max planck institut
informatik

Region Proposals

Jan Hosang, Rodrigo Benenson, Piotr Dollar, Bernt Schiele

- Who has read a proposal paper?

- Who has read a proposal paper?
- Who knows what Average Recall is?

- Who has read a proposal paper?
- Who knows what Average Recall is?
- Who has used proposals?

- Who has read a proposal paper?
- Who knows what Average Recall is?
- Who has used proposals?
- Who has written a paper about proposals or a paper using proposals?

Part I:

Motivation & different types of proposal methods

Sliding Window Approach

Slide a window and classify every location

Sliding Window Approach

Slide a window and classify every location

Sliding Window Approach

Slide a window and classify every location

Huge search space: position x scale x aspect ratio, about 10^6 windows

Problem: classification needs to be **very** fast

Conventional detection

Image

Detections with score

Sliding window

Classifier/Detector

Discard unlikely object locations

Detection proposals: a cascaded detector

Image

Image & search space

Detections with score

Proposal method

Classifier/Detector

Discard unlikely object locations

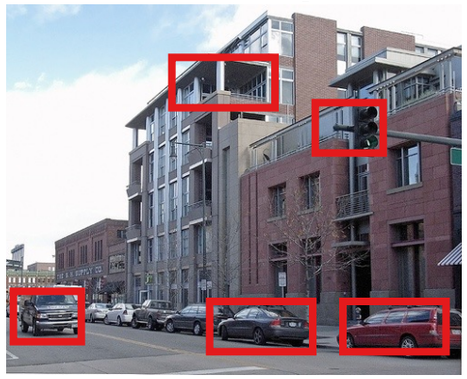
Discard unlikely object locations

Motivation: Efficiency

Image

Propose object locations

Image & search space



Score all proposals

Detections with score

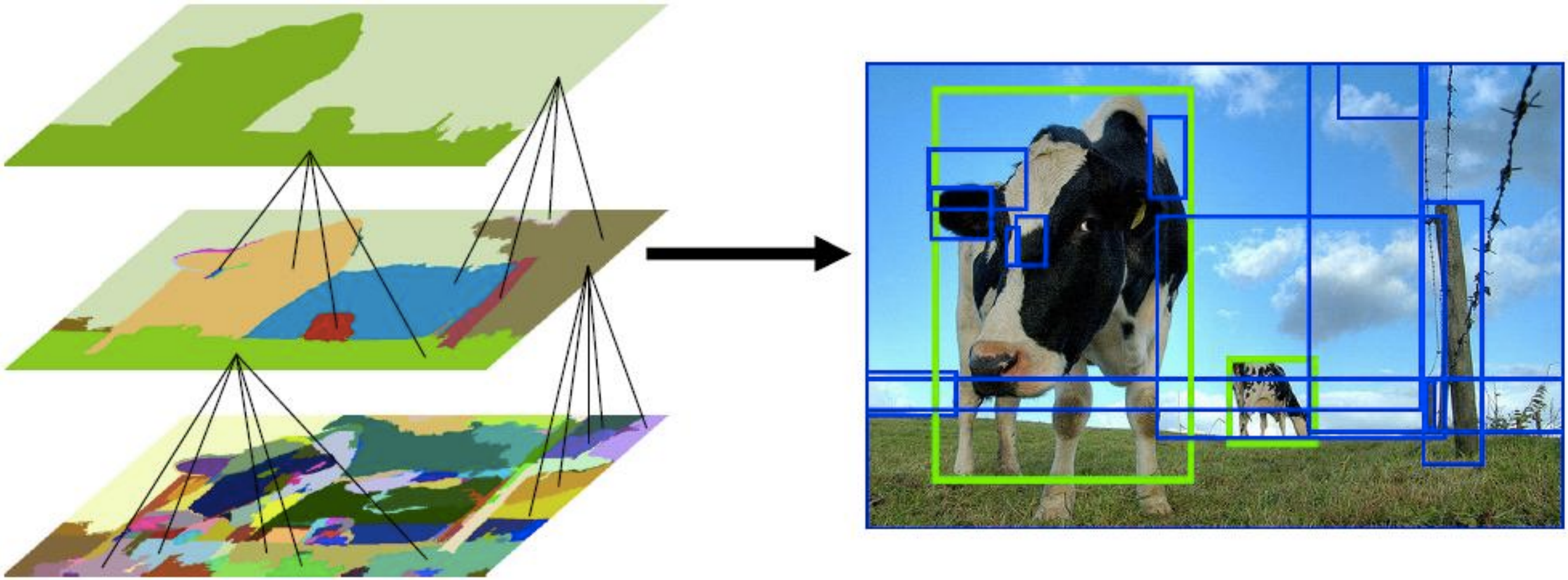
- Reducing search space allows slower classifiers
- But it comes with the cost of running a proposal method: about 0.1s — 4min per image

More side effects later

Approaches to propose object segments

- Bottom up: grouping {superpixels, pixels} into object segments
  Selective Search [ICCV ’11, IJCV ’13], Randomized Prim’s [ICCV ’13], [Rantalankila CVPR ’14], [Chang ICCV’11], CPMC [CVPR ’10, PAMI ’12], [Endres ECCV ’10, PAMI ’14], Rigor [CVPR ’14], Geodesic [ECCV ’14], MCG [CVPR ’14], …

Grouping method example: Selective Search



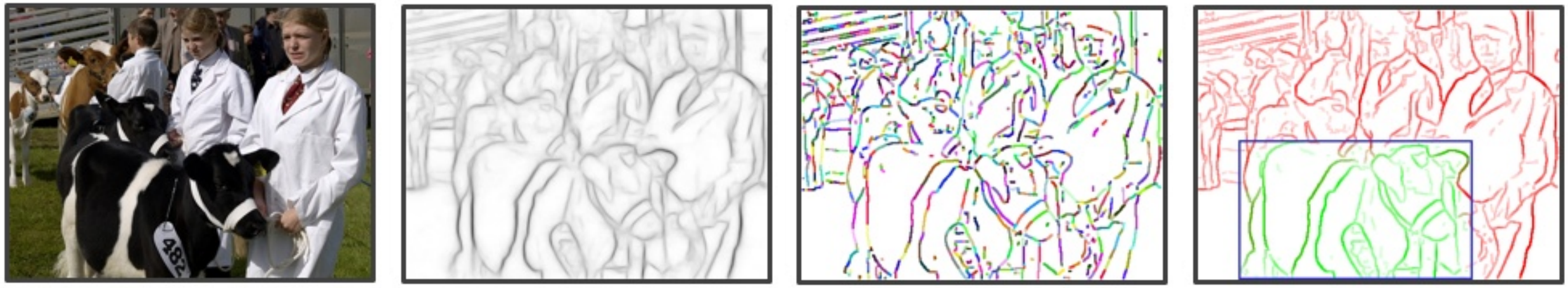
- Used information: color similarities, texture similarities, region size, region filling
- Greedy merging
- Nothing learned

