CS688: Large-Scale Image & Video Retrieval (Spring 2020)

YIN XU

KAIST

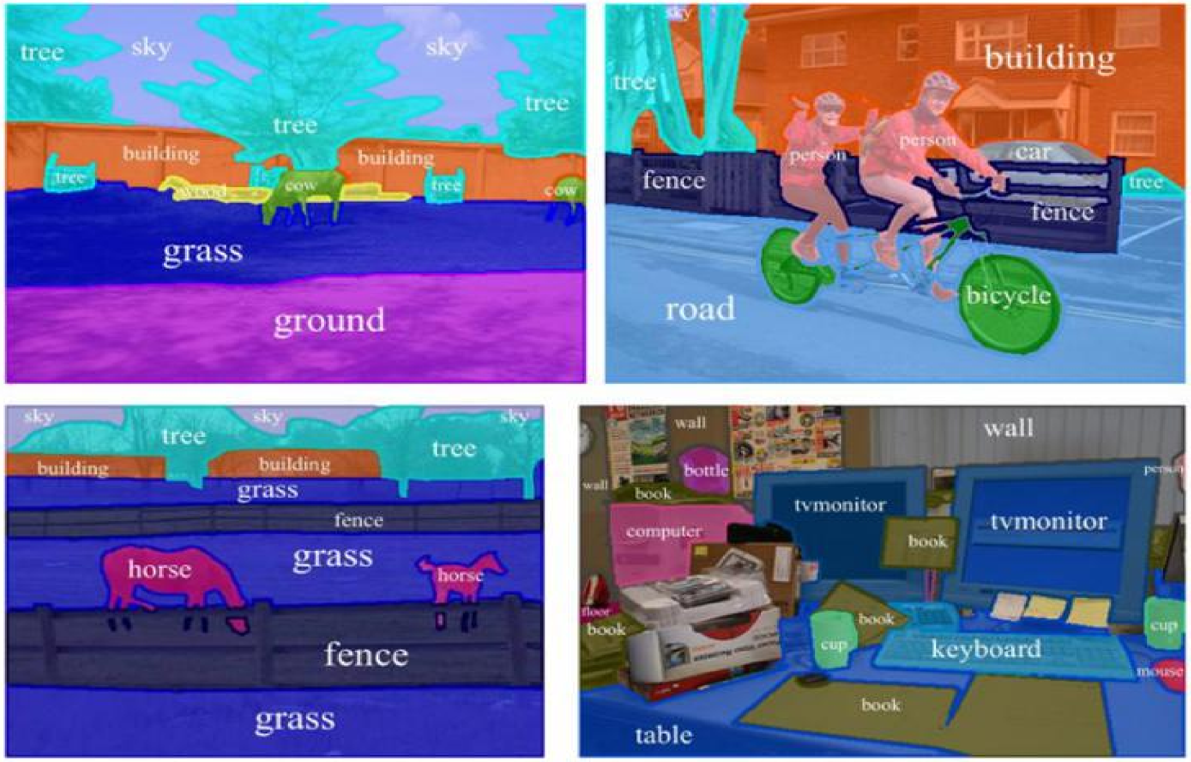
Contents

Semantic Segmentation

What is semantic segmentation?

Idea: recognizing, understanding what's in the image in pixel level.

"Two men riding on a bike in front of a building on the road. And there is a car."



Semantic Segmentation

Why semantic segmentation?

1. Robot vision and understanding
2. Autonomous driving
3. Medial image analysis

Semantic Segmentation

Interesting topics of segmentation:

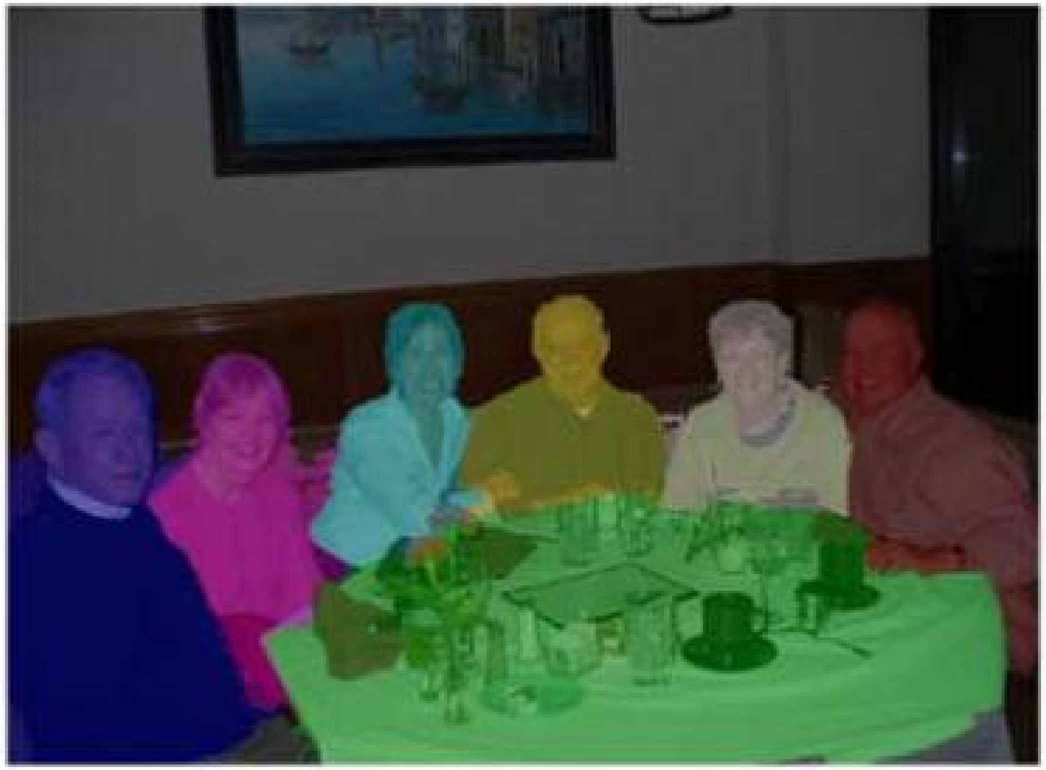
1. 2D images: (general) sematic segmentation, instance segmentation
2. 3D images: Point clouds
3. Video segmentation

