

# Parsing R-CNN for Instance-Level Human Analysis

Lu Yang<sup>1</sup>, Qing Song<sup>1</sup>, Zhihui Wang<sup>1</sup> and Ming Jiang<sup>2</sup>

<sup>1</sup>Beijing University of Posts and Telecommunications, <sup>2</sup>WiWide Inc.

{soeaver, wangzh}@bupt.edu.cn, songqing\_priv@163.com, ming@wiwide.com

## Abstract

*Instance-level human analysis is common in real-life scenarios and has multiple manifestations, such as human part segmentation, dense pose estimation, human-object interactions, etc. Models need to distinguish different human instances in the image panel and learn rich features to represent the details of each instance. In this paper, we present an end-to-end pipeline for solving the instance-level human analysis, named Parsing R-CNN. It processes a set of human instances simultaneously through comprehensive considering the characteristics of region-based approach and the appearance of a human, thus allowing representing the details of instances.*

*Parsing R-CNN is very flexible and efficient, which is applicable to many issues in human instance analysis. Our approach outperforms all state-of-the-art methods on CIHP (Crowd Instance-level Human Parsing), MHP v2.0 (Multi-Human Parsing) and DensePose-COCO datasets. Based on the proposed Parsing R-CNN, we reach the 1st place in the COCO 2018 Challenge DensePose Estimation task. Code and models are publicly available<sup>1</sup>.*

## 1. Introduction

Human part segmentation [14, 26, 29, 40, 49], dense pose estimation [16] and human-object interactions [9, 13, 21] are the most fundamental and critical tasks in analyzing human in the wild. These tasks require human details at the instance level, which involve several perceptual tasks including detection, segmentation, estimation, i.e. There is a commonality between them, which can be regarded as an instance-level human analysis task.

Due to the successful development of convolutional neural networks [11, 23, 25, 37, 42, 45], great progress has been made in instance-level human analysis, especially in human part segmentation and dense pose estimation. Several related works [29, 49] follow the two stages pipeline, Mask R-CNN [17], which detects human in the image panel and

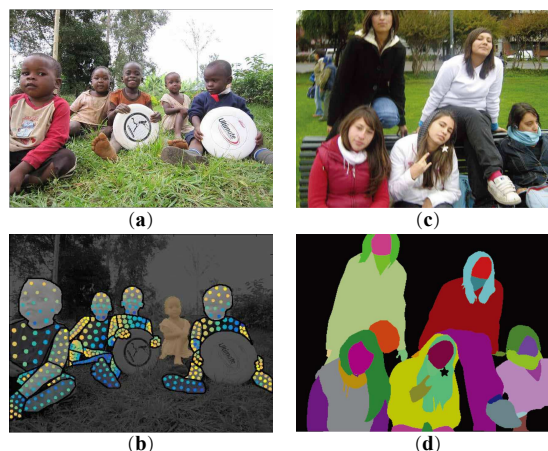


Figure 1. Example tasks of **instance-level human analysis**. (a) and (b) are samples for dense pose estimation. (c) and (d) are samples for human part segmentation.

predicts a class-aware mask in parallel with several convolutional layers. This method has achieved great success and wide application in instance segmentation [7, 17, 35, 47]. However, there are still several deficiencies in extending to the instance-level human analysis. One of the most important problems is that the design of the mask branch is used to predict a class-agnostic instance mask [17], but the instance-level human analysis requires more detailed features, which can not be well solved by existing methods. Besides, human analysis needs to correlate geometric and semantic relations between human parts / dense points, which is also missing. Therefore, in order to solve these problems, we propose Parsing R-CNN, which provides a concise and effective scheme for the instance-level human analysis tasks. This scheme can be successfully applied to the human part segmentation and dense pose estimation (Figure 1).

Our research explores the problem of instance-level human analysis from four aspects. First, to enhance feature semantic information and maintain feature resolution, proposals separation sampling is adopted. Human instances

<sup>1</sup><https://github.com/soeaver/Parsing-R-CNN>

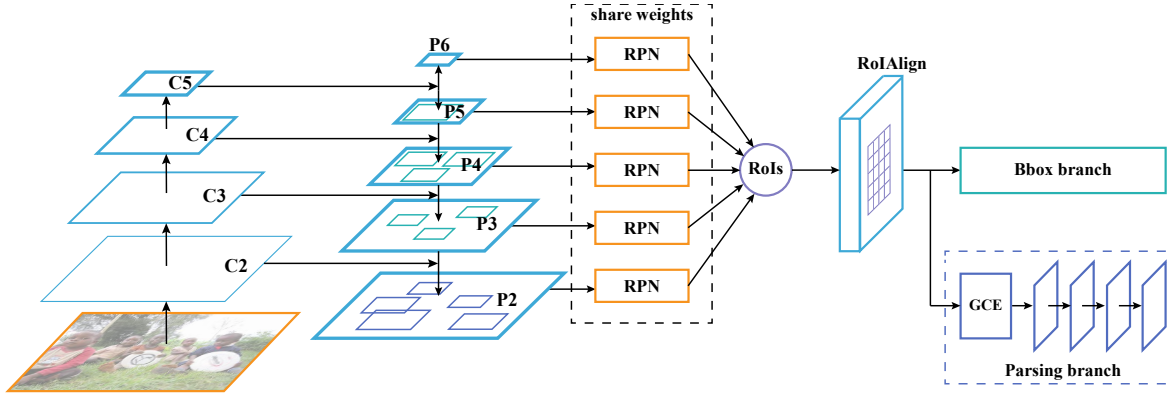


Figure 2. **Parsing R-CNN pipeline.** We adopt FPN backbone and RoIAlign operation, parsing branch is used for instance-level human analysis.

often occupy a relatively large proportion in images [33]. Therefore, RoIPool [10] operations are often performed on the coarser-resolution feature maps [32]. But this will lose a lot of details of the instance. In this work, we adopt the proposals separation sampling strategy, which using pyramid features at RPN [38] phase, but the RoIPool only performed on the finest level.

