Pose Partition Networks for Multi-person Pose Estimation

Xuecheng Nie[1](✉), Jiashi Feng[1], Junliang Xing[2], and Shuicheng Yan[1,3]

[1] ECE Department, National University of Singapore, Singapore, Singapore
niexuecheng@u.nus.edu, elefjia@nus.edu.sg, yanshuicheng@360.cn
[2] Institute of Automation, Chinese Academy of Sciences, Beijing, China
jlxing@nlpr.ia.ac.cn
[3] Qihoo 360 AI Institute, Beijing, China

Abstract. This paper proposes a novel Pose Partition Network (PPN) to address the challenging multi-person pose estimation problem. The proposed PPN is favorably featured by low complexity and high accuracy of joint detection and partition. In particular, PPN performs dense regressions from global joint candidates within a specific embedding space, which is parameterized by centroids of persons, to efficiently generate robust person detection and joint partition. Then, PPN infers body joint configurations through conducting graph partition for each person detection locally, utilizing reliable global affinity cues. In this way, PPN reduces computation complexity and improves multi-person pose estimation significantly. We implement PPN with the Hourglass architecture as the backbone network to simultaneously learn joint detector and dense regressor. Extensive experiments on benchmarks MPII Human Pose Multi-Person, extended PASCAL-Person-Part, and WAF show the efficiency of PPN with new state-of-the-art performance.

Keywords: Multi-person pose estimation · Pose partition
Dense regression

1 Introduction

Multi-person pose estimation aims to localize body joints of multiple persons captured in a 2D monocular image [7,22]. Despite extensive prior research, this problem remains very challenging due to the highly complex joint configuration, partial or even complete joint occlusion, significant overlap between neighboring persons, unknown number of persons and more critically the difficulties in allocating joints to multiple persons. These challenges feature the uniqueness of multi-person pose estimation compared with the simpler single-person setting [18,27]. To tackle these challenges, existing multi-person pose estimation

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01228-1_42) contains supplementary material, which is available to authorized users.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11209, pp. 705–720, 2018.
https://doi.org/10.1007/978-3-030-01228-1_42

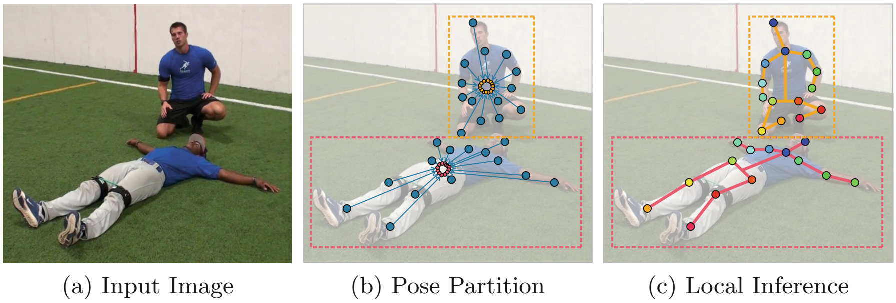

(a) Input Image (b) Pose Partition (c) Local Inference

Fig. 1. Pose Partition Networks for multi-person pose estimation. (a) Input image. (b) Pose partition. PPN models person detection and joint partition as a regression process inferred from joint candidates. (c) Local inference. PPN performs local inference for joint configurations conditioned on generated person detections with joint partitions

approaches usually perform joint detection and partition *separately*, mainly following two different strategies. The *top-down* strategy [7,8,13,20,23] first detects persons and then performs pose estimation for each single person individually. The *bottom-up* strategy [3,11,12,15,16,22], in contrast, generates all joint candidates at first, and then tries to partition them to corresponding person instances.

The top-down approaches directly leverage existing person detection models [17,24] and single-person pose estimation methods [18,27]. Thus they effectively avoid complex joint partitions. However, their performance is critically limited by the quality of person detections. If the employed person detector fails to detect a person instance accurately (due to occlusion, overlapping or other distracting factors), the introduced errors cannot be remedied and would severely harm performance of the following pose estimation. Moreover, they suffer from high joint detection complexity, which linearly increases with the number of persons in the image, because they need to run the single-person joint detector for each person detection sequentially.

In contrast, the bottom-up approaches detect all joint candidates at first by globally applying a joint detector for only once and then partition them to corresponding persons according to joint affinities. Hence, they enjoy lower joint detection complexity than the top-down ones and better robustness to errors from early commitment. However, they suffer from very high complexity of partitioning joints to corresponding persons, which usually involves solving NP-hard graph partition problems [11,22] on densely connected graphs covering the whole image.

