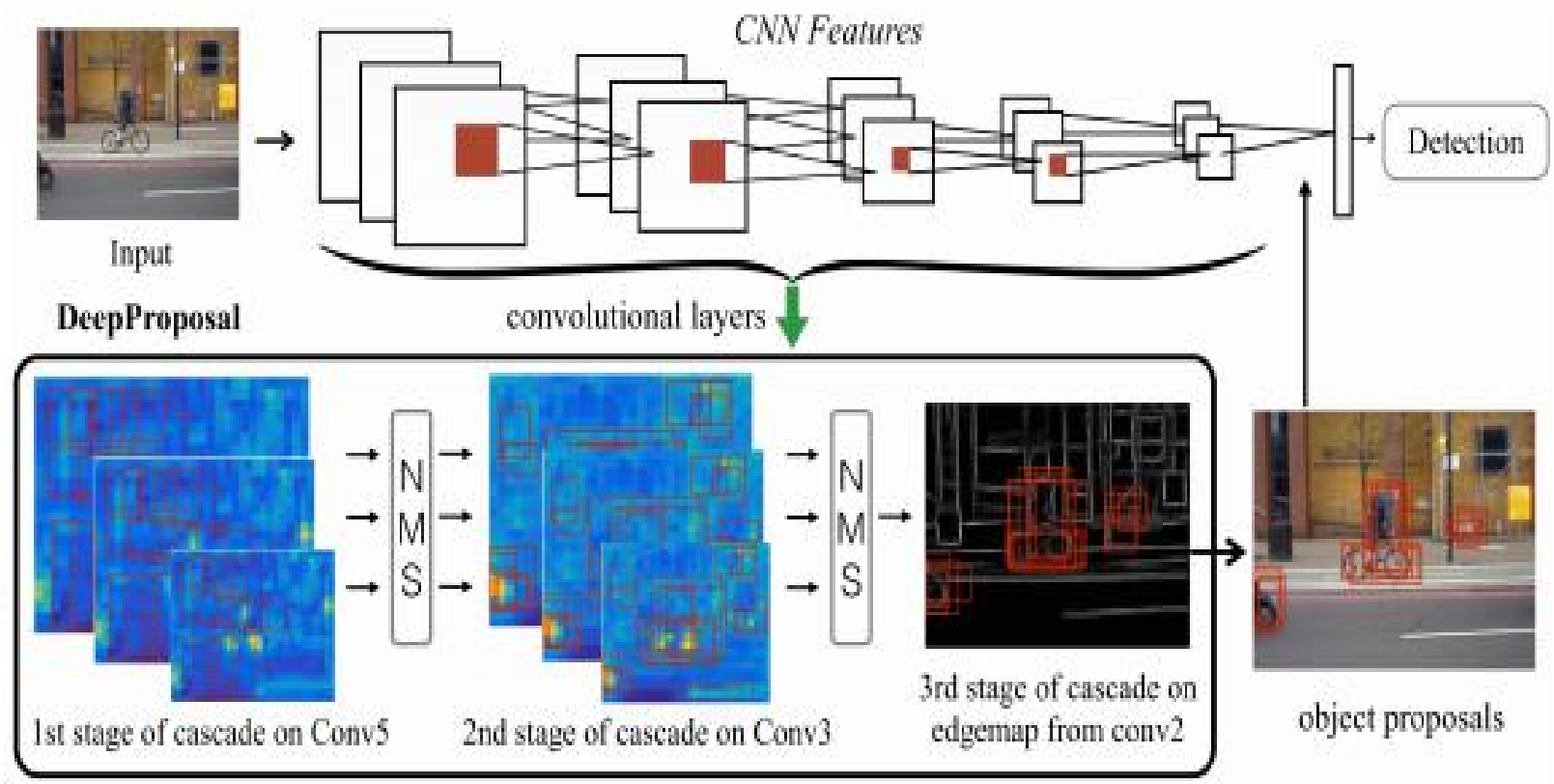
DeepProposal: Hunting Objects by Cascading Deep Convolutional Layers

A04--张磊
2015年毕业于华东政法大学经济专业
中国农业银行

Framework


CNN Features
Detection
Input
DeepProposal
convolutional layers
N
M
S
N
M
S
1st stage of cascade on Conv5
2nd stage of cascade on Conv3
3rd stage of cascade on edgemap from conv2
object proposals

Why?