Second, to obtain more detailed information to distinguish different human parts or dense points in the instance, we enlarge the RoI resolution of the parsing branch. Human analysis tasks generally distinguish between dozens or even dozens of categories. It is necessary and effective to enlarge the resolution of the feature map.

Third, we propose a geometric and context encoding module to enlarge receptive field and capture the relationship between different parts of the human body. It is a lightweight component consisting of two parts. The first part is used to obtain multi-level receptive field and context information, and the second part is used to learn geometric correlation. With this module, class-aware masks with better quality are produced.

Finally, we analyze how to improve the performance of Parsing R-CNN efficiently by increasing the capacity of parsing branch. Based on this, we propose an appropriate branch composition scheme with high accuracy and small computational overhead.

With the proposed Parsing R-CNN, we achieve state-of-the-art performance on several datasets [14, 49]. For human part segmentation, Parsing R-CNN outperforms all known top-down or bottom-up methods both on CIHP [14] (Crowd Instance-level Human Parsing) and MHP v2.0 [49] (Multi-Human Parsing) datasets. For dense pose estimation, Parsing R-CNN achieves 64.1% mAP on COCO DensePose [16] test dataset, winning the 1st place in COCO 2018 Challenge DensePose task by a very large margin.

Parsing R-CNN is general and not limited to human part segmentation and dense pose estimation. We do not see any reason preventing it from finding broader applications in other human analysis tasks, such as human-object interactions, etc.

## 2. Related Work

**Region-based Approach.** The region-based approach [10, 11, 12, 17, 19, 32, 38] is very important in object detection, which has high accuracy and good expansibility. Generally speaking, the region-based approach generates a series of candidate object regions [38, 43, 50], then performs object classification and bounding-box regression in parallel within each candidate region. RoIPool and Region Proposal Network (RPN) are proposed by Fast R-CNN [10] and Faster R-CNN [38] respectively, which enable end-to-end learning and greatly improve speed and accuracy. Mask R-CNN [17] is an important milestone that successfully extends the region-based approach to instance segmentation and pose estimation, which has become an advanced pipeline in visual recognition. Mask R-CNN is flexible and robust to many follow-up improvements, and can be extended to more visual tasks [2, 3, 13, 16, 22, 39].

**Human Part Segmentation.** Human part segmentation is a core task of human analysis, which has been extensively studied in recent years [30, 31, 34]. Recently, Zhao *et al.* [49] put forward the MHP v2.0 (Multi-Human Parsing) dataset, which contains 25,403 elaborately annotated images with 58 fine-grained semantic category labels. Gong *et al.* [14] present another large-scale dataset called Crowd Instance-level Human Parsing (CIHP) dataset, which has 38,280 diverse human images. Each image in CIHP is labeled with pixel-wise annotations on 20 categories and instance-level identification. These datasets have greatly promoted

the research of human part segmentation, and considerable progress has been made.

On the other hand, Zhao *et al.* [49] propose the Nested Adversarial Network (NAN) for human part segmentation, which consists of three GAN-like sub-nets, respectively performing semantic saliency prediction, instance-agnostic parsing and instance-aware clustering. Gong *et al.* [14] design a detection-free Part Grouping Network (PGN) for instance-level human part segmentation. Although these works have achieved good performance, the segmentation result has great room for improvement and lack of an efficient region-based pipeline to unify the solution of instance-level human analysis.

**Dense Pose Estimation.** Guler *et al.* [16] propose an innovative dataset for instance-level human analysis, DensePose-COCO, a large-scale ground-truth dataset with image-to-surface correspondences manually annotated on 50k COCO images. Dense pose estimation can be understood as providing a refined version of human part segmentation (human part segmentation is crucial for dense pose estimation) and pose estimation, where one predicts continuous part labels of each human body. They also present the DensePose-RCNN, which combines the Dense Regression approach with the Mask-RCNN [17] architecture. Cross-cascading architecture is applied to the system that further improves accuracy. DensePose-RCNN gives a concise pipeline for dense pose estimation with good accuracy. However, many problems in the task are not discussed, such as the scale of human instance, the feature resolution and so on.

We consider that we can not treat human part segmentation and dense pose estimation in isolation. They are both specific tasks of instance-level human analysis and have a lot of commonalities. Therefore, based on the successful region-based approach, we propose Parsing R-CNN, a unified solution for instance-level human analysis.

### 3. Parsing R-CNN

Our goal is to leverage a unified pipeline for instance-level human analysis, which can achieve good performance in both human part segmentation, dense pose estimation and has the high scalability to other similar tasks [13, 39]. Like Mask R-CNN, the proposed Parsing R-CNN is conceptually simple, an additional parsing branch is used to generate the output of instance-level human analysis, as shown in Figure 2. In this section, we will introduce the motivation and content of Parsing R-CNN in detail.

