Parsing R-CNN for Dense Pose Estimation

—— COCO 2018 DensePose Challenge

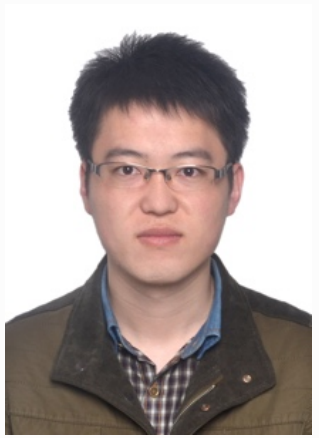
Speaker: Yang Lu

BUPT-PRIV

Team member

Yang Lu*

Song Qing*



Wang Zhihui*

* Beijing University of Posts and Telecommunications (BUPT)

BUPT-PRIV

Results

COCO 2018 DensePose Test

	AP	AP50	AP75	APm	APl
BUPT-PRIV (Ours)	**64**	**92**	**75**	**57**	**67**
PlumSix	58	89	66	50	61
ML_Lab	57	89	64	51	59
Sound of silent	57	87	66	48	61
DensePose ResNeXt101	56	89	64	51	59

BUPT-PRIV

Overview

- **Method Structure**
  - Top-down or Bottom-up?
  - End to end?

- **Parsing R-CNN**
  - Spatial & Semantic information
  - Feature map resolution
  - Receptive field
  - Branch capacity

- **Evaluation Metric**

BUPT-PRIV

- **Method Structure**

Bottom-up

Backbone → Keypoints / Dense Pose / Person Part / …… → **Grouping**

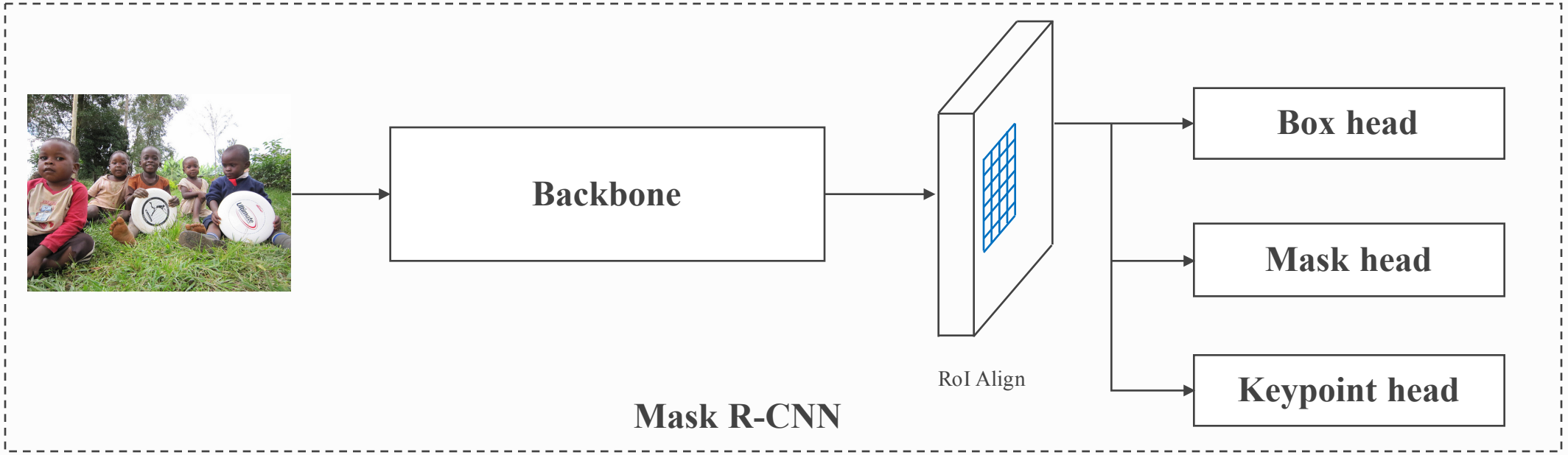
Backbone → Instance B-Box → **Instance-level task**

