Object Proposals in Computer Vision

Neelima Chavali

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Dhruv Batra, Chair
Devi Parikh
Lynn Abbott

22nd July, 2015
Blacksburg, Virginia

Keywords: Object proposals, evaluation, computer vision

Object Proposals in Computer Vision

Neelima Chavali

ABSTRACT

Object recognition is a central problem in computer vision which deals with both localizing and identifying objects in images. Object proposals have recently become an important part of the object recognition process. Object proposals are algorithms used for localizing objects in images. This thesis is a study in object proposals and is composed of three parts. First, we present a new data-driven approach for generating object proposals. Second, we release a MATLAB library which can be used to generate object proposals using all the existing algorithms. The library can also be used for evaluating object proposals using the three most commonly used metrics. Finally, we identify previously unnoticed bias in the existing protocol for evaluating object proposals and propose ways to alleviate this bias.

To my family and teachers.

Acknowledgements

The past 2.5 of years spent towards my graduate studies have been a short but wonderful journey back into academia, and I would like thank some of the many people who have contributed to making this journey very gratifying.

First of all, I would like to thank my adviser Dr. Dhruv Batra for his wonderful guidance and support through out this time. I have cherished every project he has involved me in. I am grateful to have had access to the computational infrastructure he has built for the lab. I thoroughly enjoyed his introductory and advanced courses in Machine Learning.

I would like to thank Dr. Devi Parikh for being an amazing adviser–figure. I thoroughly enjoyed interacting with her during the course "Advanced Topics in Computer Vision". I had a great time being a Teaching Assistant to her course "Computer Vision" and enjoyed every aspect of that job. And finally, it was a lot of fun working with her and Abhijit Sarkar on the project "Sound of an Image".

I would like to thank Dr. Lynn Abbott for being a member of my thesis committee and for the helpful discussions and suggestions.

I had a great time working with Harsh, Clint and Abdullah on CloudCV, with Akrit on ILSVRC 2014 detection challenge. A major chunk of this thesis was a joint work with Harsh and Aroma. Special thanks to both of them, it was a pleasure and a lot of fun working with them.

Life in the CVMLP lab was great fun and an education in itself due to the many amazing people. I would like to thank Stan for being a very affectionate and helpful friend. I dearly miss the amazing varieties of cookies and breads he baked every Monday. I would like to thank Prakriti, Qing, Peng, Qi, Aishwarya, Yash, Micheal, Xiao, Shrenik, Rama, Faruk, Senthil and several others for sharing many great moments, and helping me out on several occasions. I would like to thank my friends outside my lab: Sneha, Latha, Indira, Madhurima, Lisa, Dhiraj, Sriram, Kshitija and Mandar for making my stay in Blacksburg enjoyable.

I would like to thank Lakshmi Bala ma'am, Vinay Bade and Charles Reindorf, without whose recommendation letters my graduate school would not have happened. I would like to thank my friends from Knoxville – Anima, Ranjeet, Paul, Imelda and Duddu, for the wonderful

weekends in my first semester. I would also like to thank the warm and study–friendly environment provided by Mill mountain coffee shop in Blacksburg and Muddy's coffee shop in San Francisco from where I did a major chunk of my work towards this thesis and other projects.

I would like to express my heartfelt gratitude to my family. I would like to thank my parents – Chavali Padmavathi and C V S Sastry, my brother Gautham, my sister–in–law Ranjitha, my in–laws – Ghatty Kanyakumari and G D S Prasad, Prakasam pedananna, Sharada aamma and Aruna vadina for their constant encouragement, support and affection. Lastly, I would like to thank my husband Pavan Ghatty for being an amazing influence on me. This thesis would not have been possible without his support and company.

This work was partially supported by the National Science Foundation under grants IIS-1353694 and IIS-1350553, the Army Research Office YIP Award W911NF-14-1-0180, and the Office of Naval Research grant N00014-14-1-0679. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government or any sponsor.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"*What does it mean to see? The plain man's answer (and Aristotle's too) would be, to know what is where by looking*" – David Marr [1]

1.1 Teaching computers to see

One of the main goals of research in Artificial Intelligence is to develop computer systems that exhibit intelligent behavior [2]. Intelligent behavior can mean speech recognition, visual recognition, natural language processing, motor control, etc., to name a few. Computer vision is the branch of AI which is concerned with visual recognition, i.e., the aim of computer vision is to develop computer systems that can *see*. It aims to teach computers to *perceive* and *interpret* **what** is present in the world and **where**, through media which capture visual information like images and videos.

The field of computer vision was initially conceived as a summer undergraduate project [3] in 1966. Notwithstanding the seemingly simple definition of 'seeing', it has proved to be a tough problem to solve. Going beyond perception and interpretation of visual data, research in computer vision now encompasses the following areas:

- Computing properties of the 3D world from visual data (measurement)
- Algorithms and representations to allow a machine to **recognize objects**, people, scenes, and activities (perception and interpretation)
- Algorithms to mine, search, and interact with visual data (search and organization)

In this thesis, we focus on algorithms called *Object Proposals* which aid in **recognizing objects** in images. Object recognition involves both localizing and identifying objects present

in an image. Object proposals are algorithms which are used for localizing objects in the object recognition process.

1.2 Contributions of this thesis

In this thesis, our contributions are threefold:

1. We present a new data–driven approach for generating object proposals
2. We release a easy–to–use MATLAB library which can be used for:
    - generating object proposals using all the existing approaches
    - evaluating object proposals using the commonly used evaluation metrics
3. We identify biases in object proposal evaluation protocol and propose ways to fix alleviate this bias

1.3 Outline

In chapter 2, we provide a review of related works in object proposal literature and the protocols used in evaluating these algorithms.

In chapter 3, we discuss a new data driven approach for generating object proposals.

In chapter 4, we present our Object Proposal library and discuss its usage.

In chapter 5, we hypothesize that the current evaluation protocol for object proposals is '*game–able*'. We conduct a thorough experimental analysis to support our hypothesis and propose ways to alleviate this problem.

In chapter 6, we conclude the thesis work and propose some future directions.

1.4 Related publications

Ideas described in chapter 5 appear in the following arXiv publication:

Object-Proposal Evaluation Protocol is 'Gameable'.
Neelima Chavali, Harsh Agrawal, Aroma Mahendru, Dhruv Batra.
arXiv:1505.05836, 2015.

Chapter 2

Object Proposals: Background

'*Think outside the (sliding) box*'

2.1 Object detection

Object detection is concerned with coming up with algorithms which answer the question: "where are the instances of a particular object class in the image (if any)?" At the most abstract level, object detection can be expressed as a binary classification problem for each object class. It is typically solved as a 2 stage process.

1. **Training a binary object classifier**
   - Obtain positive and negative training examples for the given object class
   - Define features and train a binary classifier
2. **Detect objects in images**
   - Given an image, generate sub–windows from an image using either a **sliding window approach** or **object proposal approach**.
   - The score of the classifier at each of the sub-windows is an indication of the presence of the object within that sub–window of the image.

2.2 Sliding window approach

This approach employs brute force and exhaustively looks at every possible sub–window in an image. An $n * n$ image has $O(n^4)$ sub–windows. Even a low resolution image with

$300 * 300$ pixels has close to a billion sub–windows. Up until three years ago, this approach was dominantly used in the object detection pipelines.

2.3 Object proposal approach to object detection

In the last few years, the Computer Vision community has witnessed the emergence of a new class of techniques called *Object Proposal* algorithms [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14].

Object proposals are a set of candidate regions or sub–windows in an image that may potentially contain an object.

Object proposal algorithms have now replaced the sliding window approach in the object detection pipeline [15, 16, 17, 18, 19].For example, *all top-performing entries* [20, 21, 22, 23] in the ImageNet Detection Challenge 2014 [24] used object proposals. They are preferred over the previously–used sliding window paradigm because of their computational efficiency. Objects present in an image may vary in location, size, and aspect ratio, and performing an exhaustive search over such a high dimensional space is not efficient. Today we are facing image–deluge both at industry level for example in search engines, and at personal level for example massive personal albums in mobiles and laptops. This necessitates more efficient design patterns than the sliding window pattern. The object proposals paradigm bridges this necessity by focussing the computational effort on a small number of candidate windows (order of 1000s of windows).

Types of Object Proposals: Object proposals can be broadly categorized into two categories:

- **Window scoring**:
  In these methods, the space of all possible windows in an image is sampled to get a subset of the windows (*e.g.*, via sliding window). These windows are then scored for the presence of an object based on the image features from the windows. The algorithms that fall under this category are [25, 8, 4, 13].

  Objectness [25]: In this approach, windows are scored based on a combination of multiple cues such as saliency, colour contrast, edge density, location and size statistics, and how much such windows overlap with superpixel segments

  Rahtu [8]: In this approach, windows are scored similar to the Objectness approach but with more cues. They also use a different sampling strategy based on a combining super–pixels.

  Edge Boxes [4]: In this approach, they make the observation that the number of edges fully enclosed in a bounding box is indicative of the presence of an object in a window. They propose a scoring function to measure this quantity. They propose bounding boxes by empirically determining the parameters for sampling windows from

the space of all possible windows (based on the detector IOU and the IOU between neighboring samples, and NMS) and then score them.

Bing [13]: In this approach, they introduce a new feature called Binarized Norm of Gradient for image windows, SVMs are trained using this feature on object/non-object windows. New windows are scored using the SVM ouput.

- **Segment based**:
  These algorithms involve over-segmenting an image and merging the segments using some strategy. While the output of these algorithms are segments in images, they are usually converted into bounding boxes and fed to the detection pipelines. The following methods are segment based:
  CPMC [26]: In this approach, the authors generate a set of overlapping segments. Each proposal segment is the solution of a binary segmentation problem, initialised with diverse seeds. Up to 10^4 segments are generated per image, which are subsequently ranked by objectness using a trained regressor.

  Selective Search [11]: In this approach, the authors carefully engineer features and score functions that greedily merge low-level super–pixels. The authors obtain state of the art object detections on Pascal VOC and ILSVRC2011.

  Rantalankila [10]: In this approach, the authors combine Selective Search and CPMC. Starting from low-level super–pixels they propose a merging strategy, similar to Selective Search but using different features. These merged segments are then used as seeds for a CPMC-like process to generate larger segments

  Multiscale Combinatorial Grouping [6]: In this approach, the authors propose an improved multi-scale hierarchical segmentation, a new strategy to generate proposals by merging up to 4 segments, and (similar to CPMC) a new ranking procedure to select the final detection proposals.

  Geodesic Object Proposals [14]: In this approach, the authors start from an over-segmentation of the image. Classifiers are used to place seeds for a geodesic distance transform. Level sets of each of the distance transforms dene the object/background segmentations that are the proposals.

  Rigor [12]: In this approach, the authors propose an improved variant of CPMC that speeds computation considerably by re-using computation across multiple graph-cut problems and using the fast edge detectors from.

  Endres [27]: In this approach, the authors build a hierarchical segmentation from occlusion boundaries and solve graph cuts with different seeds and parameters to generate segments. The proposals are ranked based on a wide range of cues and in a way that encourages diversity.

Random Prim [9]: In this approach, the authors use similar features as Selective Search, but introduce a randomized super–pixel merging process in which all probabilities have been learned. Speed is substantially improved.

2.4 Evaluation of Object proposals

The following factors are involved in evaluating object proposals:

1. **Dataset**: The most commonly used dataset is the the PASCAL VOC [28] detection test set. This dataset contains 4952 images in which all the occurrences of 20 object classes are annotated in the form of a bounding box (or a sub–window). Note that this is a *partially annotated* dataset as only the 20 PASCAL category instances are annotated. Recently analyses have been shown on ImageNet [29], which has more categories annotated than PASCAL, but is still a partially annotated dataset.

2. **Evaluation Metric**: The metrics used for evaluating object proposals are all typically functions of intersection over union (IOU) (or Jaccard Index) between generated proposals and ground-truth annotations. For two boxes/regions b_i and b_j of an image, IOU is defined as:

$$\text{IOU}(b_i, b_j) = \frac{area(b_i \cap b_j)}{area(b_i \cup b_j)} \tag{2.1}$$

   The following metrics are commonly used:

   • **Recall @ IOU Threshold** t: Consider a ground-truth instance $g_i \subset G$, where G is the set of all the ground truth object annotations. This metric calculates the IOU of g_i with proposals from set $l_j \subset L$ of all proposals for the image of g_i. Then it checks if the maximum value of the calculated IOUs is greater than a threshold t. If so, this ground truth instance is considered 'detected' or 'recalled'. Then $Recall@t$ is measured as the average number of recalled ground truth annotations:

$$\text{Recall} \,@\, t = \frac{1}{|G|} \sum_{g_i \in G} \text{I}\,[\max_{l_j \in L} \text{IOU}(g_i, l_j) > t] \tag{2.2}$$

   Here $I[.]$ is an indicator function for the logical preposition in the arguement. 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 summarizes the area under the Recall-vs-#proposals plot for different values of t in a single plot. This metric measures AUC-vs-#proposals. It is also plotted by varying #proposals in L and plotting AUC-vs-t.