Semantic Segmentation

Semantic segmentation: a process of assigning a label to every pixel in the image

Instance segmentation: treat multiple objects of the same class as distinct individual objects (or instances)

Semantic Segmentation



Instance Segmentation

What is its realtionship to image retrieval

Segmentation-based Retrieval (mainly for object-based retrieval):

1. Avoiding large number of regions in one image

    ---- manageable regions / objects

2. Extracting simple boundary regions (avoiding disturbrance):

    ---- segmented regions can be a unit in retrieval

3. Make a robust datatset descriptor

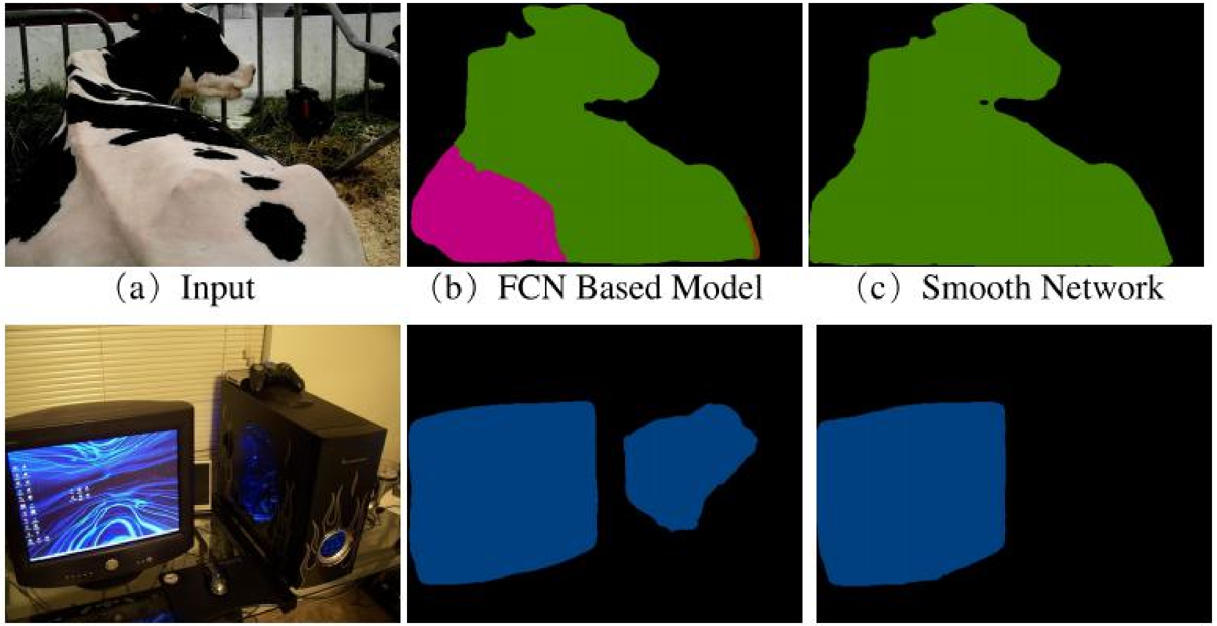
    ---- reduce search space

Challenges and solutions (2D)

- Challenges:

★ **Intra-class Inconsistency:** The same semantic label but different appearances