#### 3.1. Proposals Separation Sampling

In FPN [32] and Mask R-CNN [17], the assignment strategy is adopted to collect the RoIs (Regions of Interest) and assign them to the corresponding feature pyramid according to the scale of RoIs. Formally, large RoIs will be assigned to

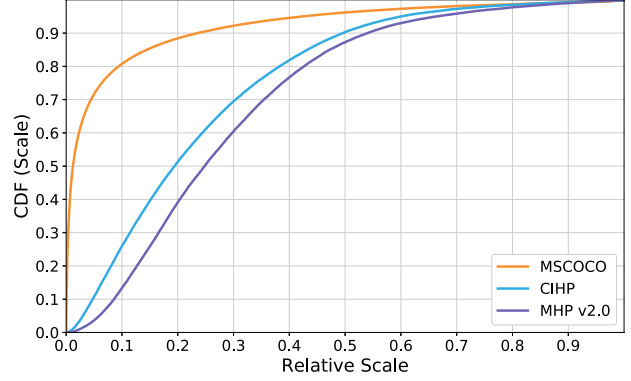


Figure 3. **Scale of instances** relative to the image (Relative Scale) *vs* fraction of instances in the dataset (CDF).

the coarser-resolution feature maps. This strategy is effective and efficient in object detection and instance segmentation. However, we find that this strategy is not the optimal solution in instance-level human analysis. Due to a small size human instance has less appearance information, only large size instances will be annotated. As shown in Figure 3, less than 20% of object instances in COCO dataset occupy more than 10% scale of the image, but this ratio is about 74% and 86% in CIHP and MHP v2.0 datasets respectively. According to the assignment strategy proposed by FPN, the most human instances will be assigned to the coarser-resolution feature maps. Instance-level human analysis often requires precise identification of some details of the human body, such as glasses and watches, or pixel areas of the left and right hand. But the coarser-resolution feature maps cannot provide more instance details, which is very harmful to human analysis.

To address this, we propose the *proposals separation sampling* (PSS) strategy that extracts features with details while preserves a multi-scale feature representation [28, 32, 36]. Our proposed change is simple: the bbox branch still adopts the scale assign strategy on the feature pyramid ( $P2-P5$ ) according to FPN [32], but the RoIPool/RoIAlign operation of parsing branch is only performed on the finest scale feature map of  $P2$ , as shown in Figure 2. In this way, we argue that object detection benefits from the pyramid representation while preserving human body details by extracting feature from the finest-resolution feature maps at parsing branch. With PSS, we observe that there has been a significant improvement in human part segmentation and dense pose estimation.

#### 3.2. Enlarging RoI Resolution

In some early region-based approaches, in order to make full use of the pre-train parameters, RoIPool operation converts an RoI into a small feature map with a fixed spatial extent of  $7 \times 7$  [10, 17, 32] (or  $14 \times 14$  followed by a convolutional layer with stride=2). This setup has been inherited in

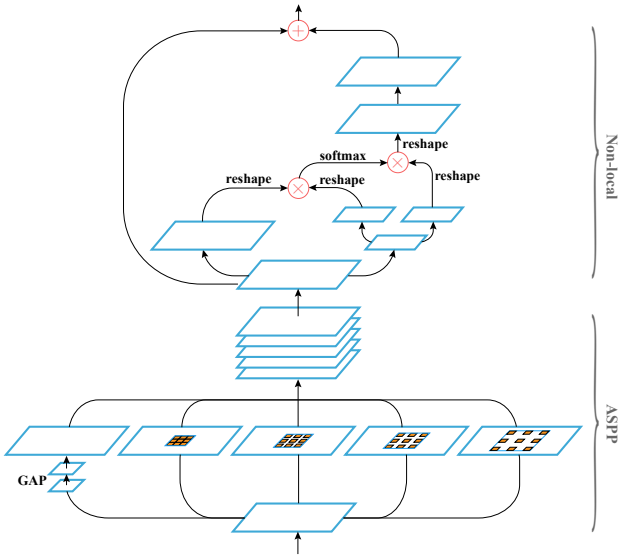


Figure 4. The proposed **Geometric and Context Encoding (GCE)** module. GAP denotes the global average pooling operator.

subsequent work and has proved its efficiency. Mask R-CNN uses  $14 \times 14$  scale RoIs in mask branch to generate segmentation masks, and the DensePose-RCNN [16] uses the same settings in the uv branch. But the most human instances occupy a large proportion of the feature maps, and too small RoI will lose a lot of detail. For example, a  $160 \times 64$  size human body whose size on  $P2$  is  $40 \times 16$ , and scaling to  $14 \times 14$  will undoubtedly reduce the prediction accuracy. In the tasks of object detection and instance segmentation, it is not very necessary to accurately predict the details of the instance. But in the instance-level human analysis, this will cause severe accuracy degradation.

In this work, we present the most simple and intuitive method: *enlarging RoI resolution (ERR)*. We employ  $32 \times 32$  RoI in parsing branch, which increases the computational cost of the branch, but improves the accuracy significantly. To address the training time and memory overhead associated with ERR, we decoupled the batch size of instance-level human analysis tasks from the detection task to a fixed value (e.g. 32) and find that this greatly increases the training speed and does not lead to accuracy degradation.

### 3.3. Geometric and Context Encoding

In previous works, the design of each branch is very succinct. A tiny FCN [37] is applied on the pooled feature grid for predicting pixel-wise masks of instances. However, using a tiny FCN in the parsing branch of instance-level human analysis will have three obvious drawbacks. First, the scale of different human parts varies greatly, which requires the feature maps capturing multi-scale information.