In this paper, we propose a novel solution, termed the *Pose Partition Network* (PPN), to overcome essential limitations of the above two types of approaches and meanwhile inherit their strengths within a unified model for efficiently and effectively estimating poses of multiple persons in a given image. As shown in Fig. 1, PPN solves multi-person pose estimation problem by simultaneously (1) modeling person detection and joint partition as a regression process over all joint candidates and (2) performing local inference for obtaining joint categorization and association conditioned on the generated person detections.

In particular, PPN introduces a dense regression module to generate person detections with partitioned joints via votes from joint candidates in a carefully designed embedding space, which is efficiently parameterized by person centroids. This pose partition model produces joint candidates and partitions by running a joint detector for only one feed-forward pass, offering much higher efficiency than top-down approaches. In addition, the produced person detections from PPN are robust to various distracting factors, *e.g.*, occlusion, overlapping, deformation, and large pose variation, benefiting the following pose estimation. PPN also introduces a local greedy inference algorithm by assuming independence among person detections for producing optimal multi-person joint configurations. This local optimization strategy reduces the search space of the graph partition problem for finding optimal poses, avoiding high joint partition complexity challenging the bottom-up strategy. Moreover, the local greedy inference algorithm exploits reliable global affinity cues from the embedding space for inferring joint configurations within robust person detections, leading to performance improvement.

We implement PPN based on the Hourglass network [18] for learning joint detector and dense regressor, simultaneously. Extensive experiments on MPII Human Pose Multi-Person [1], extended PASCAL-Person-Part [28] and WAF [7] benchmarks evidently show the efficiency and effectiveness of the proposed PPN. Moreover, PPN achieves new state-of-the-art on all these benchmarks.

We make following contributions. (1) We propose a new one feed-forward pass solution to multi-person pose estimation, totally different from previous top-down and bottom-up ones. (2) We propose a novel dense regression module to efficiently and robustly partition body joints into multiple persons, which is the key to speeding up multi-person pose estimation. (3) In addition to high efficiency, PPN is also superior in terms of robustness and accuracy on multiple benchmarks.

2 Related Work

Top-Down Multi-person Pose Estimation. Existing approaches following top-down strategy sequentially perform person detection and single-person pose estimation. In [9], Gkioxari *et al.* proposed to adopt the Generalized Hough Transform framework to first generate person proposals and then classify joint candidates based on the poselets. Sun *et al.* [25] presented a hierarchical part-based model for jointly person detection and pose estimation. Recently, deep learning techniques have been exploited to improve both person detection and single-person pose estimation. In [13], Iqbal and Gall adopted Faster-RCNN [24] based person detector and convolutional pose machine [27] based joint detector for this task. Later, Fang *et al.* [8] utilized spatial transformer network [14] and Hourglass network [18] to further improve the quality of joint detections and partitions. Despite remarkable success, they suffer from limitations from

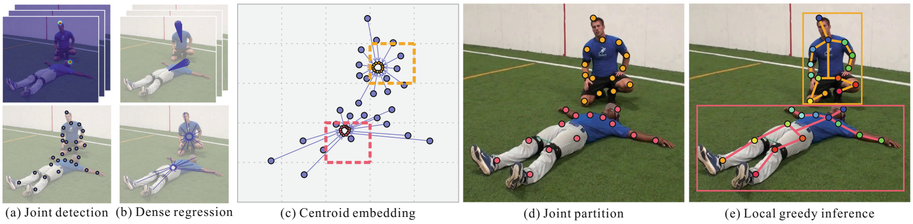


Fig. 2. Overview of the proposed Pose Partition Network for multi-person pose estimation. Given an image, PPN first uses a CNN to predict (a) joint confidence maps and (b) dense joint-centroid regression maps. Then, PPN performs (c) centroid embedding for all joint candidates in the embedding space via dense regression, to produce (d) joint partitions within person detections. Finally, PPN conducts (e) local greedy inference to generate joint configurations for each joint partition locally, giving pose estimation results of multiple persons

early commitment and high joint detection complexity. Differently, the proposed PPN adopts a one feed-forward pass regression process for efficiently producing person detections with partitioned joint candidates, offering robustness to early commitment as well as low joint detection complexity.