Approaches to propose object segments

- Bottom up: grouping {superpixels, pixels} into object segments
  Selective Search [ICCV ’11, IJCV ’13], Randomized Prim’s [ICCV ’13], [Rantalankila CVPR ’14], [Chang ICCV’11], CPMC [CVPR ’10, PAMI ’12], [Endres ECCV ’10, PAMI ’14], Rigor [CVPR ’14], Geodesic [ECCV ’14], MCG [CVPR ’14], …

- Top down: hypothesize, then score a window/segment
  Objectness [CVPR ’10, PAMI ’12], [Rahtu ICCV ’11], Bing [CVPR ’14], EdgeBoxes [ECCV ’14], [Feng ICCV ’11], [Zhang CVPR ’11], Randomized Seeds [ICCV ’13], …

Window scoring example: Edge Boxes



Extract edges Group edges

- Extract edges
- Minimize the number of edges that cross the window boundary
  - Sliding window search
  - Refine location, scale, aspect ratio
- Also nothing learned

Approaches to propose object segments

- Bottom up: grouping {superpixels, pixels} into object segments
  Selective Search [ICCV ’11, IJCV ’13], Randomized Prim’s [ICCV ’13], [Rantalankila CVPR ’14], [Chang ICCV’11], CPMC [CVPR ’10, PAMI ’12], [Endres ECCV ’10, PAMI ’14], Rigor [CVPR ’14], Geodesic [ECCV ’14], MCG [CVPR ’14], …

- Top down: hypothesize, then score a window/segment
  Objectness [CVPR ’10, PAMI ’12], [Rahtu ICCV ’11], Bing [CVPR ’14], EdgeBoxes [ECCV ’14], [Feng ICCV ’11], [Zhang CVPR ’11], Randomized Seeds [ICCV ’13], …

- Miscellanea: eg. non-parametric, convnets
  Shape sharing [ECCV ’12], Multibox [CVPR ’14], DeepMask [NIPS ’15], …

Why are convnets different?

- Many methods are complexity controlled to generalize to unseen classes
  - Simple features
  - Simple classifier, maybe even no learning
- Huge VGG convnet with tens of millions of parameters?
  - Possibility to learn n detectors inside
  - Bigger drop in performance when generalizing to new classes
- But you could also count it as window scoring

More on that later

Part II:

Detection proposal evaluation, metrics, analysis

Detection proposals: a cascaded detector

Image

Image & search space

Detections with score

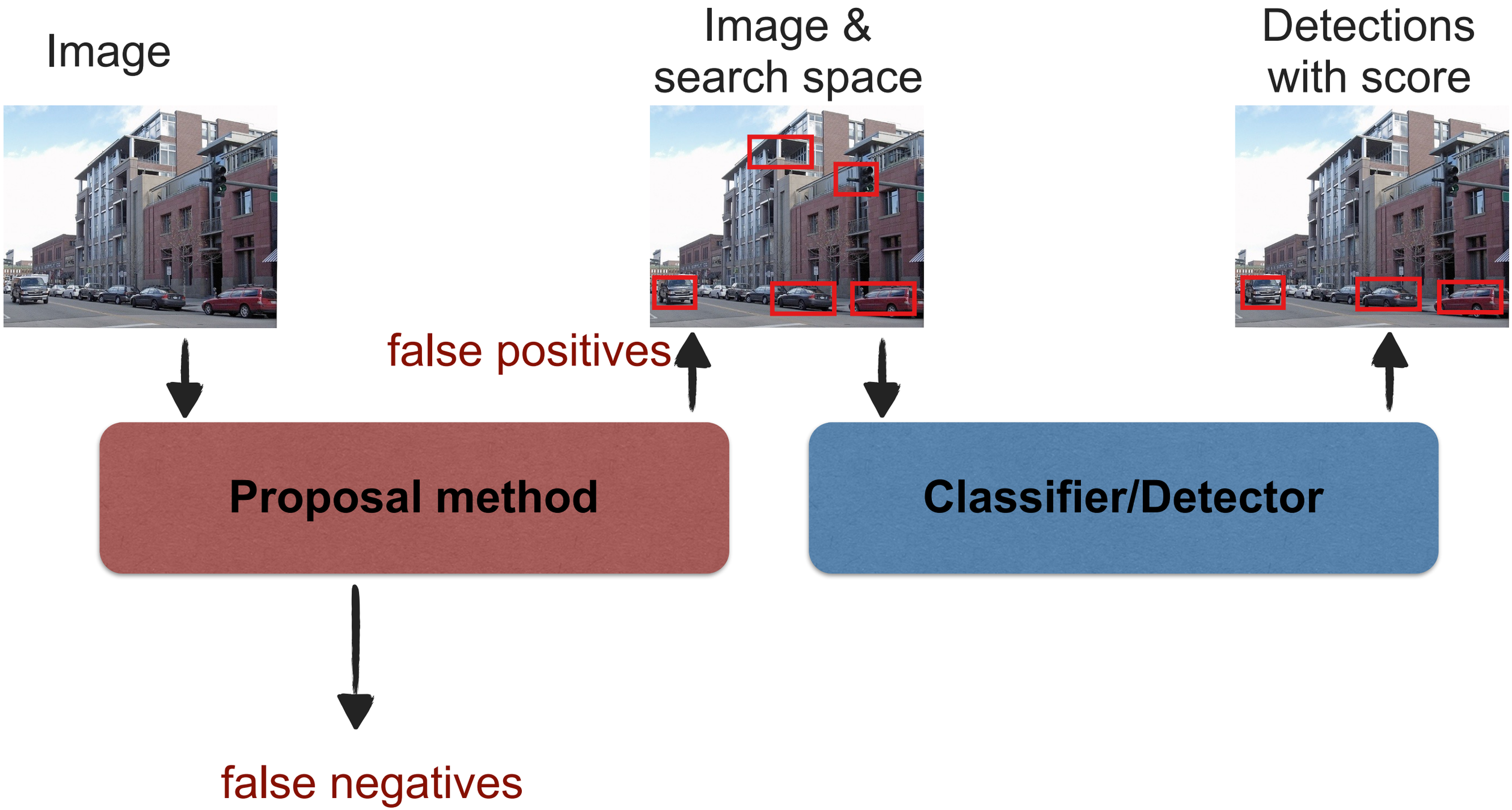
Proposal method

Classifier/Detector

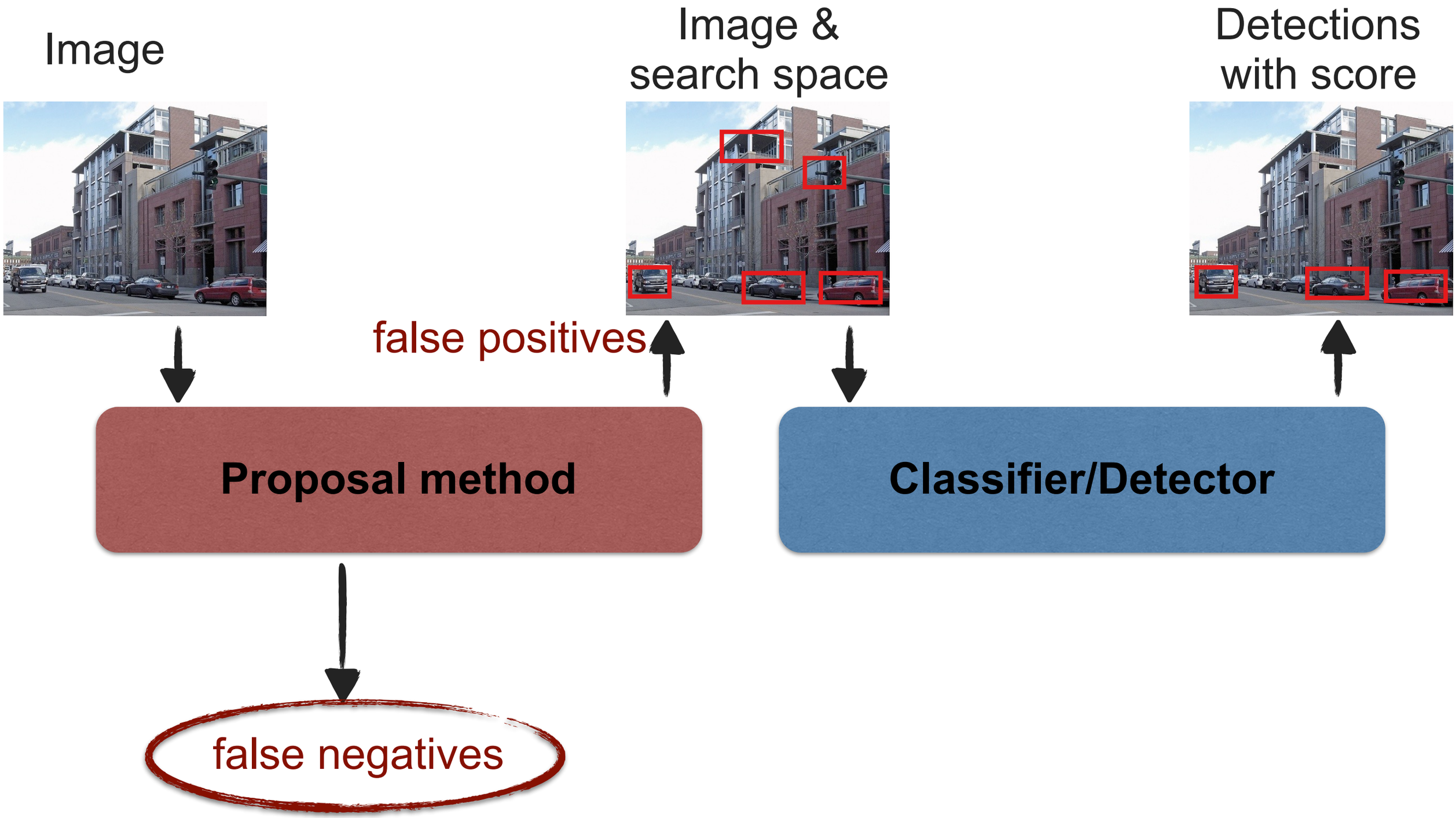
Discard unlikely object locations

Discard unlikely object locations

Detection proposals: types of mistakes



Detection proposals: types of mistakes

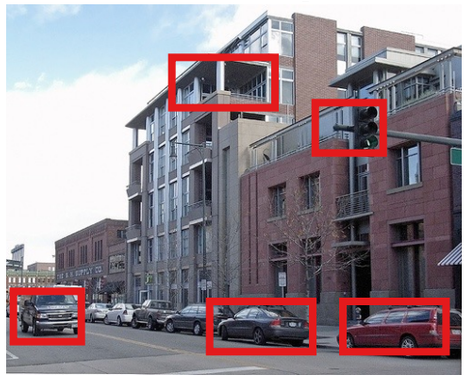


Detection proposals: evaluation

Image

Propose object locations

Image & search space



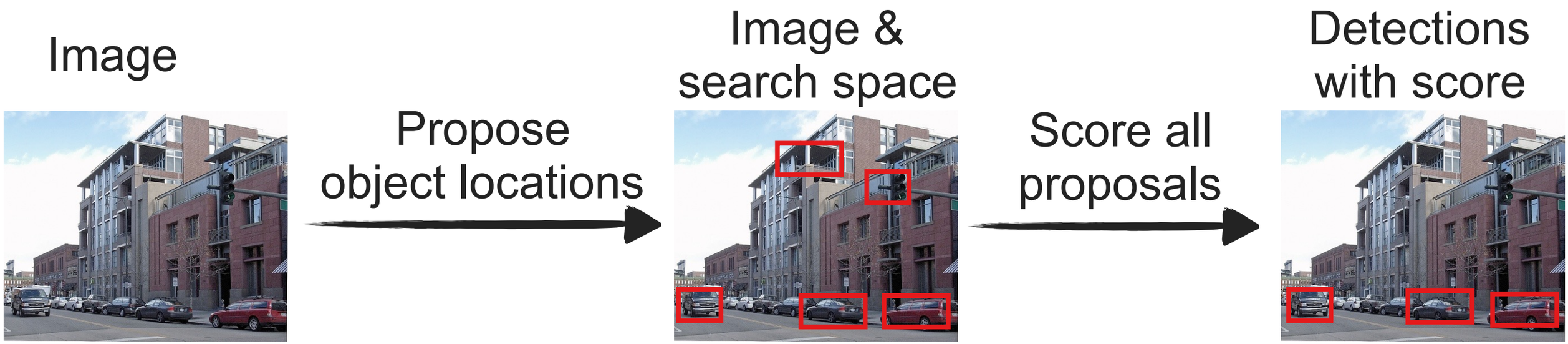
Score all proposals

Detections with score

Methods need to provide