Top-down

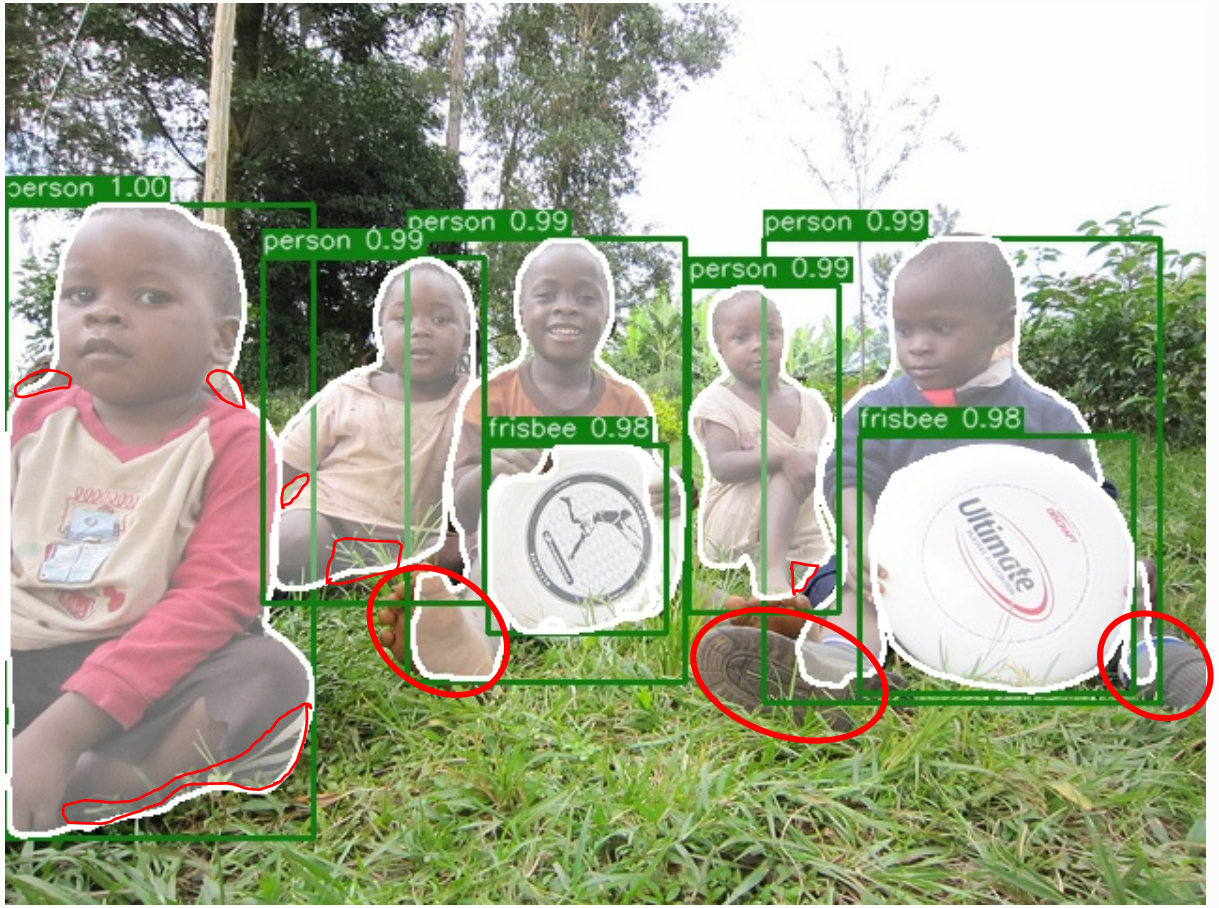
Output

BUPT-PRIV

- **Method Structure**



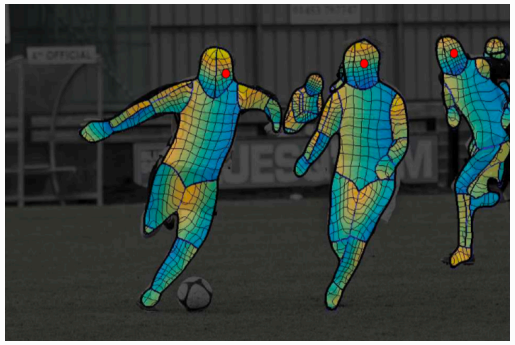
- **Motivation**



The outline of the human body is not good enough.

BUPT-PRIV

- **Motivation**



Dense Pose Estimation

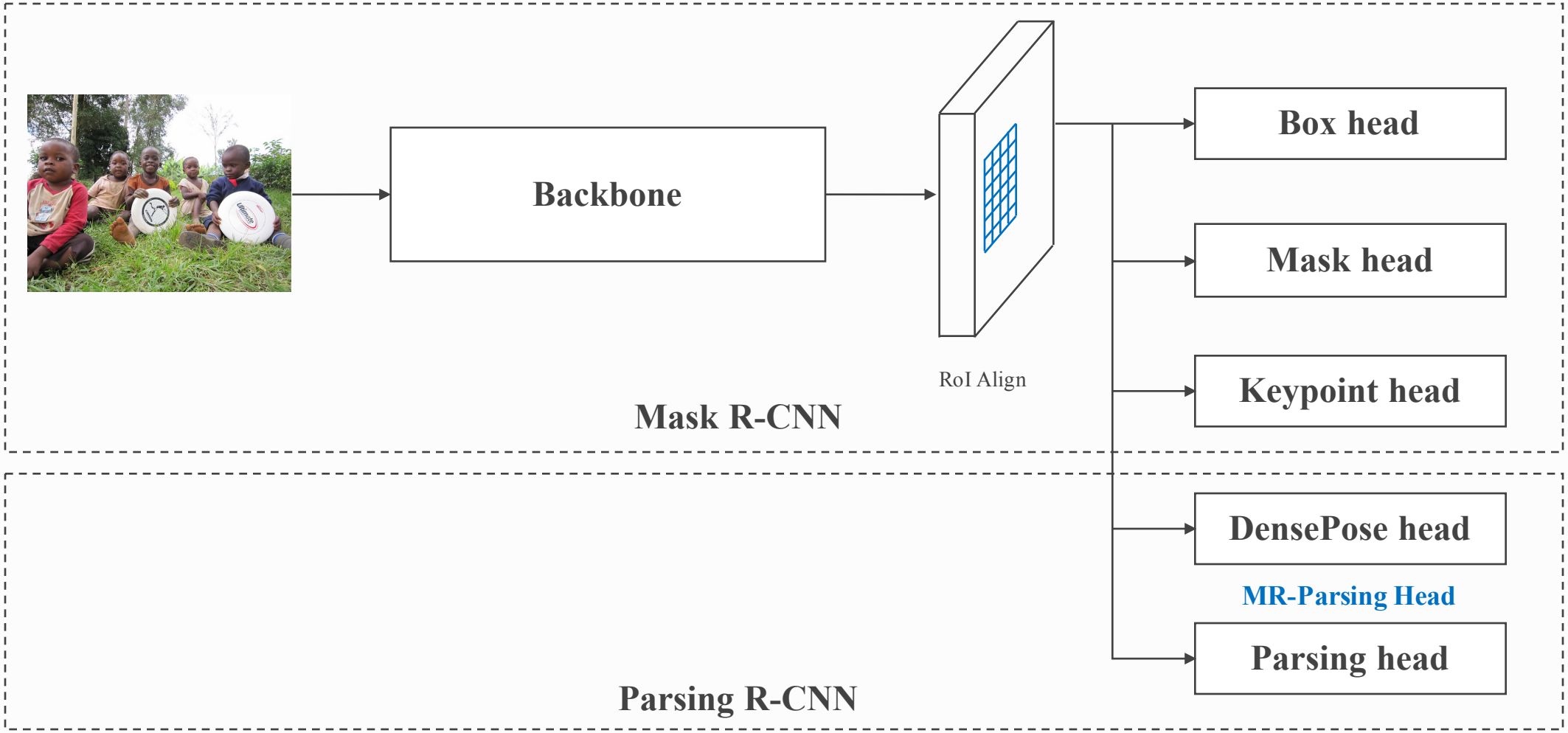
Person Part Segmentation

?

Can the Mask R-CNN handle the problem of person instance analysis well?