- We generate hypotheses in a sliding-window fashion over different activation layers and show that the final convolutional layers can find the object of interest with high recall but poor localization due to the coarseness of the feature maps.
- Instead, the first layers of the network can better localize the object of interest but with a reduced recall.

- produces state-of-the art detection performance

Two stage

- First, selection of a reduced set of promising and class-independent hypotheses
- second, a class-specific classification of each hypothesis

- it casts the detection problem to a classification problem

The aim of Object proposal generators

- These proposals can help object detection in two ways: searching objects in fewer locations to reduce the detector running time

- using more sophisticated and expensive models to achieve better performance

Sliding window

- we select a set of window sizes that best cover the training data in terms of size and aspect ratio and use them in a sliding window fashion over the selected CNN layer.

- This approach is much faster than evaluating all possible windows and avoids to select windows with sizes or aspect ratios different from the training data and therefore probably false positives

Sliding window

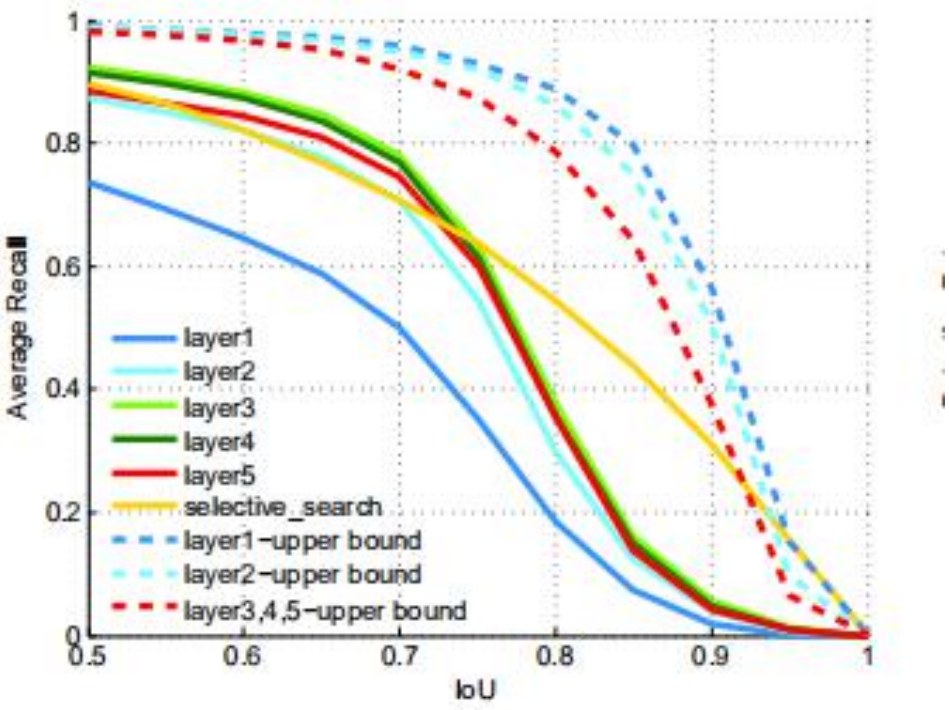
For the selection of the window sizes, we start with a

pool of windows Wall in different sizes and aspect ratios

W : {ω|ω ∈ Z^2, Z = [1..20]}.

for each window size, we compute its recall with different IoU thresholds and greedily pick one window size at a time that maximizes ∑α recall(IoU > α) over all the objects in the training set.

Using this procedure, 50 window sizes are selected for the sliding window procedure.



Multiple scales and Pooling

- For each scale, we resize the image such that min(w, h) = s where s ∈ {227, 300, 400, 600}

- Let f(x, y) be the specific channel of the feature map from a certain CNN layer and F(x, y) its integral image. Then, av_x0002_erage pooling avr of a box defined by the top left corner a = (ax, ay) and the bottom right corner b = (bx, by) is obtained as:

$$avr(a,b) = \frac{F(b_x,b_y) - F(a_x,b_y) - F(b_x,a_y) + F(a_x,a_y)}{(b_x - a_x)(b_y - a_y)}. \quad (1)$$

Other tricks

- Pyramid
- Classifier
- Non-maximal suppression

Evaluations

- We use two different evaluation metrics; the first is Detection Rate (or Recall) vs. Number of proposals. The second is IOU.

