Nondiscriminatory Treatment: a straightforward framework for multi-human parsing

Min Yan[a], Guoshan Zhang[a,*], Tong Zhang[a], Yuemin Zhang[a]

[a]*School of Electrical and Information Engineering, Tianjin University, 300072 Tianjin, China*

Abstract

Multi-human parsing aims to segment every body part of every human instance. Nearly all state-of-the-art methods follow the "detection first" or "segmentation first" pipelines. Different from them, we present an end-to-end and box-free pipeline from a new and more human-intuitive perspective. In training time, we directly do instance segmentation on humans and parts. More specifically, we introduce a notion of "indiscriminate objects with categories" which treats humans and parts without distinction and regards them both as instances with categories. In the mask prediction, each binary mask is obtained by a combination of prototypes shared among all human and part categories. In inference time, we design a brand-new grouping post-processing method that relates each part instance with one single human instance and groups them together to obtain the final human-level parsing result. We name our method as Nondiscriminatory Treatment between Humans and Parts for Human Parsing (NTHP). Experiments show that our network performs superiorly against state-of-the-art methods by a large margin on the MHP v2.0 and PASCAL-Person-Part datasets.

Keywords: Multiple Human Parsing, Mask Prototypes, Instance Segmentation

---

[*]Corresponding author
Email address: `zhanggs@tju.edu.cn` (Guoshan Zhang)

Figure 1: An example. (a) is the original picture, (b) is the bounding-box ground truth and (c)(d) are human-level mask labels. In (b), the Intersection over Union (IOU) between the two bounding boxes is 97.5% through calculation.

1. Introduction

Human parsing has become a significant aspect of human-centric analysis in recent years, which requires fine-grained semantic segmentation on pixel level. Extensive studies have been explored on parsing a single human in an image and obtained remarkable progress [1][2][3]. But in real cases, various numbers of persons are present simultaneously with interaction and occlusion, which heighten the need for better instance-level multi-human parsing methods. Multi-human parsing has many real-world applications, such as virtual reality [4], video surveillance [5], and human behavior analysis [6][7][8]. In this work, we aim at solving the task of multi-human parsing.

Due to the successful development of fully convolutional neural networks [9][10] [11][12][13], multi-human parsing has achieved great progress[14][15][16][17]. Existing works dealing with multi-human parsing can be divided into two categories, "detection first" and "segmentation first" paradigms.

"Detection first" methods consist of two stages. They detect human instances in the first stage then utilize regions-of-interest (ROIs) to parse the detected human instances in the second stage [14][16][45]. However, these methods may have the following drawbacks. 1)Human instances are often with irregular shapes, while ROIs are axis-aligned bounding-boxes, so the cropped features used to parse a single human can be excessive. We show an example in Figure

2

1 (b). 2)These methods strongly rely on the quality of the bounding boxes predicted. A Little deviation can result in huge faults. 3)Human parsing requires more detailed information in the second stage so that cropping the region to a 14×14 resolution of the conventional ROI align operation is not enough. 4)In these methods, the second stage must wait for the first stage to get accurate ROIs so that the processing procedure is slow.

On the other hand, "segmentation first" methods also have two stages. They apply a fine-grained semantic segmentation to obtain a pixel-level classification in the first stage then group pixels that belong to the same human instance in the second stage [17][48]. In the first stage, different parts demand different receptive fields due to their various sizes. Many approaches are committed to solving this problem, such as ASPP [18] , PSP [19], but lead to great computational complexity. In the second stage, some works separate different human instances via edges [17]. There is more than one boundary if a human is blocked. Figure 1 (b)(c) shows examples, both persons are in two parts, and previous approaches tend to group the apart hand by mistake.

Our research explores multi-human parsing from a brand-new point of view. We imitate the thinking process of human beings as shown in Figure 2, which views a human as a collection of parts and regards each part or human as an instance with category rather than pixels with categories. We simultaneously execute instance segmentation from two aspects, part and human, with nondiscriminatory treatment and predict their class-agnostic masks and instance categories. We named our notion "indiscriminate objects with categories". To better implement our notion, we propose a unified mask prediction module named Unified Mask Prediction Based on Prototypes (UMPP) which uses a unified prototype generation for both aspects. Finally, we design a simple grouping strategy that combines the separate parts belonging to the same human. Note that in our method, two aspects (human and part) are not completely unrelated. Both features are extracted from the same FPN structure but different levels, and share the prototypes, which can benefit from each other.

To evaluate our proposed framework, we conduct extensive experiments on

3

Figure 2: Solution formulation. According to human intuition, a human in the picture is composed of several parts with categories rather than dense pixels with semantic labels.

the MHP v2.0 [20] and PASCAL-Person-Part [21] datasets. We achieve state-of-the-art performance with 51.1 AP_{50}^p, 49.5 AP_{vol}^p, and 49.9 PCP_{50}, with a margin of 5.8 points AP_{50}^p, 2.7 points AP_{vol}^p, and 6.1 points PCP_{50} over the best previous entry on the MHP v2.0 dataset [20]. As for the PASCAL-Person-Part dataset, we also achieve state-of-the-art performance with 47.1 AP_{vol}^r and 53.9, 44.7, 31.9 AP^r with IoU thresholds of 0.5, 0.6, 0.7, separately with a margin of 4, 5.8, 6.4, 6.2 points over the best previous entry.

The main contributions of our work are concluded as follows:

• We design an end-to-end and box-free framework named NTHP for multi-human parsing keeping in line with our new notion of "indiscriminate objects with categories", which views both the humans and parts as object instances with categories rather than pixels with semantic labels.

