Object-Proposal Evaluation Protocol is 'Gameable' (Supplement)

Neelima Chavali*† Harsh Agrawal* Aroma Mahendru* Dhruv Batra
Virginia Tech
{gneelima, harsh92, maroma, dbatra}@vt.edu

Abstract

The main paper demonstrated how the object proposal evaluation protocol is 'gameable' and performed some experiments to detect this 'gameability'. In this supplement, we present additional details and results which support the arguments presented in the main paper.
In section 1, we list and briefly describe the different object proposal algorithms which we used for our experiments. Following this, details of instance-level PASCAL Context are discussed in section 2. Then we present the results on nearly-fully annotated dataset, cross dataset evaluation on other evaluation metrics in section 3. We also show the per category performance of various methods on MS COCO and PASCAL Context in section 4.

1. Overview of Object Proposal Algorithms

Table 1 provides an overview of some popular object proposal algorithms. The symbol * indicates methods we have evaluated in this paper. Note that a majority of the approaches are learning based.

2. Details of PASCAL Context Annotation

As explained in section 5.1 of the main paper, PASCAL Context provides full annotations for PASCAL VOC 2010 dataset in the form of semantic segmentations. A total of 459 classes have labeled in this dataset. We split these into three categories namely Objects/Things, Background/Stuff and Ambiguous as shown in Tables 2, 4 and 3. Most 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 manually created instance level annotations for the same.

Statistics of New Annotations: We made the following observations on our new annotations:

- The number of instances we annotated for the extra 60 categories were about the same as the number of instances for annotated for 20 PASCAL categories in the original PASCAL VOC. This shows that about half the annotations were missing and thus a lot of genuine proposal candidates are not being rewarded.
- Most non-PASCAL categories occupy a small percentage of the image. This is understandable given that the dataset was curated with these categories. The other categories just happened to be in the pictures.

3. 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. 1a shows performance on Recall-vs-IOU metric at 1000 #proposals on PASCAL 20 categories. Fig. 1b, Fig. 1c show performance on Recall-*vs*.-#proposals metric at 0.5 and 0.7 IOU respectively. Similarly in Figs. 1d,1e, 1f and Figs. 1g,1h, 1i, 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*.-#proposals in the main paper. When only PASCAL categories are annotated (Figs. 1a,1b, 1c), DMPs outperform all proposal methods. However, when other categories are also annotated (Figs. 1g,1h, 1i) or the performance is evaluated specifically on the other categories (Figs. 1d,1e, 1f), DMPs cease to be the top performers.
Finally, we also report results on different metrics PASCAL Context (Fig. 2) and NYU-Depth v2 (Fig. 3). They also show similar trends, supporting the claims made in the paper.

4. Measuring Fine-Grained Recall

We also looked at a more fine-grained per-category performance of proposal methods and DMPs. Fine grained recall can be used to answer if some proposal methods are optimized for larger or frequent categories i.e. if they perform

*Equal contribution.
†Now at Amgen Inc.

Method	Code Source	Approach	Learning Involved	Metric	Datasets
*objectness**	Source code from [1]	Window scoring	Yes 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 [2]	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 from [3]	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 [1]	Recall @ t > various IoU thresholds and # proposals, AUC	PASCAL VOC 2007 test set
*randomPrim**	Source code from [4]	Segment based	Yes supervised, train on 6 PASCAL categories	Recall @ t > various IOU thresholds using 10k and 1k proposals	Pascal VOC 2007 test set/2012 trainval set on 14 categories not used in training
*mcg**	Source code from [5]	Segment based	Yes	NA, only segments were evaluated	NA (tested on segmentation dataset)
*edgeBoxes**	Source code from [6]	Window scoring	No	AUC, Recall @ t > various IOU thresholds and # proposals, Recall vs IoU	PASCAL VOC 2007 testset
*bing**	Source code from [7]	Window scoring	Yes supervised, on PASCAL VOC 2007 train set, 20 object classes/6 object classes	Recall @ t> 0.5 vs # proposals	PASCAL VOC 2007 detection complete test set/14 unseen object categories
rantalankila	Source code from [8]	Segment based	Yes	NA, only segments are evaluated	NA (tested on segmentation dataset)
Geodesic	Source code from [9]	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 [10]	Segment based	Yes, pairwise potentials between super pixels learned on BSDS-500 boundary detection dataset	NA, only segments were evaluated	NA (tested on segmentation dataset)
endres	Source code from [11]	Segment based	Yes	NA, only segments are evaluated	NA (tested on segmentation dataset)

Table 1: Properties of existing bounding box approaches. * indicates the methods which have studied in this paper.

