Cracking BING and Beyond

Qiyang Zhao
zhaoqy@buaa.edu.cn
Zhibin Liu
liuzhibin@nlsde.buaa.edu.cn
Baolin Yin
yin@nlsde.buaa.edu.cn

State Key Laboratory of
Software Development Environment,
Beihang University,
Beijing, China

Abstract

Objectness proposal is an emerging field which aims to reduce candidate object windows without missing the real ones. Under the evaluation framework of *DR-#WIN*, lots of methods report good performance in recent years, and the best is BING in CVPR 2014. BING provides good detection rates and surprisingly high efficiencies. But what can we benefit from it from the view of computer vision research?

In this paper, we show that the success of BING is rather in combinatorial geometry than in computer vision research. The secret lies in the Achilles' heel of the *DR-#WIN* evaluation framework: the 0.5-*INT-UNION* criterion. We proposed a method to construct a rather small set of windows to "cover" all legal rectangles. On images no larger than 512×512, supposing all object rectangles are not smaller than 16×16, nearly 19K windows are sufficient to cover all possible rectangles. The amount is far less than that of all sliding windows. It can be reduced further by exploiting the prior distribution of the locations and sizes of object rectangles in a greedy way. We also proposed a hybrid scheme blending both greedy and stochastic results. On the VOC2007 test set, it recalls 95.68% objects with 1000 proposal windows. The detection rates on the first ten windows are $13.99\% \sim 40.29\%$ higher than earlier methods in average.

1 Introduction

Sliding window is a popular strategy in recent object detection methods [4][5][6]. However in generic object detection tasks, there is no specific field knowledge, so a large number of potential windows need to be checked. In recent years, *generic objectness proposal* which aims to reduce the candidate windows in the preprocessing stage [1][2], has gained wide attention [7][8][10][16][17][12]. Up to now, the state-of-the-art performance belongs to OBN [2], SEL [10], CSVM [17] and BING [12].

The popular evaluation criterion for objectness proposal methods is detection-rate/windows-amount(*DR-#WIN*), where *DR* is the percentage of groundtruth objects covered by proposal windows. An object is considered covered by a window only if the strict PASCAL-overall criterion [13] is satisfied (the intersection of a proposal window and the object rectangle is not smaller than half of their union, or the *INT-UNION* ratio > 0.5 for short). It is clear that at least half of a good proposal window belong to the object rectangle. In that sense,

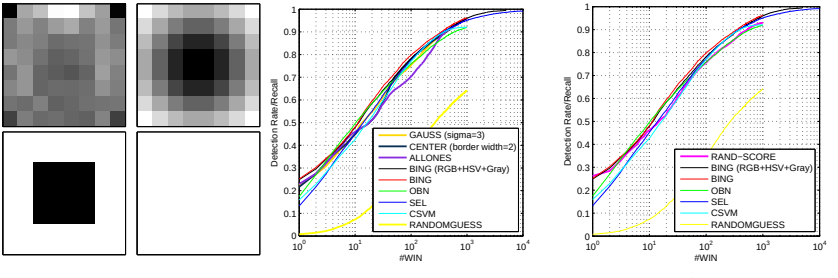
the criterion looks considerably plausible, thus has been widely used to evaluate both object detection methods and objectness proposal methods. In the following sections we call it "0.5-criterion" for short.

Under the *DR-#WIN* evaluation framework, the performance of recent objectness proposal methods is improved year by year, and all methods outperforms random guesses clearly. Amongst them, BING [12] in CVPR 2014, obtains the best performance on the VOC2007 test set. It recalls 96.2% objects with only 1,000 proposal windows. The more surprising is the method is totally a realtime one: only 0.003 seconds per image is needed on a popular PC. When the amount of proposal windows is increased to 5,000, *DR* is improved to 99.5%. It seems to be a substantial push to object detection researches.