• **Volume Under Surface (VUS)**: This measures the average recall by linearly varying t and varying the #proposals 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 need for a threshold. We first calculate the overlap between each ground truth annotation $g_i \in G$, and the 'best' object hypotheses in L. ABO is calculated as the average:

$$\text{ABO} = \frac{1}{|G|} \sum_{g_i \in G} \max_{l_j \in L} \text{IOU}(g_i, l_j) \tag{2.3}$$

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)**: This metric was recently introduced in [30]. Here, average recall (for IOU between 0.5 to 1)-vs-#proposals 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.

Table 2.1 provides an overview of all the object proposal algorithms that fall under the scope of this paper and the evaluation protocol used by them.

While a variety of approaches have been proposed for generating object proposals, there has been limited analysis and evaluation of these approaches or evaluation protocols. Hosang *et al*. [30] focus on evaluation of object proposal algorithms, in particular the stability of such algorithms on parameter changes and image perturbations. Their work shows that existing proposal algorithms generalize well to non-PASCAL categories. They show this by performing evaluation on ImageNet 200 category detection dataset. They also introduced a new evaluation metric (Average Recall). Their argument for a new metric is the need for a better localization between generated proposals and ground truth. While this is a valid and significant concern, it is orthogonal to the 'gameability'(vulnerability to manipulation) and bias in evaluation protocol, which to the best of our knowledge has not been previously addressed.

While the scope of this paper is limited to bounding box proposals in RGB images, the central thesis of the paper (*i.e.*, gameability of the evaluation protocol) is broadly applicable to other settings.

2.5 Beyond RGB proposals:

We end this chapter by noting that beyond the algorithms listed here, a wide variety of algorithms fall under the umbrella of 'object proposals'. The need for annotations on large image datasets has motivated the work on unsupervised [31] and weakly supervised object discovery [32] with object proposals. The need for object recognition in videos has motivated

work on spatio-temporal object proposals [33, 34]. Another direction of work [21, 35, 36] explored using RGB-D cuboid proposals in an object detection and semantic segmentation in RGB-D images.

Method	Code Source	Approach	Learning Involved	Metric	Datasets
*objectness**	Source code from [37]	Window scoring	Supervised, train on 6 PASCAL classes and their own custom dataset of 50 images	Recall @ t $\geq$ 0.5 vs # proposals	PASCAL VOC 07 test set, test on unseen 16 PASCAL classes
*selectiveSearch**	Source code from [38]	Segment based	No	Recall @ t $\geq$ 0.5 vs # proposals, MABO, per class ABO	PASCAL VOC 2007 test set, PASCAL VOC 2012 train val set
*rahtu**	Source code [39]	Window Scoring	Yes, two stages. Learning of generic bounding box prior on PASCAL VOC 2007 train set, weights for feature combination learnt on the dataset released with [37]	Recall @ t $\geq$ various IoU thresholds and # proposals, AUC	PASCAL VOC 2007 test set
*randomPrim**	Source code from [40]	Segment based	Yes supervised, train on 6 PASCAL categories	Recall @ t $\geq$ various IOU thresholds using 10k and 1k proposals	Pascal VOC 2007 test set/2012 train–val set on 14 categories not used in training
*mcg**	Source code from [41]	Segment based	Yes	NA, only segments were evaluated	NA
*edgeBoxes**	Source code from [42]	Window scoring	No	AUC, Recall @ t $\geq$ various IOU thresholds and # proposals, Recall vs IoU	PASCAL VOC 2007 test set
*bing**	Source code from [43]	Window scoring	Supervised, on PASCAL VOC 2007 train set, 20 object classes/6 object classes	Recall @ t $\geq$ 0.5 vs # proposals	PASCAL VOC 2007 detection complete test set/14 unseen object categories
rantalankila	Source code [44]	Segment based	Yes	NA, only segments are evaluated	NA
Geodesic	Source code from [45]	Segment based	Yes, for seed placement and mask construction on PASCAL VOC 2012 Segmentation training set	VUS at 10k and 2k windows, Recall vs IoU threshold, Recall vs proposals	PASCAL 2012 detection validation set
Rigor	Source code from [46]	Segment based	Yes, pairwise potentials between super pixels learned on BSDS-500 boundary detection dataset	NA, only segments were evaluated	NA
endres	Source code from [47]	Segment based	Yes	NA, only segments are evaluated	NA

Table 2.1: Properties of existing bounding box approaches. * indicates the methods which have studied in Chapter 5.

Chapter 3

Non parametric bounding box transfer

In this chapter we present a novel non–parametric, data–driven approach for generating object proposals. With the emergence of large image datasets such as ImageNet[48], a new family of non parametric data-driven methods have demonstrated impressive performance in several areas of computer vision [49, 50]. We attempt a similar approach for object proposals. Our approach is neither window–scoring nor, segmentation based, instead it is data–driven. We generate object proposals by transferring over the annotated bounding boxes from similar images in a large dataset. Our approach is similar in spirit to that of [50]; the key difference being that they attempt to copy over the entire image labeling, which is unlikely to be very accurate even with large datasets, while we copy over bounding box proposals which is more likely to succeed especially since these proposals simply need to be handed to a categorization algorithm downstream in the pipeline.

3.1 Approach

Our system generates object proposals as follows (depicted in Fig. 3.1):

- **Nearest neighbor retrieval**: Given a query image, finds its k nearest neighbors from a large database containing images annotated with bounding boxes. The nearest neighbor calculation is carried out by picking a feature space for image representation. We experiment with the DeCAF and GIST feature spaces. Fig. 3.2 gives a qualitative representation of 9 nearest neighbors in the DeCAF feature space for a given query image. More qualitative results can be found here.
- **Transfer labels**: The bounding boxes of the neighbors are then transferred to the query image. We explore two ways of transferring the bounding boxes:

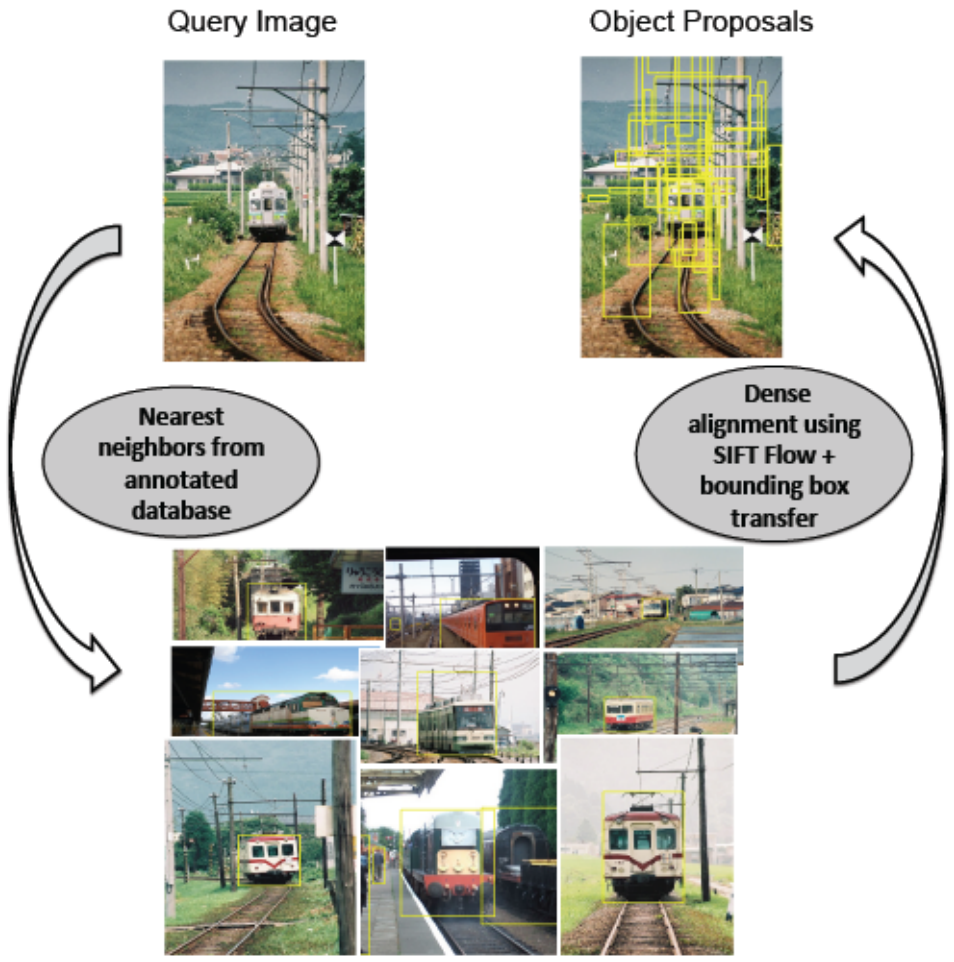


Figure 3.1: For a query image, our system finds the top matches (nine are shown here). The bounding boxes of the top matches are transferred to the input image.

Source of images: `http://host.robots.ox.ac.uk/pascal/VOC/voc2010/#rights`; Used under fair use, 2015.

(a) Query Image

Figure 3.2: A query image and 9 nearest neighbors in the DeCAF feature space Source of images: `http://host.robots.ox.ac.uk/pascal/VOC/voc2010/#rights`; Used under fair use, 2015.

1. Establish dense pixel–wise correspondence between the query image and the neighbors using SIFT–flow [51] algorithm, and use this correspondences to map the pixels from the neighbors annotations to the test image.
2. Warp the neighbors to the size of the query image and map neighbors' annotations to the test image.

3.2 Experiments and results

In our first experiment, we consider the following 4 variants of our approach:

1. Label transfer with Sift flow using DeCAF features (LT DeCAF with SF)
2. Label transfer with Sift flow using Gist features (LT Gist with SF)
3. Label transfer without Sift flow using DeCAF features (LT DeCAF No SF)
4. Label transfer without Sift flow using Gist features (LT Gist No SF)

Given a query image in the PASCAL VOC 2007 detection test set, we find the k nearest neighbors in the DeCAF and Gist feature spaces from the PASCAL 2007 detection trainval set. We vary the proposals generated by considering
$k = 5, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100$ nearest neighbors. We compare the performance of these 4 variants with one of the state-of-the-art approaches Selective Search [11] by using the evaluation protocol described in Chapter 2. For our evaluation, we use PASCAL VOC 2007 test set as the dataset, and ABO as our metric. Fig. 3.3 depicts the ABO performance of our proposed approaches. While none of our variants out–performed Selective Search, LT DeCAF no SF is the best performing one among our proposed approaches. For the sake of completion, we evaluate LT DeCAF No SF against most of the existing algorithms on ABO, Recall and AUC metrics. Note that while our approach is not the best performing algorithm, there are algorithms which perform worse than our approach. Fig. 3.4 depicts the results.

3.3 Conclusion

In this chapter, we presented a novel approach for generating object proposals by transferring bounding boxes from nearest neighbors. We considered 4 variants of generating object proposals. We compared their performance to the Selective Search algorithm using the ABO evaluation metric. **LT DeCAF no SF** was the best performing among the 4 variants, however, it did not outperform Selective Search algorithm. For the sake of completion, we compared LT DeCAF No SF with most of the existing object proposal algorithms. Based

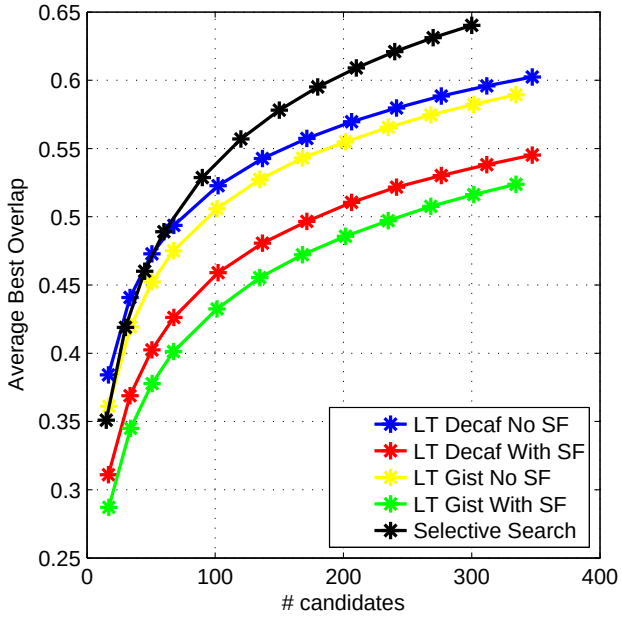
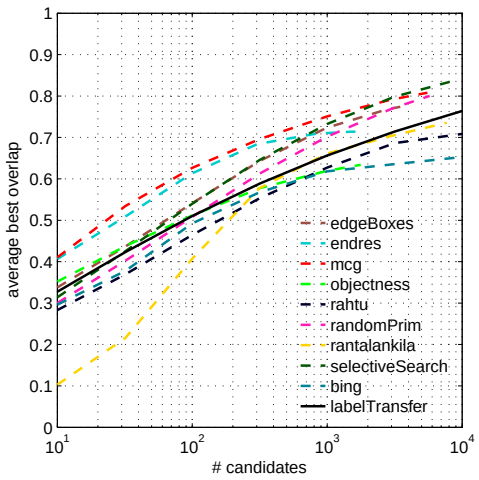
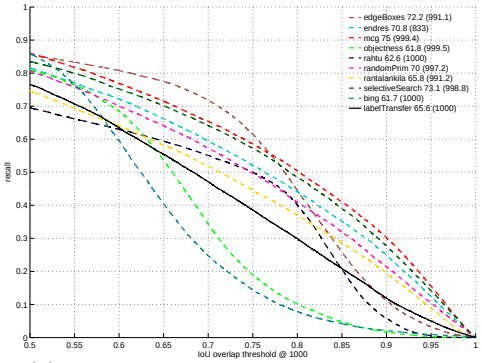