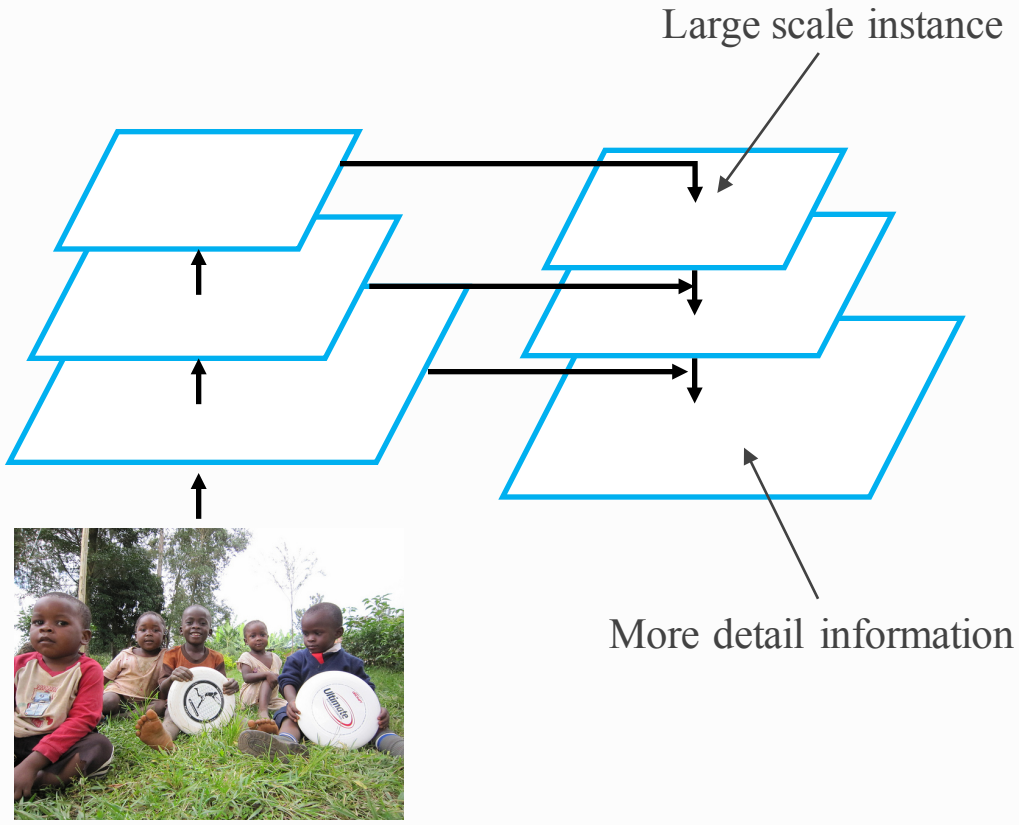
BUPT-PRIV

- **Parsing R-CNN**



- **Parsing R-CNN**

  Spatial & Semantic information



use P2 for parsing

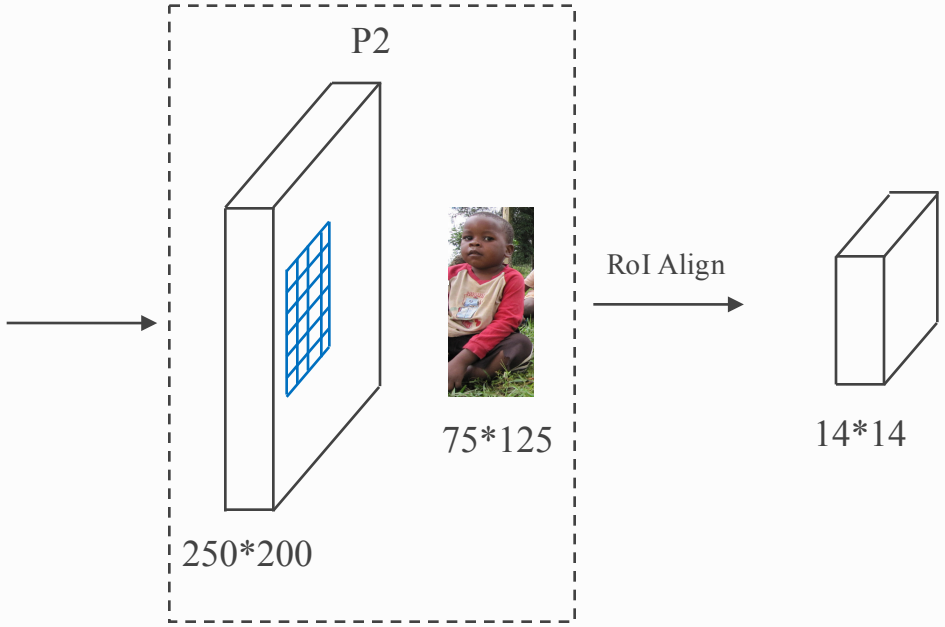
	AP	
P2-P6 (baseline)	48.9	
P2-only	48.5	-0.4
P2 for parsing	**49.8**	**+0.9**

Use P2-P6 for RPN, and only P2 for MR-Parsing Head **+0.9AP**

BUPT-PRIV

- **Parsing R-CNN**

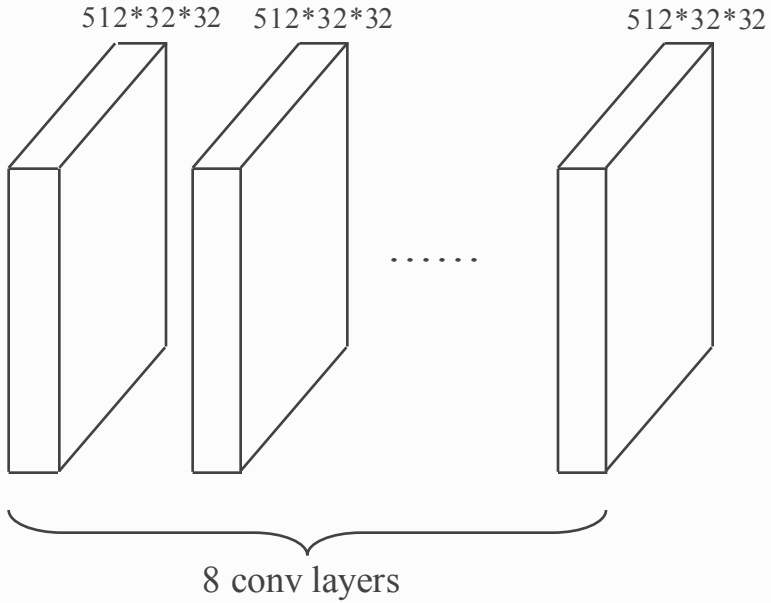
  Feature map resolution



Lost too many details !