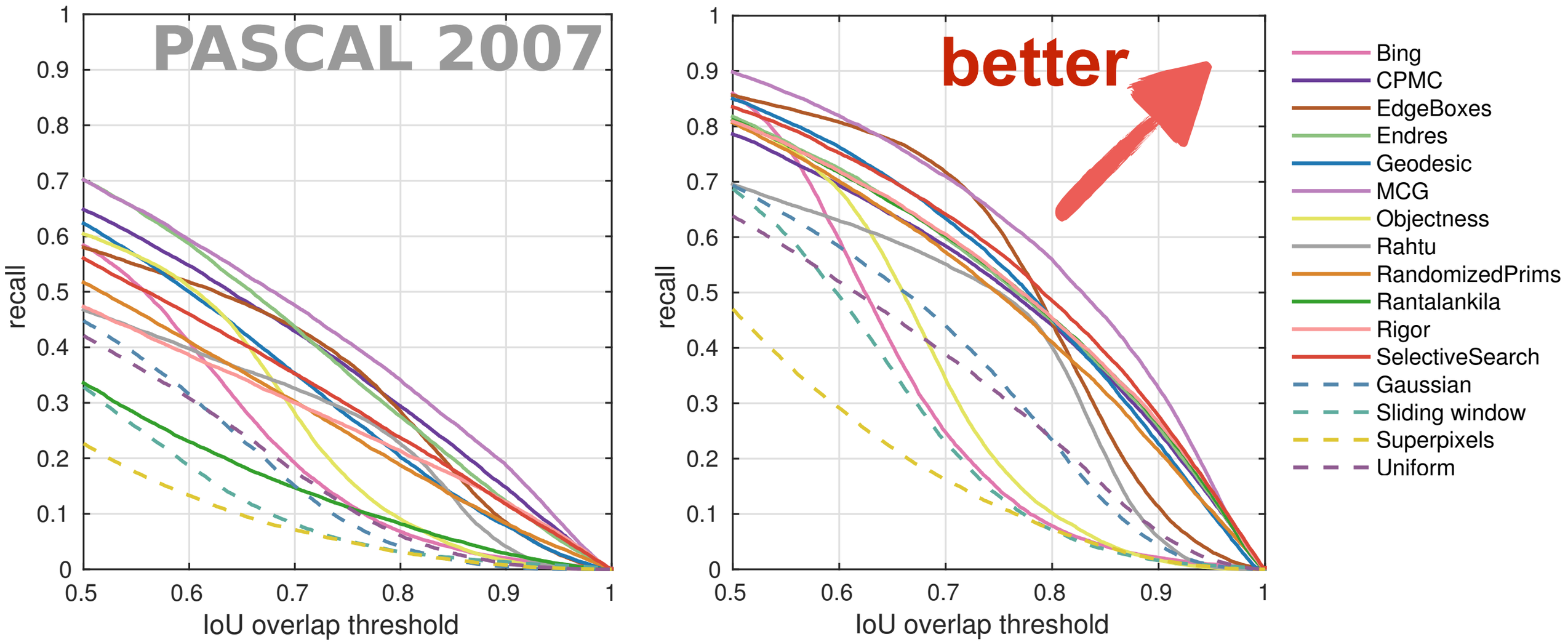
- As few FNs as possible: **high recall**
- As well localized as possible: **good localization**
- As few FPs as possible: **few proposals**

Why do these three matter?



- **High recall**
  - Missing recall cannot be undone
- **Good localization**
  - Detectors give better scores for well localized windows, ie. better AP curve
- **Few proposals**
  - This cuts down search space
  - And reduces the number of mistakes the detector can make

Typical evaluation: fix #proposals, recall vs. IoU



(a) 100 proposals per image.

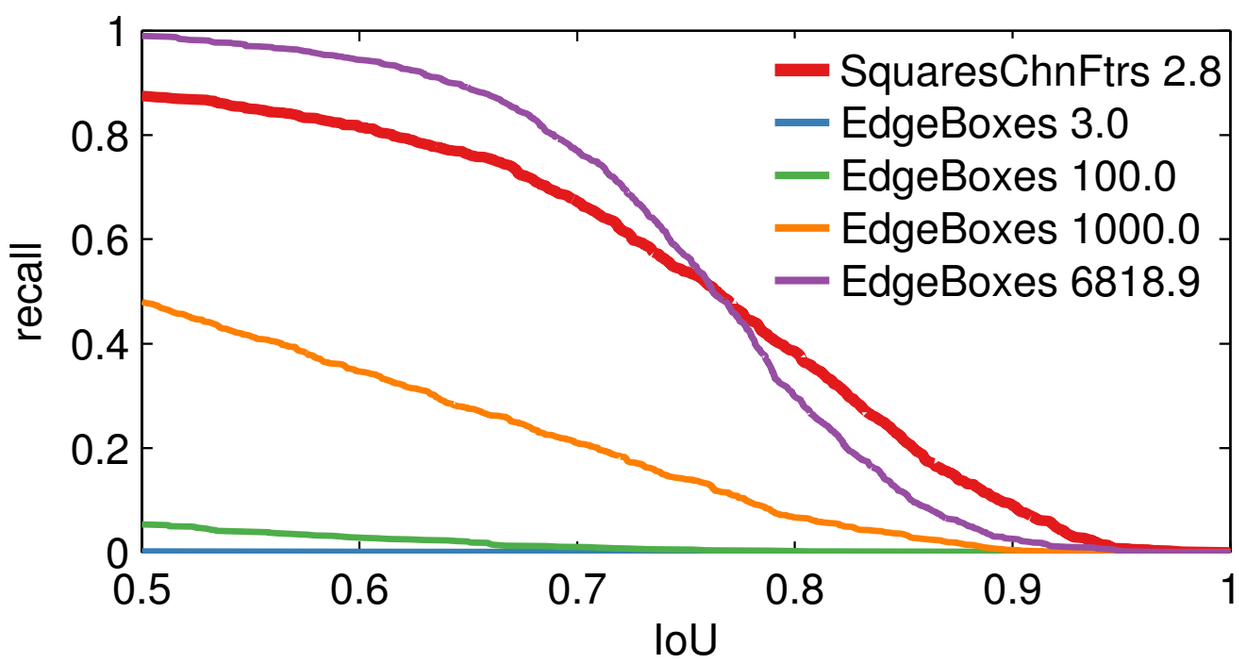
(b) 1 000 proposals per image.