• We propose a unified mask prediction module named Unified Mask Prediction Based on Prototypes (UMPP) formed by a linear combination of prototypes shared among humans and parts.

• We design a new grouping strategy in inference.

• We outperform all state-of-the-art methods on the MHP v2.0 [20] and PASCAL-Person-Part [21] datasets.

4

2. Related work

2.1. Human parsing

Human parsing has received a lot of attention in recent years. The hardest thing lies in obtaining the structure information within the human body. PCNet [22] designs a relational aggregation module and a dispersion module to deliver human structure information between different parts. Hierarchical Human Parsing [23] utilizes graph convolutional networks to understand hierarchical human layouts better. CorrPM [24] puts forward a heterogeneous non-local block to fully take advantage of the correlation between parsing, pose, and edge. All these works are within the scope of the single person parsing which can be viewed as a dense per-pixel classification problem.

Human parsing is elevated to a new level which has an unfixed number of persons in an image with the representation of the MHP dataset[15]. Nearly all methods follow the "detection first" or "segmentation first" pipeline in instance segmentation. PGN [17] adopts the "segmentation first" pipeline which appends a human-level instance-aware edge detection branch parallel with semantic segmentation and connects segments via the predicted boundary. Parsing R-CNN [16] and Unified Framework [14] adopt the "detection first" pipeline to parse distinct parts within the predicted human instances. CE2P [25] further appends a global parsing branch in parallel with the "detection first" pipeline to improve the performance. In contrast, we deal with humans and parts at the same time using the same structure.

2.2. Instance segmentation

Multi-human parsing can be viewed as a more complicated instance segmentation to some extent. Most of the methods tackling multi-human parsing are evolved from instance segmentation. Instance segmentation is one of the most common tasks in computer vision. The methods mainly follow the "top-down" or "bottom-up" pipeline. "Top-down" methods follow the principle of Mask R-CNN [26]. They first employ a detector to extract human-level features within a bounding-box. Then for each human instance, regions-of-interest

(ROIs) are cropped from the original picture-level feature maps. Finally, these ROIs are used to obtain the detailed segmentation results. Follow-up works are dedicated to improving the accuracy. PANet [27] adds another "bottom-up" path in FPN [28] to reinforce the feature representation and uses the "adaptive feature pooling" strategy to fuse the features from different levels to get better representation. Mask Scoring R-CNN [29] realizes the misalignment between the masks and the classification scores so that a MaskIoU head is appended to predict the quality of the masks predicted. Other approaches adopt the "bottom-up" strategy. For instance, Panoptic Deeplab [30] aims at panoptic segmentation but achieves good performance in instance segmentation too.

Recently, some methods that follow another pipeline named "one-stage" have obtained more interest due to their simplicity and easy-understanding nature. YOLACT [31] makes use of prototypes to generate instance masks and coefficients to get a linear combination of all predicted masks, after which a cropping operation is used to localize the objects. PolarMask [32]handles the task from a new perspective, which directly predicts the contours of instances in the polar coordinate. SOLO [33] directly predicts binary masks and mask categories by building two branches following the same backbone and FPN [28], one is the category branch, and the other is the mask branch. SOLOv2 [34] further splits the mask branch into mask feature and convolutional kernel paths to implement dynamic convolution. One-stage methods have difficulty localizing objects since it's commonly believed that convolutional operations are translation-invariant. YOLACT [31] obtains translation-variance by cropping the final mask with the predicted bounding box and SOLO [33][34] utilizes CoordConv [35] to get translation-variance. In this work, we introduce one-stage method to multi-human parsing and tackle the translation-invariant problem with CoordConv.

2.3. Prototypes

Learning prototypes (aka vocabulary or codebook) has been extensively explored in object detection[36][37]. But the prototypes in these works are used to represent features. YOLACT [31] learns prototypes specific to each image

6

Figure 3: NTHP structure. The topmost level of FPN is assigned to humans, and the rest are to parts. UMPP represents our proposed module (Unified Mask Prediction Based on Prototypes). UMPP has two types of inputs: one is the corresponding FPN output, and the other is the combination of all FPN outputs.

rather than global prototypes shared across the entire dataset. We obtain mask predictions using a linear combination of prototypes shared among humans and parts in this work.

3. Proposed method

In this section, we propose a straightforward approach extending one-stage methods original designed for instance segmentation to multi-human parsing [31][33][34]. We will introduce problem formulation, NTHP architecture, learning details, and inference procedure in detail.

3.1. Solution formulation

We consider the task from a human visual perspective. As shown in Figure 2, when a person looks at an image and does the same parsing job, he will first notice a human instance, then consider which parts belong to that human. There are three discoveries, 1) people will see an object as an instance with a category rather than a collection of pixels with categories, 2) parts and humans are both objects with little distinction from human perspective, 3) human instances are

7

made up of different part instances. Based on the first two discoveries, we introduce the notion of "indiscriminate objects with categories", which views humans and parts as indiscriminate objects and completes instance predictions with one structure. Based on the last discovery, we introduce our grouping strategy in inference time.

3.2. Overview

In this work, we directly predict the masks and their corresponding categories for both humans and parts with the same structure based on our notion of "indiscriminate objects with categories".