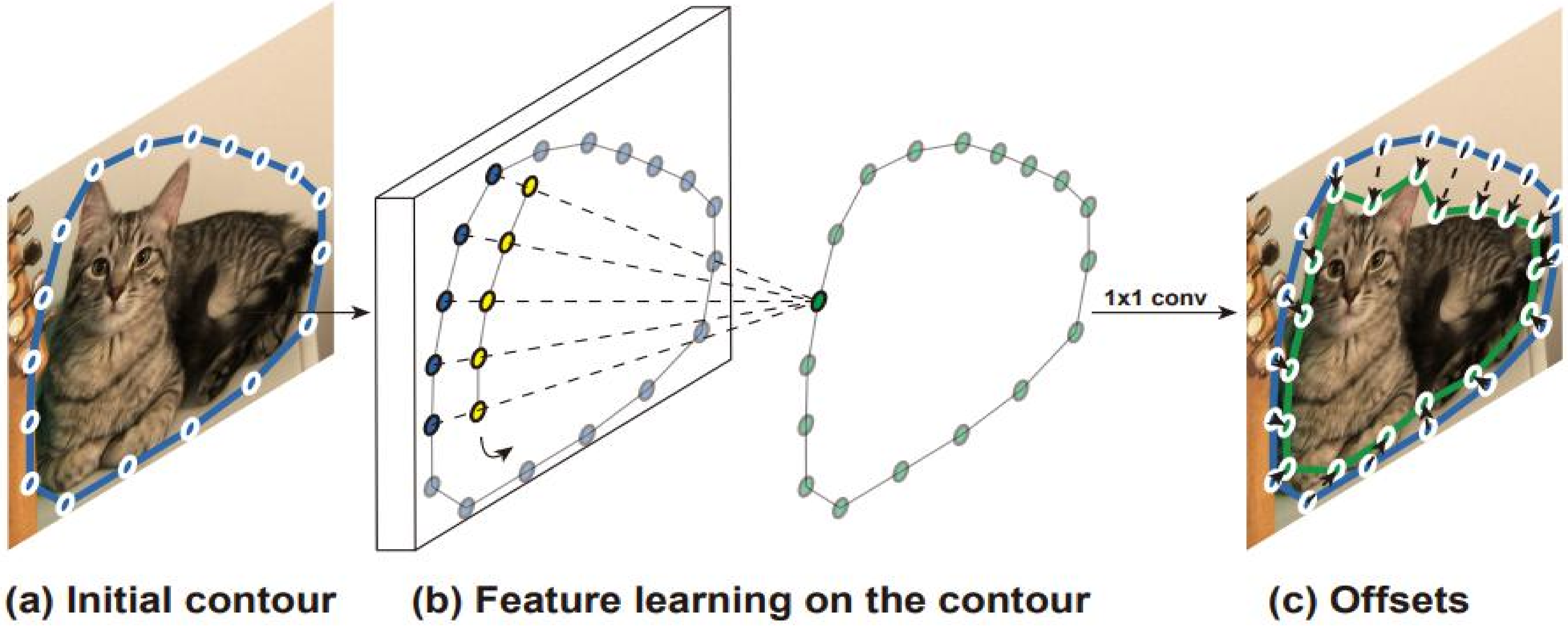
Inter-class Indistinction: Different semantic labels but with similar appearances