Figure 3.3: Evaluation of our proposed approaches on the ABO metric, and comparison to Selective Search algorithm

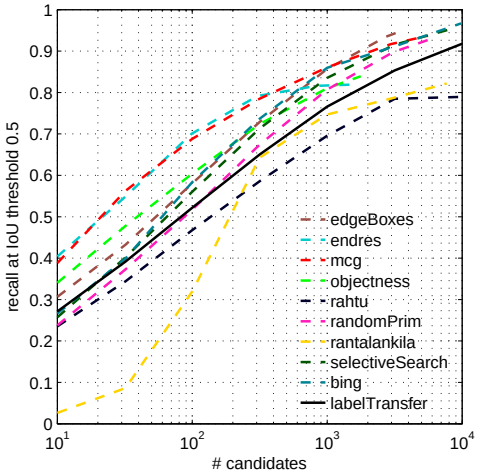
on this evaluation, we conclude that among the various object proposal algorithms reported in literature, ours is average at best. While the results in this chapter were not encouraging, it helped us segway into exploring new ideas. To facilitate evaluation of our algorithm with all the existing algorithms, we ended up building an Object Proposals Library described in Chapter 4. It also made us take a closer look at the evaluation protocol used for object proposals and helped us discover biases in the protocol. This resulted in a **arXiv** publication described in detail in Chapter 5.



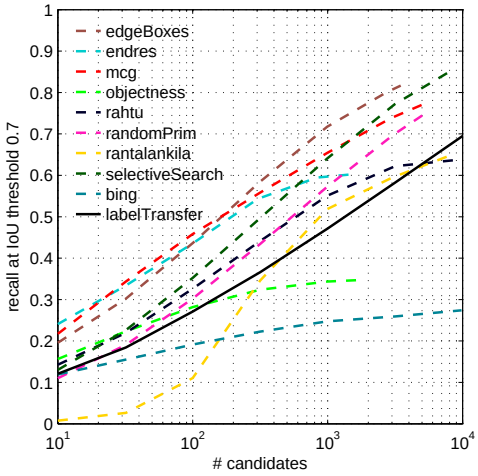
(a) Comparison of ABO performance



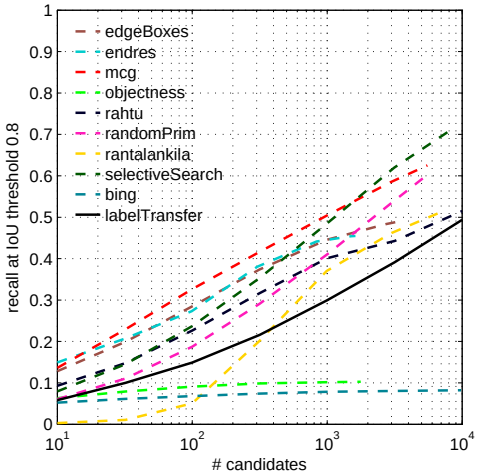
(b) Comparison of Recall at 1000 proposals



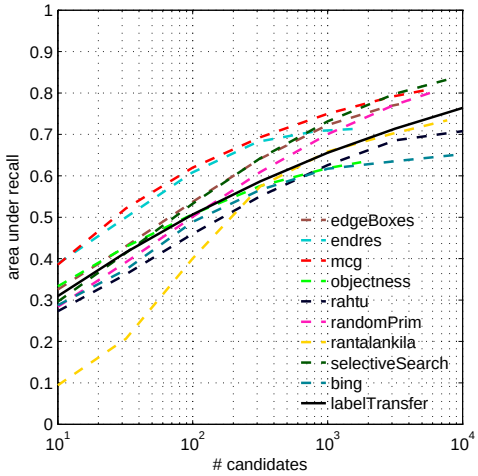
(c) Comparison of Recall at a threshold > 0.5



(d) Comparison of Recall at a threshold > 0.7



(e) Comparison of Recall at a threshold > 0.8



(f) Comparison of AUC curve

Figure 3.4: Evaluation using the most commonly used metrics. **labelTransfer** represents our approach.

Chapter 4

Object Proposals Library

In this chapter, we give provide an overview of the Object Proposals Library – a github repository for object proposal algorithms.

It is common practice in the computer vision research community for authors to make their algorithms open source. While this is a great service to the computer vision community, one downside is that we end up with *disparate* implementations of these algorithms. For example, about 10 different algorithms have been proposed in literature for generating object proposals. But these algoithms do not generate proposals in a common format. Example formats are:

- image segement regions
- bounding boxes. Even within bounding boxes, the proposals may specify
  - top–left *(x,y)* corner pixels, bottom–right *(x,y)* corner pixels of the sub–window
  - top–left *(y,x)* corner pixels, bottom–right *(y,x)* corner pixels
  - top–left *(y,x)* corner pixels, height and width, etc.

Thus, there is a lack of consistency in the output of these algorithms, even though they are all solving the same problem. In order to standardize the output of all the object proposal algorithms and have a central repository for all the existing algorithms, we built a easy–to–use Object Proposals Library. Following are its features:

- Generate proposals using all existing object proposal algorithms
- Outputs of all the algorithms are standardized into a single format
- Evaluate the object proposals using the following metrics:

1. Recall at specific threshold
2. Recall at specific number of proposals
3. Area under recall curves
4. Average Best Overlap

Fig. 4.1, Fig. 4.2 and Fig. 4.3 depict the library and its usage. The library can be accessed from here.

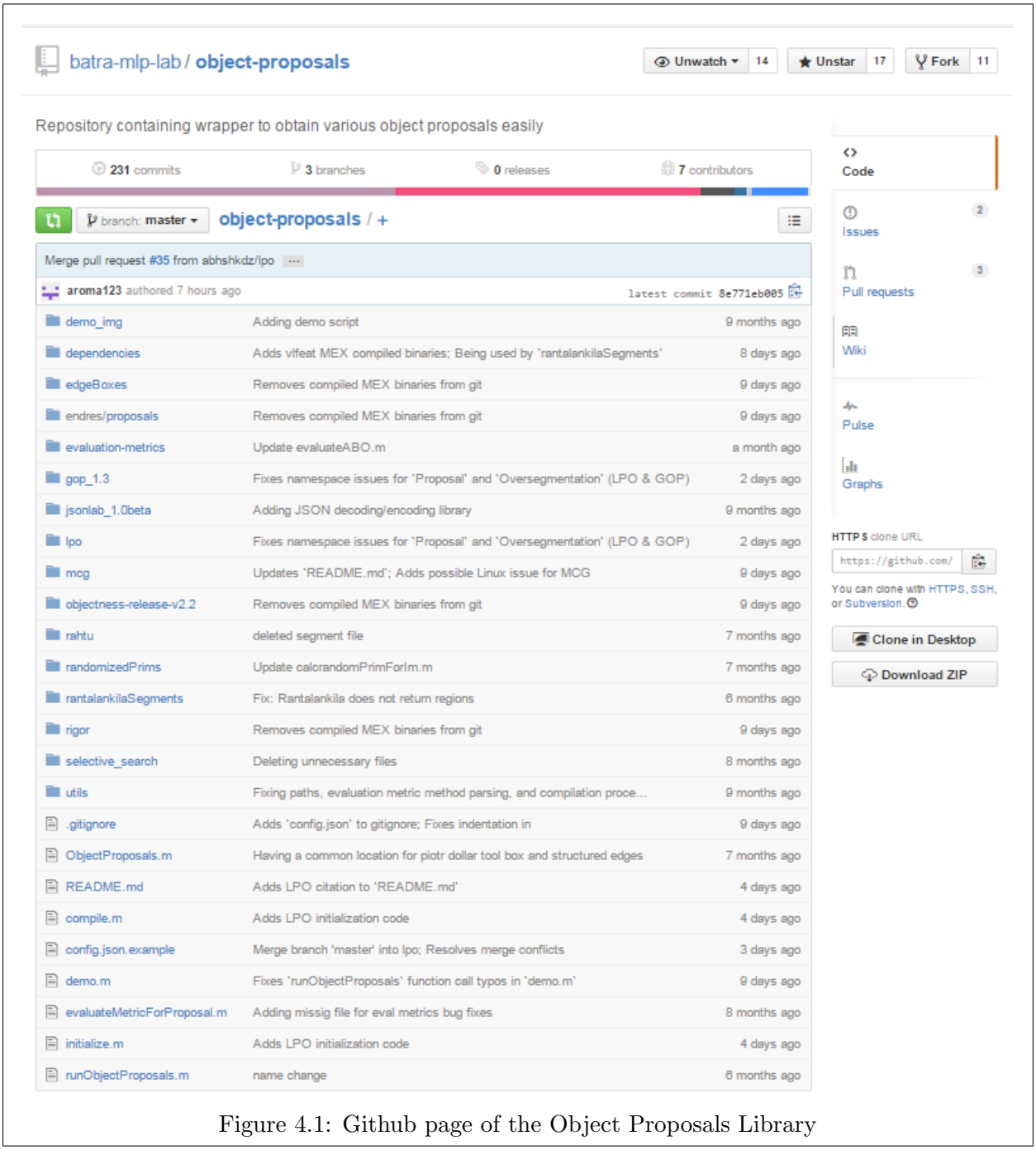


Figure 4.1: Github page of the Object Proposals Library

Generating Proposals

1. Copy over `config.json.example` to `config.json` and set `imageLocation` and `outputLocation`.
2. Initialize path variables.

```
initialize;
```

1. Generate proposals, using either of the following commands.

```
proposals = runObjectProposals('<proposalname>', 'path\to\image.jpg');

OR

im = imread('path\to\image.jpg');
proposals = runObjectProposals('<proposal name>', im);
```

1. For long-running jobs, use the following command.

```
runObjectProposals('<proposalname>');
```

This will generate proposals for all the images in `imageLocation` and save the proposals in `outputLocation`.

`<proposalname>` is the object proposal to be run. List of possible object proposal names:

- `edgeBoxes` [1]
- `endres` [2]
- `mcg` [3]
- `objectness` [4]
- `rahtu` [5]
- `randomPrim` [6]
- `rantalankila` [7]
- `selective_search` [8]
- `rigor` [9]
- `gop` [10]
- `lpo` [11]

Figure 4.2: Steps for generating proposals

Evaluating Proposals

A ground truth file needs to be generated for the dataset. We have provided the file for PASCAL 2007 test set. The following code assumes you have generated proposals for all images in the dataset for which you want to evaluate for each proposal in your `config.json` file.

Evaluation using recall curves & area under recall curves

1. Load ground truth.

```
testset=load('evaluation-metrics/data/pascal_gt_data.mat');
```

1. Generate best recall candidates.

```
compute_best_recall_candidates(testset,configjson,'<proposalame>');
```

'proposalname' is an optional argument. If not provided, the function works for all the object proposals listed above.

1. Plot RECALL/AUC curves.

```
evaluateMetricForProposal('RECALL','<proposalname>');
evaluateMetricForProposal('AUC','<proposalname');

OR

evaluateMetricForProposal('RECALL');
evaluateMetricForProposal('AUC');
```

Evaluation using ABO curves

1. Load ground truth.

```
testset=load('evaluation-metrics/data/pascal_gt_data.mat');
```

1. Generate best recall candidates.

```
compute_abo_candidates(testset,configjson);
```

1. Plot ABO curve.

```
evaluateMetricForProposal('ABO', '<proposalname');

OR

evaluateMetricForProposal('ABO');
```

Figure 4.3: Steps for evaluating proposals

Chapter 5

Object Proposal Evaluation is "Gameable"[1]

Taking a holistic view of the literature on object proposals, it becomes clear that there are actually *two distinct interpretations* of the term 'object proposals' (although no past work seems to have explicitly made this distinction):

- **Category-independent object proposals**: where the goal is to identify all the objects in the image irrespective of their category.
- **Detection proposals**: where the goal is to improve the object detection pipeline, focusing on a chosen set of object classes (*e.g.*, 20 PASCAL categories).

Notice that the former definition has an emphasis on object discovery. The latter definition has an emphasis on the ultimate performance of a detection pipeline.