We show our structure in Figure 3. Similar to [33][34], we divide the input images into several $S \times S$ grids aligned to different levels of the feature pyramid network (FPN) [28]. One of the grids (i, j) is activated if it falls into the center region of any ground-truth mask. There are two branches following FPN, category branch ($C \times S \times S$) and mask branch ($S^2 \times H \times W$), where C equals the number of classes and H, W respectively represent feature height and width. There is a one-to-one relationship between masks and categories. If a grid (i, j) is activated, its category prediction is at (i, j) of the category branch with C channels, and the class-agnostic mask is at the $(i \cdot S + j)^{th}$ channel of the mask branch. We elaborate on the category branch in section 3.3. We name our structure in mask branch Unified Mask Prediction Based on Prototypes (UMPP) and tell more detail in 3.4.

There is a principle that, in most cases, two instances in an image either have different center locations or have different object sizes [33][38]. For the location issue, it is generally accepted that the original convolutions are translation-invariant to some degree. The solution is that we add Coordconv [35] in the mask branch, the same as SOLO [33][34] and YOLACT [31]. And for the size issue, we assign objects of different sizes to different levels of FPN (five levels in total). Human instances always have bigger sizes than part instances, so we assign the topmost level to human instances and the rest to the parts. Detailed parameters for each level are shown in Table 1.

8

With the obtained instances with categories, we group part instances concerning human instances to form the human-level parsing result. We elaborate on the grouping process in section 3.6.

3.3. Category branch

There are five levels in total with output space $C \times S \times S$ in the category branch. C equals 1 for the human category prediction branch and the number of classes excluded background for parts. There are $4 \times convs$ (3×3) for feature extraction and one for prediction in each level. We share weights for those levels assigned to parts. Note that the classes predicted here are for instances rather than pixels, thus worsening the difficulty.

One puzzling question is that without the structure based on the single human, the network may have difficulty differentiating confusing categories like left-right hands and left-right arms. But our network can successfully accomplish that. The reasons are: 1) Left and right have different directional properties, and the network can learn that information if the feature is fine enough. Thanks to the specific attribute of FPN, our network can satisfy the receptive fields required for objects of various sizes thus obtaining sufficient context information. 2) We use only one FPN so that parts can also get the human-level structure information through information flow.

3.4. Unified Mask Prediction Based on Prototypes (UMPP)

In UMPP, as shown in Figure 4, we learn mask coefficients separately for humans and parts and a unified collection of prototypes to form the class-agnostic masks.

Conventional convolutional operations are adept in taking advantage of the spatial coherence to obtain context information while fc operations are good at producing semantic vectors. In our structure, the branches for mask coefficients are parallel with the category branches, also with $4 \times convs$ (3×3) for feature extraction and one *conv* (1×1) for prediction. The former $4 \times convs$ can learn the context information for each grid due to the attribute of convolutions and

9

Figure 4: Unified Mask Prediction Based on Prototypes (UMPP) Blue/yellow indicates low/high values in the prototypes. We use all levels of FPN to form prototypes. "U" in brackets represents the upsampling operation

the last 1×1 $conv$ is the re-implementation for $S\times S$ fc operations to produce semantic vectors. For each grid, we predict K coefficients in the channel direction. The output space of mask coefficients F is $K\times S\times S$. Just like the category branch, the highest level is for humans. We use $K = 256$ for both humans and parts in experiments.

We learn a unified collection of prototypes P for humans and parts by applying feature pyramid fusion proposed in [39] but adding Coordconv [35] to the highest part level. We use all levels of FPN. Each level consists of a series of convolutions, and all levels are upsampled to 1/4 scale of the input image and summed together. There is a 1×1 $conv$ with ReLU at the end. The output space is $K\times H\times W$. In the center-of-right of Figure 4, we show some of the visualization results of prototypes. The first two examples are prototypes of humans while the others are parts. We can see that our network can successfully learn the location information, thus obtaining translation-variance, i.e., two individual persons. Such phenomenon largely owes to Coordconv [35] or the padding operation according to YOLACT[31], which demonstrates that padding gives

10

the network the ability to tell how far away from the image's edge a pixel is. We also consider that using a unified expression brings mutual-benefits. In FPN, higher-level features are convoluted using lower-level features and lower-level features combine higher-level features. That is to say, part features can get human-level information and learn more accurate context information, and in the meantime, human features can obtain the fusion of the part features to form the overall information.

The final mask predictions are obtained by a linear combination of prototypes, which is implemented as a sigmoid of a single matrix multiplication:

$$M = \sigma(PF^T) \tag{1}$$

where P is the matrix of prototypes, F is the matrix of mask coefficients, and σ is the sigmoid operation. The output space of M is $S^2 \times H \times W$.

3.5. NTHP learning

3.5.1. Label assignment

There are five levels of outputs with various resolutions that concentrate on objects of different sizes in the FPN of our structure. We assign the highest level to humans and the rest to parts. Besides, low levels have high resolutions and are responsible for small objects, thus requiring more grids. We show more details in Table 1. "Scale" means root mean square of the area of the minimum bounding box of the object.

In the head after each FPN level, there are S×S grids. A grid (i, j) is activated if it falls into the center region of any ground-truth mask, and 1) its category label is the class of the corresponding ground truth, 2) its mask label is the binary mask of the corresponding ground truth. Given a ground-truth mask, we calculate its mass center (c_x, c_y), width w, height h, then the center region is controlled by constant scale factors ε:$(c_x, c_y, \varepsilon w, \varepsilon h)$, we set $\varepsilon = 0.2$ following SOLO [33]. For each ground truth, there are no more than 9 grids activated. If the number of grids exceeds, the nine closest to the center point are used.