Bottom-Up Multi-person Pose Estimation. The bottom-up strategy provides robustness to early commitment and low joint detection complexity. Previous bottom-up approaches [3,11,19,22] mainly focus on improving either the joint detector or joint affinity cues, benefiting the following joint partition and configuration inference. For joint detector, fully convolutional neural networks, *e.g.*, Residual networks [10] and Hourglass networks [18], have been widely exploited. As for joint affinity cues, Insafutdinov *et al.* [11] explored geometric and appearance constraints among joint candidates. Cao *et al.* [3] proposed part affinity fields to encode location and orientation of limbs. Newell and Deng [19] presented the associative embedding for grouping joint candidates. Nevertheless, all these approaches partition joints based on partitioning the graph covering the whole image, resulting in high inference complexity. In contrast, PPN performs local inference with robust global affinity cues which is efficiently generated by dense regressions from the centroid embedding space, reducing complexity for joint partitions and improving pose estimation.

3 Approach

3.1 Pose Partition Model

The overall pipeline for the proposed Pose Partition Network (PPN) model is shown in Fig. 2. Throughout the paper, we use following notations. Let $\mathbf{I}$ denote an image containing multiple persons, $\mathbf{p}=\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\}$ denote spatial coordinates of N joint candidates from all persons in $\mathbf{I}$ with $\mathbf{p}_v=(x_v, y_v)^\top, \forall v=1, \dots, N$, and $\mathbf{u}=\{u_1, u_2, \dots, u_N\}$ denote the labels of corresponding joint candidates, in which $u_v \in \{1, 2, \ldots, K\}$ and K is the number of joint categories. For allocating joints via local inference, we also consider the proximities between joints, denoted as $\mathbf{b} \in \mathbb{R}^{N \times N}$. Here $\mathbf{b}_{(v,w)}$ encodes the proximity between the vth joint candidate $(\mathbf{p}_v, u_v)$ and the wth joint candidate $(\mathbf{p}_w, u_w)$, and gives the probability for them to be from the same person.

The proposed PPN with learnable parameters $\varTheta$ aims to solve the multi-person pose estimation task through learning to infer the conditional distribution $\mathbb{P}(\mathbf{p}, \mathbf{u}, \mathbf{b}|\mathbf{I}, \varTheta)$. Namely, given the image $\mathbf{I}$, PPN infers the joint locations $\mathbf{p}$, labels $\mathbf{u}$ and proximities $\mathbf{b}$ providing the largest likelihood probability. To this end, PPN adopts a regression model to *simultaneously* produce person detections with joint partitions implicitly and infers joint configuration $\mathbf{p}$ and $\mathbf{u}$ for each person detection locally. In this way, PPN reduces the difficulty and complexity of multi-person pose estimation significantly. Formally, PPN introduces latent variables $\mathbf{g} = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_M\}$ to encode joint partitions, and each $\mathbf{g}_i$ is a collection of joint candidates (without labels) belonging to a specific person detection, and M is the number of joint partitions. With these latent variables $\mathbf{g}$, $\mathbb{P}(\mathbf{p}, \mathbf{u}, \mathbf{b}|\mathbf{I}, \varTheta)$ can be factorized into

$$\mathbb{P}(\mathbf{p}, \mathbf{u}, \mathbf{b}|\mathbf{I}, \varTheta) = \sum_{\mathbf{g}} \mathbb{P}(\mathbf{p}, \mathbf{u}, \mathbf{b}, \mathbf{g}|\mathbf{I}, \varTheta) = \sum_{\mathbf{g}} \underbrace{\mathbb{P}(\mathbf{p}|\mathbf{I}, \varTheta)\mathbb{P}(\mathbf{g}|\mathbf{I}, \varTheta, \mathbf{p})}_{\text{partition generation}} \underbrace{\mathbb{P}(\mathbf{u}, \mathbf{b}|\mathbf{I}, \varTheta, \mathbf{p}, \mathbf{g})}_{\text{joint configuration}}, \tag{1}$$

where $\mathbb{P}(\mathbf{p}|\mathbf{I}, \varTheta)\mathbb{P}(\mathbf{g}|\mathbf{I}, \varTheta, \mathbf{p})$ models the joint partition generation process within person detections based on joint candidates. Maximizing the above likelihood probability gives optimal pose estimation for multiple persons in $\mathbf{I}$.