Proposals methods are just bad detectors

- In the early days authors cared about class generalization
  - In the Objectness paper **[CVPR ’10]** set of training/test classes were disjoint
  - Methods had either simple features or small model capacity to generalize

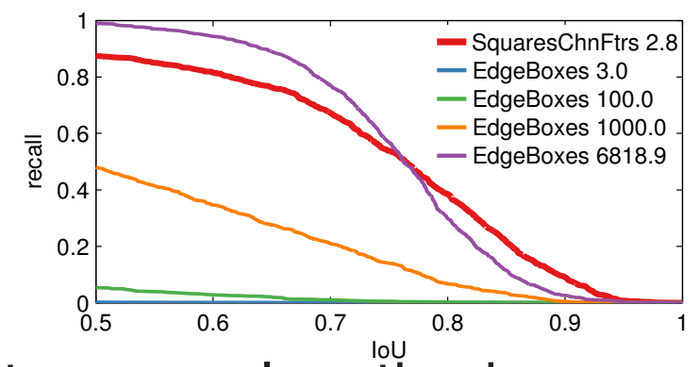
Proposals methods are just bad detectors

- In the early days authors cared about class generalization
  - In the Objectness paper **[CVPR '10]** set of training/test classes were disjoint
  - Methods had either simple features or small model capacity to generalize
- But on Pascal/COCO-type benchmarks
  - Who cares about class generalization?
  - You want high recall for few proposals
  - At the end of the day a high recall detector is the best proposal method (e.g. Faster R-CNN)



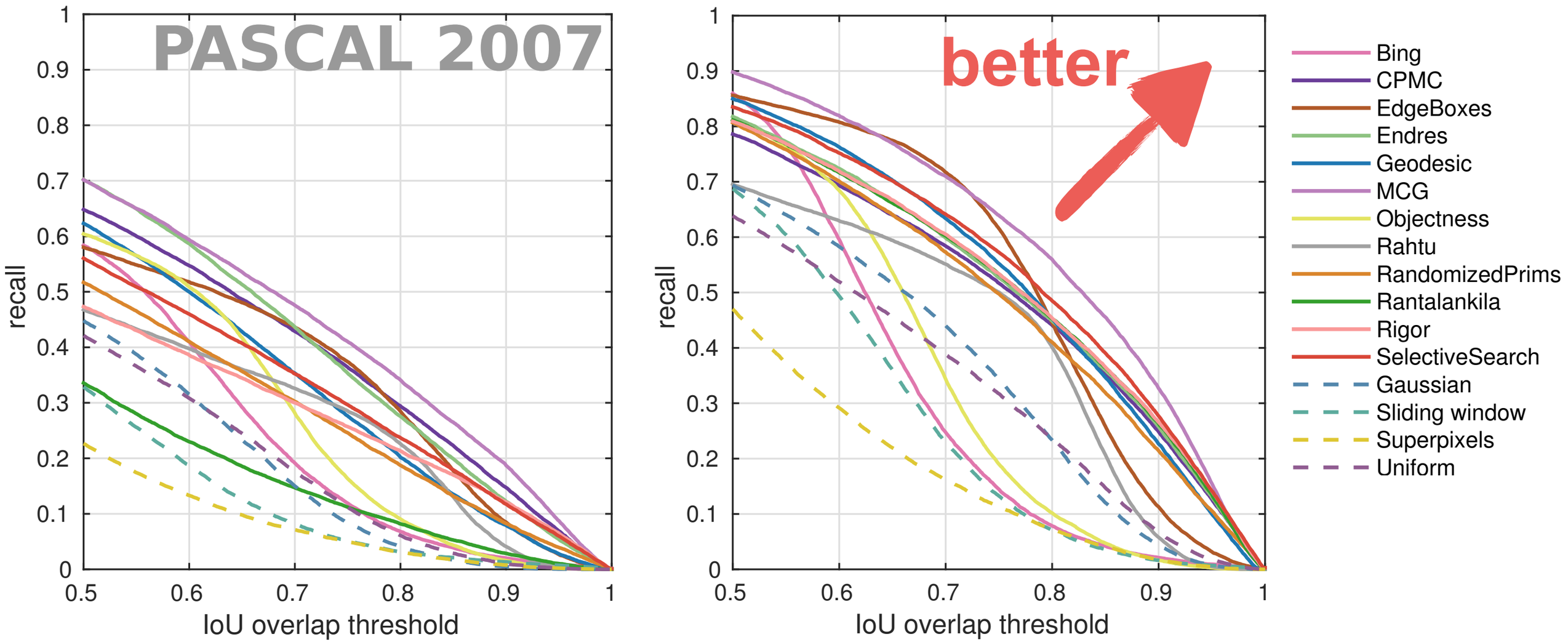
Proposals methods are just bad detectors

- In the early days authors cared about class generalization
  - In the Objectness paper **[CVPR '10]** set of training/test classes were disjoint
  - Methods had either simple features or small model capacity to generalize
- But on Pascal/COCO-type benchmarks
  - Who cares about class generalization?
  - You want high recall for few proposals
  - At the end of the day a high recall detector is the best proposal method (e.g. Faster R-CNN)



- What you want to detect might not be an object (and violate the "common appearance assumption")
  - Parts of objects: shoulder, car door, …
  - Groups of objects

