PointRend: Image Segmentation as Rendering

——from CVPR2021 By FAIR

Speaker: Gong, Qiqi

Overview

- View image segmentation as a **rendering(渲染)** problem
- Propose a **module:** PointRend, applied to instance & semantic segmentation
    - Instance segmentation: Mask R-CNN + PointRend
    - Semantic segmentation: DeepLabv3 + PointRend
    - Benchmarks: COCO & Cityscapes

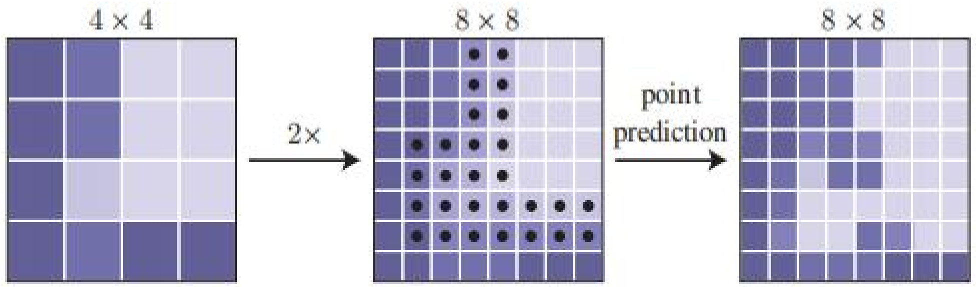
Rendering

- Rendering (渲染): A concept in computer graphics
- Displaying a model (3D) on an 2D image
- An analogy
  - Rendering: render 3D model on a regular grid
  - Segmentation: “render” segmentation output from an underlying continuous entity
  - **Core: boundary parts**

Method

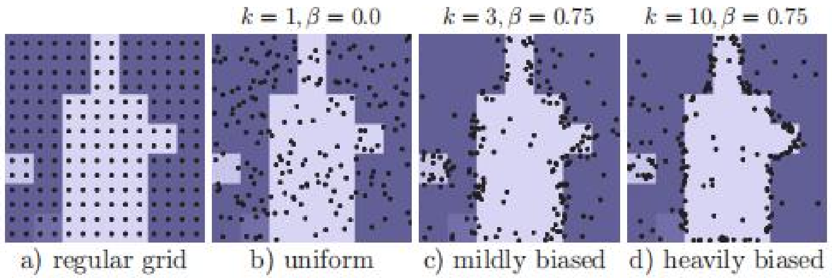
- Overview
  - Errors occur mostly on boundary parts
  - Choose *N* hard points in output mask to re-predict
- Three components:
  - **Point selection strategy**: avoid excessive computation
  - **Point-wise feature representation**: for each selected point
  - **Point head**: predict a label from point-wise feature representation

Method



- **Point selection strategy**
  - Selected points should be located more densely near areas like boundaries
  - Inference Stage:
    - 1. Upsample predicted segmentation
    - 2. Choose *N* most unceratin points (*p* closest to 0.5 for binary mask)
    - 3. Computes their point-wise feature representation and predict labels
    - Repeat 1-3 until a desired resolution
  - Complexity:
    - Desired resolution : M*M; Starting resolution: M0*M0
    - Complexity: $N \log_2 \frac{M}{M_0}$

Method



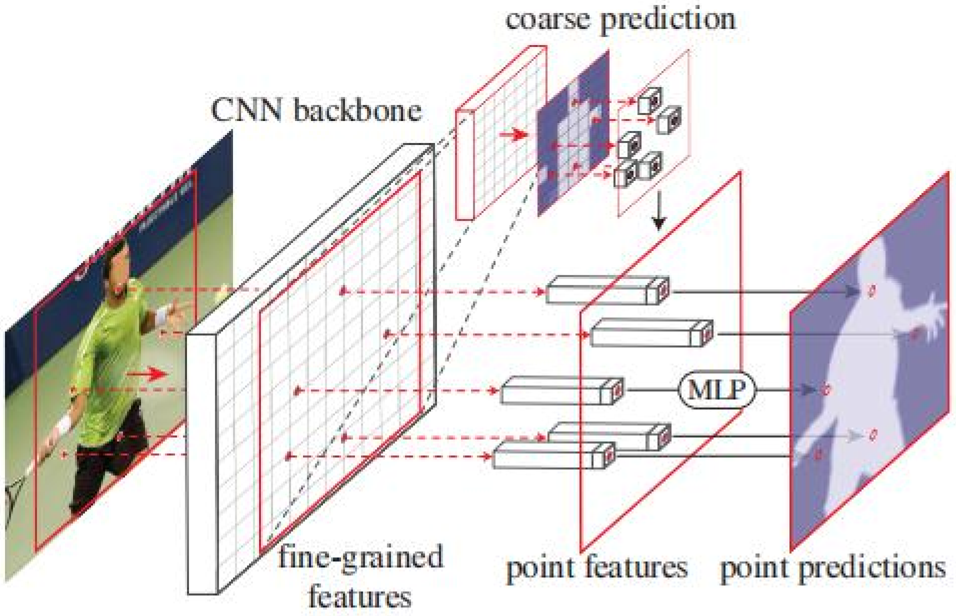
- **Point selection strategy**
  - Selected points should be located more densely near areas like boundaries
  - Training Stage:
    - Non-iterative stategy based on random sampling
    - 1.Over generation: Randomly sampling kN points (k>1) from a uniform distribution;
    - 2. Importance sampling: Select most uncertain βN ($\beta \in [0,1]$) points from kN points;
    - 3. Coverage: remaining $(1-\beta)N$ points are sampled uniformly
  - Number of selected can be difference between training and inference
  - Predictions and loss functions are only computed on the N sampled points