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 2: 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	

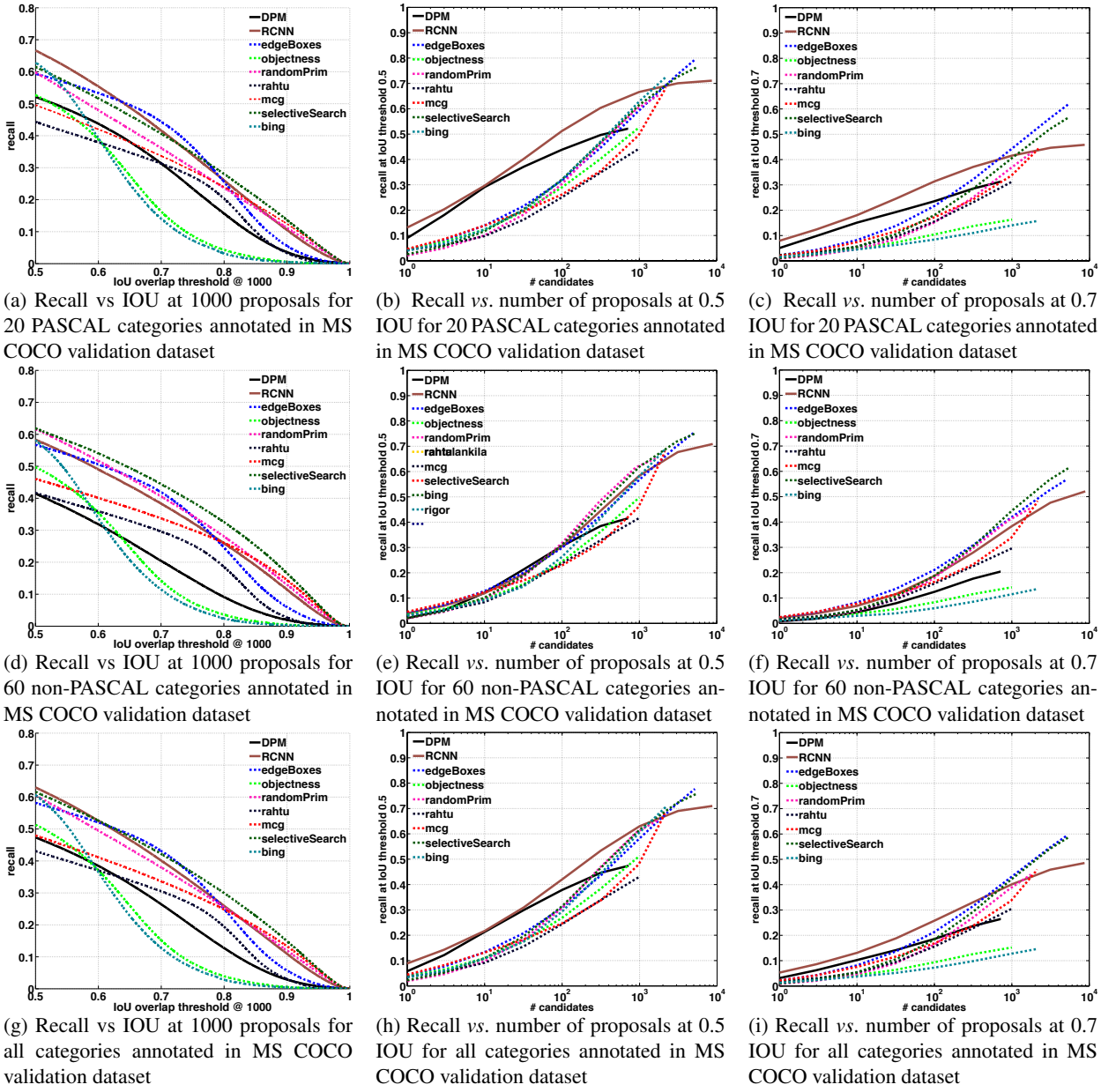
Table 3: Ambiguous Classes in PASCAL Context

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 4: Background/Stuff Classes in PASCAL Context

better or worse with respect to different object attributes like area, kinds of objects, etc. It is also easier to observe the change in performance of a particular method on frequently occurring category *vs*. rarely occurring category. We performed this experiment on instance level PASCAL Context and MS COCO datasets. 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,