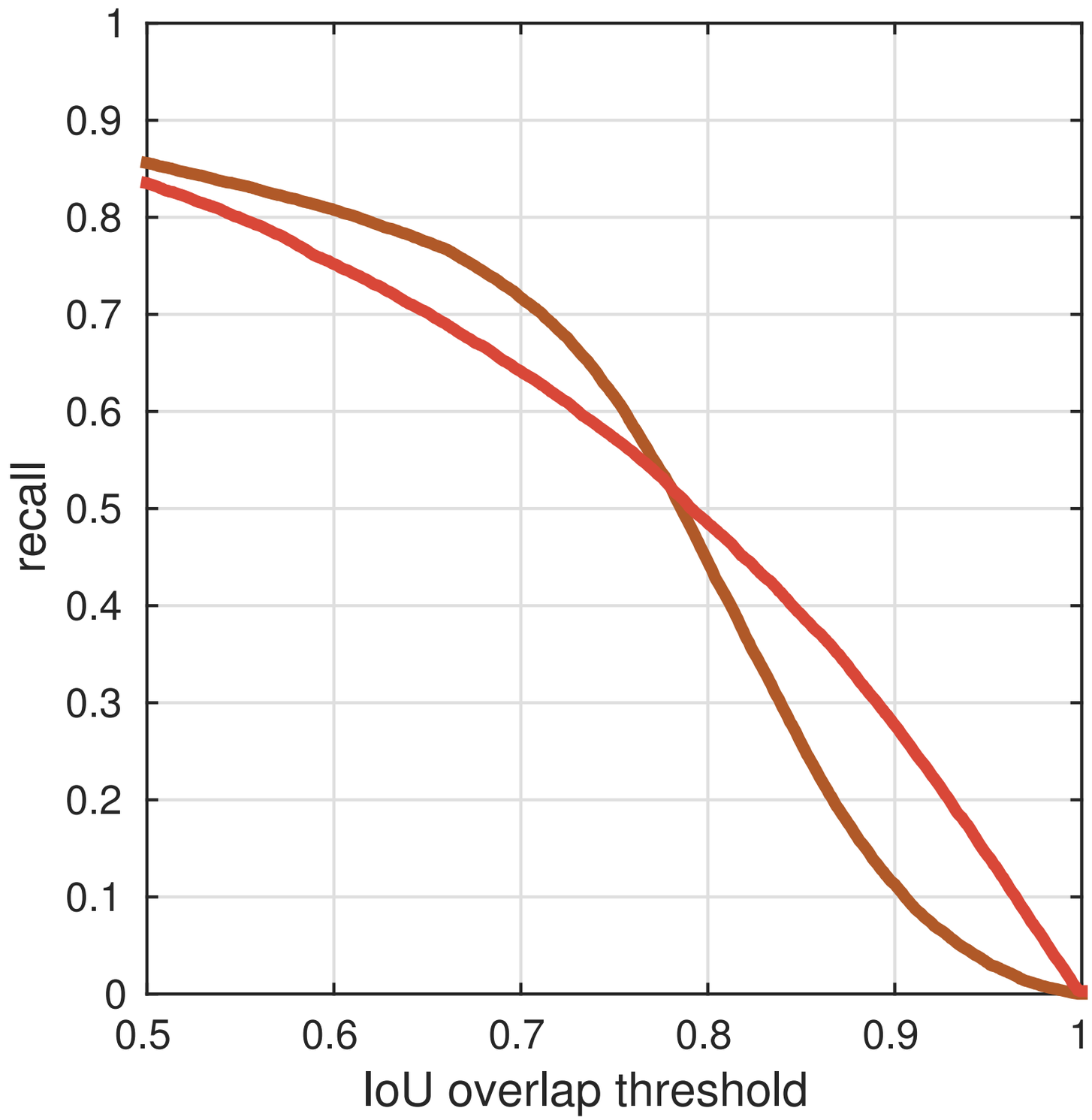
Typical evaluation: fix #proposals, recall vs. IoU



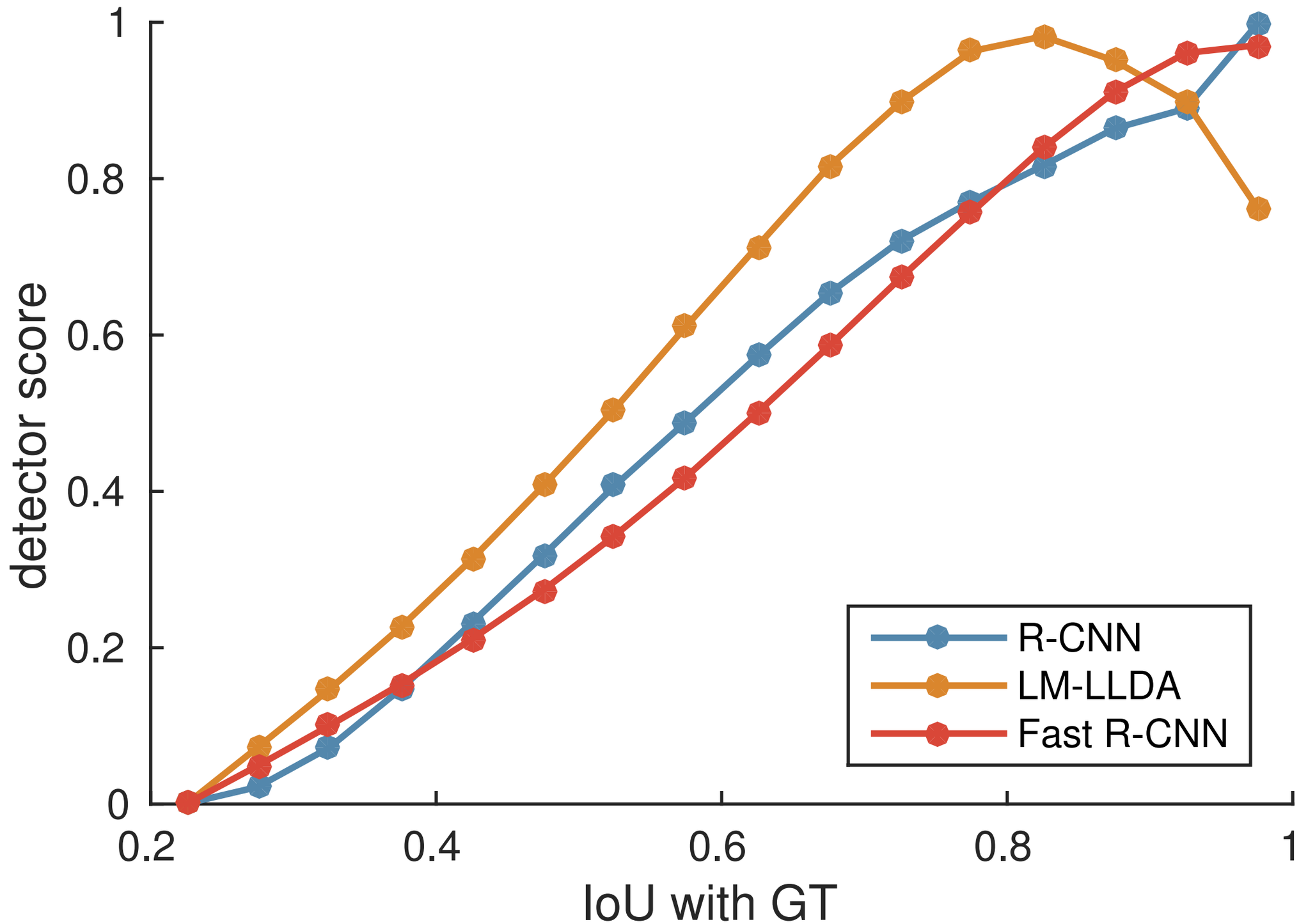
(a) 100 proposals per image.