However, with a series of experiments, we found it is not worthy of being too excited on BING. The success of BING lies mainly in combinatorial geometry and simple probabilistic models. Computer vision researchers can hardly get much enlightenment from the method. In order to get deep insight, we finished a combinatorial geometric analysis and proposed a method to construct a window set (we call it the *full cover set*) to cover all legal rectangles. The size of the *full cover set* is far less than the amount of all sliding windows. Furthermore, enlightened by the location/size hints adopted in saliency detection methods [11][3][14][15], we found it is feasible to reduce the size of *full cover set* further by exploiting the distributions of locations/sizes of object rectangles in images. A greedy scheme and a hybrid scheme were established to pursue it.

The paper is organized as follows: (1) introducing the noticeable and unnoticeable idea in BING, and cracking it with a series of experiments in Sec.2, (2) from the view of combinatorial geometry, presenting a method to construct a *full cover set*, and putting forward an upper bound on its size in Sec.3, (3) presenting a greedy scheme and a hybrid scheme to reduce the *full cover set* by exploiting the location/size hints of object rectangles in Sec.4, (4) showing experiment results on the effectiveness of these two schemes in Sec.5.

2 BING and Its Secret



(a) (b)

Figure 1: Experiments on the effectiveness of templates in BING: (a) four learned/hand-tuned templates and their performances. Top left: template learned by BING. Top right: Gaussian template. Bottom left: center-surrounded template. Bottom right: all-ones template; (b) performance of RAND-SCORE.

As in [1][2], the major motivation of BING claimed in [12] is, most objects have well defined closed boundaries in images and they exhibit different appearances from their surroundings. The authors argue that, after being resized to a fixed size (8×8 in [12]), almost all annotated

object rectangles share a common characteristics in gradients. This commonness is captured by a template W by learning from train images with a linear SVM. The subtle differences between diverse width/height configurations are also noticed and utilized in a re-weighting model. So the establishment of BING consists of two stages: calculating W in stage I, and learning the re-weighting model in stage II. Particularly, in order to improve the efficiency, BING uses several $0/1$ vectors to approximate W, and calculate the inner product of W and candidate windows by bitwise operations.

As suggested in [12], the template W from stage I should play the most important role in BING. To verify whether W weighs in here as expected, we designed several templates by hands to substitute W in BING. These templates include: (1) W_{gauss} of the complements of a gaussian filter, (2) W_{center} of zeros surrounded by ones on borders, and (3) $W_{allones}$ of ones on all entries. We still finish the training in stage II for each templates. These templates become less correlated to W in turn, but their *DR-#WIN* performances on VOC 2007 test set are very close, see Fig.1.a.

These experiments remind us that the templates, whether hand-tuned or learned, maybe do not have so strong significance as suggested in [12]. In order to get deeper insight, we discarded any templates and tried to directly assign the scores of stage I with random values. We call this method RAND-SCORE. Uniform random numbers in $[0,1]$ is adopted here.

We had anticipated the performance of RAND-SCORE should be close to random guesses, but surprisingly its performance is rather good on most random seeds, as shown in Fig.1.b. On some good seeds, we are even close to the performance of BING. Furthermore, its *DR* on the first window, 25.7%, is better than all earlier methods including BING.

The astonishing facts remind us, maybe the effect of templates in stage I of BING is not so noticeable. But what on earth makes BING performing so well? There are two key facts which are not mentioned particularly in [12] but hidden in their public code [1]: first, the sizes of proposal windows are doubled each time; second, the non-max suppression steps are chosen to be 0.25 relative to window sizes. Actually, these two facts coincide perfectly with our combinatorial geometric analysis in Sec.3. We would see that they are the most important aspects to ensure the performance of BING.

Besides this, the stage II of BING adjusts the preference of proposal windows according to different width-height configurations. It is indeed to exploit the distribution of the sizes of annotated object rectangles. This will be discussed in Sec.4.