(a) Input (b) FCN Based Model (c) Smooth Network

(d) Input (e) FCN Based Model (f) Border Network

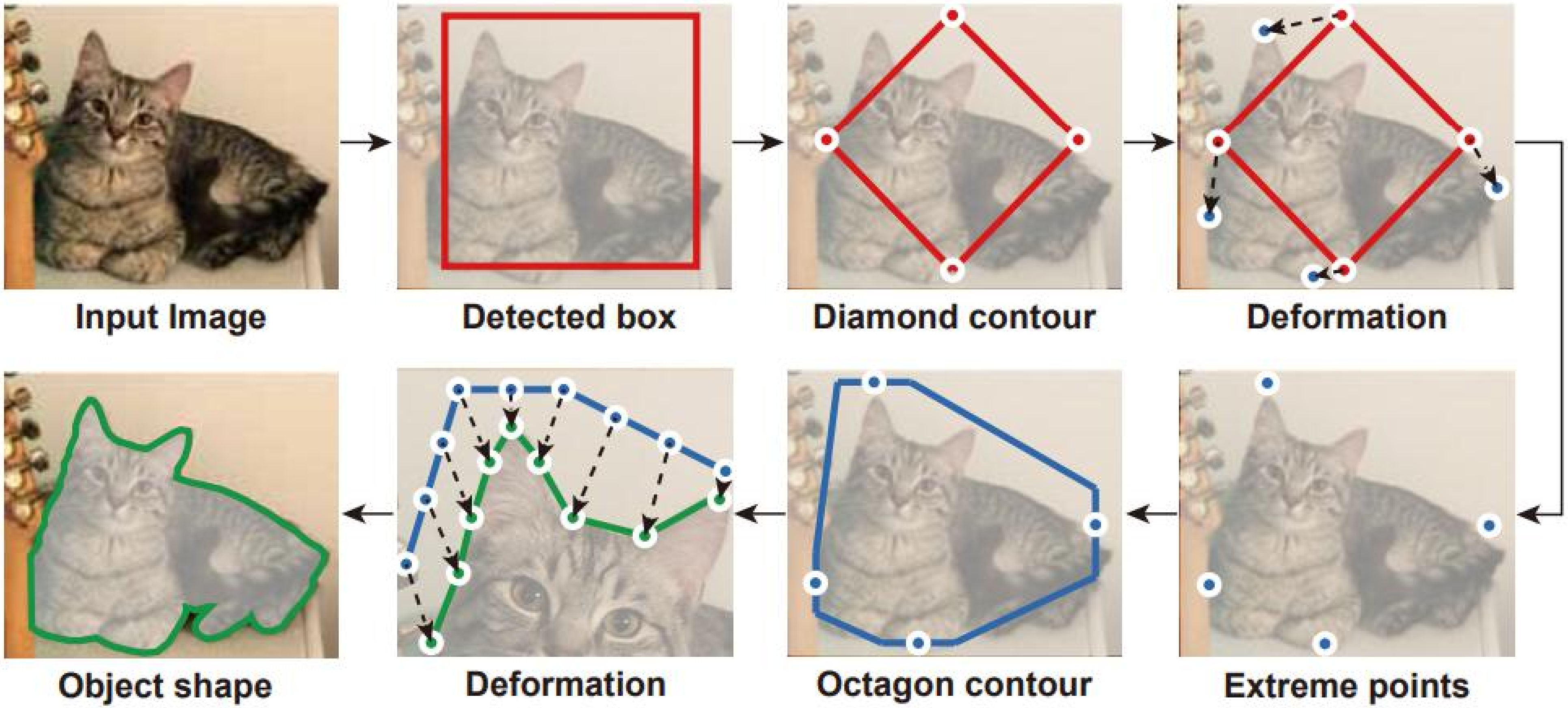
Possible Solutions: Edge-approximation



Deep Snake for Real-Time Instance Segmentation

Possible Solutions: Edge approximation

:s



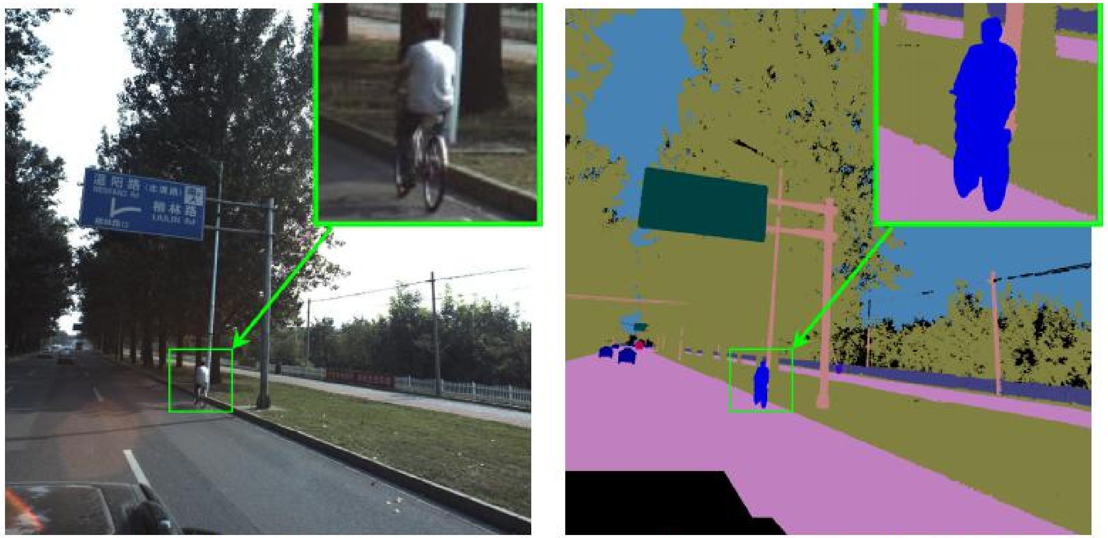
Deep Snake for Real-Time Instance Segmentation,CVPR 2020

Possible Solutions: Keypoint detection

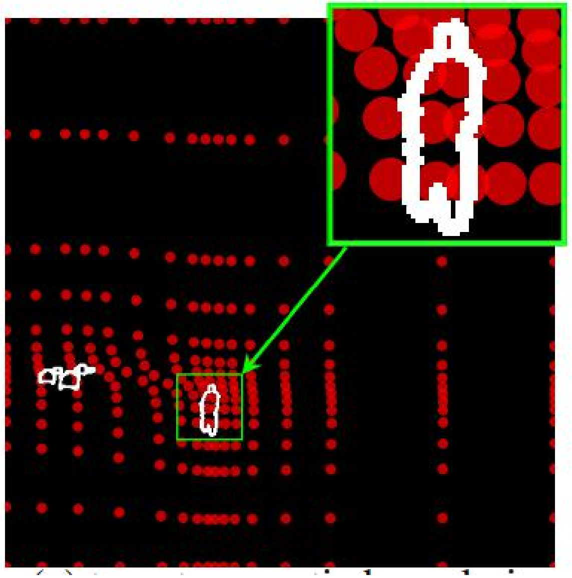
Steps:

1) compute the boundary map with given semantic labels.

2) For each pixel, find the closet pixel on the boundary.