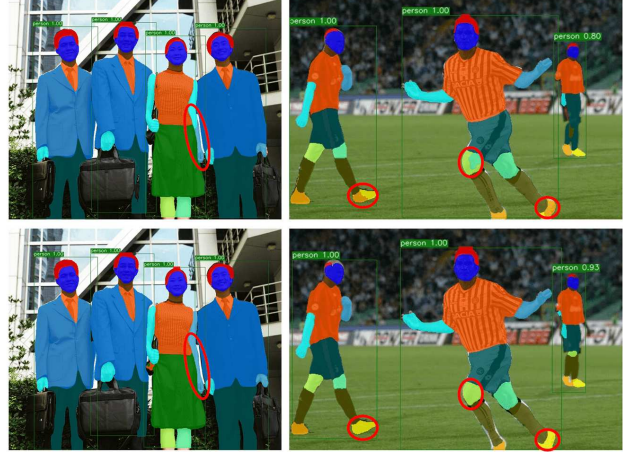


Figure 5. **Visualization results with / without GCE module.** The 1st row shows visualization results without GCE, and the 2nd shows ones with GCE. The GCE module can refine segmentation results of human instances (red circles).

Secondly, each human part is geometrically related, which requires a non-local representation [1]. Third,  $32 \times 32$  RoI needs a large receptive field, and stacking four or eight  $3 \times 3$  convolutional layers are not enough.

*Atrous spatial pyramid pooling (ASPP)* [4, 5, 6] is an effective module in semantic segmentation, where parallel atrous convolutional layers with different rates capture multi-scale information. Recently, Wang *et al.* [44] present the non-local operation and demonstrates outstanding performance on several benchmarks. Non-local operation is able to capture long-range dependencies which is of central importance in deep neural networks. For instance-level human analysis, we combine the advantages of ASPP and non-local, propose the *Geometric and Context Encoding (GCE)* module to replace FCN in parsing branch. As shown in Figure 5, the proposed GCE module can encode the geometric and context information of each instance, effectively distinguish different parts of the human body. In the GCE module, the ASPP part consists of one  $1 \times 1$  convolution and three  $3 \times 3$  convolutions with rates = (6, 12, 18). The image-level features are generated by global average pooling, which is followed by a  $1 \times 1$  convolution, and then bilinearly upsample the feature to the original  $32 \times 32$  spatial dimension. The non-local part adopts embedded Gaussian version, and a batch normalization [23] layer is added to the last convolutional layer. All the convolutional layers in GCE module have 256 channels. See Figure 4.

### 3.4. Increasing Parsing Branch Capacity

In the design of neural network for visual task, we often divide the network into several parts according to the characteristics of the features learned by different convolutional layers. For example, the layers closer to the output



	mAP <sup>bbox</sup>	mIoU	AP <sup>p</sup> <sub>50</sub>	AP <sup>p</sup> <sub>vol</sub>	PCP <sub>50</sub>
baseline	<b>67.7</b>	47.2	41.4	45.4	44.3
P2 only	66.4	47.7	42.6	45.8	45.1
PSS	67.5	<b>48.2</b>	<b>42.9</b>	<b>46.0</b>	<b>45.5</b>

Table 1. Ablation study on **proposals separation sampling (PSS)** strategy.

	fps	mIoU	AP <sup>p</sup> <sub>50</sub>	AP <sup>p</sup> <sub>vol</sub>	PCP <sub>50</sub>
baseline (14×14)	10.4	48.2	42.9	46.0	45.5
ERR (32×32)	9.1	50.7	47.9	47.6	49.7
ERR (32×32), 100 RoIs	<b>11.5</b>	50.5	47.5	47.3	49.0
ERR (64×64)	5.6	<b>51.5</b>	<b>49.0</b>	<b>47.9</b>	<b>50.8</b>

Table 2. Ablation study on **enlarging RoI resolution (ERR)** operation, the numbers in brackets are the RoI scales.

	mIoU	AP <sup>p</sup> <sub>50</sub>	AP <sup>p</sup> <sub>vol</sub>	PCP <sub>50</sub>
baseline	50.7	47.9	47.6	49.7
ASPP only	51.9	51.1	48.3	51.4
Non-local only	50.5	47.0	47.6	48.9
GCE	<b>52.7</b>	<b>53.2</b>	<b>49.7</b>	<b>52.6</b>

Table 3. Ablation study on **Geometric and Context Encoding (GCE)** module.

strongly respond to entire objects while other neurons are more likely to be activated by local texture and patterns. The region-based approach handles each RoI in parallel, so the branch of each task can be understood as an independent neural network. Increasing capacity in different locations of the network will bring different performance improvements. For example, Dai *et al.* [8] consider that it is most efficient to replace the standard convolutional layers with deformable convolutional layers in the last stage of ResNet [18, 20] or Aligned-Inception-ResNet. Wang *et al.* [44] suggest adding non-local operation in the first three stages of the network to achieve better results. In this work, we divide parsing branch into three parts: before GCE, GCE module and after GCE respectively. Through experiments we find that it is most efficient to increasing parsing branch capacity after GCE. Although GCE module can learn multi-scale information and geometric relations, we conjecture that the features learned by GCE need to be further refined to represent instance-level human information, such as human part segmentation and dense pose estimation.

## 4. Experiments

In this section, we compare the performance of Parsing R-CNN on three datasets, two human part segmentation datasets, and one dense pose estimation dataset.

### 4.1. Implementation Details

We implement the Parsing R-CNN based on Pytorch on a server with 8 NVIDIA Titan X GPUs. We adopt FPN and RoIAlign in all architectures, each of which is trained end-to-end. A mini-batch involves 2 images per GPU and

	mIoU	AP <sup>p</sup> <sub>50</sub>	AP <sup>p</sup> <sub>vol</sub>	PCP <sub>50</sub>
baseline	52.7	53.2	49.7	52.6
4conv + GCE	52.8	54.9	50.5	54.2
GCE + 4conv (IPBC)	<b>53.5</b>	58.5	<b>51.7</b>	56.5
4conv + GCE + 4conv	53.1	<b>58.8</b>	51.6	<b>56.7</b>

Table 4. Ablation study on **Increasing Parsing Branch Capacity (IPBC)** structure. 4conv denotes four convolutional layers with 3×3 kernels.