3 A Combinatorial Geometric Analysis

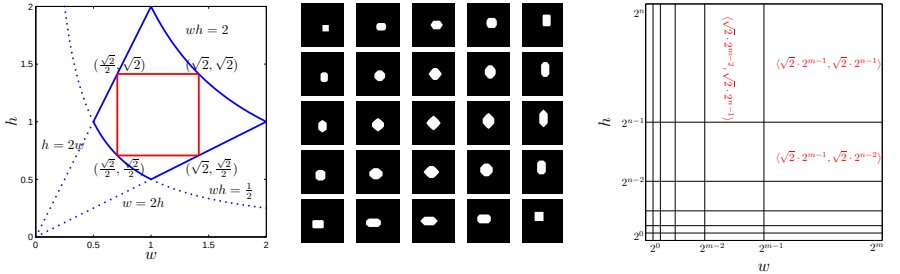
Consider a question: how many windows are required at least to "cover" all legal rectangles in a image? Here we say a proposal window covers a object rectangle if and only if the 0.5-criterion is satisfied. This is an atypical covering problem in combinatorial geometry [9], due to the varying sizes of covering rectangles we considered.

We solve the problem as follows. First we consider a unit window (both its width and height are 1) on an infinite plane. What sizes are feasible for rectangles to be covered by this unit window? For this we have

Lemma 3.1. *The feasible size $\langle w,h\rangle$ is in a region surrounded by two curves $w\cdot h=\frac{1}{2}$ and $w\cdot h=2$, and two lines $w-2h=0$ and $h-2w=0$.*

[1] http://mmcheng.net/bing/

Particularly, all sizes $\frac{\sqrt{2}}{2} < w,h < \sqrt{2}$ are feasible, as bounded in the red square in Fig.2.a. It is taken as the basis of our following analysis due to its regular shape.



(a) (b) (c)

Figure 2: Illustrations of the construction of the *full cover set*: (a) feasible scope (bounded by solid blue curves and lines) of $\langle w,h\rangle$ to get covered by a unit window; (b) feasible locations (in white) for 25 typical $\langle w,h\rangle$'s to be covered by a unit window. h changes from $\sqrt{2}/2$ to $\sqrt{2}$(top to bottom).w changes from $\sqrt{2}/2$ to $\sqrt{2}$(left to right); (c) the way to construct the *full cover set*. Texts in red are the sizes of proposal windows chosen to cover all rectangles sized in the corresponding 2D intervals.

Then for a feasible size configuration $\langle w,h\rangle$, what are the feasible locations of these $\langle w,h\rangle$-sized rectangles to get covered by a unit window? It is easy to see, when $w = h = \sqrt{2}/2$, all feasible locations of these rectangles make up a $\left(1-\sqrt{2}/2\right)$ square. The case for $w = h = \sqrt{2}$ is the same except that the square size changes to $\left(\sqrt{2}-1\right)$. For arbitrary $\langle w,h\rangle$, we have

Lemma 3.2. *For arbitrary $\frac{\sqrt{2}}{2} \le w,h \le \sqrt{2}$, the feasible locations of $\langle w,h\rangle$-sized rectangles make up a region which contains a $\left(1-\frac{\sqrt{2}}{2}\right)$ square. Furthermore, the borders of this square are parallel to the unit window.*

Its proof is straightforward but space-consuming, so we omit it here. We emphasize in Lemma 3.2 that the square borders are parallel to the unit window, so we can easily put these "cover squares" together to cover any rectangles, especially the whole image.

Based on Lemma 3.2, easily we have

Lemma 3.3. *For $2^i \le w \le 2^{i+1}, 2^j \le h \le 2^{j+1}$(here i,j are non-negative integers), the feasible locations to get covered by a rectangle of the width $2^i \cdot \sqrt{2}$ and height $2^j \cdot \sqrt{2}$ make up a region which contains a rectangle subregion of the width $\left(1-\frac{\sqrt{2}}{2}\right)\cdot 2^i \cdot \sqrt{2}$ and height $\left(1-\frac{\sqrt{2}}{2}\right)\cdot 2^j \cdot \sqrt{2}$. Again, their borders are parallel to the proposal rectangle.*