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

(b) Recall *vs*. number of proposals at 0.5 IOU for 20 PASCAL categories annotated in MS COCO validation dataset

(c) Recall *vs*. number of proposals at 0.7 IOU for 20 PASCAL categories annotated in MS COCO validation dataset

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

(e) Recall *vs*. number of proposals at 0.5 IOU for 60 non-PASCAL categories annotated in MS COCO validation dataset

(f) Recall *vs*. number of proposals at 0.7 IOU for 60 non-PASCAL categories annotated in MS COCO validation dataset

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

(h) Recall *vs*. number of proposals at 0.5 IOU for all categories annotated in MS COCO validation dataset

(i) Recall *vs*. number of proposals at 0.7 IOU for all categories annotated in MS COCO validation dataset

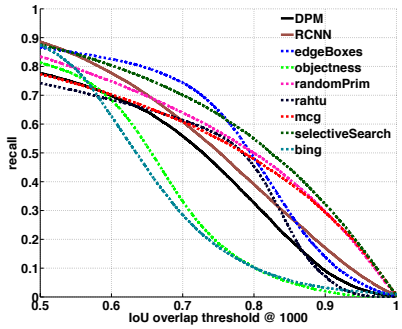
Figure 1: Performance of various object proposal methods on different evaluation metrics when evaluated on MS COCO dataset.

- Membership in 'super-categories' defined in MS COCO dataset (electronics, animals, appliance, *etc*.). 10 pre-defined clusters of objects of different kind (These clusters are the subset of 11 super-categories defined in MS COCO dataset for classifying individual classes in groups of similar objects.)

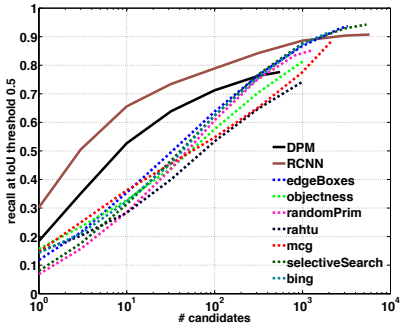
Now, we present the plots of recall for all 80 (20 PASCAL + 60 non-PASCAL) categories for the modified PASCAL Context dataset and MS COCO. Note that the non-PASCAL 60 categories are different for both the datasets.

Trends: Fig. 4 shows the performance of different proposal methods and DMPs along each of these dimensions.

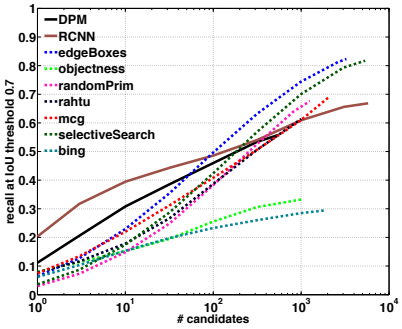
In Fig. 4a, we see that recall steadily improves perhaps as expected, bigger objects are typically easier to find than smaller objects. In Fig. 4b, we see that the recall generally increases as the number of instances increase except for one outlier category. This category was found to be 'pole' which appears to be quite difficult to recall, since poles are often occluded and have a long elongated shape, it is not surprising that this number is pretty low. Finally, in Fig. 4c we observe that some super-categories (*e.g*. outdoor objects) are hard to recall while others (*e.g*. animal, electronics) are relatively easier to recall. It can be seen in Fig. 5, the trends on MS COCO are almost similar to PASCAL Context.



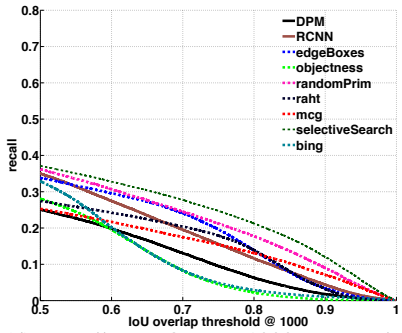
(a) Recall vs IOU at 1000 proposals for 20 PASCAL categories annotated in PASCAL Context dataset



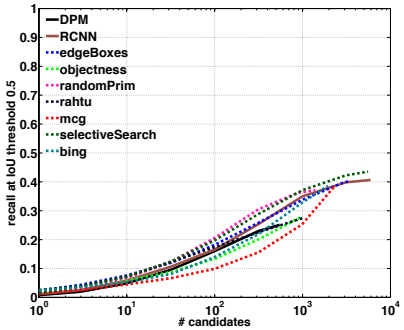
(b) Recall *vs*. number of proposals at 0.5 IOU for 20 PASCAL annotated in PASCAL Context dataset