- **Parsing R-CNN**

  Feature map resolution



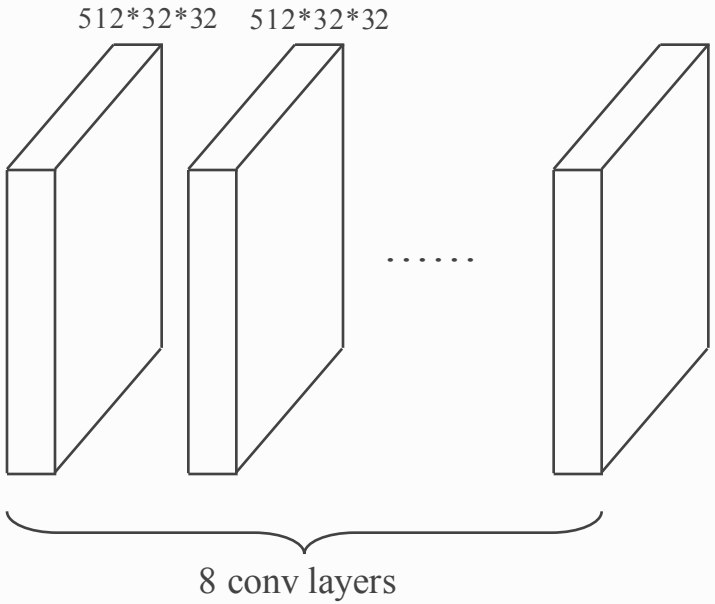
enlarging the roi size

	AP	
14*14 (Original size)	49.8	
32*32 (Ours)	**52.2**	**+2.4**
48*48 (Ours)	52.8	+3.0

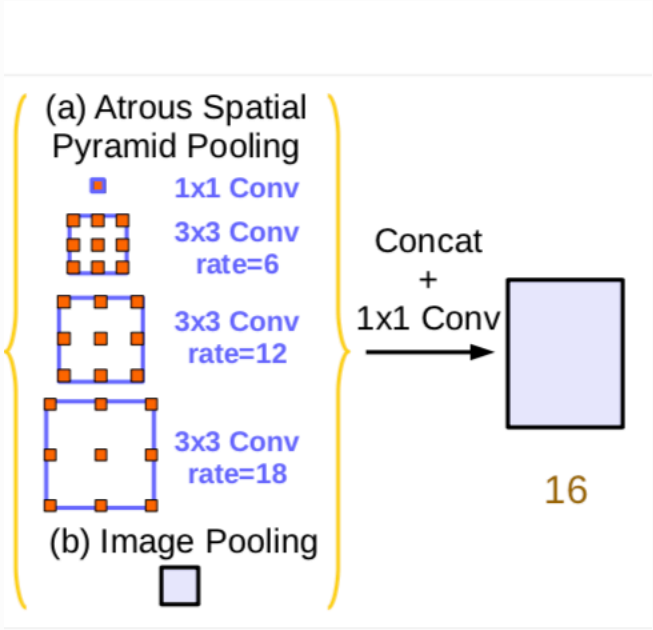
Use 32 * 32 roi size and output size is 128 * 128 **+2.4 AP**

- **Parsing R-CNN**

  Receptive field



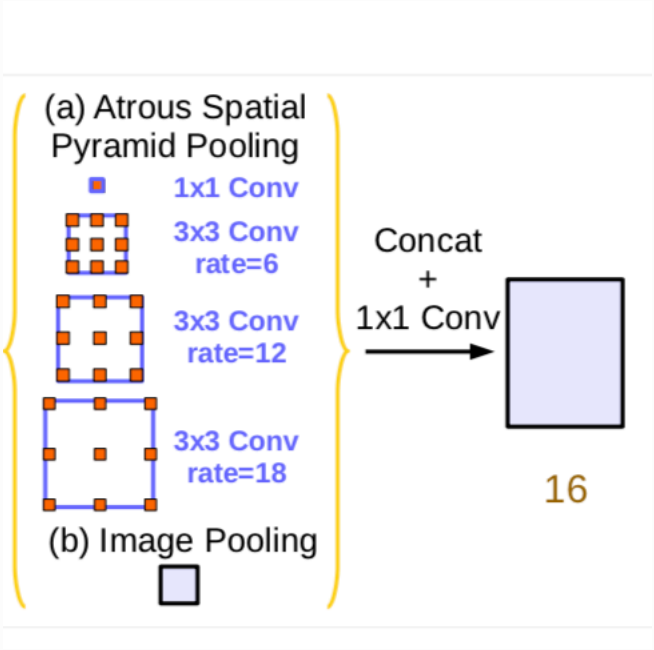
RF = 17 < 32 = roi size



From DeepLab V3

BUPT-PRIV

- **Parsing R-CNN**
  Receptive field



From DeepLab V3

use aspp module for paring

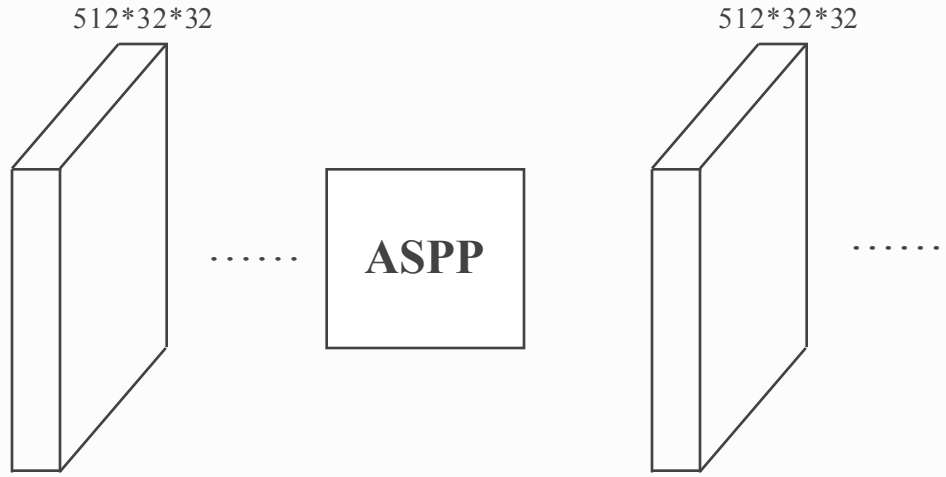
	AP	
8 conv layers	52.2	
aspp module	**52.9**	**+0.7**
ppm module (PSPNET)	52.4	+0.2

Use aspp module to instead of 8 conv layers **+0.7 AP**

BUPT-PRIV

• Parsing R-CNN

Branch capacity



Before ASPP
Semantic space transformation ?