Then for an image of the width M and height N, how many windows of the width $2^i \cdot \sqrt{2}$ and height $2^j \cdot \sqrt{2}$ are needed to cover all $2^i \le w \le 2^{i+1}, 2^j \le h \le 2^{j+1}$ rectangle windows? Denote the amount with $s(i,j)$, then we have

Lemma 3.4. $s(i,j) = \lceil \frac{M-2^i}{\left(1-\frac{\sqrt{2}}{2}\right)\cdot 2^i\cdot\sqrt{2}} \rceil \cdot \lceil \frac{N-2^j}{\left(1-\frac{\sqrt{2}}{2}\right)\cdot 2^j\cdot\sqrt{2}} \rceil$

Its proof follows the simple scheme to put the minimum "cover squares" together to cover the image. Now we are ready to construct a *full cover set* to cover all legal rectangles. Without loss of generality, we suppose the image size is $M = 2^m, N = 2^n$. If the object rectangles' widths and heights start from 1, the amount of all windows in the *full cover set* is

$$\begin{aligned}
\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} s(i,j) &= \sum_{i=0}^{m-1}\sum_{j=0}^{n-1} \lceil \frac{2^m - 2^i}{\left(1-\frac{\sqrt{2}}{2}\right)\cdot 2^i \cdot \sqrt{2}} \rceil \cdot \lceil \frac{2^n - 2^j}{\left(1-\frac{\sqrt{2}}{2}\right)\cdot 2^j \cdot \sqrt{2}} \rceil && (1)\\
&= \sum_{i=1}^{m}\sum_{j=1}^{n} \lceil \frac{2^i - 1}{\sqrt{2}-1} \rceil \cdot \lceil \frac{2^j - 1}{\sqrt{2}-1} \rceil && (2)\\
&\le \sum_{i=1}^{m}\sum_{j=1}^{n} (\frac{2^i - 1}{\sqrt{2}-1} + 1) \cdot (\frac{2^j - 1}{\sqrt{2}-1} + 1) && (3)\\
&\le (\sqrt{2}+1)^2 \cdot 2^{m+n+2} + (\sqrt{2}+1)\cdot(n2^{m+1} + m2^{n+1}) + mn && (4)\\
&= O(MN) && (5)
\end{aligned}$$

Particularly, when $m = n = 9$, to say, both the image's width and height are 512, the amount $6,002,500$ is far less than that of all possible sliding windows.

Furthermore, there are restrictions on the object rectangle sizes in real applications usually. If all their widths/heights are restricted to be at least 2^k, then the amount of needed windows is

$$\begin{aligned}
\sum_{i=k}^{m-1}\sum_{j=k}^{n-1} s(i,j) &= \sum_{i=1}^{m-k}\sum_{j=1}^{n-k} \lceil \frac{2^i - 1}{\sqrt{2}-1} \rceil \cdot \lceil \frac{2^j - 1}{\sqrt{2}-1} \rceil && (6)\\
&\le \sum_{i=1}^{m-k}\sum_{j=1}^{n-k} (\frac{2^i - 1}{\sqrt{2}-1} + 1) \cdot (\frac{2^j - 1}{\sqrt{2}-1} + 1) && (7)\\
&\le (\sqrt{2}+1)^2 \cdot 2^{m+n-2k+2} && (8)\\
&\quad +(\sqrt{2}+1)\cdot((n-k)2^{m-k+1} + (m-k)2^{n-k+1}) + (m-k)(n-k) && (9)\\
&= O(2^{-2k}\cdot MN) && (10)
\end{aligned}$$