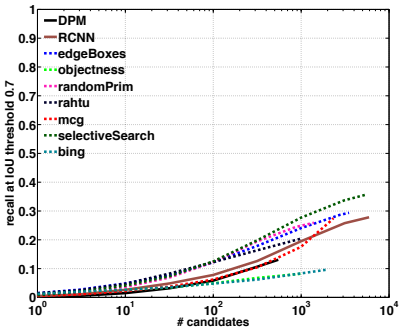
(c) Recall *vs*. number of proposals at 0.7 IOU for 20 PASCAL categories annotated in PASCAL Context dataset



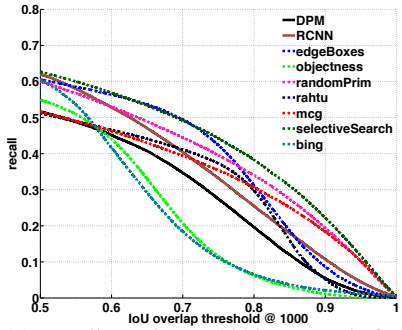
(d) Recall vs IOU at 1000 proposals for non-PASCAL categories annotated in PASCAL Context dataset



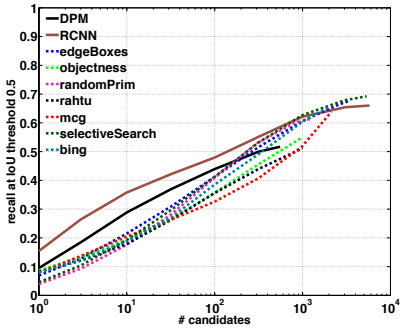
(e) Recall *vs*. number of proposals at 0.5 IOU for non-PASCAL annotated in PASCAL Context dataset



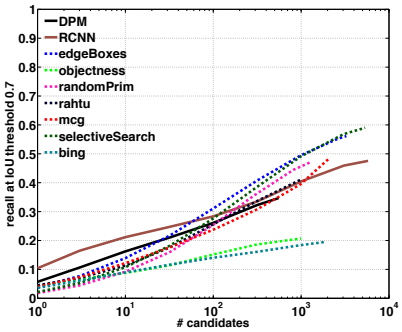
(f) Recall *vs*. number of proposals at 0.7 IOU for non-PASCAL categories annotated in PASCAL Context dataset



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



(h) Recall *vs*. number of proposals at 0.5 IOU for all categories annotated in PASCAL Context dataset



(i) Recall *vs*. number of proposals at 0.7 IOU for all categories annotated in PASCAL Context dataset

Figure 2: Performance of various object proposal methods on different evaluation metrics when evaluated on PASCAL Context dataset

References

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

[2] 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. 2

[3] 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. 2

[4] S. Manen, M. Guillaumin, and L. V. Gool, "Prime object proposals with randomized primś algorithm." https://github.com/smanenfr/rp#rp. 2

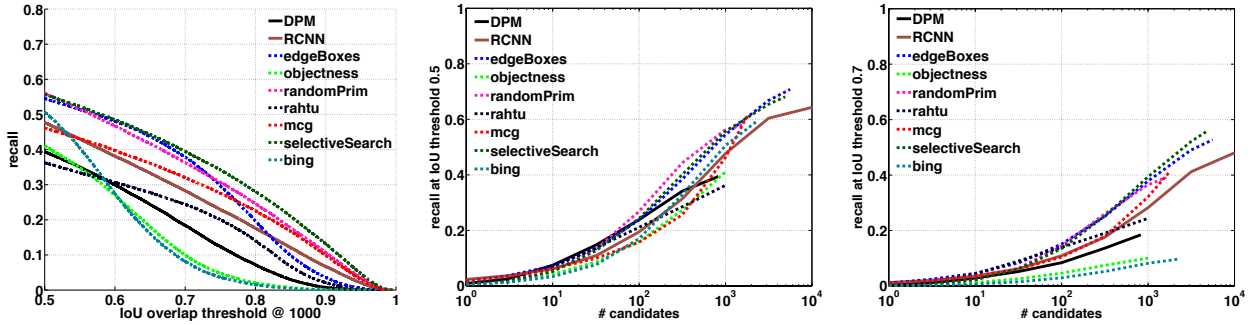
[5] P. Arbelaez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping." https://github.com/jponttuset/mcg/archive/v2.0.zip. 2

[6] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges." https://github.com/pdollar/edges. 2