Method

- **Point-wise Representation**
    - Combining fine-grained and coarse prediction features
    - Fine-grained Features:
        - Extract a feature vector at each sampled point from CNN feature maps
        - Can be extracted from one or more feature maps
        - Deficiencies:
            - Do not contain region-specific information
            - May only contain relatively low-level information
    - Coarse Features:

Method

- **Point-wise Representation**
  - Coarse Features:
    - A K-dimensional vector **at each point** (a K-class prediction)
- **Point Head**
  - Using a simple MLP



Experiments: Instance Segmentation

- Architecture
  - Mask R-CNN
  - ResNet-50 + FPN
  - Mask head adjustment
- Training: 14^2 points, k=3, β=0.75
- Inference: N=28^2

mask head	output resolution	COCO AP	COCO AP*	Cityscapes AP
4× conv	28×28	35.2	37.6	33.0
PointRend	28×28	36.1 (+0.9)	39.2 (+1.6)	35.5 (+2.5)
PointRend	224×224	**36.3** (+1.1)	**39.7** (+2.1)	**35.8** (+2.8)

Table 1: **PointRend *vs.* the default 4× conv mask head for Mask R-CNN [19].** Mask AP is reported. AP* is COCO mask AP evaluated against the higher-quality LVIS annotations [16] (see text for details). A ResNet-50-FPN backbone is used for both COCO and Cityscapes models. PointRend outperforms the standard 4× conv mask head both quantitively and qualitatively. Higher output resolution leads to more detailed predictions, see Fig. 2 and Fig. 6.

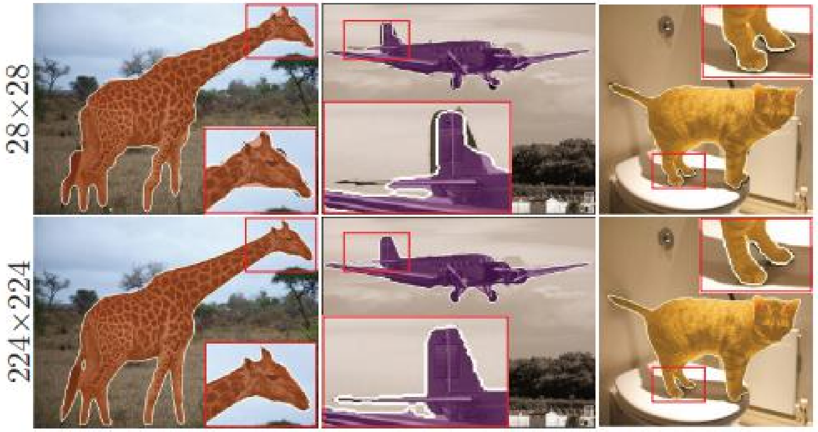


Figure 6: **PointRend inference with different output resolutions.** High resolution masks align better with object boundaries.