Table 1: Label assignment

Pyramid	F1	F2	F3	F4	F5
Object	Part	Part	Part	Part	Human
Grids (S)	40	36	24	16	20
Scale	<96	48∼192	96∼384	≥192	-

3.5.2. Loss function

We use the following training loss function:

$$L = L_{cp} + \lambda L_{mp} + L_{ch} + \lambda L_{mh} \qquad (2)$$

where L_{cp} is Focal loss [40] for the category classification for parts, L_{mp} is Dice Loss [41] for the mask prediction for parts, L_{ch} is Focal loss [40] for the category classification for humans and L_{mh} is Dice Loss [41] for the mask prediction for humans. λ is set to 3 in experiments. Note that we calculate classification loss for each grid but mask loss only for grids that have instance labels.

3.6. Grouping strategy in inference time

Our structure obtains four types of information, categories of part instances, masks of part instances, categories of human instances, and masks of human instances. In inference time, we need to group the parts based on humans. We propose the following steps:

Figure 5: Grouping strategy in inference time. & means the "and" operation.

(1) Assign the category with the highest classification score to the corresponding part mask. Pick k_{part} part masks with the highest category scores. We

12

predefine two thresholds, n_{part} for the max number of instances in an image (to decrease the memory cost), s_{part} for the minimum score value. k_{part} can be calculated by:

$$k_{part} = \min(num(score_{part} > s_{part}), n_{part}) \qquad (3)$$

where num means the number of masks that meets the criteria, and $score$ means the mask score, who is the product of the category score and the segmentation score. We use n_{part}=200, s_{part}=1/3.

(2) Pick k_{human} human masks whose scores are bigger than s_{human} and apply matrix NMS [34] to select the best ones. We use s_{human}=0.1 in experiments.

(3) For each selected part instance, calculate its overlapping ratio r_{part} towards every human instance:

$$r_{part} = \frac{area(intersection(part, human))}{area(part)} \qquad (4)$$

We show its pseudocode in Algorithm 1

(4) For each human instance, first pick the part instances whose overlapping ratio r_{part} is bigger than r_{human}, then assign every pixel with the category label using the selected part instances sorted by scores, finally use the 'and' operation between the human and the combination of the selected part instance masks to get the final result. r_{human} =2/3 in experiments. We use the product of the class-agnostic human-level mask score and the mean of category scores of pixels within the human mask as our final score.

$$score_{parsing} = score_{human} * \text{mean}(score_{pixel}) \qquad (5)$$

where $score_{parsing}$ is the final score for human parsing, $score_{human}$ is the human-level instance score, and $score_{pixel}$ is the category scores of pixels within the human mask.

The process is illustrated in Figure 5

Algorithm 1 Pseudocode of computing overlapping ratio r_{part}

```
# mm: matrix multiplication
# num_h: number of masks for humans
# num_p: number of masks for parts
# h_masks: binary masks for humans (num_h×h×w)
# p_masks: binary masks for parts (num_p×h×w)
h_masks = h_masks.reshape(num_h, h*w)
p_masks = p_masks.reshape(num_p, h*w)
inter_matrix = mm(h_masks, p_masks.permute(1,0))
part_matrix = p_masks.sum((-2,-1)).unsqueeze(0).expand(num_h,num_p)
ratio = inter_matrix/part_matrix
```

4. Experiments

We conduct comprehensive experiments and compare our method with state-of-the-art methods on the MHP v2.0 [20] and PASCAL-Person-Part [21] datasets.

4.1. Datasets

4.1.1. MHP v2.0 dataset

The MHP v2.0 [20] dataset is the most challenging dataset for multi-human parsing. It contains 15,403 training images, 5,000 validation images with 59 part classes. Each image contains 2-26 persons, with 3 on average. It has the maximum number of classes in multi-human parsing to the best of our knowledge.

4.1.2. PASCAL-Person-Part dataset

The PASCAL-Person-Part [21] dataset contains 1,716 images for training and 1,817 for testing. The annotations include six human parts: Head, Torso, Upper arms, Lower arms, Upper legs, and Lower legs. Each image contains 2.2 persons on average.

4.2. Experimental settings

4.2.1. Implementation details

We implement the NTHP based on Pytorch end-to-end on a server with 2 NVIDIA GeForce GTX 1080Ti GPUs. A mini-batch involves 6 images. We

use Group Normalization (GN) [13] with group size 32. The shorter side of the image scales randomly from [544, 864] pixels, and the longer side is set to 1333 pixels. The inference is on a single scale of 1333 pixels for the longer side and 800 pixels for the shorter side. All models are trained with ResNet50 [42] as backbone. As for the MHP dataset [20], we trained for 12 epochs with an initial learning rate of 0.001 per GPU per image, which is decreased by 10 at the 9th and again at the 11th epoch. Weight decay is 0.0001 and momentum is 0.9. Values of experiments with longer learning schedule are 36, 0.001, 10, 27, 33, 0.0001 and 0.9. As for the PASCAL-Person-Part [21] dataset, we train for 54 epochs and decrease the learning rate at the 45th and the 51th epoch. Other settings are the same as the MHP dataset.

4.2.2. Evaluation metric

We use separate evaluation metrics for the MHP v2.0 dataset [20] and the PASCAL-Person-Part dataset [21] for a fair comparison with other networks.