After ASPP
High-level semantic feature ?

add 4 conv layers after aspp

	AP	
aspp module	52.9	
4 conv before aspp	53.0	+0.1
4 conv after aspp	**53.9**	**+1.0**
both	54.0	+1.1

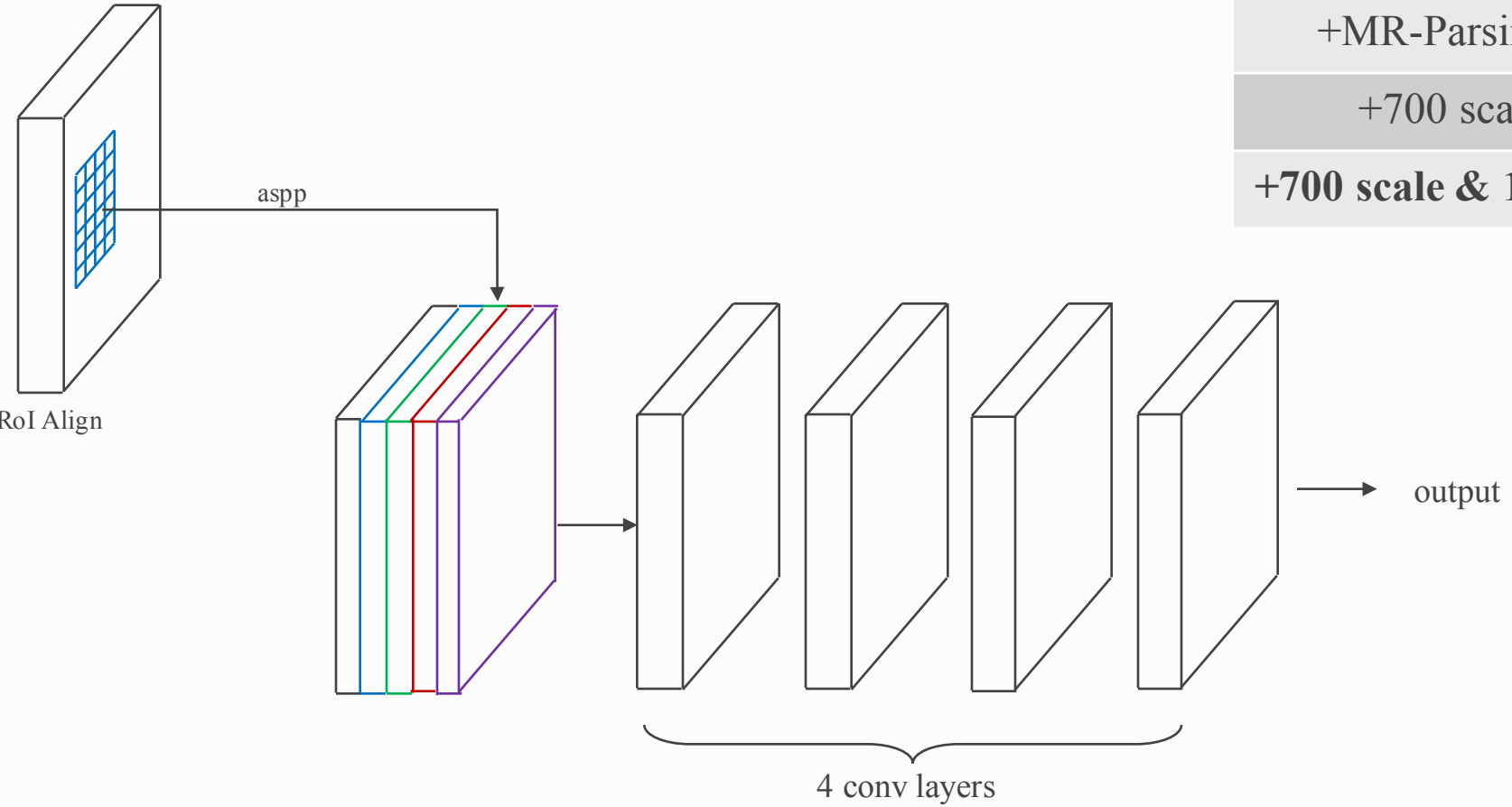
Use 4 conv layers after aspp module **+1.0 AP**

BUPT-PRIV

- **Parsing R-CNN**

Small testing scale and less roi for speeding up

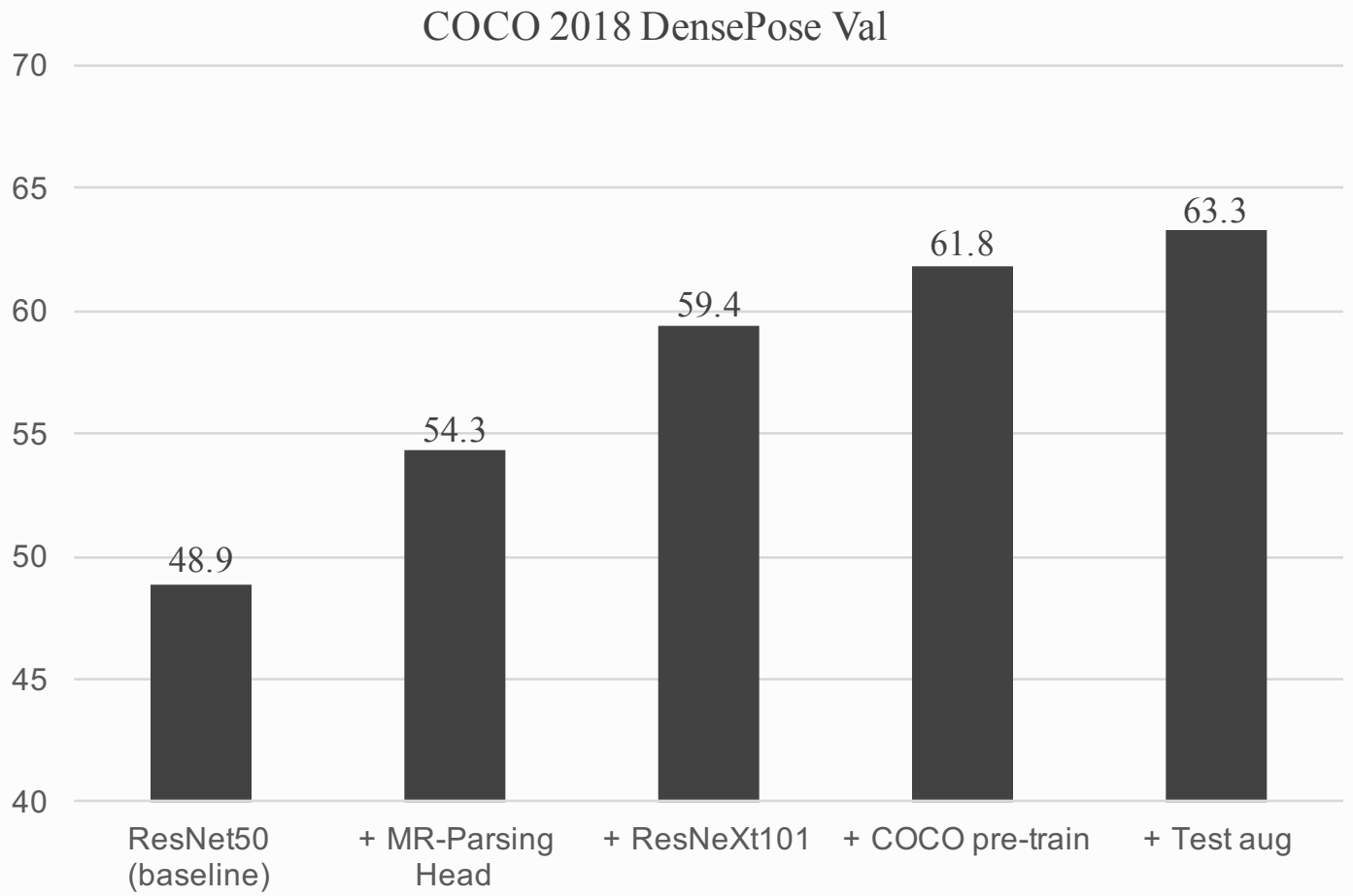
	AP	
ResNet50 (baseline)	48.9	
+MR-Parsing Head	53.9	+5.0
+700 scale test	54.4	+5.5
+700 scale & 100 rois test	**54.3**	**+5.4**