Experiments: Instance Segmentation

- Ablation Experiments

selection strategy	COCO AP	COCO AP*	Cityscapes AP
regular grid	35.7	39.1	34.4
uniform (k=1, β=0.0)	35.9	39.0	34.5
mildly biased (k=3, β=0.75)	**36.3**	**39.7**	**35.8**
heavily biased (k=10, β=1.0)	34.4	37.5	34.1

Table 4: **Training-time point selection strategies** with 14^2 points per box. Mildly biasing sampling towards uncertain regions performs the best. Heavily biased sampling performs even worse than uniform or regular grid sampling indicating the importance of coverage. AP* is COCO mask AP evaluated against the higher-quality LVIS annotations [16] (see text for details).

mask head	backbone	COCO AP	COCO AP*
4× conv	R50-FPN	37.2	39.5
PointRend	R50-FPN	**38.2** (+1.0)	**41.5** (+2.0)
4× conv	R101-FPN	38.6	41.4
PointRend	R101-FPN	**39.8** (+1.2)	**43.5** (+2.1)
4× conv	X101-FPN	39.5	42.1
PointRend	X101-FPN	**40.9** (+1.4)	**44.9** (+2.8)

Table 5: **Larger models and a longer 3× schedule [18].** PointRend benefits from more advanced models and the longer training. The gap between PointRend and the default mask head in Mask R-CNN holds. AP* is COCO mask AP evaluated against the higher-quality LVIS annotations [16] (see text for details).

Experiments: Semantic Segmentation

- Architecture
  - SemanticFPN: ResNet-101
  - DeepLabv3: ResNet-103
  - Inference: N=8096

method	output resolution	mIoU
DeeplabV3-OS-16	64×128	77.2
DeeplabV3-OS-8	128×256	77.8 (+0.6)
DeeplabV3-OS-16 + PointRend	1024×2048	**78.4** **(+1.2)**

Table 6: **DeeplabV3 with PointRend** for Cityscapes semantic segmentation outperforms baseline DeepLabV3. Dilating the res_4 stage during inference yields a larger, more accurate prediction, but at much higher computational and memory costs; it is still inferior to using PointRend.

method	output resolution	mIoU
SemanticFPN P_2-P_5	256×512	77.7
SemanticFPN P_2-P_5 + PointRend	1024×2048	**78.6** **(+0.9)**
SemanticFPN P_3-P_5	128×256	77.4
SemanticFPN P_3-P_5 + PointRend	1024×2048	**78.5** **(+1.1)**

Table 7: **SemanticFPN with PointRend** for Cityscapes semantic segmentation outperform the baseline SemanticFPN.

Experiments: Semantic Segmentation

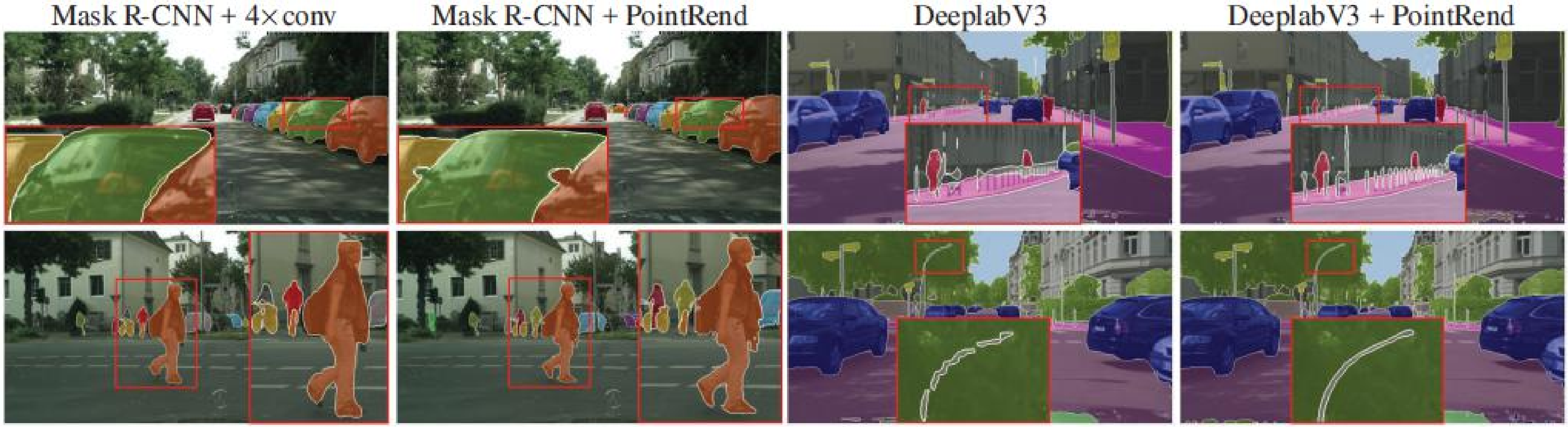


Figure 8: **Cityscapes example results for instance and semantic segmentation.** In instance segmentation larger objects benefit more from PointRend ability to yield high resolution output. Whereas for semantic segmentation PointRend recovers small objects and details.