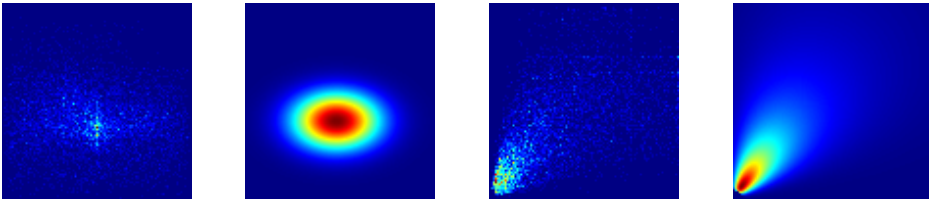
Particularly, when the widths/heights of all object rectangles are at least 16, the amount is $19,600$. While on the restriction of 32, we need only $4,225$ windows. Again, these amounts are both far less than that of all possible sliding windows. According to our analysis, the window amounts sufficient to satisfy all 0.5-criterions are far less than what people imagined before. We call it the Achilles' heel of the *DR-#WIN* evaluation framework.

Recall that in BING, the widths/heights of proposal windows are doubled each time, in the same way as in Lemma 3.1-3.4. Its non-max suppression step, 0.25 relative to the normalized size 8, is very close to the step $(1-\frac{\sqrt{2}}{2}) \approx 0.29$ in Lemma 3.2. These two settings meets our analysis well: they cut off many redundant windows which has no fresh object rectangles to cover. According to Lemma 3.2, increasing suppression steps would leave some object rectangles uncovered, while reducing suppression steps would increase the windows' amount unnecessarily. Combined with our experiments on templates in Sec.2, it can be concluded these settings are the most important aspects bringing success to BING.

4 A Greedy Scheme and Its Extension

In the above, we considered to cover all legal rectangles, implicitly supposing all the sizes and locations of object rectangles are uniformly distributed. However it is not the truth in real images. Many literatures on saliency detection have found that object rectangles of interest prefer close to image centers [11][3][14][15], and their sizes are usually not too large or too small. This is also pointed out in [2]: the locations of object rectangles are considered to be Gaussian distributed, while the widths and heights seem to be equal with high probabilities.

In other words, the amount obtained in Lemma 3.3 is an upper bound to the minimum cover set of any distributions of object rectangle sizes/locations, but this upper bound is too loose in most cases.



(a) (b) (c) (d)

Figure 3: Empirical distribution and fitted distribution of locations and sizes of object rectangle in VOC2007: (a) empirical distribution of locations; (b) estimated Gaussian distribution of locations.; (c) empirical distribution of sizes; (b) estimated Gamma-Gaussian distribution of sizes.

When knowing the distribution of object rectangles' sizes and locations, it is unnecessary to consider all sizes/locations. Only those possible locations/sizes will work. Furthermore, we should pay more attention to those "hot" locations/sizes. It would be a more "economical" way to construct the cover set than what we have done in Sec.3.

Different from the observations in [2], we found the height/width ratios are close to 1.4. And a compound density, the product of Gamma density and Gaussian density, is more suitable to fit the joint distribution of widths and heights. as shown in Fig.3. According to our observation, it is reasonable to consider the two VOC2007 datasets share a common underlying distribution of object rectangles' locations and sizes.

By exploiting the underlying distribution of object rectangles' locations and sizes, we designed a greedy scheme to produce an identical cover set for all images. In other words, these proposal windows are the same and completely independent to diverse images. It should be pointed out that in our scheme, all image sizes are resized to a fixed size first, and so are object rectangles' locations and sizes accordingly. In each round, we pick the "hottest" window according to the left object rectangles, and thereafter remove those object rectangles covered by the new window. The scheme is outlined in the Algorithm 1.

The greedy scheme is very successful in reducing the size of the *full cover set*, as to be shown in Sec.5. However when we try to use it in the "train-predict" way, its performance is much worse than expected, mainly due to the huge difference between low-probability sample spaces of the training set and test set. We proposed a hybrid scheme to address this. It is a combination of our greedy scheme and RAND-SCORE in Sec.2. We replace the windows in the greedy set with the results of RAND-SCORE with increasing probabilities from the exponential family. When the number of windows increases, the chance of replacement

gets larger, as shown in the Algorithm 2. Roughly speaking, we utilize the greedy scheme to address those "hot" windows, and leave the low probability space to RAND-SCORE. We found the blending parameters $\alpha = 0.0006, \beta = 2$ work well in our experiments.