To evaluate our network on the MHP dataset [20], we use the metric named Average Precision based on Part (AP^p) which uses an average of part-level pixel IoU of different semantic part categories within a person instance to determine if one instance is a true positive and Percentage of Correctly Parsed Body Parts (PCP) which is the ratio between the correctly parsed categories and the total number of categories within a person. We use AP^p with an IOU threshold of 0.5 (AP^p_{50}), the average of AP^p with IOU thresholds ranging from 0.1 to 0.9 with a step size of 0.1 (AP^p_{vol}), and PCP with an IOU threshold of 0.5 (PCP_{50}). All these metrics are proposed by [15] to evaluate the network performance on the MHP dataset.

As for the PASCAL-Person-Part dataset, we use Mean Average Precision (AP^r) first proposed for evaluating instance segmentation results by SDS [43] and is used by nearly all methods to compare the performance of instance-level multi-human parsing result on the PASCAL-Person-Part dataset [21]. We use AP^r separately with IOU thresholds of 0.5, 0.6, 0.7, and AP^r_{vol} as an average of AP^r with IOU thresholds ranging from 0.1 to 0.9 with a step size of 0.1.

15

4.3. Experimental results

All our experiments are implemented on the MHP v2.0 dataset [20] with ResNet50 [42] as the backbone and trained for 12 epochs.

4.3.1. Unified prototype generation structure

We conduct experiments on two settings of using 1) a unified collection of prototypes for humans and parts, as shown in Figure 4, 2) two groups of prototypes. As for the second set, we use the same structure as the first set but duplicate it twice to deal with humans and parts separately. We show the result in Table 2.

From Table 2, we can see that adopting a unified structure brings a better result. Besides, the second setting costs more memory since it doubles the feature pyramid fusion process.

Table 2: Unified vs. Separate prototype generation structures.

Prototypes	AP_{50}^p	AP_{vol}^p	PCP_{50}
1) Unified	**41.8**	**46.2**	**41.9**
2) Separate	41.5	**46.2**	41.8

4.3.2. Not sharing weights between humans and parts

We also conduct a series of experiments on whether to share weights across different levels between humans and parts on mask coefficient and category branches.

We show the results in Table 3. We can see that not sharing weights between them obtains the best results. For the category branch, we consider that this is because the classes predicted between humans and parts are different so that the network needs to learn diverse features. As for the mask coefficient branch, the smallest unit predicted in prototypes is in part level, and sometimes humans are fused by a combination of parts, thus requiring different features.

16

Table 3: Share weights or not between humans and parts. Yes or no represents whether adding deformable convolutions or not.

Mask coefficients	Category	AP_{50}^p	AP_{vol}^p	PCP_{50}
no	no	**41.8**	**46.2**	**41.9**
yes	no	41.3	46.1	41.6
yes	yes	41.1	45.9	41.5

4.3.3. Prototype generation structure

We compare two choices of mask prototype generation structure, 1) feature pyramid fusion proposed in [39] with Coordconv [35] as shown in Figure 4, 2) the structure similar to the one used in YOLACT [31]. As for the second choice, we use $4 \times convs$ (3×3) for feature extraction and one (1×1) for prediction after the finest level of FPN with the final resolution 1/4 of the original image.

Our network utilizes the unique attribute of FPN [28], which concentrates on small objects on lower levels but big ones on higher levels. The features obtained by choice two are not enough to generate prototypes with a large range of scales. On the contrary, choice one obtains detailed information and semantic information simultaneously. As shown in Table 4, using the finest level alone degrades the performance by a large margin.

Table 4: Prototype generation structure.

Generation choice	AP_{50}^p	AP_{vol}^p	PCP_{50}
1) choice one	**41.8**	**46.2**	**41.9**
2) choice two	32.1	41.4	33.7

4.3.4. Other experiments

We find that adding deformable convolutions [44] in our network can obtain considerable improvement. We add deformable convolutions [44] in the backbone and the prototype generation module. Besides, increasing iterations is a

common method to improve the performance. We also try training our network for 36 epochs. We show the results in Table 5.

Table 5: Other experiments on MHP v2.0. DCN means deformable convolutions. Yes or no represents whether adding deformable convolutions or not.

Baseline	DCN	Epochs	AP_{50}^p	AP_{vol}^p	PCP_{50}
	no	12	41.8	46.2	41.9
ResNet50	yes	12	46.0	47.7	45.2
	yes	36	**51.1**	**49.5**	**49.9**

4.4. Comparisons with the state-of-the-art methods

We compare NTHP with state-of-the-art methods on the MHP v2.0 [20] and PASCAL-Person-Part [21] datasets.

We show the results on MHP v2.0 in Table 6. For a fair comparison with the best previous method RP R-CNN [45], we use ResNet50 [42] as the backbone. From Table 6, we can see that we outperform all state-of-the-art methods with 46.0 AP_{50}^p, 47.7 AP_{vol}^p, 45.2 PCP_{50} by training for only 12 epochs. Our best results are obtained by training for 36 epochs with 51.1 AP_{50}^p, 49.5 AP_{vol}^p, 49.9 PCP_{50}, with a margin of 5.8 points AP_{50}^p, 2.7 points AP_{vol}^p, and 6.1 points AP_{50}^p over the best previous entry. Note that we do not use flipping operation in training or any test-time augmentation on the MHP v2.0 dataset. We visualize good or bad results in Figure 6. We contrast the part-level and human-level instance segmentation results. And to demonstrate our classification performance, we use the same color to represent instances with the same category in the second and third lines.

We show the results on the PASCAL-Person-Part dataset [21] in Table 7. On this dataset, we use the metric AP^r for a fair comparison. All previous methods use multi-scale and flip training. Besides, MNC [47] was pre-trained on the Pascal VOC 2011/SBD dataset [49], and Holistic [48] and PGN [17] was pre-trained on the Pascal VOC dataset [50]. PGN [17] further uses test-

Table 6: Multi-human parsing results on the MHP v2.0 val set. *denotes longer learning schedule.