A good designed head is very important for instance-level parsing

BUPT-PRIV

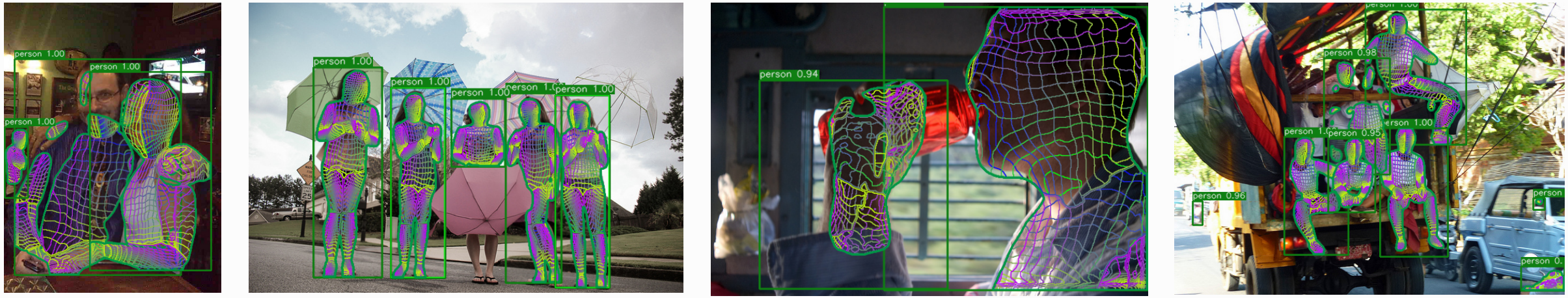
- **Parsing R-CNN**



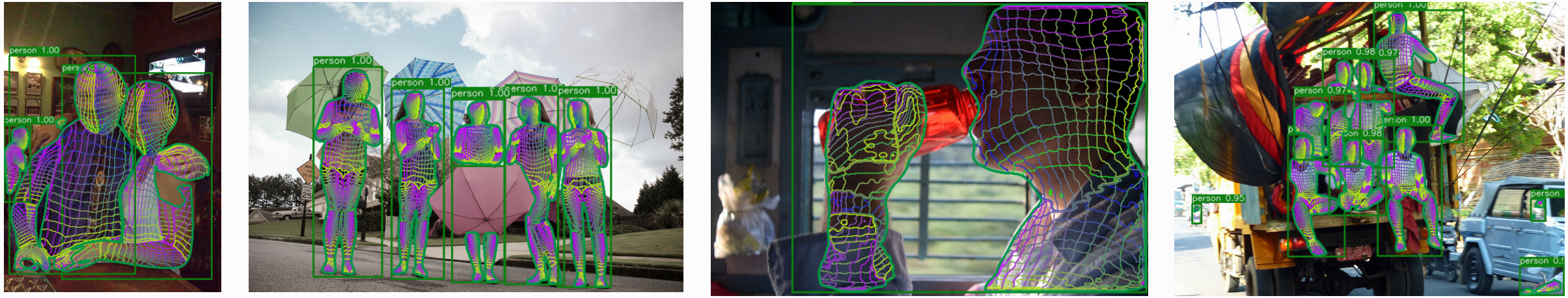
We get 64.1 mAP on the test set with single model.

BUPT-PRIV

- **Parsing R-CNN**



DensePose ResNeXt101



Parsing R-CNN (ours)