	LR	mIoU	AP <sup>p</sup> <sub>50</sub>	AP <sup>p</sup> <sub>vol</sub>	PCP <sub>50</sub>
ImageNet [41]	1x	53.5	58.5	51.7	56.5
	2x	55.3	61.8	53.3	59.3
	3x	<b>56.3</b>	<b>63.7</b>	<b>53.9</b>	<b>60.1</b>
COCO [33]	1x	55.9	63.1	53.5	60.4
	2x	57.1	64.7	54.2	61.9
	3x	<b>57.5</b>	<b>65.4</b>	<b>54.6</b>	<b>62.6</b>

Table 5. Results of different **pretrained models and maximum iterations** on CIHP<sub>val</sub>.

and each image has 512 sampled RoIs for bbox branch and 32 sampled RoIs for parsing branch. We train using image scales randomly sampled from [512, 864] pixels; inference is on a single scale of 800 pixels. For CIHP dataset, we train on `train` for 45k iterations, with a learning rate of 0.02 which is decreased by 10 at the 30k and 40k iteration. For MHP v2.0 dataset, the max iteration is half as long as the CIHP dataset with the learning rate change points scaled proportionally. For DensePose-COCO, we train for 130k iterations, starting from a learning rate of 0.002 and reducing it by 10 at 100k and 120k iterations. Other details are identical as in Mask R-CNN [15, 17].

### 4.2. Experiments on Human Part Segmentation

**Metrics and Baseline.** We evaluate the performance of human part segmentation from two scenarios. For semantic segmentation, we follow [24] to generate multi-person mask and adopt the standard mean intersection over union (mIoU) [37] to evaluate the performance. For instance-level performance, we use the Average Precision based on part (AP<sup>p</sup>) [49] for multi-human parsing evaluation, which uses part-level pixel IoU of different semantic part categories within a person instance to determine if one instance is a true positive. We report the AP<sup>p</sup><sub>50</sub> and AP<sup>p</sup><sub>vol</sub>. The former has a IoU threshold equal to 0.5, and the latter is the mean of the AP<sup>p</sup> at IoU thresholds ranging from 0.1 to 0.9, in increments of 0.1. In addition, we also report Percentage of Correctly parsed semantic Parts (PCP) metric [49].

For a fair comparison, our baseline adopts ResNet-50-FPN [18, 20, 32, 46] as backbone. The parsing branch consists of a stack of eight 3×3 512-d convolutional layers, followed by a deconvolution [48] layer and 2× bilinear upscaling. Following [17], the feature map resolution after RoIAlign is 14×14, so the output resolution is 56×56. During training, we apply a per-pixel softmax [37] as the

Baseline	PSS	ERR	GCE	IPBC	3x LR	COCO	mIoU	AP <sub>50</sub> <sup>p</sup>	AP <sub>vol</sub> <sup>p</sup>	PCP <sub>50</sub>
ResNet50										
✓							47.2	41.4	45.4	44.3
✓	✓						48.2	42.9	46.0	45.5
✓	✓	✓					50.7	47.9	47.6	49.7
✓	✓	✓	✓				52.7	53.2	49.7	52.6
✓	✓	✓	✓	✓			53.5	58.5	51.7	56.5
✓	✓	✓	✓	✓	✓		56.3	63.7	53.9	60.1
✓	✓	✓	✓	✓	✓	✓	<b>57.5</b>	<b>65.4</b>	<b>54.6</b>	<b>62.6</b>
Δ							<b>+10.3</b>	<b>+24.0</b>	<b>+9.2</b>	<b>+18.3</b>

Table 6. **Human part segmentation results on CIHP v1.1**. We adopt ResNet50-FPN as backbone, and gradually add *Proposals separation sampling* (PSS), *Enlarging RoI Resolution* (ERR), *Geometric and Context Encoding* (GCE) and *Increasing Parsing Branch Capacity* (IPBC). 3x LR denotes that we increase the number of iterations to three times of standard. We also report the performance of pretraining the whole model on COCO keypoint annotations (COCO).

Baseline	PSS	ERR	GCE	IPBC	3x LR	COCO	mIoU	AP <sub>50</sub> <sup>p</sup>	AP <sub>vol</sub> <sup>p</sup>	PCP <sub>50</sub>
ResNet50										
✓							28.7	10.1	33.4	21.8
✓	✓						29.8	10.6	33.8	22.2
✓	✓	✓					32.3	14.0	34.1	27.4
✓	✓	✓	✓				33.7	17.4	36.3	30.5
✓	✓	✓	✓	✓			34.3	20.0	37.6	32.7
✓	✓	✓	✓	✓	✓		36.2	24.5	39.5	37.2
✓	✓	✓	✓	✓	✓	✓	<b>37.0</b>	<b>26.6</b>	<b>40.3</b>	<b>40.0</b>
Δ							<b>+8.3</b>	<b>+16.5</b>	<b>+7.1</b>	<b>+18.2</b>

Table 7. **Human part segmentation results on MHP v2.0 v1.1**, we adopt ResNet50-FPN as backbone.

multinomial cross-entropy loss.