Mehods	Epochs	AP_{50}^p	AP_{vol}^p	PCP_{50}
MH-Parser [15]	-	17.9	36.0	26.9
Parsing R-CNN [16]	75	24.5	39.5	37.2
NAN [20]	~80	25.1	41.7	32.2
M-CE2P [25]	150	34.5	42.7	43.8
SemaTree [46]	200	34.4	42.5	43.5
RP R-CNN [45]	150	45.3	46.8	43.8
NTHP (ours)	12	46.0	47.7	45.2
NTHP* (ours)	36	**51.1**	**49.5**	**49.9**

Table 7: Multi-human parsing results on the PASCAL-Person-Part test dataset. #denotes using pretraining on other datasets. †denotes test-time augmentation.

Methods	Epochs	AP_{vol}^r	IoU threshold		
			0.5	0.6	0.7
MNC#[47]	~117	36.7	38.8	28.1	19.3
Holistic#[48]	~100	38.4	40.6	30.4	19.1
PGN#†[17]	~80	39.2	39.6	29.9	20.0
Unified#†[14]	~600	43.1	48.1	38.3	25.7
NTHP (ours)	54	43.9	49.1	40.0	28.1
NTHP#(ours)	54	**47.1**	**53.9**	**44.7**	**31.9**

time augmentation (multi-scale and flip strategy). Unified [14] employs the same setting as PGN [17]. Unlike our experiments on the MHP v2.0 dataset [20], we add flip operation in training due to the limited number of available images. From Table 7, we can see that with a shallower backbone ResNet50 and less training epochs and without any pre-training or test-time augmentation operation, our network outperform all state-of-the-art methods with 43.9 AP^r_{vol} and 49.1, 40.0, 28.1 AP^r with IoU thresholds of 0.5, 0.6, 0.7. We also show our result with pretraining on the MHP v2.0 dataset [20]. We obtain 47.1 AP^r_{vol} and 53.9, 44.7, 31.9 AP^r with IoU thresholds of 0.5, 0.6, 0.7, separately with a margin of 4, 5.8, 6.4, 6.2 points over the best previous entry. We also visualize good or bad results in Figure 7.

5. Conclusions

In this work, we design a straightforward and simple framework for multi-human parsing. Divergent from previous methods, we simultaneously conduct two types of instance segmentation for humans and parts using the same structure based on our newly proposed notion of "indiscriminate objects with categories". We also design a unified mask prediction module named UMPP which first generates a collection of prototypes shared among humans and parts, then makes a linear combination of them to get the binary masks. Besides, a simple post-processing strategy is developed to get the final results in a mutually beneficial way. We conduct extensive experiments on the challenging MHP v2.0 and PASCAL-Person-Part datasets and outperform all state-of-the-art methods.

Acknowledgements

Figure 6: Visualization of NTHP on the MHP v2.0 dataset. Pictures on the first line are original images, pictures on the second, fourth, and sixth lines are predictions, and those on the third, fifth, and last lines are ground truths.

Figure 7: Visualization of NTHP on the PASCAL-Person-Part dataset. Pictures on the first line are original images, pictures on the second, fourth, and sixth lines are predictions, and those on the third, fifth, and last lines are ground truths.

References

[1] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, T. L. Berg, Parsing clothing in fashion photographs, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012, pp. 3570–3577. `doi:10.1109/CVPR.2012.6248101`.

[2] J. Dong, Q. Chen, W. Xia, Z. Huang, S. Yan, A deformable mixture parsing model with parselets, in: Proceedings of the IEEE International Conference on Computer Vision, Institute of Electrical and Electronics Engineers Inc., 2013, pp. 3408–3415. `doi:10.1109/ICCV.2013.423`.

[3] X. Liang, C. Xu, X. Shen, J. Yang, J. Tang, L. Lin, S. Yan, Human Parsing with Contextualized Convolutional Neural Network, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (1) (2017) 115–127. `doi:10.1109/TPAMI.2016.2537339`.

[4] J. Lin, X. Guo, J. Shao, C. Jiang, Y. Zhu, S. C. Zhu, A virtual reality platform for dynamic human-scene interaction, in: SA 2016 - SIGGRAPH ASIA 2016 Virtual Reality Meets Physical Reality: Modelling and Simulating Virtual Humans and Environments, Association for Computing Machinery, Inc, New York, NY, USA, 2016, pp. 1–4. `doi:10.1145/2992138.2992144`.
URL `https://dl.acm.org/doi/10.1145/2992138.2992144`

[5] S. Liu, C. Wang, R. Qian, H. Yu, R. Bao, Y. Sun, Surveillance video parsing with single frame supervision, in: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Vol. 2017-Janua, Institute of Electrical and Electronics Engineers Inc., 2017, pp. 1013–1021. `arXiv:1611.09587, doi:10.1109/CVPR.2017.114`.

[6] L. Fan, W. Wang, S. C. Zhu, X. Tang, S. Huang, Understanding human gaze communication by spatio-temporal graph reasoning, in: Proceedings

of the IEEE International Conference on Computer Vision, Vol. 2019-Octob, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 5723–5732. `arXiv:1909.02144`, `doi:10.1109/ICCV.2019.00582`.

[7] P. Zhou, M. Chi, Relation parsing neural network for human-object interaction detection, in: Proceedings of the IEEE International Conference on Computer Vision, Vol. 2019-Octob, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 843–851. `doi:10.1109/ICCV.2019.00093`.