Surprisingly, despite the two different interpretations and goals of the term 'object proposals', there exists only a single evaluation protocol (with a few different evaluation metrics). Following is the widely adopted evaluation protocol:

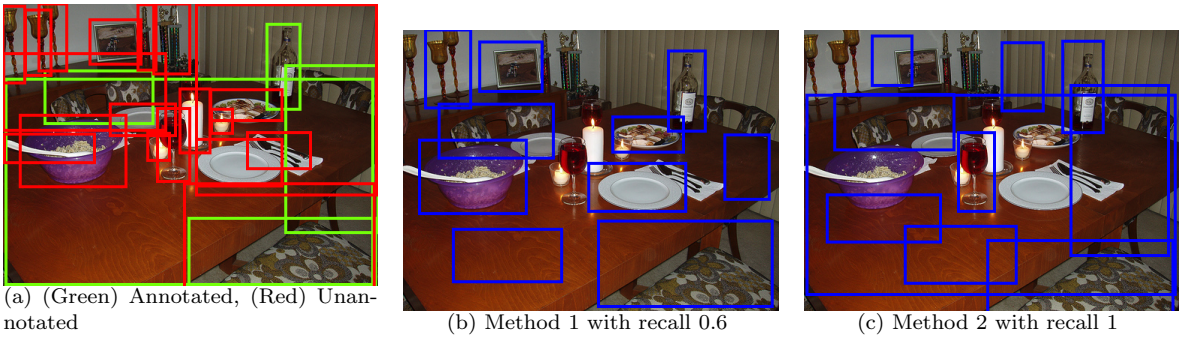
1. Generate proposals on a dataset: The most commonly used dataset for evaluation today is the PASCAL VOC [28] detection test sets. Note that this is a *partially annotated* dataset where only the 20 PASCAL category instances are annotated.
2. Measure the performance of the generated proposals: typically in terms of 'recall' of the annotated instances. Commonly used metrics are described in Section 2.4.

The central emphasis of this chapter is that the current evaluation protocol for object proposal methods is suitable only for detection proposals and is a *biased protocol* for category-independent object proposals. By evaluating only on a specific set of object categories,

[1] Object proposal evaluation is vulnerable to manipulation (both intentional and unintentional)

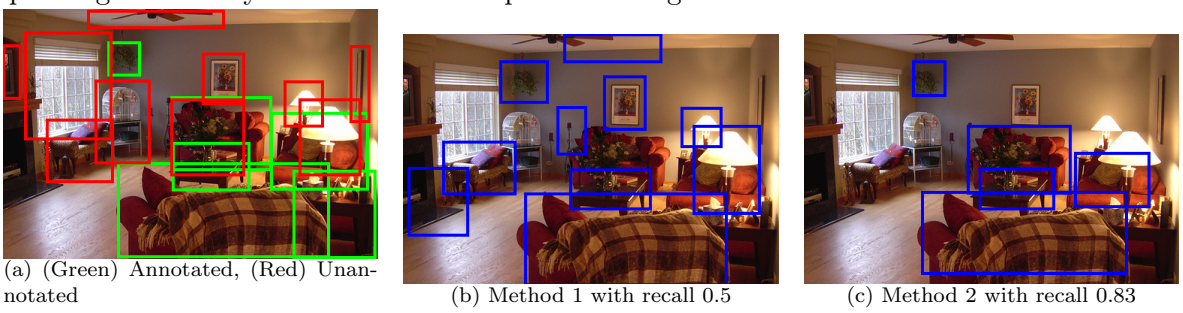
we fail to capture the performance of the proposal algorithm on *all the remaining object categories that are present in the test set, but not annotated in the ground truth.*

Figs. 5.1, 5.2 illustrate this idea on images from PASCAL VOC 2010. Column (a) shows the ground-truth object annotations (in green, the annotations natively present in the dataset for the 20 PASCAL categories; in red, the annotations that we added to the dataset by marking objects originally annotated 'background'). We can see that dataset contains annotations for PASCAL categories (chairs, tables, bottles, *etc.*) but does not contain annotations for other objects present in the image like picture frames, plates, glasses *etc.*



(a) (Green) Annotated, (Red) Unannotated (b) Method 1 with recall 0.6 (c) Method 2 with recall 1

Figure 5.1: (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 that 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.



(a) (Green) Annotated, (Red) Unannotated (b) Method 1 with recall 0.5 (c) Method 2 with recall 0.83

Figure 5.2: (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 plates, glasses that Method 2 missed. Clearly the recall for Method 1 *should* be higher. However, the calculated recall for Method 2 is significantly higher, which feels counter-intuitive. This is because Method 2 recalls more PASCAL category objects.

Source of images: `http://host.robots.ox.ac.uk/pascal/VOC/voc2010/#rights`; Used under fair use, 2015.

Columns (b) and (c) show the outputs of two object proposal systems (top row shows the case when both methods produce the same number of proposals; bottom row shows unequal

number of proposals). We can see that proposal method in Column (b) seems to be more "complete", in the sense that it covers or discovers a large number of instances. For instance, in the top row it detects a number of non-PASCAL categories but misses out on finding the table. In both examples, the method in Column (c) achieves a higher accuracy metric, *even in the bottom row, when it recalls fewer objects*. The reason is that Column (c) recalls/discovers instances of the 20 PASCAL categories, which are the only ones originally annotated.

While intuitive (and somewhat obvious) in hindsight, we believe this is an important finding because it makes the current protocol *gameable* or susceptible to manipulation (both intentional and unintentional). Our thorough evaluation of all popular object proposal techniques suggests that no current technique seems to have 'gamed' or exploited the bias in the protocol. However, caution must be exercised while evaluating these and future algorithms, lest we over-fit as a community to a specific set of object classes and unknowingly lose the category independence of object proposals.

To summarize, the contributions of this chapter are:

- We make an explicit distinction between the two mutually co-existing but different interpretations of object proposals.
- We report the bias and 'gameability' of current object proposal evaluation protocols.
- We demonstrate this gameability via a simple thought experiment where we propose a 'fraudulent' object proposal method that *outperforms all existing object proposal techniques* on current metrics.
- We conduct a thorough evaluation of existing object proposal methods on three densely annotated datasets.
- We propose two ways of improving the current evaluation protocol to truly reward the category-independence of object proposals:

  1. fully annotated datasets, and
  2. evaluation metrics that look at per class performance and bias capacity of proposal generators.

  For the former, we introduce a densely-annotated version of PASCAL VOC 2010 where we annotated *all instances of a lage set of object categories* occurring in all images.

5.1 A Thought Experiment: How to Game the Evaluation Protocol

Let us conduct a thought experiment to demonstrate that the object proposal evaluation protocol can be 'gamed'.

Imagine yourself reviewing a paper claiming to introduce a new object proposal method – called DMP.

Before we divulge the details of DMP, consider the performance of DMP shown in Fig. 5.3 on the PASCAL VOC 2007 dataset, under the AUC-vs-#proposals metric. As we can clearly see, the proposed method DMP *significantly* outperforms all existing object proposal methods [4, 5, 6, 7, 8, 9, 11, 13, 14] (which seem to have little variation over one another). This result seems to indicate that a significant advancement has been made in the field of object proposals generation. In fact, the improvement in AUC at at some points in the curve (*e.g.*, at M=10) seems to be *an order of magnitude* larger than all previous incremental improvements reported in the literature!

So what is our proposed state-of-art technique DMP? It is a mixture-of-experts model, consisting of 20 experts, where each expert is a DeCAF-based [52] objectness detector. At this point, you the savvy reader, are probably already beginning to guess what we did.

DMP stands for 'Detector Masquerading as Proposal' generator. We trained object detectors for the 20 PASCAL categories (in this case with RCNN[15]), and then used these 20 detectors to produce the top-M most confident detections, and declared them to be 'object proposals'.

The point of this *ad absurdum* experiment is to demonstrate the following fact – Clearly, no one would consider a collection of 20 object detectors to be a category-independent object proposal generation method. However, our existing evaluation protocols declare them to be state-of-art.

Why did this happen? Because the protocol today involves evaluating a proposal generator on *a partially annotated* dataset such as PASCAL. The protocol does not reward recall of non-PASCAL categories; in fact, early recall (near the top of the list of candidates) of non-PASCAL objects results in a penalty for the proposal generator! As a result, a proposal generator that tunes itself to these 20 PASCAL categories (either explicitly via training or implicitly via design choices or hyper-parameters) will be declared a better proposal generator when it may not be (as illustrated by DMP). Since the asymptotic limit of this evaluation protocol is an absurd proposal generator, we should be cautious of methods proposing incremental improvements on this protocol.

This thought experiment exposes the inability of the existing protocol to evaluate category independence. There are two ways of alleviating this problem:

- Modify the dataset, *i.e.* use a fully or more densely annotated dataset.

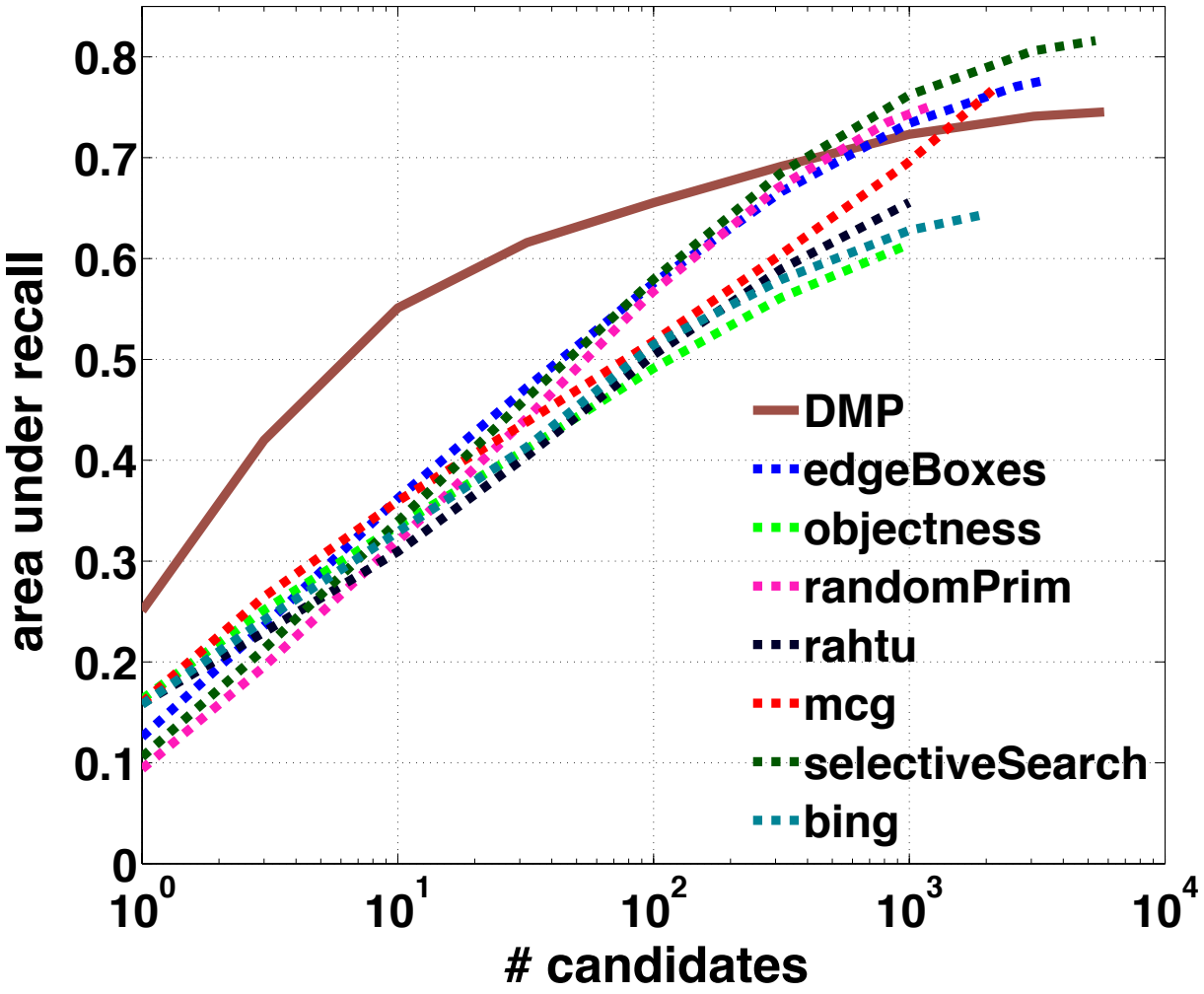


Figure 5.3: Performance of different object proposal methods (dashed lines) and our proposed 'fraudulent' method (DMP) on the PASCAL VOC 2007 dataset. We can see that DMP *significantly* outperforms all other proposal generators. See text for details.

- Modify the metric for evaluation to be used with a partially annotated dataset, because collecting a fully-annotated dataset might be expensive and tedious.

We explore both these directions next.

5.2 Modifying the Dataset

As our first analysis, we evaluate the performance of various object proposal methods and two DMPs (RCNN and DPM[53]) on three different denser-annotated datasets containing many more object categories. This is to quantify how much the performance of our 'fraud-

ulent' proposal generators (DMPs) drops once the bias towards the 20 PASCAL categories is diminished (or completely removed).