- If we use IoU of 0.5, thismeasure is too lose because a detector, for working properly, needs also good alignment with the object. Thus we evaluate our method for an overlap of 0.7 as well.

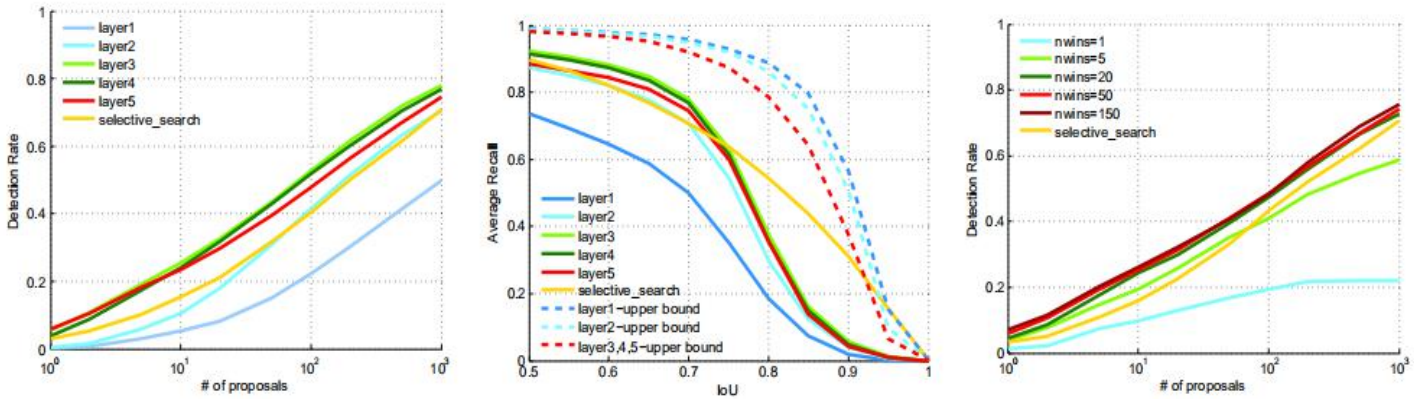


Figure 2: (**Left**) Recall versus number of proposals for IoU=0.7. (**Middle**) recall versus overlap for 1000 proposals for different layers. (**Right**) Recall versus number of proposals at IoU=0.7 on layer 5 for different number of window sizes. All are reported on the PASCAL VOC 2007 test set.

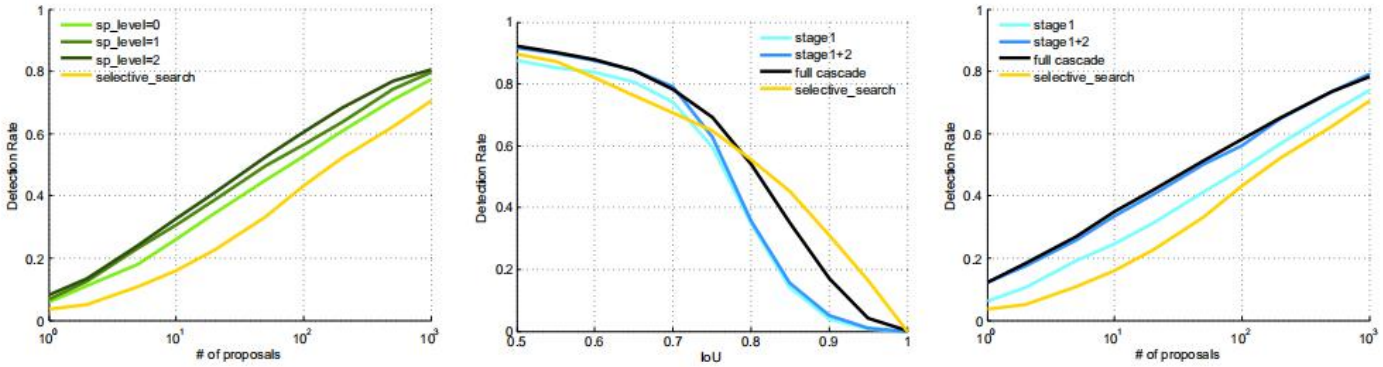


Figure 3: (**Left**) Recall versus number of proposals in IoU=0.7 for different spatial pyramid levels (**Middle**) Recall versus IoU for 1000 proposals for different stages of the cascade. (**Right**) Recall versus number of proposals in IoU=0.7 for the different stages of the cascade. All are reported on the PASCAL VOC 2007 test set.