[8] T. Wang, T. Yang, M. Danelljan, F. S. Khan, X. Zhang, J. Sun, Learning human-object interaction detection using interaction points, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2020, pp. 4115–4124. `arXiv:2003.14023`, `doi:10.1109/CVPR42600.2020.00417`.

[9] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2014, pp. 580–587. `arXiv:1311.2524`, `doi:10.1109/CVPR.2014.81`.

[10] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, Tech. Rep. 6 (2017). `doi:10.1145/3065386`.
URL `http://code.google.com/p/cuda-convnet/`

[11] E. Shelhamer, J. Long, T. Darrell, Fully Convolutional Networks for Semantic Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (4) (2017) 640–651. `arXiv:1411.4038`, `doi:10.1109/TPAMI.2016.2572683`.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the Inception Architecture for Computer Vision, in: Proceedings of the IEEE

Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2016-Decem, IEEE Computer Society, 2016, pp. 2818–2826. `arXiv:1512.00567`, `doi:10.1109/CVPR.2016.308`.

[13] Y. Wu, K. He, Group normalization, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 11217 LNCS, Springer Verlag, 2018, pp. 3–19. `doi:10.1007/978-3-030-01261-8_1`.
URL `https://doi.org/10.1007/978-3-030-01261-8{_}1`

[14] H. Qin, W. Hong, W. C. Hung, Y. H. Tsai, M. H. Yang, A top-down unified framework for instance-level human parsing, in: 30th British Machine Vision Conference 2019, BMVC 2019, 30th British Machine Vision Conference, BMVC 2019, 2020.

[15] J. Li, J. Zhao, Y. Wei, C. Lang, Y. Li, T. Sim, S. Yan, J. Feng, Multi-human parsing in the wild, arXiv `arXiv:1705.07206`.
URL `http://arxiv.org/abs/1705.07206`

[16] L. Yang, Q. Song, Z. Wang, M. Jiang, Parsing R-CNN for instance-level human analysis, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2019-June, IEEE Computer Society, 2019, pp. 364–373. `arXiv:1811.12596`, `doi: 10.1109/CVPR.2019.00045`.

[17] K. Gong, X. Liang, Y. Li, Y. Chen, M. Yang, L. Lin, Instance-Level Human Parsing via Part Grouping Network, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 11208 LNCS, Springer Verlag, 2018, pp. 805–822. `arXiv:1808.00157`, `doi:10.1007/978-3-030-01225-0_47`.
URL `https://doi.org/10.1007/978-3-030-01225-0{_}47`

[18] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Con-

volution, and Fully Connected CRFs, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (4) (2018) 834–848. arXiv:1606.00915, doi:10.1109/TPAMI.2017.2699184.

[19] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network (2017). arXiv:1612.01105, doi:10.1109/CVPR.2017.660.

[20] J. Zhao, J. Li, Y. Cheng, T. Sim, S. Yan, J. Feng, Understanding humans in crowded scenes: Deep nested adversarial learning and a new benchmark for multi-human parsing, in: MM 2018 - Proceedings of the 2018 ACM Multimedia Conference, Association for Computing Machinery, Inc, New York, NY, USA, 2018, pp. 792–800. arXiv:1804.03287, doi:10.1145/3240508.3240509.
URL https://dl.acm.org/doi/10.1145/3240508.3240509

[21] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, A. Yuille, Detect what you can: Detecting and representing objects using holistic models and body parts, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2014, pp. 1979–1986. arXiv:1406.2031, doi:10.1109/CVPR.2014.254.

[22] X. Zhang, Y. Chen, B. Zhu, J. Wang, M. Tang, Part-aware context network for human parsing, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2020, pp. 8968–8977. doi:10.1109/CVPR42600.2020.00899.

[23] W. Wang, H. Zhu, J. Dai, Y. Pang, J. Shen, L. Shao, Hierarchical human parsing with typed part-relation reasoning, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2020, pp. 8926–8936. arXiv:2003.04845, doi:10.1109/CVPR42600.2020.00895.

[24] Z. Zhang, C. Su, L. Zheng, X. Xie, Correlating edge, pose with parsing, in: Proceedings of the IEEE Computer Society Conference on Computer

Vision and Pattern Recognition, IEEE Computer Society, 2020, pp. 8897–8906. `arXiv:2005.01431, doi:10.1109/CVPR42600.2020.00892.`

[25] T. Ruan, T. Liu, Z. Huang, Y. Wei, S. Wei, Y. Zhao, T. Huang, Devil in the details: Towards accurate single and multiple human parsing, in: arXiv, AAAI, 2018.

[26] K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, Vol. 2017-Octob, Institute of Electrical and Electronics Engineers Inc., 2017, pp. 2980–2988. `doi:10.1109/ICCV.2017.322.`

[27] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path Aggregation Network for Instance Segmentation, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2018, pp. 8759–8768. `arXiv:1803.01534, doi:10.1109/CVPR.2018.00913.`

[28] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Vol. 2017-Janua, Institute of Electrical and Electronics Engineers Inc., 2017, pp. 936–944. `doi:10.1109/CVPR.2017.106.`

[29] Z. Huang, L. Huang, Y. Gong, C. Huang, X. Wang, Mask scoring R-CNN, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2019-June, IEEE Computer Society, 2019, pp. 6402–6411. `arXiv:1903.00241, doi:10.1109/CVPR.2019.00657.`

[30] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, L. C. Chen, Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,