**Component Ablation Studies on CIHP.** We investigate various options of the proposed Parsing R-CNN in Section 3. In addition, we also study two other methods to improve performance: increasing the number of iterations and COCO pretraining. Our ablation study on CIHP [14] v1.1 from the baseline gradually to all components incorporated is shown in Table 6.

**1) Proposals Separation Sampling.** *Proposals separation sampling* (PSS) strategy improves the mIoU about 1.0 than the baseline. We also only adopt the P2 feature map both for bbox branch and parsing branch, the mIoU is reduced by 0.5 and bbox mAP is much worse. As shown in Table 1, instance-level metrics are promoted to a certain extent with PSS, which indicates that the proposed strategy is effective.

**2) Enlarging RoI Resolution.** In Table 2, we employ  $32 \times 32$  and  $64 \times 64$  RoI scales respectively, and find that the performance can be significantly improved than the original  $14 \times 14$  scale. The ERR ( $32 \times 32$ ) yields 2.8 improvement in terms of mIoU. For instance-level metrics, the improvements are even greater: 5.0, 1.7, 4.4 respectively. Moreover, the RoIs of parsing branch is parallel, so the speed is reduced by only 12%. And we can increase the inference speed by reducing the number of RoIs. If we use 100 RoIs at inference phase, the speed can be greatly improved and the performance basically does not drop. Relative to  $32 \times 32$ , the  $64 \times 64$  RoI scale can continue to improve the performance, but considering speed / accuracy trade-offs we consider that

	method	mIoU	AP <sub>50</sub> <sup>p</sup>	AP <sub>vol</sub> <sup>p</sup>	PCP <sub>50</sub>
CIHP	PGN (R101) <sup>†</sup> [14]	55.8	—	—	—
	CE2P (R101) [40]	<b>63.7</b>	—	—	—
	Parsing R-CNN (R50)	57.5	65.4	54.6	62.6
	Parsing R-CNN (X101)	59.8	69.1	55.9	66.2
	Parsing R-CNN (X101) <sup>†</sup>	61.1	<b>71.2</b>	<b>56.5</b>	<b>67.7</b>
MHP v2.0	Mask R-CNN [17]	—	14.9	33.8	25.1
	MH-Parser [27]	—	17.9	36.0	26.9
	NAN [49]	—	25.1	41.7	32.2
	CE2P (R101) [40]	<b>41.8</b>	<b>33.3</b>	42.2	—
	Parsing R-CNN (R50)	37.0	26.6	40.3	40.0
	Parsing R-CNN (X101)	40.3	30.2	41.8	44.2
	Parsing R-CNN (X101) <sup>†</sup>	<b>41.8</b>	32.5	<b>42.7</b>	<b>47.9</b>

Table 8. **Results of state-of-the-art methods on CIHP and MHP v2.0 v1.1**. <sup>†</sup> denotes using test-time augmentation.

using ERR ( $32 \times 32$ ) is efficient.

**3) Geometric and Context Encoding.** GCE module is the core component of Parsing R-CNN, which can significantly improve the mIoU about 2.0 than stacking of eight  $3 \times 3$  512-d convolutional layers, and it is even more lightweight. With or without Non-local operation, the ASPP part can still yield 1.2 improvement in terms of mIoU. But without ASPP part, only Non-local operation will cause performance degradation than the baseline. Results are shown In Table 3.

**4) Increasing Parsing Branch Capacity.** We divide the parsing branch into three parts: before GCE, GCE module and after GCE. As shown in Table 4, we find that the part before GCE is not necessary. It is most efficient to increasing parsing branch capacity after GCE, which significantly improves the semantic segmentation and instance-level metrics

Baseline	PSS	ERR	GCE	IPBC	COCO	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>M</sub>	AP <sub>L</sub>
ResNet50										
✓						48.9	84.9	50.8	43.8	50.6
✓	✓					50.9	86.1	53.4	46.4	52.4
✓	✓	✓				53.4	86.7	57.0	49.2	54.8
✓	✓	✓	✓			54.2	87.2	59.5	47.2	55.9
✓	✓	✓	✓	✓		55.0	87.6	59.8	50.6	56.6
✓	✓	✓	✓	✓	✓	<b>58.3</b>	<b>90.1</b>	<b>66.9</b>	<b>51.8</b>	<b>61.9</b>
Δ						<b>+9.4</b>	<b>+5.2</b>	<b>+16.1</b>	<b>+8.0</b>	<b>+11.3</b>
ResNeXt101										
✓						55.5	89.1	60.8	50.7	56.8
✓	✓	✓	✓	✓		59.1	91.0	69.4	53.9	63.1
✓	✓	✓	✓	✓	✓	<b>61.6</b>	<b>91.6</b>	<b>72.3</b>	<b>54.8</b>	<b>64.8</b>
Δ						<b>+6.1</b>	<b>+2.5</b>	<b>+11.5</b>	<b>+4.1</b>	<b>+8.0</b>

Table 9. **Dense pose estimation results on DensePose-COCO val**. We adopt ResNet50-FPN and ResNeXt101-32x8d-FPN as backbone respectively. The baseline is DensePose-RCNN.

(+0.8, +5.3, +2.0, +3.9 respectively). Considering speed / accuracy trade-offs, we adopt the GCE followed by four  $3 \times 3$  512-d convolutional layers as parsing branch.

**5) Increasing iterations and COCO pretraining.** Increasing iterations is a common method for improving performance. As shown in Table 5, we investigate the results of twice or three times as long as the standard schedule on CIHP val and find the improvements are obvious. We further pre-train the Parsing R-CNN models on the COCO keypoints annotations<sup>2</sup>, and initialize the parsing branch with the pose estimation weights. This strategy can further improve the performance about 1.1 to 2.4 in terms of mIoU. Combining these two methods, Parsing R-CNN yields 4.0 improvement in terms of mIoU. And for instance-level metrics, the improvements are 6.9, 2.9, 6.1 respectively.