Algorithm 1 Greedy scheme to produce an identical cover set for all images

Input: Image set $\{I_i\}$ and annotated object rectangle sets $\{O_i\}$; amount N of wanted proposal windows; minimum width/height 2^L and maximum width/height 2^H of annotated object rectangles

for $j \leftarrow 1$ to N **do**
  for all possible locations $\langle x, y \rangle$ **do**
    for all $\langle w, h \rangle$ in $\{\sqrt{2} \cdot 2^L, \sqrt{2} \cdot 2^{L+1}, \ldots, \sqrt{2} \cdot 2^{H-1}\}^2$ **do**
      if $\langle x, y, w, h \rangle$ has not been chosen **then**
        calculate the detection rate $DR(x, y, w, h)$ for the window $\langle x, y, w, h \rangle$;
        save all annotated object rectangles covered by $\langle x, y, w, h \rangle$ in a set $C_{x,y,w,h}$
      end if
    end for
  end for
  $\langle x_j, y_j, w_j, h_j \rangle \leftarrow \arg\max DR(x, y, w, h)$
  remove all elements in C_{x_j, y_j, w_j, h_j} from annotated rectangles
end for

Output: N proposal windows $\{\langle x_j, y_j, w_j, h_j \rangle\}$.

Algorithm 2 Hybrid scheme blending greedy results and RAND-SCORE results

Input: Greedy results $\{\langle x_j^G, y_j^G, w_j^G, h_j^G \rangle\}$, and RAND-SCORE results $\{\langle x_j^R, y_j^R, w_j^R, h_j^R \rangle\}$; amount N of wanted proposal windows; blending parameter α, β

for $j \leftarrow 1$ to N **do**
  generate a uniformly random value v in $[0, 1]$
  if $v < e^{-\alpha \cdot (j-1)^\beta}$ **then**
    $\langle x_j, y_j, w_j, h_j \rangle \leftarrow \langle x_j^G, y_j^G, w_j^G, h_j^G \rangle$
  else
    $\langle x_j, y_j, w_j, h_j \rangle \leftarrow \langle x_j^R, y_j^R, w_j^R, h_j^R \rangle$
  end if
end for

Output: N proposal windows $\{\langle x_j, y_j, w_j, h_j \rangle\}$.

5 Experiments

There are two aims in our experiments: (1) check if the greedy scheme is effective in reducing the size of the *full cover set*; (2) compare the hybrid scheme with BING and other methods. We adopt PASCAL VOC2007 which is designed for testing object detection methods. The dataset includes 2,501 training images and 4,952 test images. All annotated object rectangles are parallel to image borders, and belong to 20 categories.

The evaluation criterion is *DR-#WIN* as in other literatures. Four experiments are finished: (1) produce a cover set for the training set and collect its *DR*'s on the test set, (2)

produce a cover set for the test set and collect its *DR*'s on the training set, (3) produce a cover set for all images, then collect its *DR*'s on the training set and test set respectively, (4) collect the performance of our hybrid scheme on the test set. The results are shown in Fig.4 and Fig.5.

In the first three experiments, our greedy scheme performs considerably well. All *full cover sets* are reduced to about 1,000 windows, and the first windows have 0.3+ *DR*'s in all experiments. Nevertheless, the greedy scheme does not perform well on unexploited sets except for its superiority to other methods on the first few *DR*'s. The *DR*'s become worse gradually when window number increases, even than RAND-SCORE in Sec.2.

The reason is straightforward. In VOC2007, there are only 18,333 annotated object rectangles("difficult" objects are excluded as in [12]). Compared with the whole space of possible rectangles, these samples are too few to deliver sufficient information on the distribution of locations and sizes. However those spots of high probabilities are conveyed well both by the training set and test set, so we obtain high *DR*'s on first few windows. On the other hand, the window space of low probabilities are much larger than that of high probabilities, therefore the distributions of low probability spaces on training set and test set are completely different.