We begin by *creating* a nearly fully-annotated dataset by building on the effort of PASCAL Context [54]; followed by evaluation on other partial-but-denser-annotated datasets MS COCO [55] and NYU-Depth V2 [56].

5.2.1 PASCAL Context

This dataset was introduced by Mottaghi *et al*. [54]. It contains additional annotations for all images of PASCAL VOC 2010 dataset [57]. The annotations are semantic segmentation maps, where *every single pixel* previously annotated 'background' in PASCAL was assigned a category label. In total, annotations have been provided for 459 categories. This includes the original 20 PASCAL categories and new classes such as keyboard, fridge, picture, cabinet, plate, clock.

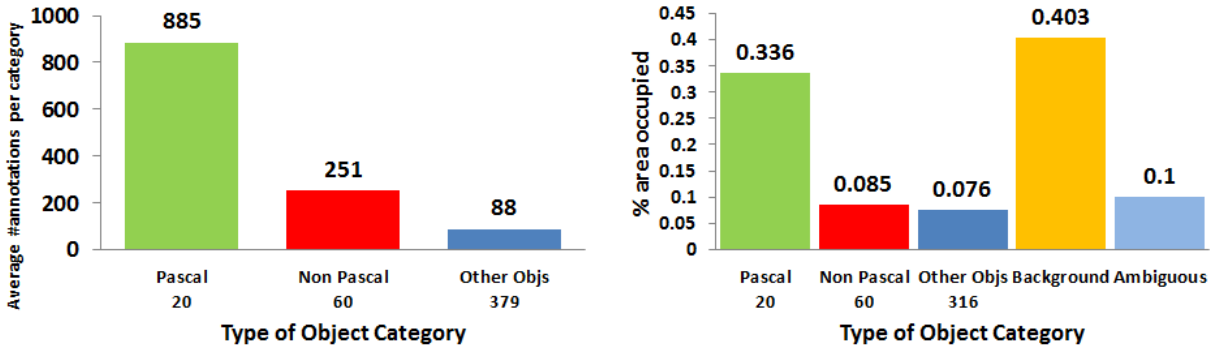
Unfortunately, as of the time of writing this paper, the dataset contains only category-level semantic segmentations, not *instance-level* segmentations. For our task, we needed instance-level bounding box annotations, which cannot be reliably extracted from category-level segmentation masks because the masks for several instances (of say chairs) may be merged together into a single 'blob' in the category-level mask.

Creating Instance-Level Annotations for PASCAL Context: Thus, we created instance-level bounding box annotations for all images in PASCAL Context dataset. First, out of the 459 category labels in PASCAL Context, we identified 379 categories to be 'thing' categories, and ignored the remaining 'stuff' categories or 'ambiguous' categories (*e.g.*, a 'tree' may be a 'thing' or 'stuff' depending on viewpoint of the camera) – neither of these lend themselves to bounding-box-based object detection. A complete list is available in the supplemental material.

Next, we selected the 60 most frequent non-PASCAL categories from this list of 'things' and manually annotated all instances of these categories. This manual annotation was performed with the aid of the semantic segmentation maps already present in the PASCAL context annotations. Examples annotations are shown in Fig. 5.5, and more examples are available in the supplement.

Statistics of New Annotations: Fig. 5.4 shows some statistics of our new annotations. Specifically, Fig. 5.4a shows average number of instances for the 20 PASCAL categories, the 60 new non-PASCAL categories, and the other ignored 'thing' objects. It is interesting to note that the number of annotations for the new 60 categories we annotated were about the same as the number of instances for 20 PASCAL categories. This is a good indicator of the number of proposal candidates which are not being rewarded due to partially annotated nature of the PASCAL VOC 2010 dataset.

Fig. 5.4b shows the average size (percentage of image-area) of different types of objects – 20 PASCAL categories, 60 new categories, remaining (316) 'thing' categories, 'stuff' (or background) categories, and 'ambiguous' categories. We can see that most non-PASCAL categories occupy a small percentage of the image. This is understandable given that the dataset was curated for the 20 PASCAL categories. The other categories just happened to be in the pictures. Unfortunately, this also makes them difficult to be detected.



(a) Average number of annotations for different object categories.

(b) Fraction of image-area occupied by different object categories.

Figure 5.4: Distribution of object classes in PASCAL Context with respect to different attributes.

(a) PASCAL Context annotations [54].

(b) Our augmented annotations.

Figure 5.5: Augmenting PASCAL Context with instance-level annotations. (Green = PASCAL 20 categories; Red = new objects)

Source of image: `http://host.robots.ox.ac.uk/pascal/VOC/voc2010/#rights`; Used under fair use, 2015.

5.2.2 MS COCO

The Microsoft Common Objects in Context (MS COCO) dataset [55] contains 91 common object categories with 82 of them having more than 5,000 labeled instances. Overall, the dataset has 2,500,000 labeled instances in 328,000 images. The dataset not only has sig-

nificantly higher number of instances per category than the PASCAL VOC dataset, but also considerably more object instances per image (7.7) as compared to ImageNet (3.0) and PASCAL (2.3).

5.2.3 NYU-Depth V2

NYU-Depth V2 dataset [56] is comprised of video sequences from a variety of indoor scenes as recorded by both the RGB and Depth cameras from the Microsoft Kinect. It features 1449 densely labeled pairs of aligned RGB and depth images where each object is labeled with a class and an instance number. We used these 1449 densely annotated RGB images for evaluating object proposal algorithms. To the best of our knowledge, this is the first paper to compare proposal methods on such a dataset.

5.2.4 Evaluating Proposals on Different Datasets

Experimental Setup: On MS COCO and PASCAL Context datasets we conducted experiments as follows:

- Use the existing evaluation protocol for evaluation, *i.e.*, evaluate only on the 20 PASCAL categories.
- Evaluate on all the annotated classes.
- For the sake of completeness, we also report results on all the classes except the PASCAL 20 classes.

On NYU-Depth V2 we only evaluate the performance on all categories. This is because only 8 (of the 20) PASCAL categories are present in NYU-Depth V2 (since it comprises of indoor scenes).

The two DMPs we use are based on two popular object detectors - DPM[53] and RCNN[15]. We train a DPM on 20 pascal categories and use it as an object proposal method. To generate large number of proposals, we chose a low value of threshold in Non-Maximum Suppression (NMS). Proposals are generated for each category and a score is assigned to them by the corresponding DPM for that category. These proposals are then merge-sorted on the basis of this score. Top M proposals are selected from this sorted list where M is the number of proposals to be generated.

Another (stronger) DMP is RCNN which is a detection pipeline that uses 20 SVMs (each for one PASCAL category) trained on decaf features extracted on selective search boxes. Since RCNN itself uses selective search proposals, it should be viewed as a trained *reranker* of selective search boxes. As a consequence, it ultimately equals selective search performance

once the number of candidates become large. We used the pretrained SVM models released with the RCNN code, which were trained on the 20 classes of PASCAL VOC 2007 trainval set. For every test image, we generate the Selective Search proposals using the 'FAST' mode and calculate the 20 SVM scores for each proposal. The 'objectness' score of a proposal is then the maximum of the 20 SVM scores. All the proposals are then sorted by this score and top M proposals are selected.

Results and Observations: We now explore how changing the evaluation protocol affects the results of the thought experiment from Section 5.1.

Figs. 5.6, 5.7 compare the performance of DMPs with a number of existing object proposal methods [4, 5, 6, 7, 8, 9, 11, 13, 14] on PASCAL Context and MS COCO respectively.

We can see in column (a) that when evaluated on only 20 PASCAL categories the DMPs trained on these categories appear to significantly outperform all proposal generators. However, we can see that they are not category independent because they suffer a big drop in performance when evaluated on 60 non-PASCAL categories in column (b).

Notice that on PASCAL context, *all proposal generators* suffer a drop in performance between the 20 PASCAL categories and 60 non-PASCAL categories. As we previously discussed, this is due to the fact that the non-PASCAL categories tend to be generally smaller than the PASCAL categories (which were the main targets of the dataset curators). However, the DMPs suffer the biggest drop.

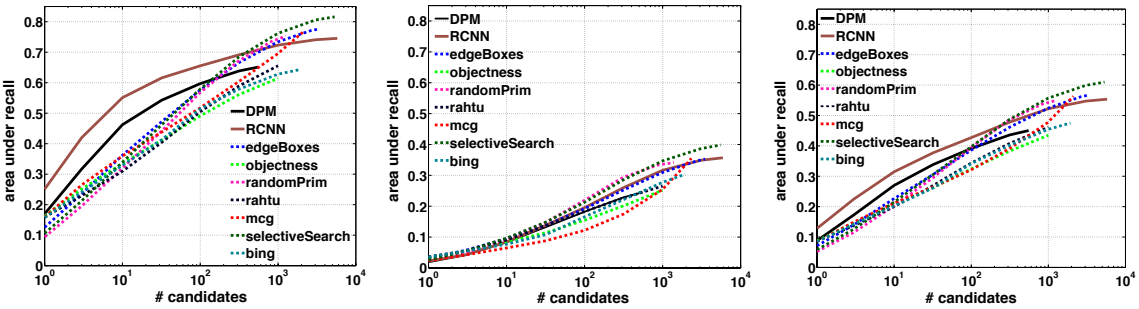
It is interesting to note that due to the ratio of instances of 20 PASCAL categories vs other 60 categories, DMPs continue to slightly outperform proposal generators when evaluated on all categories, as shown in Column (c).

Fig. 5.8 shows results for NYU-Depth V2. We can see that when many classes in the test dataset are not PASCAL classes, DMPs tend to perform poorly, although interesting still not as poor as the worst proposal generators. Results on other evaluation criteria are in the supplementary document.

Take-away messages:

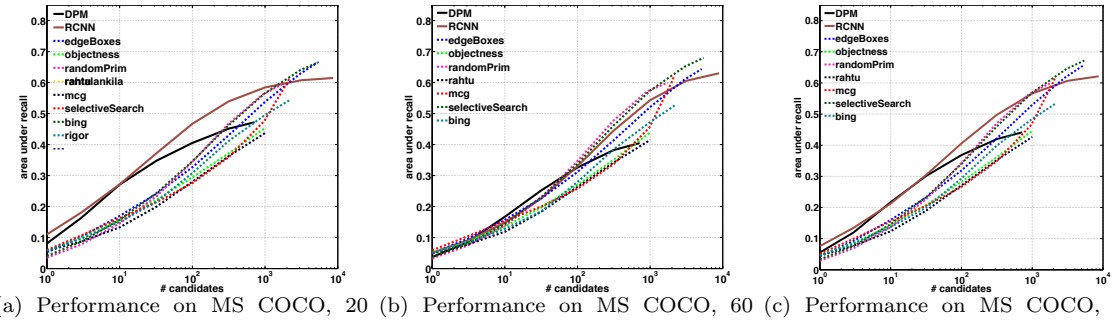
1. The drop in performance from after the adding new non-PASCAL categories exposes DMPs' poor generalization on "unseen" categories. This drop can be used to detect and guard against 'biased' proposal generators.

2. The slight advantage that DMPs continue to hold over existing proposals (even when all categories are annotated) is due to the imbalance in the number of annotations of PASCAL classes and non-PASCAL classes. This exposes the problem of a metric which is class agnostic, used on a dataset which is unbalanced, and suggests the need for dataset with more coverage.



(a) Performance on PASCAL Context, 20 PASCAL classes annotated. (b) Performance on PASCAL Context, 60 non-PASCAL classes annotated. (c) Performance on PASCAL Context, all classes annotated.

Figure 5.6: Performance of different methods on PASCAL Context with different sets of annotations.



(a) Performance on MS COCO, 20 PASCAL classes annotated. (b) Performance on MS COCO, 60 non-PASCAL classes annotated. (c) Performance on MS COCO, all classes annotated.

Figure 5.7: Performance of different methods on MS COCO with different sets of annotations.

5.3 Modifying the Metric

To recap what we have discussed so far – we began by introducing a 'fraudulent' object proposal algorithm that could exploit or fool the existing evaluation protocols. We discussed how one way of detecting such bias is by changing the annotations in the dataset. However, annotating datasets is an expensive and time-consuming process. In this section, we analyze whether we can continue to use existing partially annotated datasets but modify the metrics used.

5.3.1 Measuring Fine-Grained Recall

One way of developing a fine-grained understanding of the performance of various proposal generation methods and to detect bias is by plotting Recall per category. In the previous section, we saw that the performance of DMPs was different on non-PASCAL categories as compared to 20 PASCAL classes on the modified PASCAL Context dataset.