(b) 1 000 proposals per image.

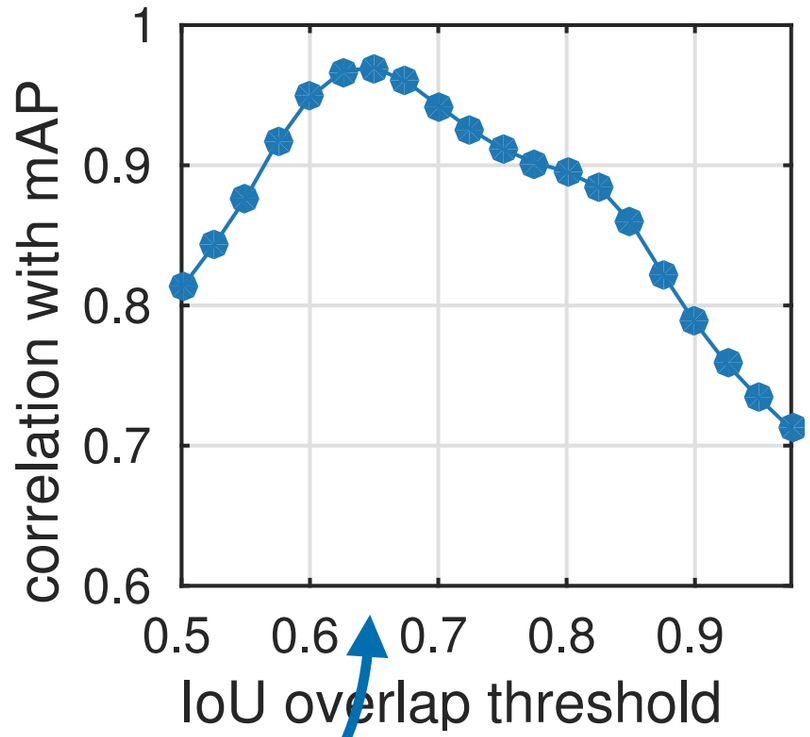
But how important is recall vs. localization?