The performance of our hybrid scheme is much better. In most time, its *DR*'s are higher than OBN and CSVM, and close to SEL and BING. It recalls 95.68% objects with 1000 proposal windows. Especially, its *DR*'s are 13.99% $\sim$ 40.29% (relatively) higher than all other methods in average on the first ten windows.

Finally, there is another concern that maybe our scheme exploits the 0.5-criterion too much so most object rectangles are just ok to be recalled. An effective measure for this is the average best *INT-UNION* ratios (collect the best *INT-UNION* ratio for each object rectangle then average them). Our average best *INT-UNION* ratios are very close to BING: they are only 2.49% (relatively) lower than BING(RGB+HSV+Gray) on first 1000 proposal windows in average. It indicates the 0.5-criterion is not exploited in our scheme much more than in BING.

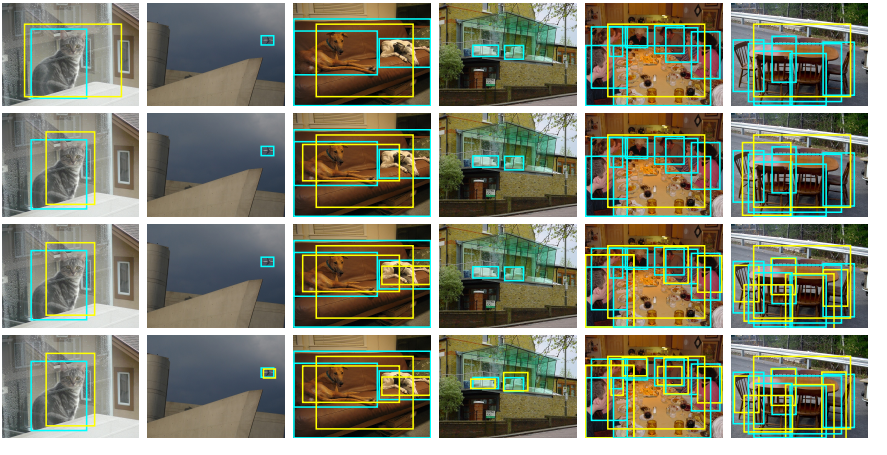


Figure 4: Qualitative results of our hybrid scheme on VOC2007 test set for increasing number of proposal windows. Each row shows the results obtained with 1, 10, 100 and 1000 windows (top to bottom). Groundtruth object rectangles are marked in cyan, and the best proposal windows are marked in yellow. See Fig.5 for quantitative results.

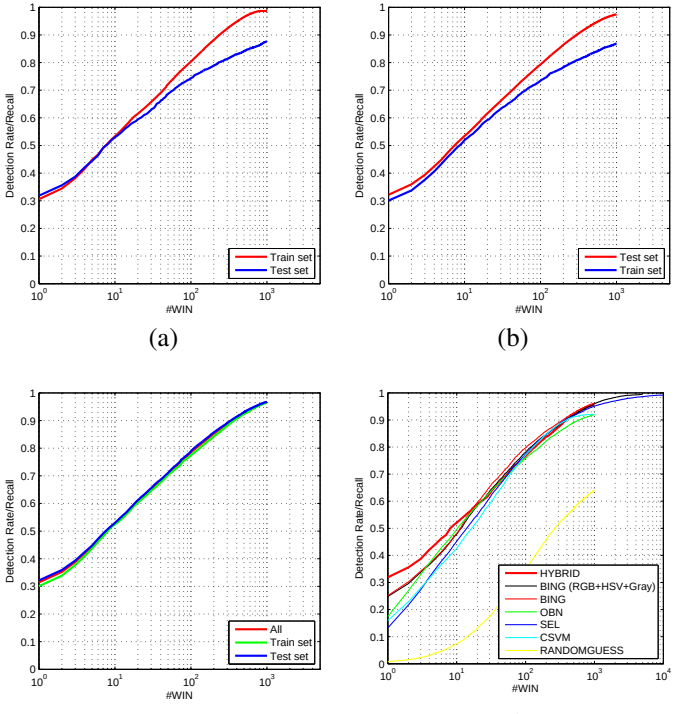


Figure 5: Performance of greedy scheme and hybrid scheme: (a) cover set of training images, and its performance on test images; (b) cover set of test images, and its performance on training images; (c) cover set of all images, and its performance on two sets respectively; (d) comparison of hybrid scheme with other methods.