To gain other such insights, we look at a more fine-grained per category performance of proposal methods and DMPs. We evaluate by plotting recall values and area under the recall curve for all 80 (20 PASCAL + 60 non-PASCAL) categories for the modified PASCAL-

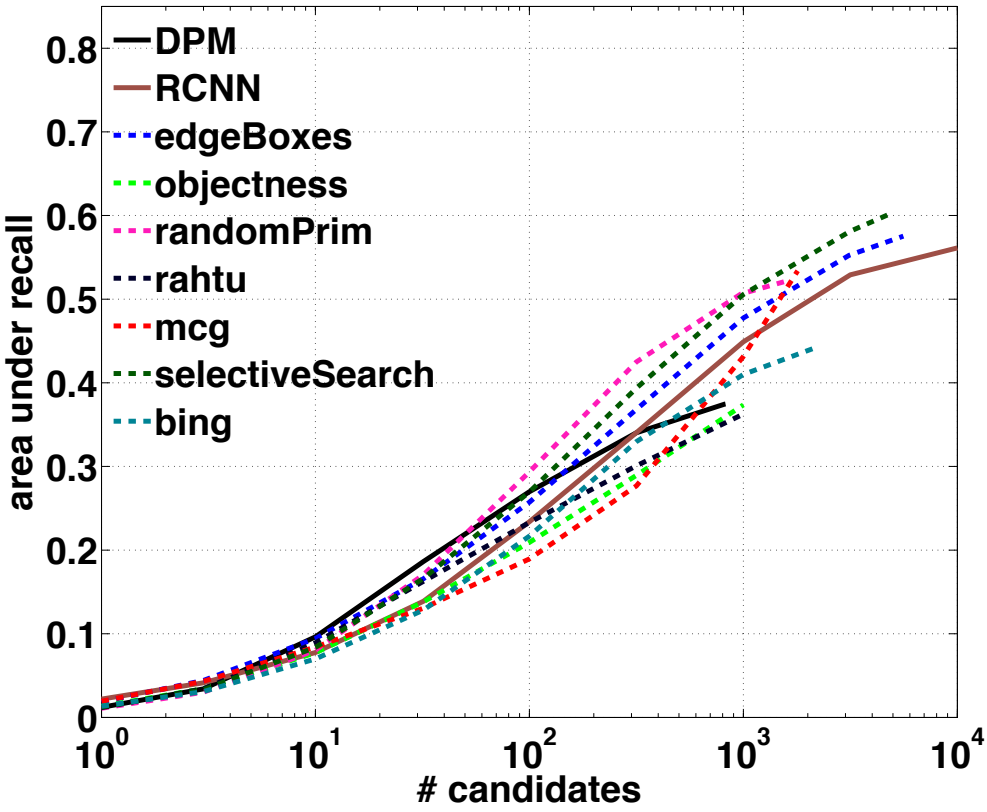


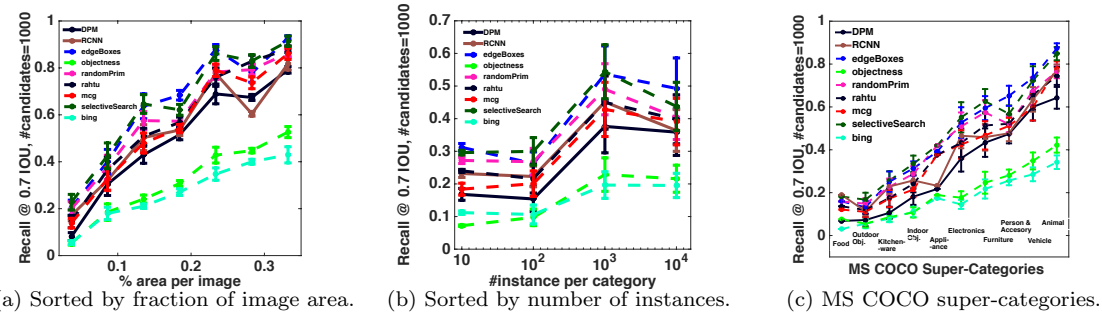
Figure 5.8: Performance on NYU-Depth V2, all classes annotated

Context dataset. We sorted/clustered all categories on the basis of:

- Average size (fraction of image area) of the category,
- Frequency (Number of instances) of the category,
- Membership in 'super-categories' defined in MS COCO dataset (electronics, animals, appliance, *etc.*).

Trends: Fig. 5.9 shows the performance of different proposal methods and DMPs along each of these dimensions. These plots can be used to answer if some proposal methods are optimized for larger or frequent categories. Interestingly, we noticed that all methods follow similar trends with respect to different attributes, suggesting that either no such optimization has taken place, or all methods have been optimized. It is reasonable to believe the former.

In Fig. 5.9a, we find that recall value steadily improves as the relative size of object to image increases for all proposal methods as well as DMPs. This shows that perhaps as expected, bigger objects are typically easier to find than smaller objects. In Fig. 5.9b, we see that the recall values generally increases as the number of instances increase except for one outlier



(a) Sorted by fraction of image area. (b) Sorted by number of instances. (c) MS COCO super-categories.

Figure 5.9: Recall at 0.7 IOU for categories sorted/clustered by (a) size, (b) number of instances, and (c) MS COCO 'super-categories'.

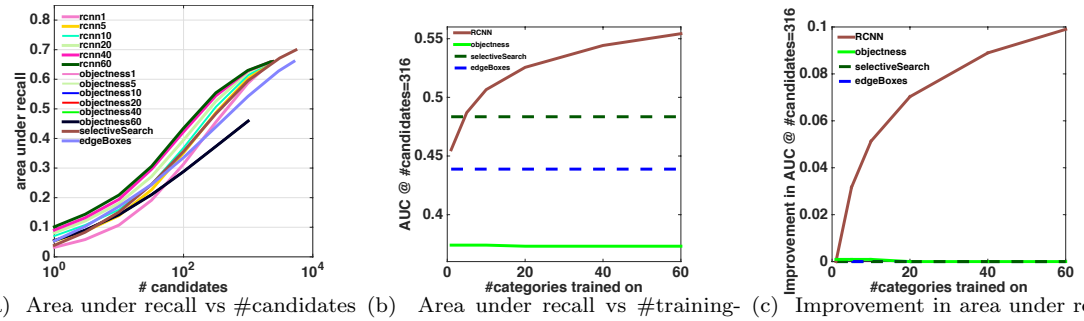
category. This category was found to be 'pole' which appears to be quite difficult to recall, since poles are often occluded and have a long elongated shape.

Overall, while this fine-grained analysis is useful, we found that it can not be used to detect a 'fraudulent' proposal generator, since DMPs behave similar to the other methods.

5.3.2 Assessing Bias Capacity

Some proposal methods rely on explicit training to learn "objectness" parameters (similar to DMPs). Depending upon which and how many categories are trained on, these methods could have a biased view of "objectness". One way of measuring the *bias capacity* in an proposal method is to plot the performance *vs*. the number of categories trained on. A method that involves little or not 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 'bias-prone' methods like DMPs that are tuned to specific classes.

In this experiment, we compared the performance of one DMP method (RCNN[15]), one learning-based proposal method (Objectness[7]), and two non-learning based proposal methods (Selective Search[11], EdgeBoxes[4]) as a function of the number of 'seen' categories (the categories trained on). Method names 'rcnnTrainN', 'objectnessTrainN' indicate that they were trained on images that contain annotations for only N categories. Once trained, these methods were evaluated on a randomly-chosen set of 2396 images taken from MS COCO[55] dataset. Fig. 5.10 shows the results. We can see that with even a modest increase in training data, performance improvement of RCNN is much more than objectness. There is no change in Selective Search [11] and EdgeBoxes [4] since there is no training involved at all. It is thus reasonable to conclude that object independent methods are less prone to be affected by training with more data as compared to methods like DMPs. This analysis could hence serve as a diagnostic tool for differentiating between such methods.



(a) Area under recall vs #candidates for various #trained categories (b) Area under recall vs #training-categories. (c) Improvement in area under recall vs #training-categories.

Figure 5.10: 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.

5.4 Conclusion

In this chapter, we make an explicit distinction between the two mutually co-existing but different interpretations of object proposals. The current evaluation protocol for object proposal methods is suitable only for detection proposals and is a biased 'gameable' protocol for category-independent object proposals. By evaluating only on a specific set of object categories, we fail to capture the performance of the proposal algorithm on all the remaining object categories that are present in the test set, but not annotated in the ground truth. We demonstrate this gameability via a simple thought experiment where we propose a 'fraudulent' object proposal method that outperforms all existing object proposal techniques on current metrics. We conduct a thorough evaluation of existing object proposal methods on three densely annotated datasets. We introduce a densely-annotated version of PASCAL VOC 2010 where we annotated all instances of all object categories occurring in all images. We hope this dataset will be broadly useful.

Furthermore, since densely annotating the dataset is a tedious and costly task; we proposed a set of diagnostic tools to plug the vulnerability of the current protocol.

Fortunately, we find that none of existing proposal methods seem to be biased, most of the existing algorithms do generalize well to different datasets and in our experiments even on densely annotated datasets. In that sense, our findings are consistent with results in [30]. However, that should not prevent us from recognizing and safeguarding against the flaws in the protocol, lest we over-fit as a community to a specific set of object classes.

Chapter 6

Conclusion

In this chapter, we summarize the work done towards this thesis. Object proposals have quickly become the de–facto pre–processing step in a number of vision pipelines. Specifically in the object detection pipeline, they represent a more efficient design pattern when compared to the previously used sliding window approach. This makes them an ideal choice for real time and low power object detection. In this thesis we studied various aspects of object proposals.

In chapter 2, we summarized various object proposal algorithms. We also described in detail the protocol used for evaluating them.

In chapter 3, we proposed a new approach for generating object proposals by transfering the bounding boxes from the nearest neighbors in a large annotated dataset. While our approach did not out–perform the state–of–art, it helped us segway into exploring new ideas described in the later chapters.

In chapter 4, we provided an overview of the Object Proposals Library. It is a central repository for generating object proposals using all the existing algorithms ,and evaluating them using the most commonly used metrics.

In chapter 5, our contributions are the following:

- We make an explicit distinction between the two mutually co-existing but different interpretations of object proposals.
- We report the bias and 'gameability' of current object proposal evaluation protocols.
- We demonstrate this gameability via a simple thought experiment where we propose a 'fraudulent' object proposal method that *outperforms all existing object proposal techniques* on current metrics.
- We conduct a thorough evaluation of existing object proposal methods on three densely

annotated datasets.

- We propose two ways of improving the current evaluation protocol to truly reward the category-independence of object proposals:

  1. fully annotated datasets, and
  2. evaluation metrics that look at per class performance and bias capacity of proposal generators.

  For the former, we introduce a densely-annotated version of PASCAL VOC 2010 where we annotated *all instances of a large set of object categories* occurring in all images.

6.1 Future work

It would be interesting to see how our label transfer work can be improved by better distance metrics and by performing region based nearest neighbor search rather than full image.

It would also be interesting to see if our analysis for evaluating object proposals can be extended to other kinds of proposals like spatio–temporal proposals, RGBD proposals, etc. And if people really care about detection proposals, then the recall metrics can and should directly be optimized for in the same spirit as [58], [59] and follow on work in [60].

Bibliography

[1] D. Marr, "Vision: A computational investigation into the human representation and processing of visual information," *WH San Francisco: Freeman and Company*, 1982.

[2] E. C. Hildreth and J. M. Hollerbach, "Artificial intelligence: computational approach to vision and motor control," *Comprehensive Physiology*, 1987.

[3] S. Papert, "The summer vision project," 1966. `http://dspace.mit.edu/bitstream/handle/1721.1/6125/AIM-100.pdf`.

[4] C. L. Zitnick and P. Dollar, "Edge boxes: Locating object proposals from edges," in *The IEEE European Conference on Computer Vision (ECCV)*, 2014.

[5] I. Endres and D. Hoiem, "Category-independent object proposals with diverse ranking," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 36, no. 2, pp. 222–234, 2014.

[6] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[7] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2012.

[8] E. Rahtu, J. Kannala, and M. B. Blaschko, "Learning a category independent object detection cascade," in *The IEEE International Conference on Computer Vision (ICCV)*, 2011.

[9] S. Manen, M. Guillaumin, and L. Van Gool, "Prime object proposals with randomized prim's algorithm," in *The IEEE International Conference on Computer Vision (ICCV)*, 2013.

[10] P. Rantalankila, J. Kannala, and E. Rahtu, "Generating object segmentation proposals using global and local search," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[11] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision (IJCV)*, 2013.

[12] A. Humayun, F. Li, and J. M. Rehg, "Rigor- recycling inference in graph cuts for generating object regions," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[13] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[14] P. Krähenbühl and V. Koltun, "Geodesic object proposals," in *The IEEE European Conference on Computer Vision (ECCV)*, 2014.

[15] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *The IEEE European Conference on Computer Vision (ECCV)*, 2014.

[17] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov, "Scalable, high-quality object detection," *arXiv preprint arXiv:1412.1441*, 2014.

[18] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," in *The IEEE International Conference on Computer Vision (ICCV)*, 2013.

[19] R. G. Cinbis, J. Verbeek, and C. Schmid, "Segmentation Driven Object Detection with Fisher Vectors," in *The IEEE International Conference on Computer Vision (ICCV)*, 2013.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *arXiv preprint arXiv:1409.4842*, 2014.

