Notes of “Object-Proposal Evaluation Protocol is ‘Gameable’ ”

Jihang Li

February 4, 2020

Contents

Abstract

The choice of using a partially annotated dataset for evaluation of object proposals is problematic. To alleviate this problem:

1. Introduce a nearly-fully annotated version of PASCAL VOC dataset, which serves as test-bed to check if object proposal techniques are over-fitting to a particular list of categories.
2. Perform an exhaustive evaluation of object proposal methods on the introduced nearly-fully annotated PASCAL dataset and perform cross-dataset generalization experiments.
3. Introduce a diagnostic experiment to detect the **bias capacity** in an object proposal algorithm.

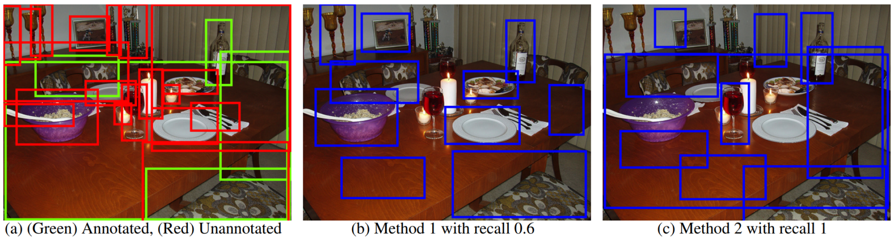
1 Introduction

Despite the different goals between object proposal and detection, there exists only a single evaluation protocol:

1. Generate proposals on a dataset.
2. Measure the performance of the generated proposals, typically using recall.

Contributions:

- Report the "gameability" of the current object proposal evaluation protocol.
- Present a simple technique for producing state-of-art object proposals.
- Propose three ways of improving the current evaluation protocol to measure the category independence of object proposals:
    1. Evaluation on fully annotated datasets.
    2. Cross-dataset evaluation on densely annotated datasets.
    3. A **new evaluation** metric that quantifies the **bias capacity** of proposal generators.
- Thoroughly evaluate existing proposal methods on this near-fully and two densely annotated datasets.


(a) (Green) Annotated, (Red) Unannotated

(b) Method 1 with recall 0.6

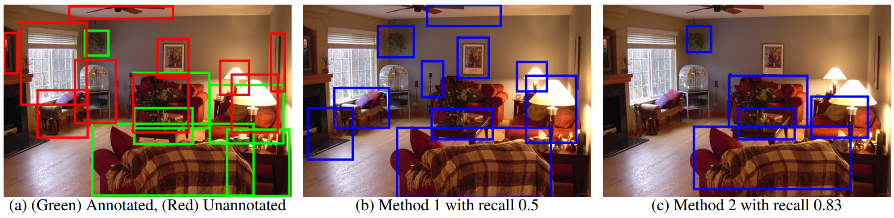
(c) Method 2 with recall 1

Figure 1: (a) shows PASCAL annotations natively present in the dataset in green. Other objects that are not annotated but present in the image are shown in red; (b) shows Method 1 and (c) shows Method 2. Method 1 visually seems to recall more categories such as plates, glasses, etcthat Method 2 missed. Despite that, the computed recall for Method 2 is higher because it recalled all instances of PASCAL categories that were present in the ground truth. Note that the number of proposals generated by both methods is equal in this figure.

2 Related Work

Evaluating Proposals:

- Hosang *et al.* focus on evaluation of object proposal algorithms, in particular the stability of such algorithms on parameter changes and image perturbations.



(a) (Green) Annotated, (Red) Unannotated

(b) Method 1 with recall 0.5

(c) Method 2 with recall 0.83

Figure 2: Figure 2: (a) shows PASCAL annotations natively present in the dataset in green. Other objects that are not annotated but present in the image are shown in red; (b) shows Method 1 and (c) shows Method 2. Method 1 visually seems to recall more categories such as lamps, picture, etc. that Method 2 missed. Clearly the recall for Method 1 should be higher. However, the calculated

- Hosang *et al.* presents an analysis of various proposal methods regarding proposal repeatability, ground truth annotation recall, and their impact on detection performance. They also introduced a new evaluation metric: **Average Recall**.
- Pont-Tuset *et al.* analyzes the state-of-the-art methods in segment-based object proposals, focusing on the challenges faced when going from PASCAL VOC to MS COCO. Also analyzes how aligned the proposal methods are with the **bias** observed in MS COCO towards small objects and the center of the image and propose a method to boost their performance.