As shown in Table 6, with these proposed components, the metrics of our Parsing R-CNN all exceed the baseline by a big margin. For semantic segmentation, Parsing R-CNN attains 57.6% mIoU which outperforms the baseline by a massive 10.3 points. For instance-level metrics, the improvement of Parsing R-CNN is more significant, which improves AP<sub>50</sub><sup>p</sup> by 24.0 points, AP<sub>vol</sub><sup>p</sup> by 9.2 points, and PCP<sub>50</sub> by 18.3 points.

**Component Ablation Studies on MHP v2.0.** We also gradually add *Proposals separation sampling* (PSS), *Enlarging RoI Resolution* (ERR), *Geometric and Context Encoding* (GCE) and *Increasing Parsing Branch Capacity* (IPBC) for ablation studies on MHP v2.0 [49] val, the results are shown in Table 7. There are 59 semantic categories in the MHP v2.0 dataset, and some of them are small-scale, so the baseline is worse than CIHP dataset. Parsing R-CNN is also significantly improving for MHP v2.0 dataset, which yields 10.3 improvement in terms of mIoU. For instance-level metrics, Parsing R-CNN improves AP<sub>50</sub><sup>p</sup> by 16.5 points, AP<sub>vol</sub><sup>p</sup> by 7.1 points, and PCP<sub>50</sub> by 18.2 points.

<sup>2</sup>Parsing R-CNN (without ERR) achieves 66.2% AP on COCO val, which yields 0.8 improvement than Mask R-CNN [17] with s1x LR.

**Comparisons with State-of-the-Art Methods.** Parsing R-CNN significantly improves the performance of human part segmentation. In order to further prove its effectiveness, we compare the proposed Parsing R-CNN to the state-of-the-art methods on CIHP and MHP v2.0 datasets, respectively.

For CIHP dataset, Parsing R-CNN uses ResNet-50-FPN outperforms the PGN [14] which using ResNet-101 by 1.7 points in terms of mIoU (Table 8). It is worth noting that PGN adopts multi-scale inputs and left-right flipped images to improve performance, while the result of Parsing R-CNN is without test-time augmentation. We also report the performance of Parsing R-CNN using ResNeXt-101-32x8d-FPN backbone, which attains 59.8% mIoU. Moreover, using ResNeXt-101-32x8d-FPN we report the results with multi-scale testing and horizontal flipping. This gives us a single model result of 61.1% mIoU. Because PGN only reports the Average Precision based on region (AP<sup>r</sup>), we can not directly compare the instance-level metrics. But by the result of semantic segmentation, we can also infer that Parsing R-CNN is superior to PGN on human parts segmentation task.

For MHP v2.0 dataset, we also report the results of Parsing R-CNN using ResNet-50-FPN and ResNeXt-101-32x8d-FPN (with or without test-time augmentation) backbones. In Table 8, compared with the previous state-of-the-art methods [17, 27, 49]<sup>3</sup>, Parsing R-CNN further improves results, with a margin of 7.4 points AP<sub>50</sub><sup>p</sup>, 1.0 points AP<sub>vol</sub><sup>p</sup> and 15.7 points PCP<sub>50</sub> over the best previous entry. Unfortunately, all the methods do not give the metric of semantic segmentation.

### 4.3. Experiments on Dense Pose Estimation

**Metrics and Baseline.** Following [16], we adopt the Average Precision (AP) at a number of *geodesic point similarity* (GPS) thresholds ranging from 0.5 to 0.95 as the evaluation metric. The structure of baseline model is exactly the same as the one of human part segmentation. We only replace the

<sup>3</sup>All the previous state-of-the-art methods only report the results evaluated on MHP v2.0 test.





Figure 6. Images in each row are **visual results of Parsing R-CNN** using ResNet50-FPN on CIHP val, MHP v2.0 val and DensePose-COCO val, respectively.

	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>M</sub>	AP <sub>L</sub>
DensePose-RCNN	56	89	64	51	59
yuchen.ma	57	87	66	48	61
ML-LAB	57	89	64	51	59
Min-Byeonguk	58	89	66	50	61
<b>Parsing R-CNN (ours)</b>	<b>64</b>	<b>92</b>	<b>75</b>	<b>57</b>	<b>67</b>

Table 10. **2018 COCO Challenge results** of Dense Pose Estimation task on *test*.

per-pixel softmax loss with the dense pose estimation losses.

**Component Ablation Studies on DensePose-COCO.** Like human part segmentation, we adopt the proposed Parsing R-CNN for dense pose estimation. Corresponding results are shown in Table 9. We adopt ResNet50-FPN and ResNeXt101-32x8d-FPN as backbone respectively. With ResNet50-FPN, Parsing R-CNN outperforms the baseline (DensePose-RCNN) by a good margin. Combining all the proposed components, our method achieves 55.0% AP, which yields 6.1 improvement than DensePose-RCNN. With COCO pretraining, Parsing R-CNN further improves 3.3 points AP. Parsing R-CNN also shows significant improvement of AP<sub>75</sub> (50.8% vs 66.9%), which indicates that our method is more accurate in points localization on the surface. As shown in Table 9, our Parsing R-CNN still increases the performance of dense pose estimation, when the model is upgraded from ResNet50 to ResNeXt101-32x8d, showing

good generalization of the Parsing R-CNN framework.