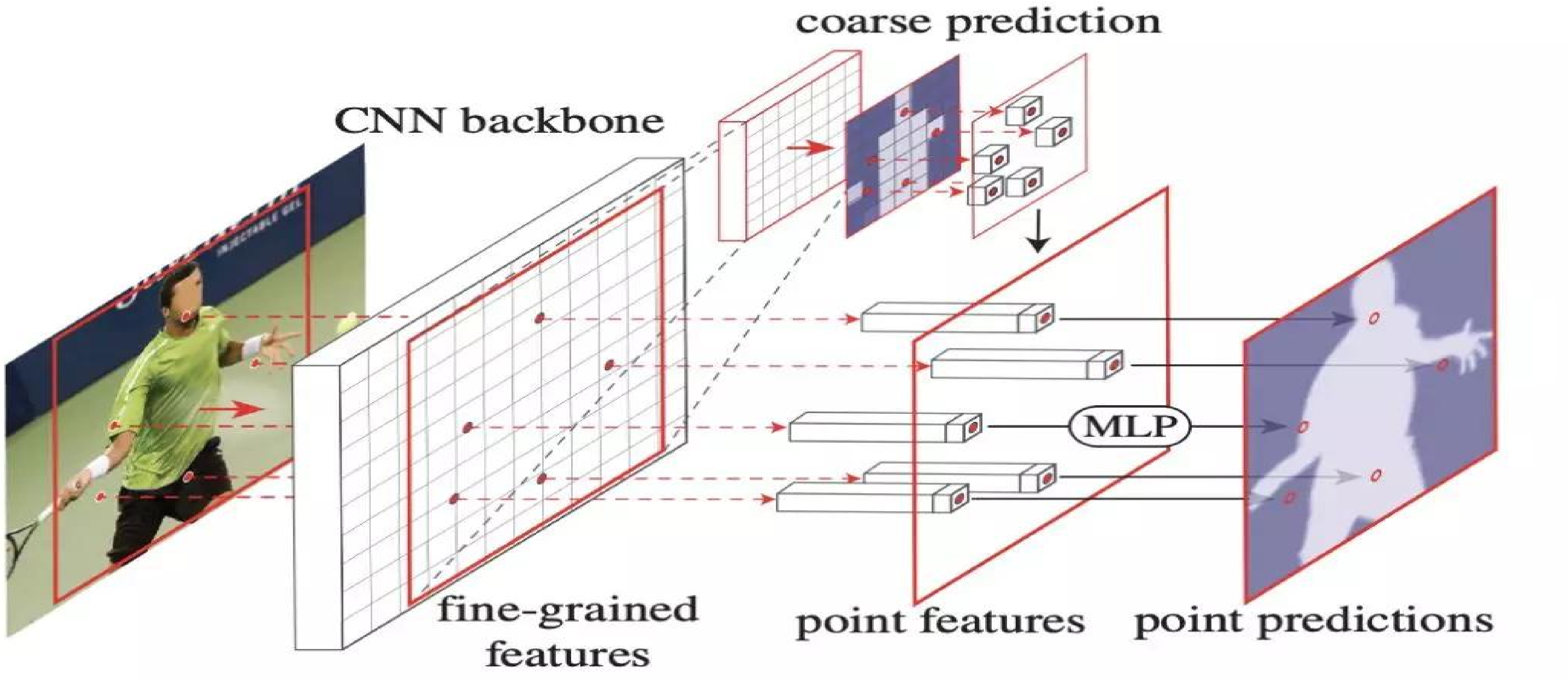


(a) original 2710×2710 image (b) ground truth labels



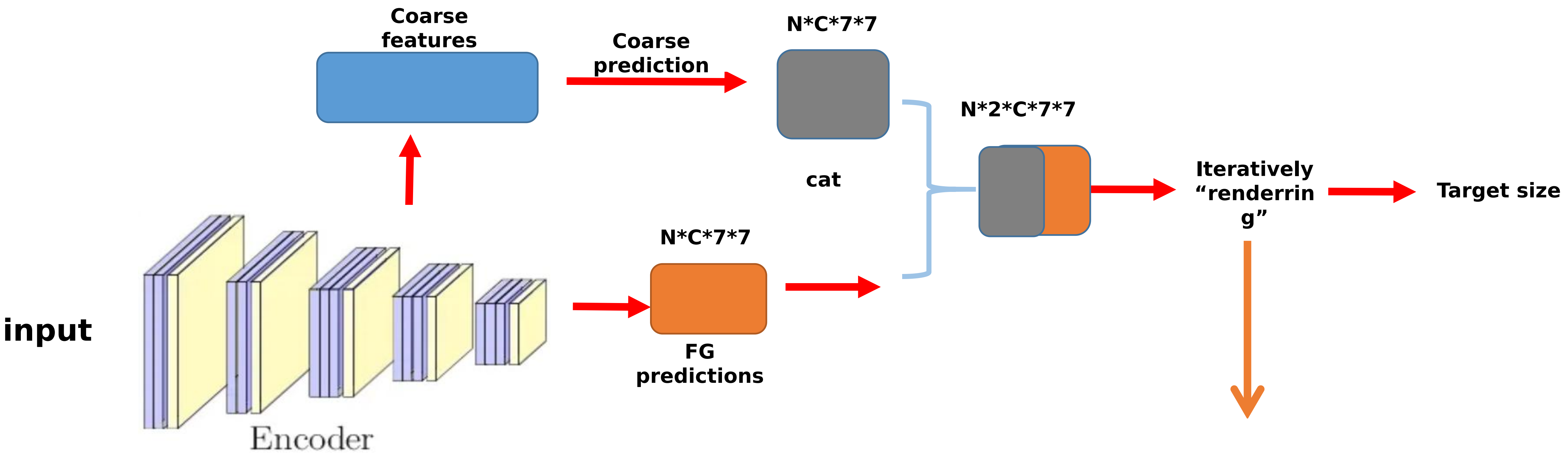
Efficient Segmentation: Learning Downsampling Near Semantic Boundaries, ICCV 2019

PointRend: Image Segmentation as Rendering



upsampling +correction

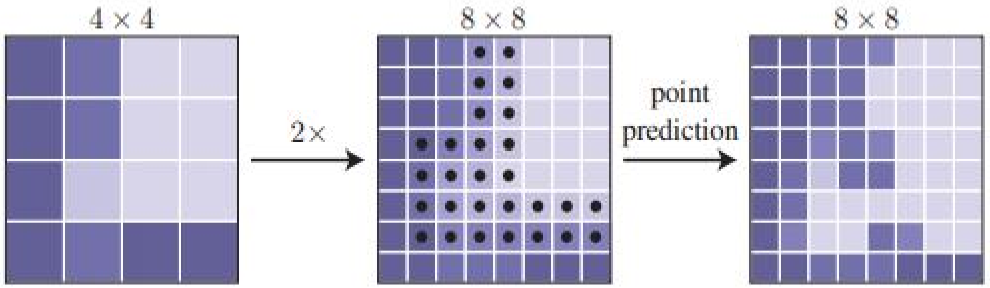
PointRend: Image Segmentation as Rendering (CVPR 2020)



From 7*7 to 224*224:

----$log^{\frac{224}{7}}$ =5 iterations

PointRend: Image Segmentation as Rendering



Steps:

1) Upsample (Bilinear Interpolation)
2) Uncertainty calculation:
   --- the difference between the most & second most confidence
   --- set a threshold 0.5

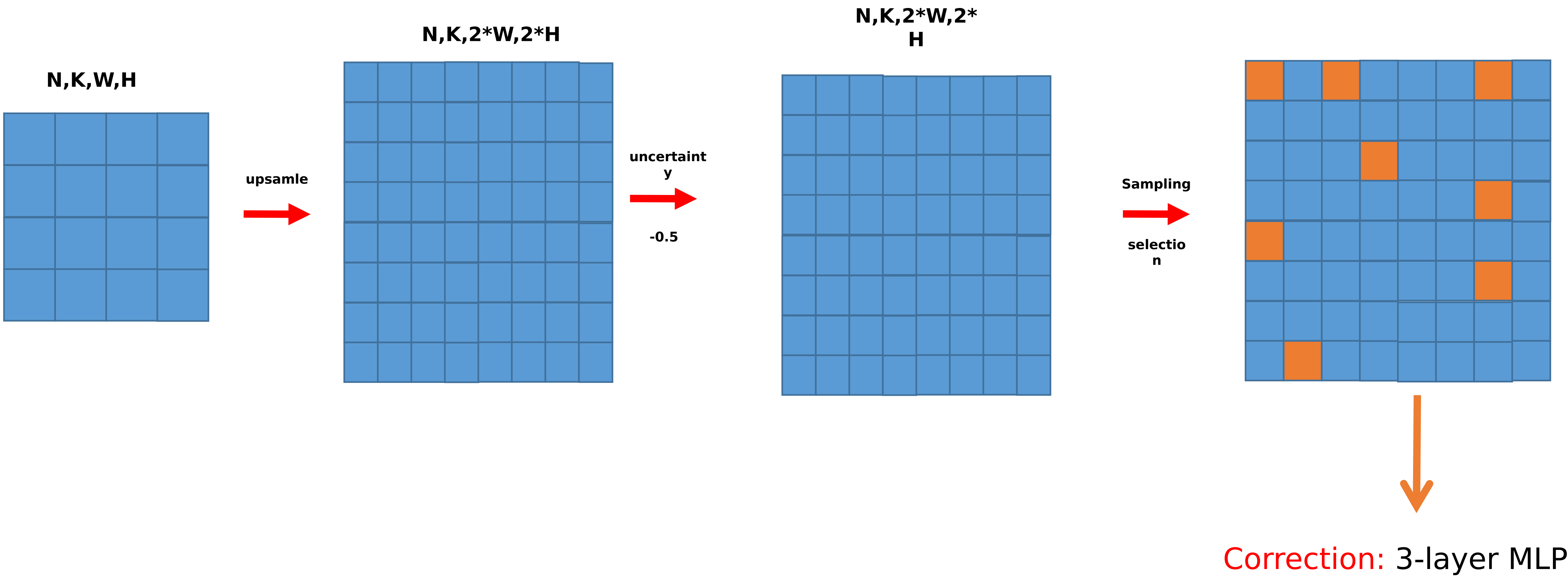
3) Generate k*N points from uniform distribution and then select the top $\beta * \mathrm{N}$ ones (uncertain).

4) Feed selected pixels into 3-layer MLP

Notes:

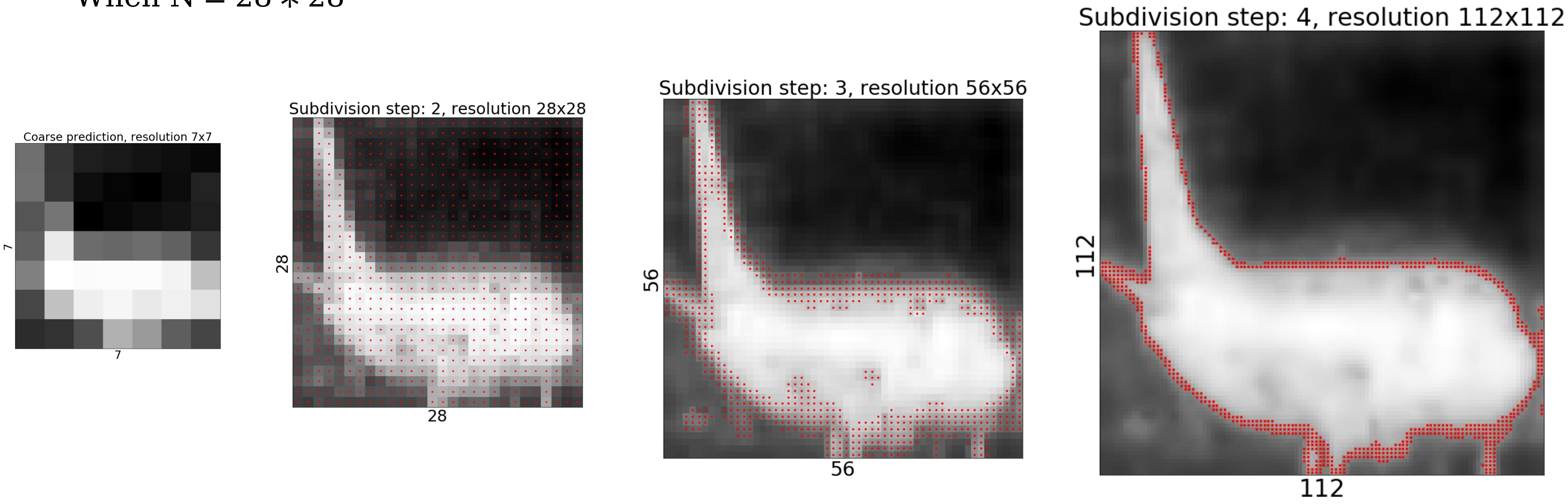
Last step of segmentation:
- ---map all vectors to a K-d space (with conv1*1)
- ---using argmax($\hat{y}$) (pixel classification)
  - ----use the indices as its classification