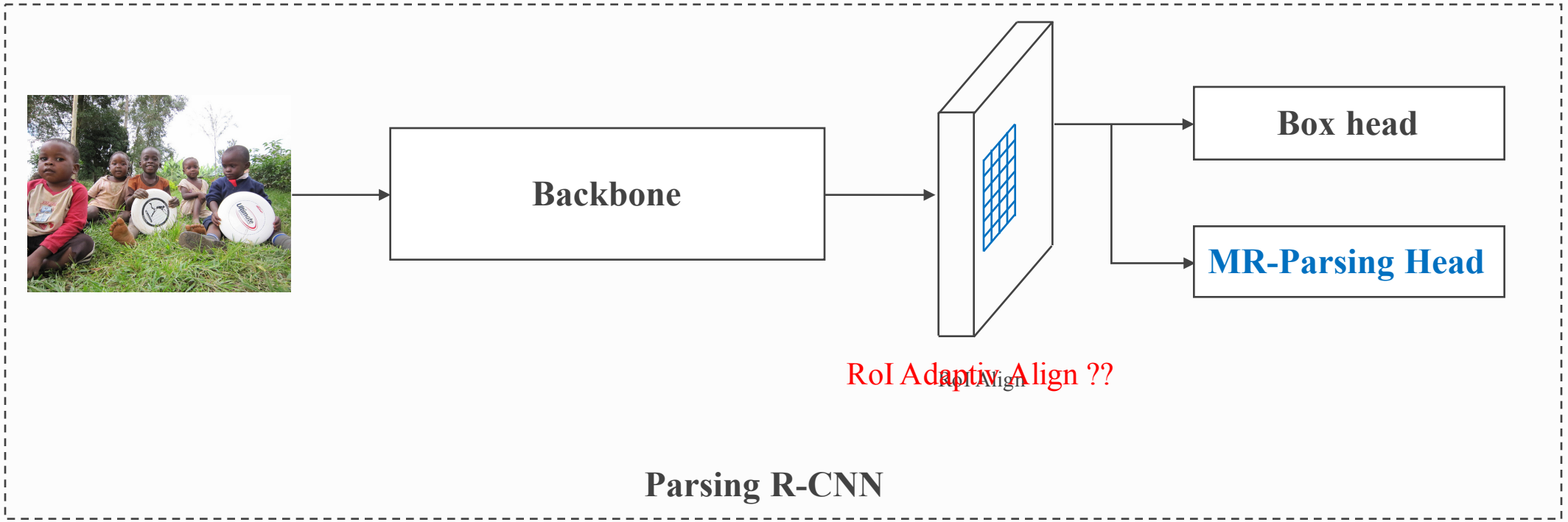
BUPT-PRIV

- **Parsing R-CNN**

What's the next?

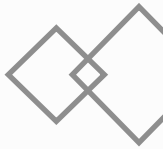
BUPT-PRIV

- **Parsing R-CNN**



- **Parsing R-CNN**

And One More Thing…



BUPT-PRIV

- **Evaluation Metric**

```
Average Precision  (AP) @[ OGPS=0.50:0.95 | area=   all | maxDets= 20 ] = 0.749
Average Precision  (AP) @[ OGPS=0.50      | area=   all | maxDets= 20 ] = 0.939
Average Precision  (AP) @[ OGPS=0.55      | area=   all | maxDets= 20 ] = 0.935
Average Precision  (AP) @[ OGPS=0.60      | area=   all | maxDets= 20 ] = 0.929
Average Precision  (AP) @[ OGPS=0.65      | area=   all | maxDets= 20 ] = 0.916
Average Precision  (AP) @[ OGPS=0.70      | area=   all | maxDets= 20 ] = 0.900
Average Precision  (AP) @[ OGPS=0.75      | area=   all | maxDets= 20 ] = 0.871
Average Precision  (AP) @[ OGPS=0.80      | area=   all | maxDets= 20 ] = 0.810
Average Precision  (AP) @[ OGPS=0.85      | area=   all | maxDets= 20 ] = 0.689
Average Precision  (AP) @[ OGPS=0.90      | area=   all | maxDets= 20 ] = 0.440
Average Precision  (AP) @[ OGPS=0.95      | area=   all | maxDets= 20 ] = 0.065
Average Precision  (AP) @[ OGPS=0.50:0.95 | area=medium | maxDets= 20 ] = 0.698
Average Precision  (AP) @[ OGPS=0.50:0.95 | area= large | maxDets= 20 ] = 0.771
Average Recall     (AR) @[ OGPS=0.50:0.95 | area=   all | maxDets= 20 ] = 0.828
Average Recall     (AR) @[ OGPS=0.50      | area=   all | maxDets= 20 ] = 0.974
Average Recall     (AR) @[ OGPS=0.75      | area=   all | maxDets= 20 ] = 0.921
Average Recall     (AR) @[ OGPS=0.50:0.95 | area=medium | maxDets= 20 ] = 0.751
Average Recall     (AR) @[ OGPS=0.50:0.95 | area= large | maxDets= 20 ] = 0.839
```

We get **74.9** AP on the test set !!!