[7] 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. 2

[8] 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. 2

[9] P. Krähenbühl and V. Koltun, "Geodesic object proposals." http://www.philkr.net/home/gop. 2

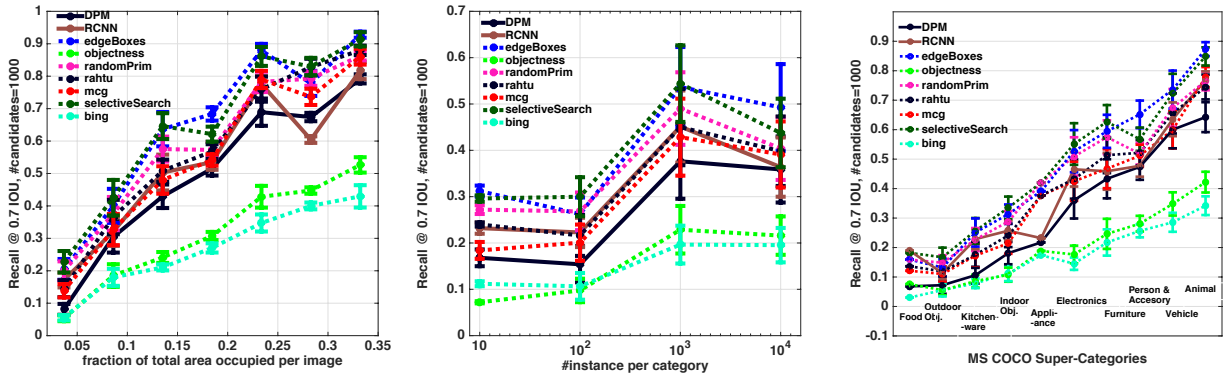


(a) Recall vs IOU at 1000 proposals for all categories annotated in the NYU2 dataset

(b) Recall *vs*. number of proposals at 0.5 IOU for all categories annotated in the NYU2 dataset

(c) Recall *vs*. number of proposals at 0.7 IOU for all categories annotated in the NYU2 dataset

Figure 3: Performance of various object proposal methods on different evaluation metrics when evaluated on NYU2 dataset containing annotations for all categories

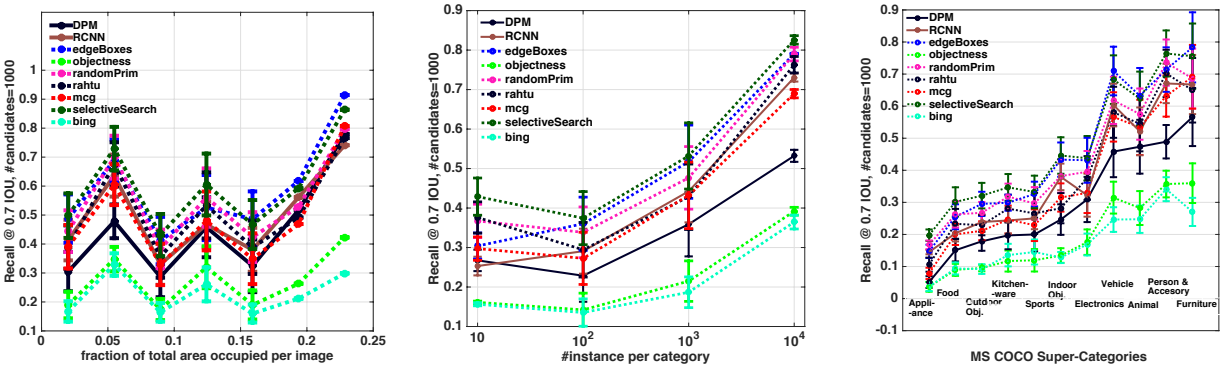


(a) Sorted by size.

(b) Sorted by the number of instances.

(c) MS COCO super-categories.

Figure 4: Recall at 0.7 IOU for categories sorted/clustered by (a) size, (b) number of instances, and (c) MS COCO 'super-categories' evaluated on PASCAL Context.



(a) Sorted by size.

(b) Sorted by the number of instances.

(c) MS COCO super-categories.

Figure 5: Recall at 0.7 IOU for categories sorted/clustered by (a) size, (b) number of instances, and (c) MS COCO 'super-categories' evaluated on PASCAL Context and MS COCO.

[10] 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. 2

[11] I. Endres and D. Hoiem, "Category-independent object proposals with diverse ranking." http://vision.cs.uiuc.edu/proposals/data/PROP_code.tar.gz. 2