**COCO 2018 Challenge.** With Parsing R-CNN, we participated in the COCO 2018 DensePose Estimation Challenge, and reached the 1st place over all competitors. Table 10 summarizes the entries from the leaderboard of COCO 2018 Challenge. Our entry only utilizes a single model (ResNeXt101-32x8d), and attains 64.1% AP on DensePose-COCO test which surpasses the 2nd place by 6 points.

Qualitative results are illustrated in Figure 6. Images in each row are visual results of Parsing R-CNN using ResNet50-FPN on CIHP val, MHP v2.0 val and DensePose-COCO val, respectively.

## 5. Conclusion

We present a novel region-based approach Parsing R-CNN for instance-level human analysis, which achieves state-of-the-art results on several challenging benchmarks. Our approach explores the problem of instance-level human analysis from four aspects, and verified the effectiveness on human part segmentation and dense pose estimation tasks. Based on the proposed Parsing R-CNN, we reach the 1st place in the COCO 2018 Challenge DensePose Estimation task. In the future, we will extend Parsing R-CNN to more applications of instance-level human analysis.



## References

- [1] A. Buades, B. Coll, and J. Morel. A non-local algorithm for image denoising. In *CVPR*, 2005. 4
- [2] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 2
- [3] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 2
- [4] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2016. 4
- [5] L. Chen, G. Papandreou, F. Schroff, and H. Adam. Re-thinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*, 2017. 4
- [6] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 4
- [7] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016. 1
- [8] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *ICCV*, 2017. 5
- [9] C. Gao, Y. Zou, and J. Huang. ican: Instance-centric attention network for human-object interaction detection. In *BMVC*, 2018. 1
- [10] R. Girshick. Fast r-cnn. In *ICCV*, 2015. 2, 3
- [11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 2
- [12] R. Girshick, F. N. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. In *CVPR*, 2015. 2
- [13] G. Gkioxari, R. Girshick, P. Dollár, and K. He. Detecting and recognizing human-object interactions. In *CVPR*, 2018. 1, 2, 3
- [14] K. Gong, X. Liang, Y. Li, Y. Chen, and L. Lin. Instance-level human parsing via part grouping network. In *ECCV*, 2018. 1, 2, 3, 6, 7
- [15] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large mini-batch sgd: Training imagenet in 1 hour. *arXiv:1706.02677*, 2017. 5
- [16] R. Guler, N. Neverova, and I. Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018. 1, 2, 3, 4, 7
- [17] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017. 1, 2, 3, 5, 6, 7
- [18] K. He, X. Zhang, S. Ren, , and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5
- [19] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 2
- [20] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, 2016. 5
- [21] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. In *CVPR*, 2018. 1
- [22] R. Hu, P. Dollár, K. He, T. Darrell, and R. Girshick. Learning to segment every thing. In *CVPR*, 2018. 2
- [23] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 1, 4
- [24] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. *arXiv:1801.00868*, 2018. 5
- [25] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1
- [26] J. Li, J. Zhao, Y. Chen, S. Roy, S. Yan, J. Feng, and T. Sim. Multi-human parsing machines. In *ACM MM*, 2018. 1
- [27] J. Li, J. Zhao, Y. Wei, C. Lang, Y. Li, T. Sim, S. Yan, and J. Feng. Multi-human parsing in the wild. *arXiv:1705.07206*, 2017. 6, 7
- [28] Z. Li and F. Zhou. Fssd: Feature fusion single shot multibox detector. *arXiv:1712.00960*, 2017. 3
- [29] X. Liang, K. Gong, X. Shen, and L. Lin. Look into person: Joint human parsing and pose estimation network and a new benchmark. *TPAMI*, 2018. 1
- [30] X. Liang, S. Liu, X. Shen, J. Yang, L. Liu, J. Dong, L. Lin, and S. Yan. Deep human parsing with active template regression. *TPAMI*, 2015. 2
- [31] X. Liang, C. Xu, X. Shen, J. Yang, S. Liu, J. Tang, L. Lin, and S. Yan. Human parsing with contextualized convolutional neural network. In *ICCV*, 2015. 2
- [32] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2, 3, 5
- [33] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 5
- [34] S. Liu, X. Liang, X. Shen, J. Yang, C. Xu, L. Lin, X. Cao, and S. Yan. Matching-cnn meets knn: Quasiparametric human parsing. In *CVPR*, 2015. 2
- [35] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018. 1
- [36] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. Ssd: Single shot multibox detector. In *ECCV*, 2016. 3
- [37] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 4, 5
- [38] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2
- [39] G. Rohit, G. Georgia, T. Lorenzo, P. Manohar, and T. Du. Detect-and-track: Efficient pose estimation in videos. In *CVPR*, 2018. 2, 3
- [40] T. Ruan, T. Liu, Z. Huang, Y. Wei, S. Wei, Y. Zhao, and T. Huang. Devil in the details: Towards accurate single and multiple human parsing. In *AAAI*, 2019. 1, 6
- [41] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 5
- [42] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 1
- [43] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 2013. 2
- [44] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *CVPR*, 2018. 4, 5
- [45] Y. Wu and K. He. Group normalization. In *ECCV*, 2018. 1

- [46] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. [5](#)
- [47] L. Yi, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017. [1](#)
- [48] M. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. In *ECCV*, 2014. [5](#)
- [49] J. Zhao, J. Li, Y. Cheng, and J. Feng. Understanding humans in crowded scenes: Deep nested adversarial learning and a new benchmark for multi-human parsing. In *ACM MM*, 2018. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#)
- [50] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. [2](#)