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
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

Sliding window

Classifier/Detector

Discard unlikely object locations

Detections with score
Detection proposals: a cascaded detector

Image

Proposal method

Discard unlikely object locations

Image & search space

Classifier/Detector

Discard unlikely object locations

Detections with score
Motivation: Efficiency

• 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
- 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

1. **Image**
   - Discard unlikely object locations

2. **Proposal method**
   - Discard unlikely object locations

3. **Image & search space**
   - Classifier/Detector

4. **Detections with score**
   - Discard unlikely object locations
Detection proposals: types of mistakes

Image & search space

Proposal method

Classifier/Detector

false positives

false negatives

false negatives
Detection proposals: types of mistakes

- **Image**
- **Proposal method**
- **Image & search space**
- **Classifier/Detector**
- **Detections with score**

**Types of mistakes:**
- False positives
- False negatives
Detection proposals: evaluation

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)

![Graph showing recall versus IoU for different proposal methods]
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?

![Graph showing the relationship between IoU overlap threshold and recall. The graph displays two curves, one in red and one in brown, illustrating how recall decreases as the IoU overlap threshold increases.]
There is no one threshold, Better localization helps linearly

![Graph showing detector score vs IoU with GT for R-CNN, LM-LLDA, and Fast R-CNN]
Actually, we’re interested in mAP

\[
AR = 2 \int_{0.5}^{1} \text{recall}(o) \, do = \text{average}_{o \in [0.5, 1.0]} \text{recall}(o)
\]
Trends across number of proposals

![Graph showing trends across number of proposals](image)

- Bing
- CPMC
- EdgeBoxes
- Endres
- Geodesic
- MCG
- Objectness
- Rahtu
- RandomizedPrims
- Rantalankila
- Rigor
- SelectiveSearch
- Gaussian
- Sliding window
- Superpixels
- Uniform

The PASCAL VOC 2007 test set, on which the PASCAL Results in figure 6 and 7 present a consistent trend for a fixed number of proposals per image.

We evaluate recall on the full PASCAL VOC 2007 test set, which contains annotations for all categories at test time. We care about categories and subsets of data, so one proposal cannot cover two objects. Note that while we compute a matching between proposals and ground truth, the goal is to measure maximum recall.

In contrast to [12], we report a novel metric, the average recall (AR) between IoU proposals is varied (figure 7a, 7b). Finally, we define and report a novel metric, the average recall (AR) between IoU proposals, the fraction of ground truth annotations covered for a fixed IoU threshold, proposal recall as the number of proposals per image.

Another common and complementary metric is, as the intersection over union (IoU) threshold is varied for a fixed number of proposals, the fraction of ground truth annotations covered.

Since most metrics (class confusion, background confusion, etc.) do not apply. Instead, one of the primary metrics for evaluating proposals is, for a fixed number of proposals per image.

Context (MS COCO) [6] has more objects per image, smaller objects, and finer-grained versions of the PASCAL ones. They include additional types of animals (e.g. crustaceans), food items (e.g. hot-dogs), household items (e.g. diapers), and other diverse object categories.

Although ImageNet has more classes than PASCAL, it is still similar in statistics like number of objects per image and size of objects. Microsoft Common Objects in Context (MS COCO) has only 180 object categories, yet detection proposal methods may mark large image areas including a lot of background. We evaluate the recall of this dataset to further investigate potential biases of proposal methods.

We evaluate the recall on all annotations excluding the "crowd" annotations which may mark large image areas including a lot of background. This is to look into the potential biases of proposal methods. We evaluate the recall on all categories and all ground truth bounding boxes, including "difficult" ones, since our evaluation we include all categories in over 20 000 unconstrained images. For the purpose of proposal evaluation we include all object categories present in the PASCAL test set, which includes annotations for 20 000 images. It should be noted that these images (e.g. hot-dogs), household items (e.g. diapers), and other diverse object categories.

We evaluate the recall on the full PASCAL VOC 2007 test set to compare how different methods perform against each other.

The PASCAL VOC 2007 test set, on which the PASCAL Results in figure 6 and 7 present a consistent trend for a fixed number of proposals per image.

We evaluate recall on the full PASCAL VOC 2007 test set, which contains annotations for all categories at test time. We care about categories and subsets of data, so one proposal cannot cover two objects. Note that while we compute a matching between proposals and ground truth, the goal is to measure maximum recall.

In contrast to [12], we report a novel metric, the average recall (AR) between IoU proposals is varied (figure 7a, 7b). Finally, we define and report a novel metric, the average recall (AR) between IoU proposals, the fraction of ground truth annotations covered for a fixed IoU threshold, proposal recall as the number of proposals per image.

Another common and complementary metric is, as the intersection over union (IoU) threshold is varied for a fixed number of proposals, the fraction of ground truth annotations covered.

Since most metrics (class confusion, background confusion, etc.) do not apply. Instead, one of the primary metrics for evaluating proposals is, for a fixed number of proposals per image.

Context (MS COCO) [6] has more objects per image, smaller objects, and finer-grained versions of the PASCAL ones. They include additional types of animals (e.g. crustaceans), food items (e.g. hot-dogs), household items (e.g. diapers), and other diverse object categories.
Average Recall caveat

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

\[
AR = 2 \int_{0.5}^{1} \text{recall}(o) \, do \\
= \text{average}_{o \in [0.5,1.0]} \text{recall}(o)
\]

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

<table>
<thead>
<tr>
<th></th>
<th>AR</th>
<th># proposals</th>
<th>mean AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>EdgeBoxes AR</td>
<td>53</td>
<td>1000</td>
<td>67.8</td>
</tr>
<tr>
<td>+ nms oracle</td>
<td>53</td>
<td>~10</td>
<td>75.0</td>
</tr>
<tr>
<td>+ gt oracle</td>
<td>100</td>
<td>~1003</td>
<td>71.2</td>
</tr>
<tr>
<td>+ both oracles</td>
<td>100</td>
<td>~10</td>
<td>78.1</td>
</tr>
</tbody>
</table>
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 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?