PointRend: Image Segmentation as Rendering



PointRend: Image Segmentation as Rendering

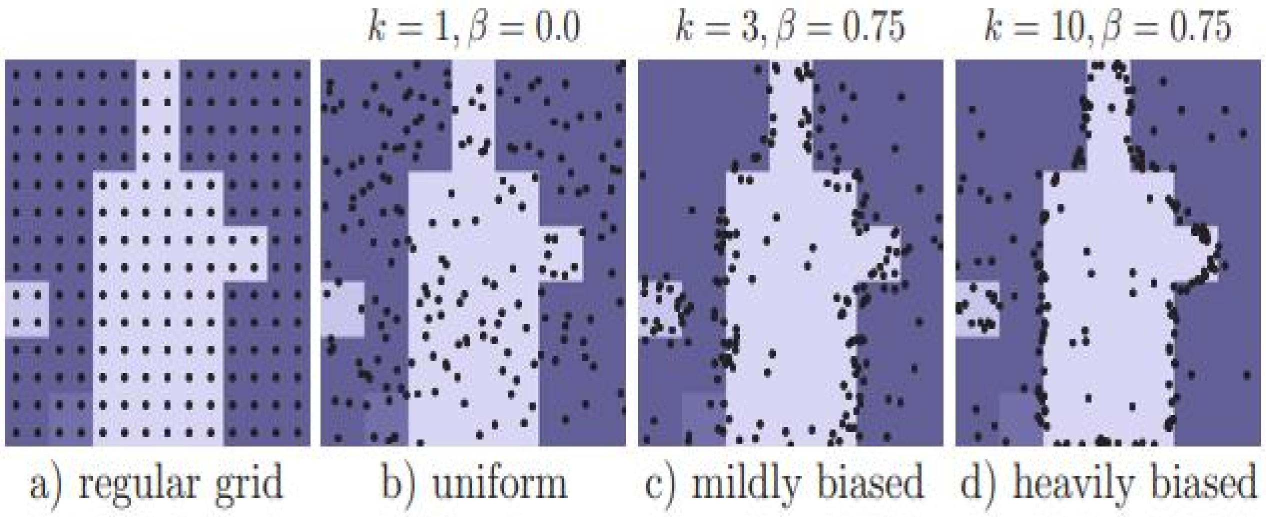
When $\mathrm{N} = 28 * 28$



Sampling Steps: from 7*7 to 112*112

PointRend: Image Segmentation as Rendering

Key-point Sampling



segmentation

PointRend: Image Segmentation as Rendering

Point Rend (Segementation)

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)**

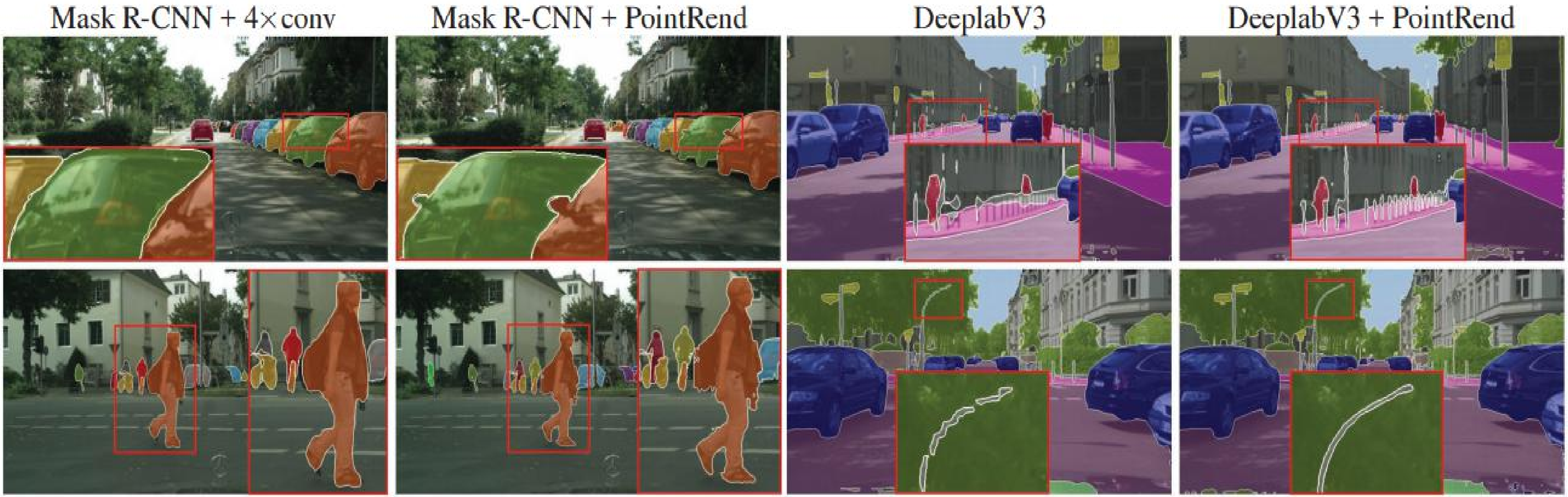
method	output resolution	mIoU
SemanticFPN P_2-P_5	256×512	77.7
SemanticFPN P_2-P_5 + PointRend	1024×2048	**78.6** **(+0.9)**
SemanticFPN P_3-P_5	128×256	77.4
SemanticFPN P_3-P_5 + PointRend	1024×2048	**78.5** **(+1.1)**