IEEE Computer Society, 2020, pp. 12472–12482. `arXiv:1911.10194,` `doi:10.1109/CVPR42600.2020.01249.`

[31] D. Bolya, C. Zhou, F. Xiao, Y. J. Lee, YOLACT: Real-time instance segmentation, in: Proceedings of the IEEE International Conference on Computer Vision, Vol. 2019-Octob, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 9156–9165. `arXiv:1904.02689,` `doi:10.1109/ICCV.2019.00925.`

[32] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. Shen, P. Luo, PolarMask: Single shot instance segmentation with polar representation, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2020, pp. 12190–12199. `arXiv:1909.13226,` `doi:10.1109/CVPR42600.2020.01221.`

[33] X. Wang, T. Kong, C. Shen, Y. Jiang, L. Li, SOLO: Segmenting Objects by Locations, in: arXiv, Springer, Cham, 2019, pp. 649–665. `arXiv:1912.04488,` `doi:10.1007/978-3-030-58523-5_38.`
URL `http://link.springer.com/10.1007/978-3-030-58523-5{_}38`

[34] X. Wang, R. Zhang, T. Kong, L. Li, C. Shen, SOLOv2: Dynamic and Fast Instance Segmentation, Tech. rep. (2020). `arXiv:2003.10152.`
URL `http://arxiv.org/abs/2003.10152`

[35] F. P. S. Rosanne Liu, Joel Lehman, Piero Molino, Eric Frank, Alex Sergeev, J. Yosinski., An intriguing failing of convolutional neural networks and the coordconv solution, NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 9628–9639.

[36] X. Ren, D. Ramanan, Histograms of sparse codes for object detection, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2013, pp. 3246–3253. `doi:10.1109/CVPR.2013.417.`

[37] S. Agarwal, D. Roth, Learning a sparse representation for object detection, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 2353, Springer Verlag, 2002, pp. 113–127. `doi:10.1007/3-540-47979-1_8`.
URL `https://link.springer.com/chapter/10.1007/3-540-47979-1{_}8`

[38] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2016-Decem, IEEE Computer Society, 2016, pp. 779–788. `arXiv:1506.02640`, `doi:10.1109/CVPR.2016.91`.

[39] A. Kirillov, R. Girshick, K. He, P. Dollar, Panoptic feature pyramid networks, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2019-June, IEEE Computer Society, 2019, pp. 6392–6401. `arXiv:1901.02446`, `doi:10.1109/CVPR.2019.00656`.

[40] T. Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal Loss for Dense Object Detection, in: Proceedings of the IEEE International Conference on Computer Vision, Vol. 2017-Octob, Institute of Electrical and Electronics Engineers Inc., 2017, pp. 2999–3007. `doi:10.1109/ICCV.2017.324`.

[41] F. Milletari, N. Navab, S. A. Ahmadi, V-Net: Fully convolutional neural networks for volumetric medical image segmentation, in: Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016, Institute of Electrical and Electronics Engineers Inc., 2016, pp. 565–571. `arXiv:1606.04797`, `doi:10.1109/3DV.2016.79`.

[42] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2016-Decem, IEEE Computer

Society, 2016, pp. 770–778. `arXiv:1512.03385`, `doi:10.1109/CVPR.2016.`
`90`.

[43] B. Hariharan, P. Arbeláez, R. Girshick, J. Malik, Simultaneous detection and segmentation, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 8695 LNCS, Springer Verlag, 2014, pp. 297–312. `arXiv:1407.1808`, `doi:10.1007/978-3-319-10584-0_20`.
URL `http://www.eecs.berkeley.edu/Research/Projects/CS/vision/shape/sds`.

[44] X. Zhu, H. Hu, S. Lin, J. Dai, Deformable convnets V2: More deformable, better results, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2019-June, IEEE Computer Society, 2019, pp. 9300–9308. `arXiv:1811.11168`, `doi:10.1109/CVPR.2019.00953`.

[45] L. Yang, Q. Song, Z. Wang, M. Hu, C. Liu, X. Xin, W. Jia, S. Xu, Renovating Parsing R-CNN for Accurate Multiple Human Parsing, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 12357 LNCS, Springer Science and Business Media Deutschland GmbH, 2020, pp. 421–437. `arXiv:2009.09447`, `doi:10.1007/978-3-030-58610-2_25`.
URL `https://link.springer.com/chapter/10.1007/978-3-030-58610-2{_}25`

[46] R. Ji, D. Du, L. Zhang, L. Wen, Y. Wu, C. Zhao, F. Huang, S. Lyu, Learning semantic neural tree for human parsing, Tech. rep. (2019). `arXiv:1912.09622`, `doi:10.1007/978-3-030-58601-0_13`.
URL `https://isrc.iscas.ac.cn/`

[47] J. Dai, K. He, J. Sun, Instance-Aware Semantic Segmentation via Multitask Network Cascades, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2016-

Decem, IEEE Computer Society, 2016, pp. 3150–3158. `arXiv:1512.04412`, `doi:10.1109/CVPR.2016.343`.

[48] Q. Li, A. Arnab, P. H. S. Torr, Holistic, Instance-Level Human Parsing, arXiv`arXiv:1709.03612`.
URL `http://arxiv.org/abs/1709.03612`

[49] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, J. Malik, Semantic contours from inverse detectors, in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 991–998. `doi:10.1109/ICCV.2011.6126343`.

[50] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, International Journal of Computer Vision 88 (2) (2010) 303–338. `doi:10.1007/s11263-009-0275-4`.
URL `http://www.flickr.com/`