Inverse cascade

- we start from a coarse spatial window resolution, and throughout the layers we select and spatially refine the window hypotheses until a reduced and spatially well localized set of hypotheses, we call our method coarse-to-fine inverse cascade.

- We found that a cascade with 3 layers is an optimal trade-off between complexity of the method and gain obtained from the cascading strategy.

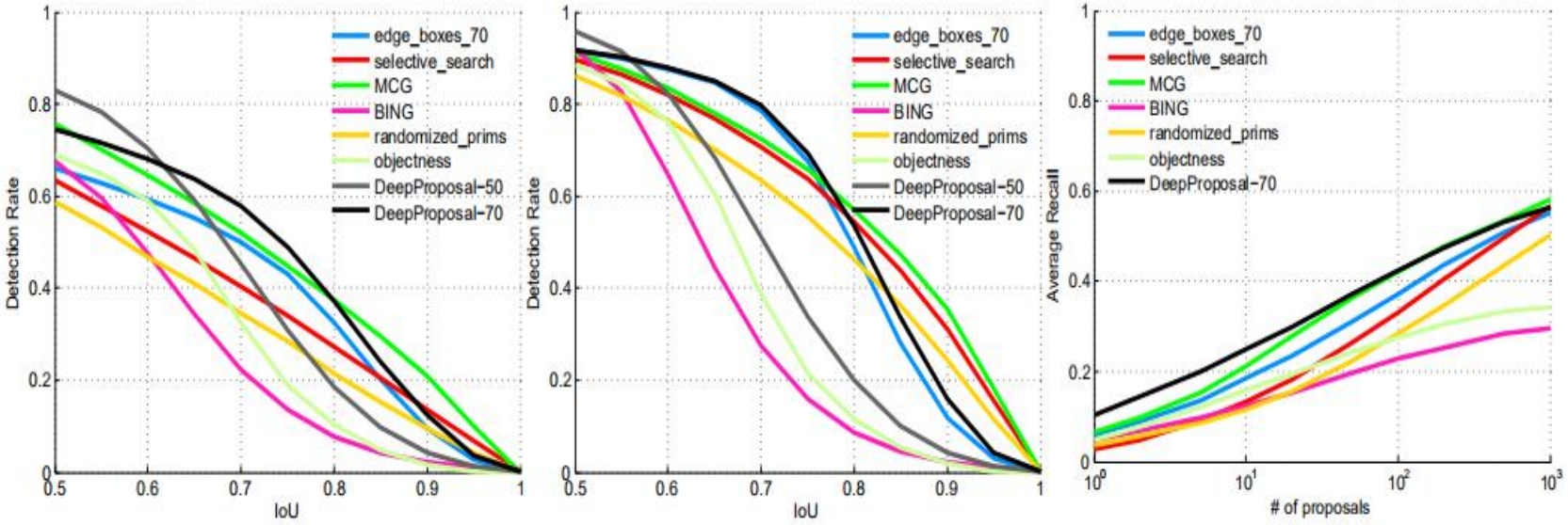
Stage of Inverse cascade

- Stage 1: Dense Sliding Window on Layer 5
- Stage 2: Re-scoring Selected Windows on Layer 3
- Stage 3: Local Refinement on Layer 2

Details of Stage 3

- The main objective of this stage is to refine the localization obtained from the previous stage of the cascade.
- We train a structured random forest on the second layer of the convolutional features to estimate contours similarly to Deep Contour
- A greedy iterative search tries to maximize the score of a proposal over different locations and aspect ratios using the scoring function used EdgeBoxes.

The end


Detection Rate
IoU
edge_boxes_70
selective_search
MCG
BING
randomized_prims
objectness
DeepProposal-50
DeepProposal-70
Detection Rate
IoU
edge_boxes_70
selective_search
MCG
BING
randomized_prims
objectness
DeepProposal-50
DeepProposal-70
Average Recall
# of proposals
edge_boxes_70
selective_search
MCG
BING
randomized_prims
objectness
DeepProposal-70