Point Rend: instance

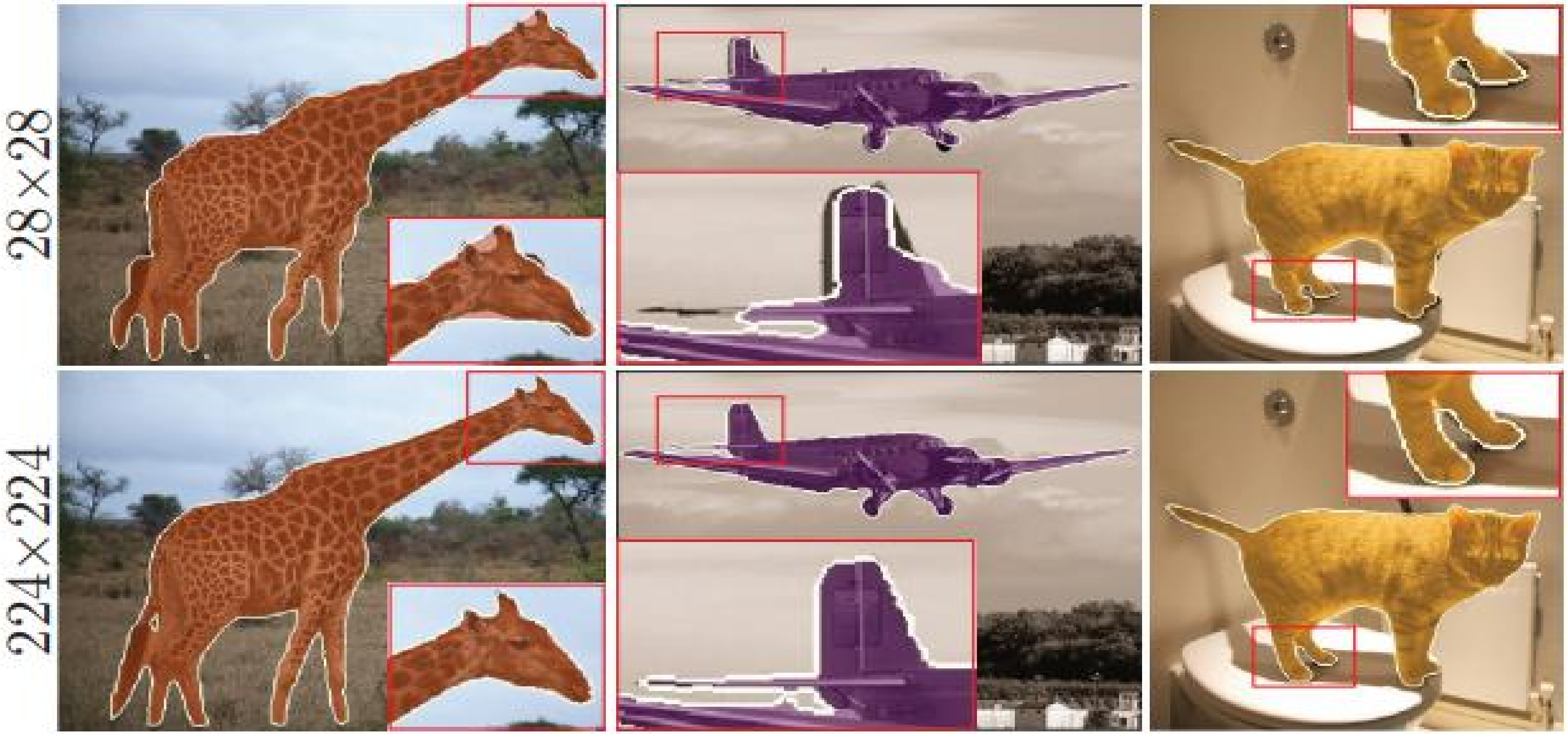
		COCO	
mask head	backbone	AP	AP*
4× conv	R50-FPN	37.2	39.5
PointRend	R50-FPN	**38.2** **(+1.0)**	**41.5** **(+2.0)**
4× conv	R101-FPN	38.6	41.4
PointRend	R101-FPN	**39.8** **(+1.2)**	**43.5** **(+2.1)**
4× conv	X101-FPN	39.5	42.1
PointRend	X101-FPN	**40.9** **(+1.4)**	**44.9** **(+2.8)**

PointRend: Image Segmentation as Rendering

Point Rend (Segementation) **Point Rend: instance**



PointRend: Image Segmentation as Rendering



PointRend: Image Segmentation as Rendering

PointRend: Image Segmentation as Rendering

Summary:

Problem: inconsistent segmentation around edge regions
Method: key-point detection + pixel-wise correction

Components: 1) Sampling method: coarse prediction + uncertainty
2) Pixel correction : 3-layer MLP

3) Process: iteratively implement upsampling +correction

Personal thinkings:

Ads: 1) Fine-grained segmentation 2) edge preservation

Dis: may not that useful in general semenatics.

Q & A