[21] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[22] W. Ouyang, P. Luo, X. Zeng, S. Qiu, Y. Tian, H. Li, S. Yang, Z. Wang, Y. Xiong, C. Qian, *et al.*, "Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection," *arXiv preprint arXiv:1409.3505*, 2014.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," 2014.

[25] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 34, pp. 2189–2202, Nov 2012.

[26] J. Carreira and C. Sminchisescu, "CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 34, 2012.

[27] I. Endres and D. Hoiem, "Category independent object proposals," in *Proceedings of the 11th European Conference on Computer Vision: Part V*, ECCV'10, (Berlin, Heidelberg), pp. 575–588, Springer-Verlag, 2010.

[28] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results." http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[29] J. Hosang, R. Benenson, and B. Schiele, "How good are detection proposals, really?," in *arXiv preprint arXiv:1406.6962*, 2014.

[30] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?," *arXiv preprint arXiv:1502.05082*, 2015.

[31] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu, "Unsupervised joint object discovery and segmentation in internet images," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1939–1946, 2013.

[32] Z. Shi, T. M. Hospedales, and T. Xiang, "Bayesian joint topic modelling for weakly supervised object localisation," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[33] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid, "Spatio-temporal object detection proposals," in *The IEEE European Conference on Computer Vision (ECCV)*, 2014.

[34] K. Fragkiadaki, P. Arbelaez, P. Felsen, and J. Malik, "Spatio-temporal moving object proposals," *arXiv preprint arXiv:1412.6504*, 2014.

[35] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning Rich Features from RGB-D Images for Object Detection and Segmentation," *arXiv preprint arXiv:1407.5736*, 2014.

[36] D. Banica and C. Sminchisescu, "CPMC-3D-O2P: semantic segmentation of RGB-D images using CPMC and second order pooling," *arXiv preprint arXiv:1312.7715*, 2013.

[37] B. Alexe, T. Deselaers, and V. Ferrari, "Objectness measure v2.2." http://groups.inf.ed.ac.uk/calvin/objectness/objectness-release-v2.2.zip.

[38] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition." http://koen.me/research/downloads/SelectiveSearchCodeIJCV.zip.

[39] E. Rahtu, J. Kannala, and M. B. Blaschko, "Learning a category independent object detection cascade." http://www.ee.oulu.fi/research/imag/object_detection/ObjectnessICCV_ver02.zip.

[40] S. Manen, M. Guillaumin, and L. V. Gool, "Prime object proposals with randomized primś algorithm." https://github.com/smanenfr/rp#rp.

[41] P. Arbelaez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping." https://github.com/jponttuset/mcg/archive/v2.0.zip.

[42] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges." https://github.com/pdollar/edges.

[43] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps." https://github.com/varun-nagaraja/BING-Objectness.

[44] P. Rantalankila, J. Kannala, and E. Rahtu, "Generating object segmentation proposals using global and local search." http://www.cse.oulu.fi/~erahtu/ObjSegProp/spagglom_01.zip.

[45] P. Krähenbühl and V. Koltun, "Geodesic object proposals." http://www.philkr.net/home/gop.

[46] A. Humayun, F. Li, and J. M. Rehg, "Rigor: Recycling inference in graph cuts for generating object regions." http://cpl.cc.gatech.edu/projects/RIGOR/resources/rigor_src.zip.

[47] I. Endres and D. Hoiem, "Category-independent object proposals with diverse ranking." http://vision.cs.uiuc.edu/proposals/data/PROP_code.tar.gz.

[48] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[49] D. Kuettel and V. Ferrari, "Figure-ground segmentation by transferring window masks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 558–565, 2012.

[50] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 33, no. 12, pp. 2368–2382, 2011.

[51] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. Freeman, "Sift flow: dense correspondence across different scenes," in *The IEEE European Conference on Computer Vision (ECCV)*, pp. 28–42, 2008. `people.csail.mit.edu/celiu/ECCV2008/`.

[52] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," *arXiv preprint arXiv:1310.1531*, 2013.

[53] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 32, no. 9, pp. 1627–1645, 2010.

[54] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[55] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *The IEEE European Conference on Computer Vision (ECCV)*, 2014.

[56] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *The IEEE European Conference on Computer Vision (ECCV)*, 2012.

[57] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results." http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[58] A. Guzman-Rivera, D. Batra, and P. Kohli, "Multiple Choice Learning: Learning to Produce Multiple Structured Outputs," in *Advances in Neural Information Processing Systems*, 2012.

[59] A. Guzman-Rivera, P. Kohli, D. Batra, and R. Rutenbar, "Efficiently enforcing diversity in multi-output structured prediction," in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, 2014.

[60] A. Guzman-Rivera, P. Kohli, B. Glocker, J. Shotton, T. Sharp, A. Fitzgibbon, and S. Izadi, "Multi-output learning for camera relocalization," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1114–1121, 2014.

Appendix A

Supplement

In Chapter 5, we demonstrated how the object proposal evaluation protocol is gameable and performed some experiments to resolve this gameability. Here we present additional details and results which support the arguments presented in Chapter 5.

Section A.1 provides details of the annotation we created for PASCAL Context.

Section A.2 presents extended results for comparison of different proposal methods on densely annotated datasets under different evaluation metrics.

Finally, we show the per category performance of various methods on MS COCO (to accompany the PASCAL Context results in the main paper) in section A.3.

A.1 Details of PASCAL Context Annotation

As explained before, PASCAL Context provides full annotations for PASCAL VOC 2010 dataset in the form of semantic segmentations. A total of 459 object classes have labeled in this dataset. We split these into three categories namely Objects/Things, Background/Stuff and Ambiguous as shown in Tables A.1, A.3 and A.2. Most object classes (396) were put in the 'Objects' category. 20 of these are PASCAL categories. Of the remaining 376, we selected the most frequently occurring 60 categories and added instance level annotations for the same.

A.2 Evaluation of Proposals on Other Metrics

In this section, we show the performance of different proposal methods and DMPs on MS COCO dataset on various metrics. Fig. A.1a shows performance on Recall-vs-IOU metric

Object/Thing Classes in PASCAL Context Dataset							
accordion	candleholder	drainer	funnel	lightbulb	pillar	sheep	tire
aeroplane	cap	dray	furnace	lighter	pillow	shell	toaster
airconditioner	car	drinkdispenser	gamecontroller	line	pipe	shoe	toilet
antenna	card	drinkingmachine	gamemachine	lion	pitcher	shoppingcart	tong
ashtray	cart	drop	gascylinder	lobster	plant	shovel	tool
babycarriage	case	drug	gashood	lock	plate	sidecar	toothbrush
bag	casetterecorder	drum	gasstove	machine	player	sign	towel
ball	cashregister	drumkit	giftbox	mailbox	pliers	signallight	toy
balloon	cat	duck	glass	mannequin	plume	sink	toycar
barrel	cd	dumbbell	glassmarble	map	poker	skateboard	train
baseballbat	cdplayer	earphone	globe	mask	pokerchip	ski	trampoline
basket	cellphone	earrings	glove	mat	pole	sled	trashbin
basketballbackboard	cello	egg	gravestone	matchbook	pooltable	slippers	tray
bathtub	chain	electricfan	guitar	mattress	postcard	snail	tricycle
bed	chair	electriciron	gun	menu	poster	snake	tripod
beer	chessboard	electricpot	hammer	meterbox	pot	snowmobiles	trophy
bell	chicken	electricsaw	handcart	microphone	pottedplant	sofa	truck
bench	chopstick	electronickeyboard	handle	microwave	printer	spanner	tube
bicycle	clip	engine	hanger	mirror	projector	spatula	turtle
binoculars	clippers	envelope	harddiskdrive	missile	pumpkin	speaker	tvmonitor
bird	clock	equipment	hat	model	rabbit	spicecontainer	tweezers
birdcage	closet	extinguisher	headphone	money	racket	spoon	typewriter
birdfeeder	cloth	eyeglass	heater	monkey	radiator	sprayer	umbrella
birdnest	coffee	fan	helicopter	mop	radio	squirrel	vacuumcleaner
blackboard	coffeemachine	faucet	helmet	motorbike	rake	stapler	vendingmachine
board	comb	faxmachine	holder	mouse	ramp	stick	videocamera
boat	computer	ferriswheel	hook	mousepad	rangehood	stickynote	videogameconsole
bone	cone	fireextinguisher	horse	musicalinstrument	receiver	stone	videoplayer
book	container	firehydrant	horse-drawncarriage	napkin	recorder	stool	videotape
bottle	controller	fireplace	hot-airballoon	net	recreationalmachines	stove	violin
bottleopener	cooker	fish	hydrovalve	newspaper	remotecontrol	straw	wakeboard
bowl	copyingmachine	fishtank	inflatorpump	oar	robot	stretcher	wallet
box	cork	fishbowl	ipod	ornament	rock	sun	wardrobe
bracelet	corkscrew	fishingnet	iron	oven	rocket	sunglass	washingmachine
brick	cow	fishingpole	ironingboard	oxygenbottle	rockinghorse	sunshade	watch
broom	crabstick	flag	jar	pack	rope	surveillancecamera	waterdispenser
brush	crane	flagstaff	kart	pan	rug	swan	waterpipe
bucket	crate	flashlight	kettle	paper	ruler	sweeper	waterskateboard
bus	cross	flower	key	paperbox	saddle	swimring	watermelon
cabinet	crutch	fly	keyboard	papercutter	saw	swing	whale
cabinetdoor	cup	food	kite	parachute	scale	switch	wheel
cage	curtain	forceps	knife	parasol	scanner	table	wheelchair
cake	cushion	fork	knifeblock	pen	scissors	tableware	window
calculator	cuttingboard	forklift	ladder	pencontainer	scoop	tank	windowblinds
calendar	disc	fountain	laddertruck	pencil	screen	tap	wineglass
camel	disccase	fox	ladle	person	screwdriver	tape	wire
camera	dishwasher	frame	laptop	photo	sculpture	tarp	
cameralens	dog	fridge	lid	piano	scythe	telephone	
can	dolphin	frog	lifebuoy	picture	sewer	telephonebooth	
candle	door	fruit	light	pig	sewingmachine	tent	

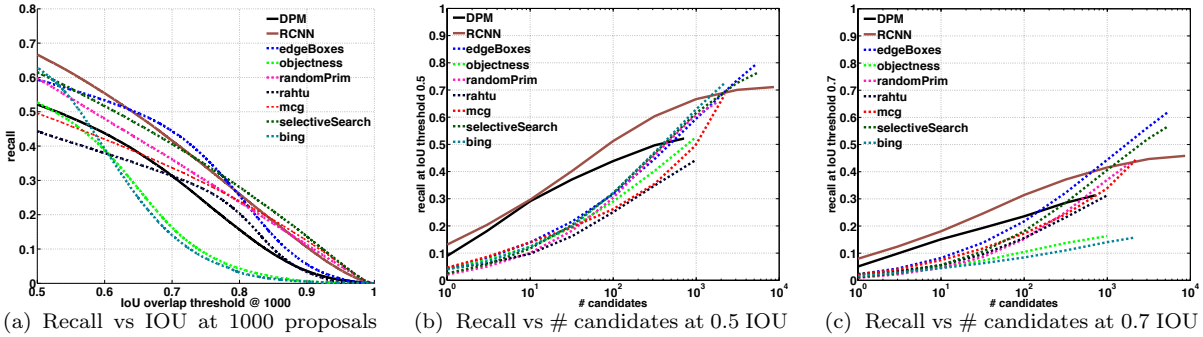
Table A.1: Object/Thing Classes in PASCAL Context

Ambiguous Classes in PASCAL Context Dataset			
artillery	escalator	ice	speedbump
bedclothes	exhibitionbooth	leaves	stair
clothestree	flame	outlet	tree
coral	guardrail	rail	unknown
dais	handrail	shelves	

Table A.2: Ambiguous Classes in PASCAL Context

Background/Stuff Classes in PASCAL Context Dataset			
atrium	floor	parterre	sky
bambooweaving	foam	patio	smoke
bridge	footbridge	pelage	snow
building	goal	plastic	stage
ceiling	grandstand	platform	swimmingpool
concrete	grass	playground	track
controlbooth	ground	road	wall
counter	hay	runway	water
court	kitchenrange	sand	wharf
dock	metal	shed	wood
fence	mountain	sidewalk	wool

Table A.3: Background/Stuff Classes in PASCAL Context



(a) Recall vs IOU at 1000 proposals for 20 PASCAL categories annotated in MS COCO validation dataset

(b) Recall vs # candidates at 0.5 IOU for 20 PASCAL categories annotated in MS COCO validation dataset

(c) Recall vs # candidates at 0.7 IOU for 20 PASCAL categories annotated in MS COCO validation dataset

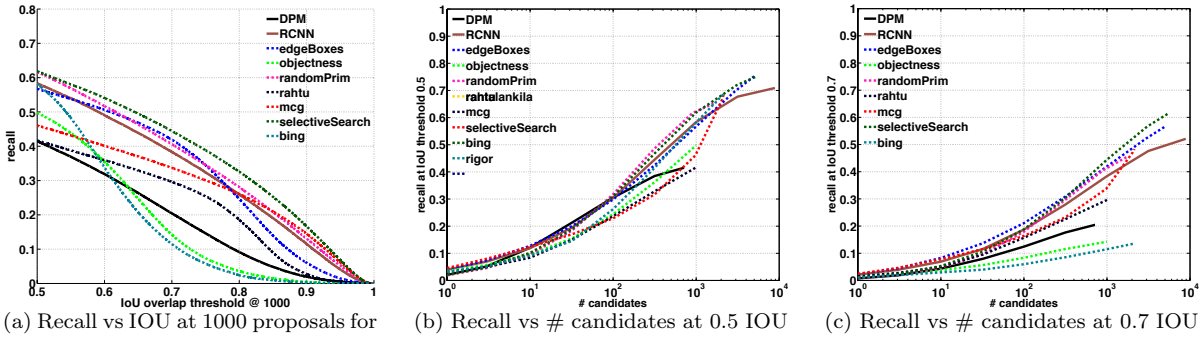
Figure A.1: Performance of various object proposal methods on different evaluation metrics when evaluated on MS COCO dataset containing annotations for only 20 PASCAL categories

at 1000 #candidates on PASCAL 20 categories. Fig. A.1b, Fig. A.1c show performance on Recall-vs-#candidates metric at 0.5 and 0.7 IOU respectively.