3 Evaluating Object Proposals

Widely used evaluation metrics:

- **Recall @ IOU Threshold** t: For each ground-truth instance, this metric checks whether the "best" proposal from list $has IOU greater than a threshold$. If so, this ground truth instance is considered "detected" or "recalled". Then average recall is measured over all the ground truth instance:

  $$\text{Recall } @t = \frac{1}{|G|}\sum_{g_i \in G} I[\max_{l_j \in L} \text{IOU}(g_i, l_j) > t], \tag{2}$$

  where $I[\cdot]$ is an indicator function for the logical preposition in the argument. Object proposals are evaluated using this metric in two ways:

  - plotting Recall-*vs.*-#proposals by fixing t
  - plotting Recall-*vs.*-t by fixing the #proposals in L

- **Area Under the recall Curve (AUC)**: AUC summarized the area under the Recall-*vs.*-#proposals plot for different values of t in a single plot. This metric measures:

  - AUC-*vs.*-#proposals
  - AUC-*vs.*-t by varying #proposals in L

- **Volume Under Surface (VUS)**: Measures the average recall by linearly varying t and varying the #proposal in L on either linear or log scale. Thus it merges both kinds of AUC plots into one.
- **Average Best Overlap (ABO)**: This metric eliminates the needs for a threshold. The overlap between each ground truth annotation $g_i \in G$ and the 'best' object hypotheses in L:

$$\mathrm{ABO} = \frac{1}{|G|} \sum_{g_i \in G} \max_{l_j \in L} \mathrm{IOU}(g_i, l_j) \tag{3}$$

  ABO is typically is calculated on a per class basis. Mean Average Best Overlap (MABO) is defined as the mean ABO over all classes.
- **Average Recall (AR)**: AR-*vs.*-#proposals (for IOU between 0.5 to 1) in L is plotted. AR also summarizes proposal performance across different values of t. AR was shown to correlate with ultimate detection performance better than other metrics.

4 A Thought Experiment: How to Game the Evaluation Protocol

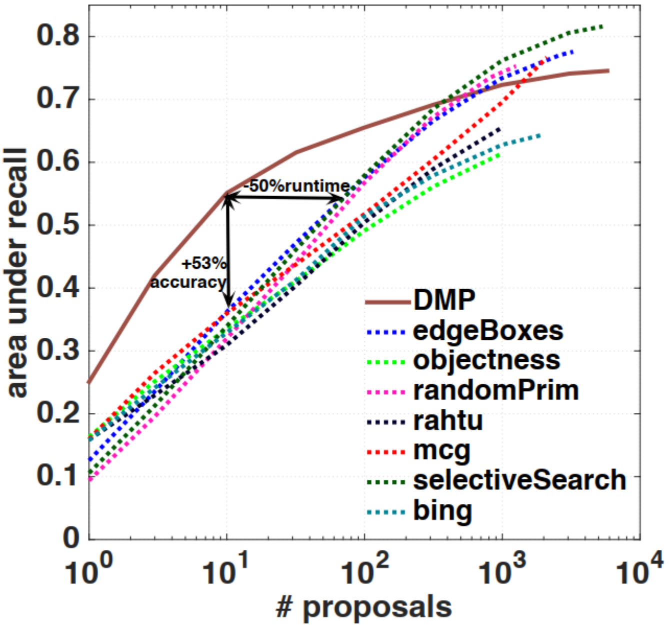


Figure 3: Performance of different object proposal methods (dashed lines) and our proposed 'fraudulent' method (DMP) on the PASCAL VOC 2010 dataset. We can see that DMP significantly outperforms all other proposal generators.

6 Bias Inspection

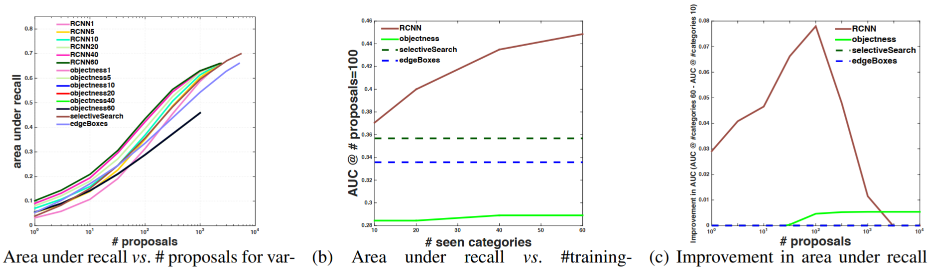
Datasets can be unbalanced. Some categories can be more frequent than others while others can be hard to detect (due to choices made in dataset collection). These issues need to be resolved for perfectly unbiased evaluation. However, generating unbiased datasets is an expensive and time-consuming process. Hence, to detect the bias without getting unbiased datasets, we need a method which can measure performance of proposal methods in a way that category specific biases can be accounted for and the extent or the capacity of this bias can be measured.

6.1 Assessing Bias Capacity

One way of measuring the **bias capacity** in a proposal method to plot the **performance vs. the number of 'seen' categories** while evaluating on some held-out set. A method that involves little or no training will be a flat curve on this plot.

Biased methods such as DMPs will get better and better as more categories are seen in training. Thus, this analysis can help us find biased or 'gamebility-prone' methods like DMPs that are/can be tuned to specific classes.

To the best of our knowledge, no previous work has attempted to measure bias capacity by varying the number of 'object' categories seen at training time.



(a) Area under recall *vs.* # proposals for various #seen categories

(b) Area under recall *vs.* #training-categories.

(c) Improvement in area under recall from #seen categories =10 to 60 *vs.* # proposals.

Figure 6: Performance of RCNN and other proposal generators vs number of object categories used for training. We can see that RCNN has the most 'bias capacity' while the performance of other methods is nearly (or absolutely) constant.

The key observation to make here (Figure 6) is that with even a modest increase in 'seen' categories with the same amount of increased training data, performance improvement of RCNN is significantly more than objectness.