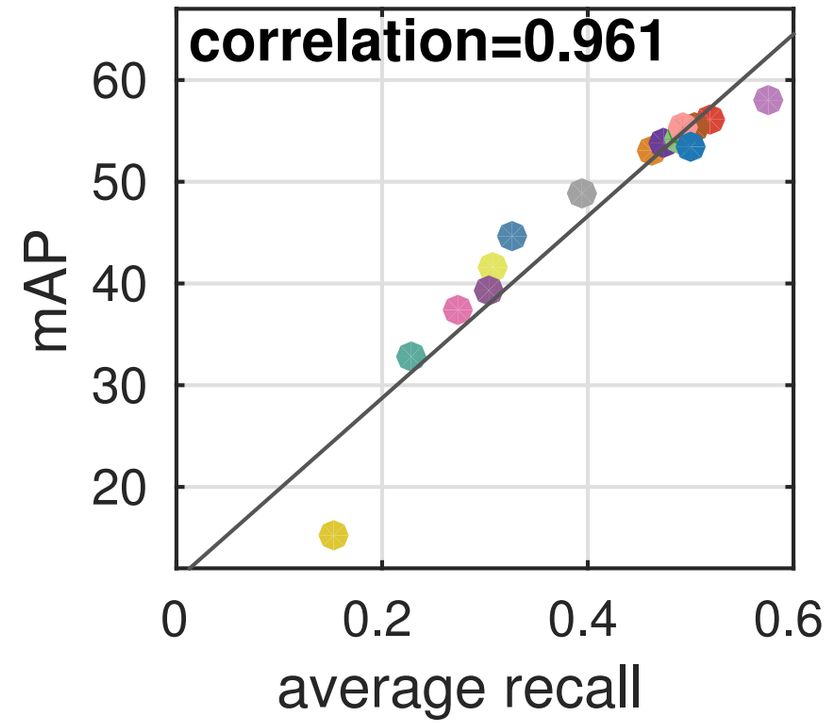


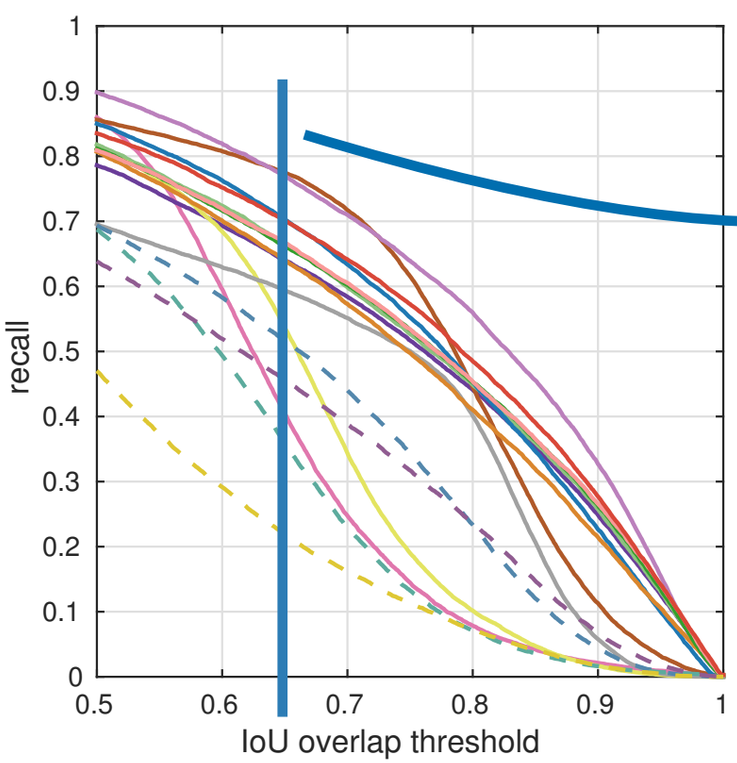
There is no one threshold, Better localization helps linearly



Actually, we're interested in mAP



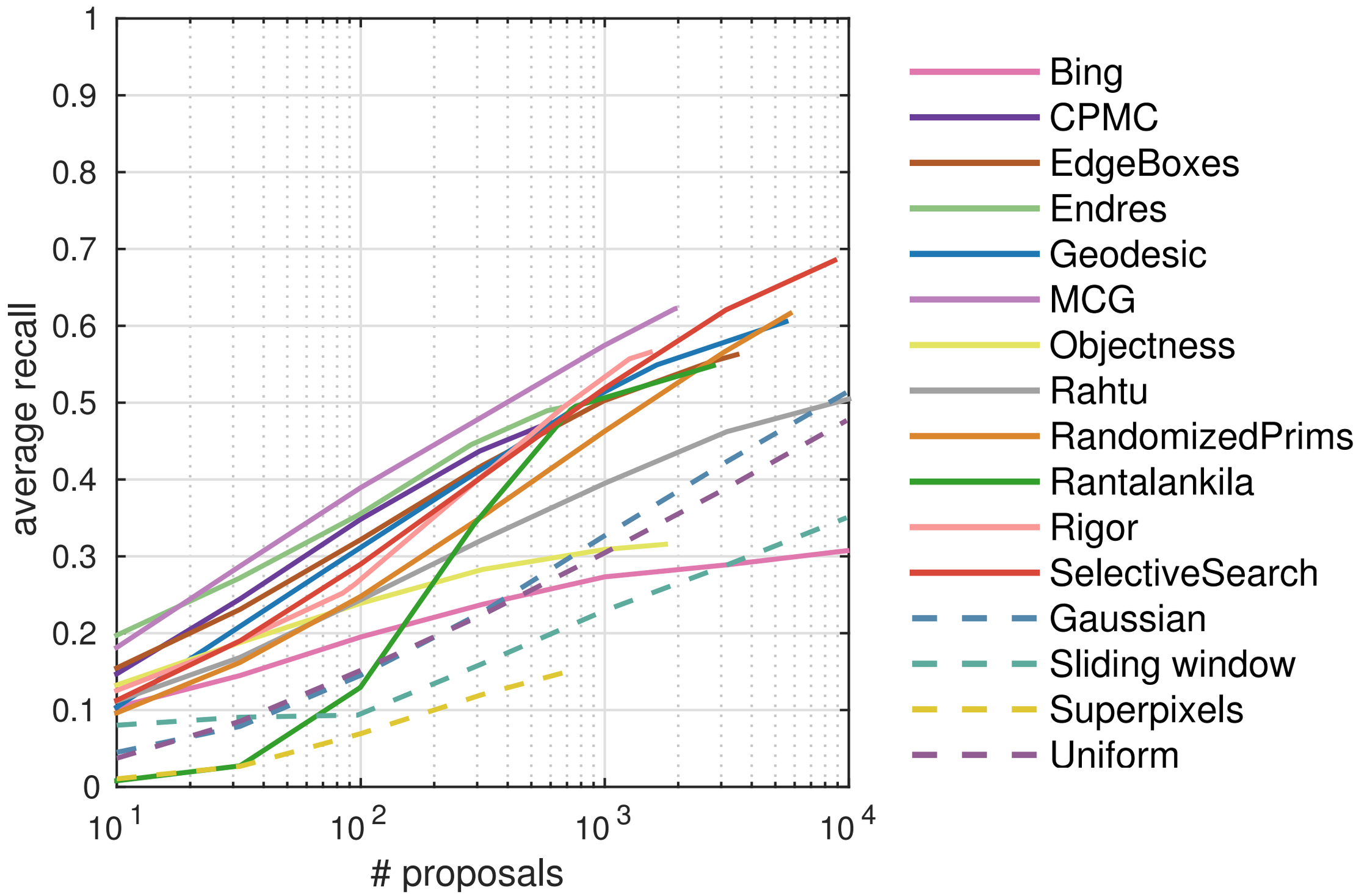




$$\mathrm{AR} = 2 \int_{0.5}^{1} \mathrm{recall}(o)\,\mathrm{d}o$$

$$= \underset{o \in [0.5, 1.0]}{\mathrm{average}} \quad \mathrm{recall}(o)$$

Trends across number of proposals



Average Recall caveat

- Correlates well if you don't change the number of proposals

$$\mathrm{AR} = 2 \int_{0.5}^{1} \mathrm{recall}(o)\,\mathrm{d}o$$
$$= \underset{o \in [0.5, 1.0]}{\mathrm{average}} \quad \mathrm{recall}(o)$$

- But extreme cases don't look as you'd expect

	AR	# proposals	mean AP
EdgeBoxesAR	53	1000	67.8
+ nms oracle	53	~10	75.0
+ gt oracle	100	~1003	71.2
+ both oracles	100	~10	78.1

Take-home messages

- Proposals should provide high recall and good localization with few proposals
  - In papers present full range of operating points
  - AR is a good summary number, but has limitations when it comparing across different number of proposals
- Sometimes a detector is the better proposal method
  - If you care about few classes and don't need to generalize
  - If the proposal methods don't work on your targets

Thanks for your attention!

Questions?