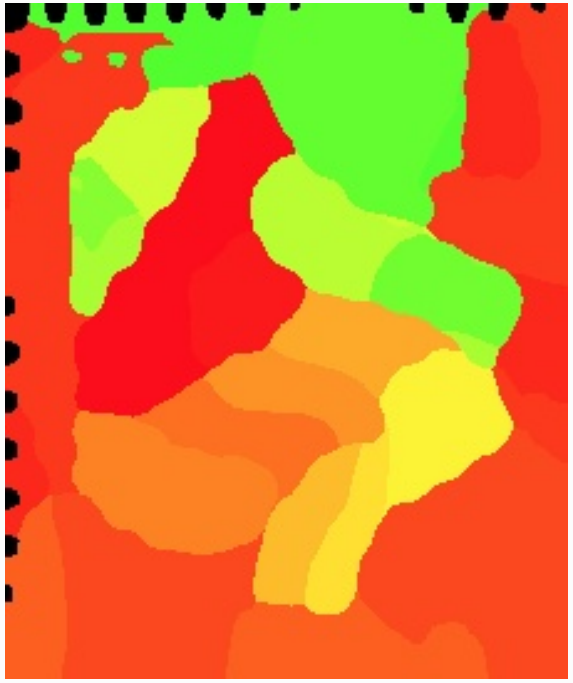
BUPT-PRIV

- **Evaluation Metric**

Mask

*

Index



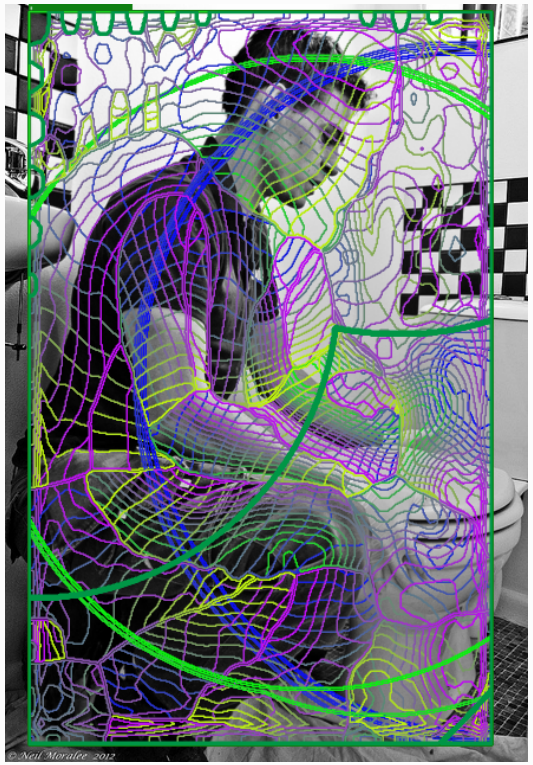
=

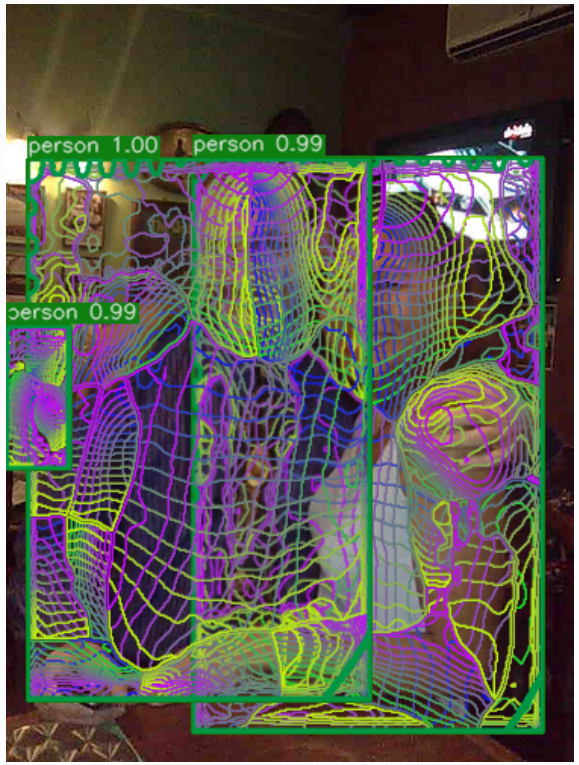
Index Refine

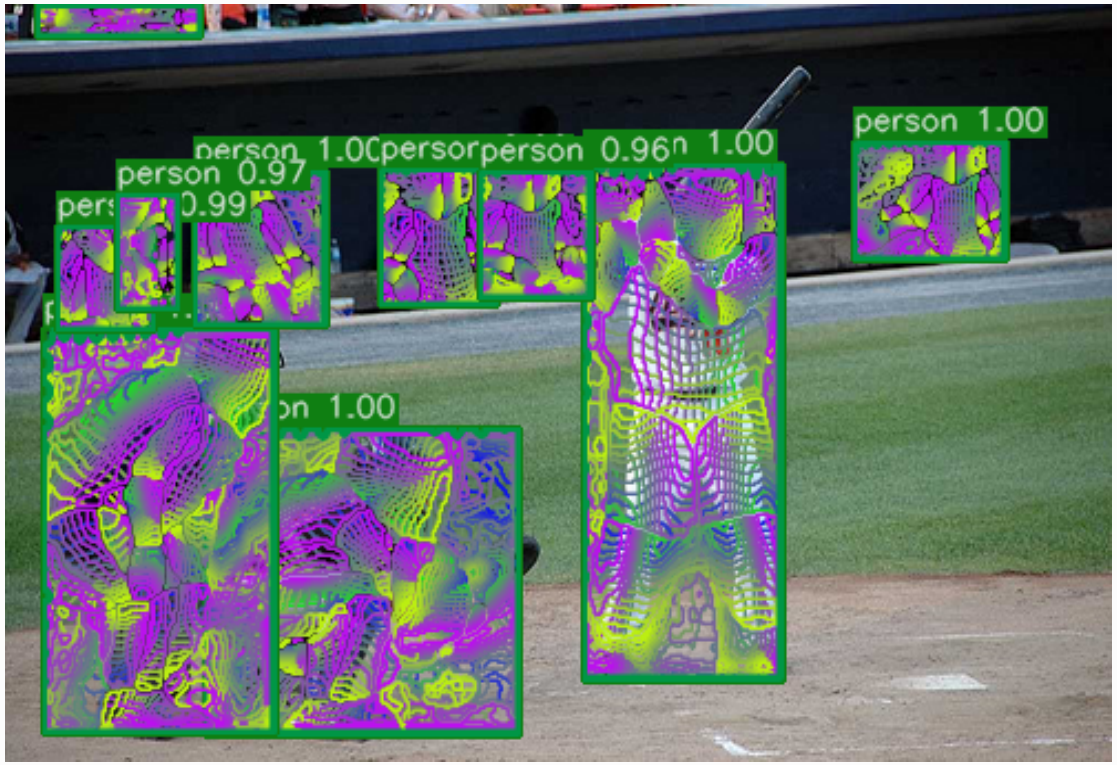
https://github.com/facebookresearch/DensePose/blob/master/detectron/core/test.py

```
948         # Removed squeeze calls due to singleton dimension issues
949         CurAnnIndex = np.argmax(CurAnnIndex, axis=0)
950         CurIndex_UV = np.argmax(CurIndex_UV, axis=0)
951         CurIndex_UV = CurIndex_UV * (CurAnnIndex>0).astype(np.float32)
```

- **Evaluation Metric**







BUPT-PRIV

- **Evaluation Metric**

COCO 2018 DensePose Val

	AP (GPS)	AP (GPS without Index refine)
DensePose ResNet50	48.9	61.5
DensePose ResNeXt101	55.5	67.2
Parsing R-CNN ResNeXt101	61.8	71.7
Parsing R-CNN ResNeXt101 +Test Aug	63.3	73.7

Without index refine, we can get unreasonabe high scores.

- **Evaluation Metric**

$$GPS_j^M = GPS_j * IoU_j^{mask} = GPS_j * \frac{I_{gt} \cap I_{pre}}{I_{gt} \cup I_{pre}}, I \text{ is class-agnostic index}$$

COCO 2018 DensePose Val

	AP (GPS)	AP (GPS without Index refine)	AP (GPS^M)	AP (GPS^M without Index refine)	Δ (GPS - GPS^M)
DensePose ResNet50	48.9	61.5	39.3	0.43	9.6
DensePose ResNeXt101	55.5	67.2	46.1	0.82	9.4
Parsing R-CNN ResNeXt101	61.8	71.7	51.7	0.64	9.9
Parsing R-CNN ResNeXt101 +Test Aug	63.3	73.7	53.1	0.73	9.8

The proposed new metric GPS^M

BUPT-PRIV

Thanks

Speaker: Yang Lu

BUPT-PRIV