Similarly in Fig. A.2 and Fig. A.3, we can see the performance of all proposal methods and DMPs on these three metrics where 60 non-PASCAL and all categories respectively are annotated in the MS COCO dataset.

These metrics also demonstrate the same trend as shown by the AUC-vs-#candidates in the main paper. When only PASCAL categories are annotated (Fig. A.1), DMPs outperform all proposal methods. However, when other categories are also annotated (Fig. A.3) or the performance is evaluated specifically on the other categories (Fig. A.2), DMPs cease to be the top performers.

For the sake of completeness, we also report results on different metrics PASCAL Context (Fig. A.4, Fig. A.5 and Fig. A.6) and NYU-Depth v2 (Fig. A.7). They also show similar trends backing our claim.

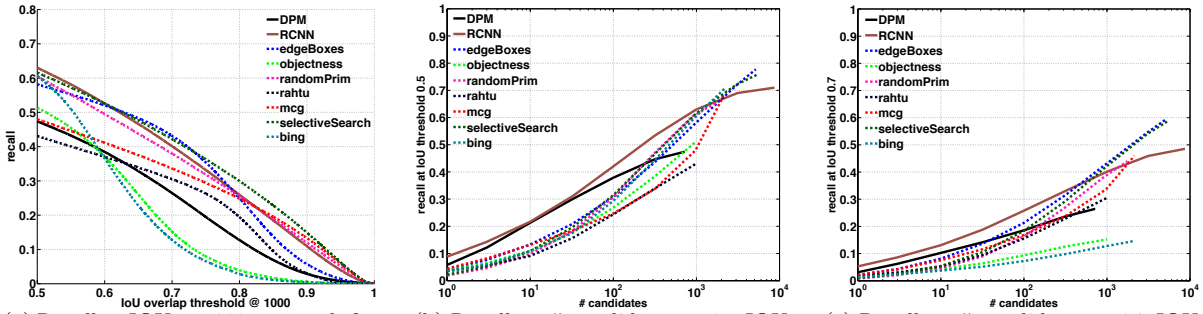


(a) Recall vs IOU at 1000 proposals for 60 non-PASCAL categories annotated in MS COCO validation dataset

(b) Recall vs # candidates at 0.5 IOU for 60 non-PASCAL categories annotated in MS COCO validation dataset

(c) Recall vs # candidates at 0.7 IOU for 60 non-PASCAL categories annotated in MS COCO validation dataset

Figure A.2: Performance of various object proposal methods on different evaluation metrics when evaluated on MS COCO dataset containing annotations for only 60 non-PASCAL categories

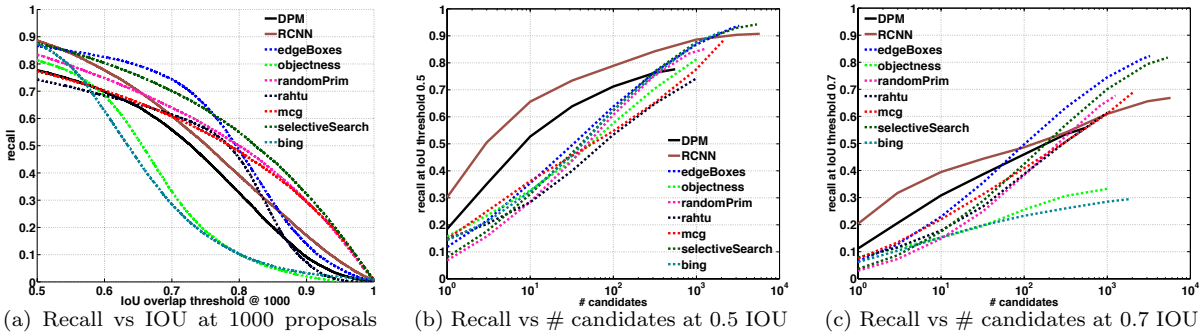


(a) Recall vs IOU at 1000 proposals for all categories annotated in MS COCO validation dataset

(b) Recall vs # candidates at 0.5 IOU for all categories annotated in MS COCO validation dataset

(c) Recall vs # candidates at 0.7 IOU for all categories annotated in MS COCO validation dataset

Figure A.3: Performance of various object proposal methods on different evaluation metrics when evaluated on MS COCO dataset containing annotations for all categories

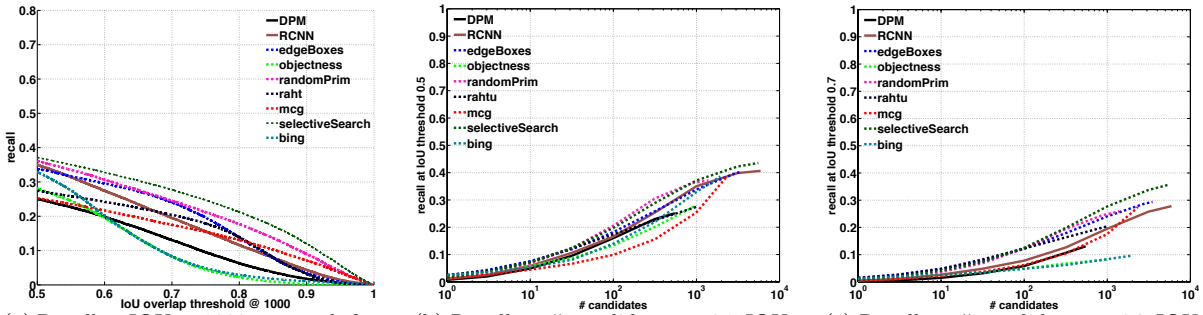


(a) Recall vs IOU at 1000 proposals for 20 PASCAL categories annotated in PASCAL Context dataset

(b) Recall vs # candidates at 0.5 IOU for 20 PASCAL annotated in PASCAL Context dataset

(c) Recall vs # candidates at 0.7 IOU for 20 PASCAL categories annotated in PASCAL Context dataset

Figure A.4: Performance of various object proposal methods on different evaluation metrics when evaluated on PASCAL Context dataset containing annotations for only 20 PASCAL categories

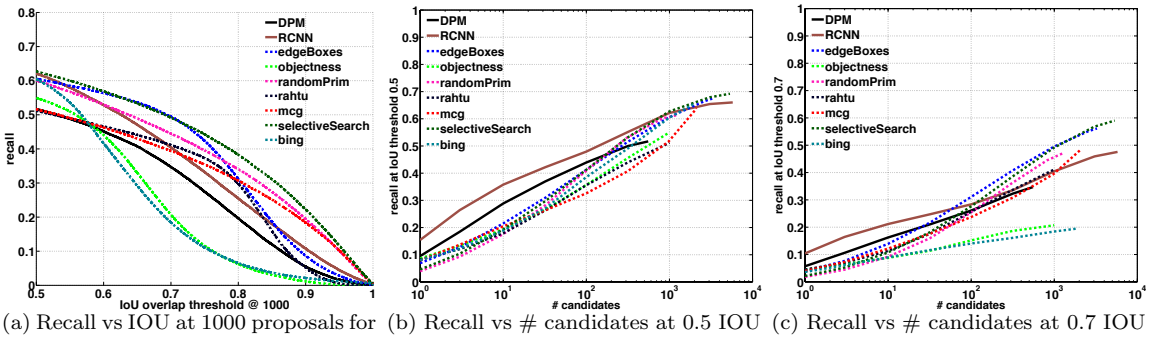


(a) Recall vs IOU at 1000 proposals for non-PASCAL categories annotated in PASCAL Context dataset

(b) Recall vs # candidates at 0.5 IOU for non-PASCAL annotated in PASCAL Context dataset

(c) Recall vs # candidates at 0.7 IOU for non-PASCAL categories annotated in PASCAL Context dataset

Figure A.5: Performance of various object proposal methods on different evaluation metrics when evaluated on PASCAL Context dataset containing annotations for only non-PASCAL categories

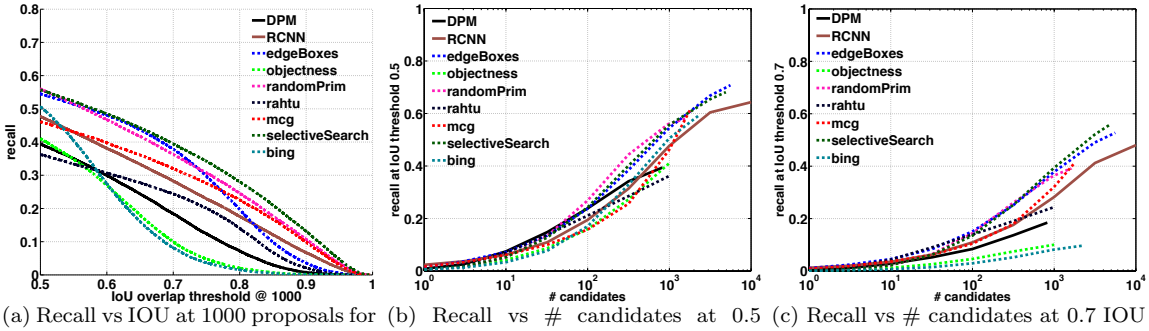


(a) Recall vs IOU at 1000 proposals for all categories annotated in PASCAL Context dataset

(b) Recall vs # candidates at 0.5 IOU for all categories annotated in PASCAL Context dataset

(c) Recall vs # candidates at 0.7 IOU for all categories annotated in PASCAL Context dataset

Figure A.6: Performance of various object proposal methods on different evaluation metrics when evaluated on PASCAL Context dataset containing annotations for all categories



(a) Recall vs IOU at 1000 proposals for all categories annotated in the NYU2 dataset

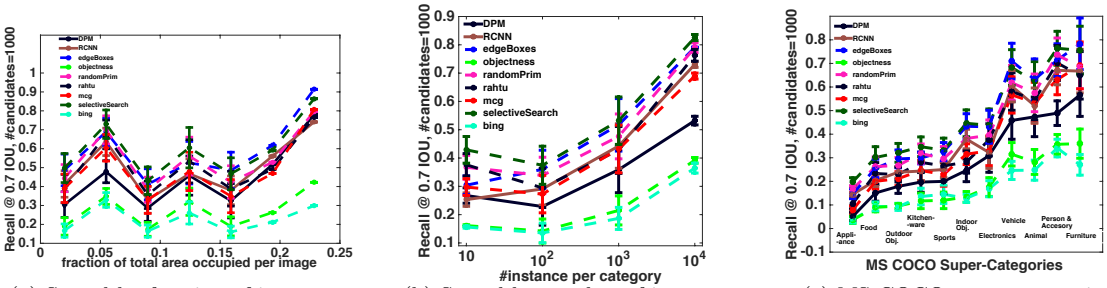
(b) Recall vs # candidates at 0.5 IOU for all categories annotated in the NYU2 dataset

(c) Recall vs # candidates at 0.7 IOU for all categories annotated in the NYU2 dataset

Figure A.7: Performance of various object proposal methods on different evaluation metrics when evaluated on NYU2 dataset containing annotations for all categories

A.3 Measuring Fine-Grained Recall

To look at a more fine-grained per category performance of proposal methods and DMPs, we presented the plots of recall values for all 80 (20 PASCAL + 60 non-PASCAL) categories for the modified PASCAL Context dataset. We have done the same experiment on MS COCO data set. 20 PASCAL categories are the same. However, the other 60 categories are different for MS COCO. It can be seen in Fig. A.8, the trends on MS COCO are more or less similar to PASCAL Context.



(a) Sorted by fraction of image area. (b) Sorted by number of instances. (c) MS COCO super-categories.

Figure A.8: Recall at 0.7 IOU for categories sorted/clustered by (a) size, (b) number of instances, and (c) MS COCO 'super-categories'.