6 Conclusion and Discussion

Under the evaluation framework of *DR-#WIN*, given sufficient samples conveying the full distribution of locations and sizes of object windows, our greedy scheme performs rather well. The hybrid scheme is also competitive in detection rates in the "train-predict" mode. And how about their time consumption? Nearly zero because the major computations are to resize proposal windows for specific images. But what can we benefit from it for object detection researches? We argue it needs a bigger picture to answer this question because it depends on whether the 0.5-criterion is actually effective and objective.

At last, we argue that if the 0.5-criterion is still adopted in evaluating objectness proposal methods, then the baseline should not be the dull random-guess method described in other literatures. The reasonable baseline should be RAND-SCORE or our hybrid scheme. Both of them bring more challenges to future researches.

Acknowledgement: This work is supported by the State Key Laboratory of Software Development Environment (No.SKLSDE-2013ZX-29, SKLSDE-2013ZX-34).

References

[1] Alexe B., Deselaers T., and Ferrari V. What is an object? In *Proc. CVPR*, 2010.

[2] Alexe B., Deselaers T., and Ferrari V. Measuring the objectness of image windows. *TPAMI*, 34(11)): 2189-2202, 2012.

[3] Gao D., Mahadevan V., and Vasconcelos N. On the plausibility of the discriminant center-surround hypothesis for visual saliency. *J. of Vision*, 8(7), 2008.

[4] Dalal N. and Triggs B. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005.

[5] Felzenszwalb P. F., Girshick R. B., McAllester D., and Ramanan D. Object detection with discriminatively trained partbased models. *TPAMI*, 32(9): 1627-1645, 2010.

[6] Lampert C. H., Blaschko M. B., and Hofmann T. Beyond sliding windows: Object localization by efficient subwindow search. In *Proc. CVPR*, 2008.

[7] Endres I. and Hoiem D. Category independent object proposals. In *Proc. ECCV*, 2010.

[8] Endres I. and Hoiem D. Category-independent object proposals with diverse ranking. *TPAMI*, 36(2): 222-234, 2014.

[9] Pach J. and Agarwal P. *Combinatorial Geometry*. John wiley & Sons, ISBN: 9780471588900, 1995.

[10] Uijlings J., van de Sande K., Gevers T., and Smeulders A. Selective search for object recognition. *IJCV*, 104(2): 154-171, 2013.

[11] Itti L., Koch C., and Niebur E. A model of saliency-based visual attention for rapid scene analysis. *TPAMI*, 20(11): 1254-1259, 1998.

[12] Cheng M. M., Zhang Z. M., Lin W. Y., and Torr P. Bing: Binarized normed gradients for objectness estimation at 300fps. In *Proc. CVPR*, 2014.

[13] Everingham M., Van Gool L., Williams C. K. I., Winn J., and Zisserman A. The pascal visual object classes (voc) challenge. *IJCV*, 88(2): 303-338, 2010.

[14] Liu T., Yuan Z., Sun J., Wang J., Zheng N., Tang X., and Shum H. Y. Learning to detect a salient object. *TPAMI*, 33(2): 353-367, 2011.

[15] Mahadevan V. and Vasconcelos N. Biologically inspired object tracking using center-surround saliency mechanisms. *TPAMI*, 35(3): 541-554, 2013.

[16] van de Sande K., Uijlings J., Gevers T., and Smeulders A. Segmentation as selective search for object recognition. In *Proc. ICCV*, 2011.

[17] Zhang Z., Warrell J., and Torr P. H. Proposal generation for object detection using cascaded ranking svms. In *Proc. CVPR*, 2011.