However, directly maximizing the above likelihood is computationally intractable. Instead of maximizing *w.r.t.* all possible partitions $\mathbf{g}$, we propose to maximize its lower bound induced by a single "optimal" partition, inspired by the EM algorithm [6]. Such approximation could reduce the complexity significantly without harming performance. Concretely, based on Eq. (1), we have

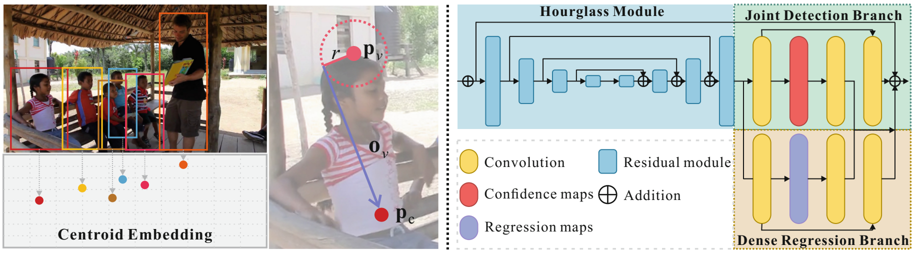
$$\mathbb{P}(\mathbf{p}, \mathbf{u}, \mathbf{b}|\mathbf{I}, \varTheta) \geq \mathbb{P}(\mathbf{p}|\mathbf{I}, \varTheta) \left\{ \max_{\mathbf{g}} \mathbb{P}(\mathbf{g}|\mathbf{I}, \varTheta, \mathbf{p}) \right\} \mathbb{P}(\mathbf{u}, \mathbf{b}|\mathbf{I}, \varTheta, \mathbf{p}, \mathbf{g}). \tag{2}$$

Here, we find the optimal solution by maximizing the above induced lower bound $\mathbb{P}(\mathbf{p}, \mathbf{u}, \mathbf{b}, \mathbf{g}|\mathbf{I}, \varTheta)$, instead of maximizing the summation. The joint partitions $\mathbf{g}$ disentangle independent joints and reduce inference complexity— only the joints falling in the same partition have non-zero proximities $\mathbf{b}$. Then $\mathbb{P}(\mathbf{p}, \mathbf{u}, \mathbf{b}, \mathbf{g}|\mathbf{I}, \varTheta)$ is further factorized as

$$\mathbb{P}(\mathbf{p}, \mathbf{u}, \mathbf{b}, \mathbf{g}|\mathbf{I}, \varTheta) = \mathbb{P}(\mathbf{p}, \mathbf{g}|\mathbf{I}, \varTheta) \times \prod_{\mathbf{g}_i \in \mathbf{g}} \mathbb{P}(\mathbf{u}_{\mathbf{g}_i}|\mathbf{I}, \varTheta, \mathbf{p}, \mathbf{g}_i)\mathbb{P}(\mathbf{b}_{\mathbf{g}_i}|\mathbf{I}, \varTheta, \mathbf{p}, \mathbf{g}_i, \mathbf{u}), \tag{3}$$

where $\mathbf{u}_{\mathbf{g}_i}$ denotes the labels of joints falling in the partition $\mathbf{g}_i$ and $\mathbf{b}_{\mathbf{g}_i}$ denotes their proximities. In the above probabilities, we define $\mathbb{P}(\mathbf{p}, \mathbf{u}, \mathbf{b}, \mathbf{g}|\mathbf{I}, \varTheta)$ as a Gibbs distribution:

$$\mathbb{P}(\mathbf{p}, \mathbf{u}, \mathbf{b}, \mathbf{g}|\mathbf{I}, \varTheta) \propto \exp\{-E(\mathbf{p}, \mathbf{u}, \mathbf{b}, \mathbf{g})\}, \tag{4}$$



(a) (b)

Fig. 3. (a) Centroid embedding via dense joint regression. Left image shows centroid embedding results for persons and right one illustrates construction of the regression target for a pixel (Sect. 3.3). (b) Architecture of Pose Partition Network. Its backbone is an Hourglass module (in blue block), followed by two branches: joint detection (in green block) and dense regression for joint partition (in yellow block) (Color figure online)

where $E(\mathbf{p}, \mathbf{u}, \mathbf{b}, \mathbf{g})$ is the energy function for the joint distribution $\mathbb{P}(\mathbf{p}, \mathbf{u}, \mathbf{b}, \mathbf{g}|\mathbf{I}, \Theta)$. Its explicit form is derived from Eq. (3) accordingly:

$$E(\mathbf{p}, \mathbf{u}, \mathbf{b}, \mathbf{g}) = -\varphi(\mathbf{p}, \mathbf{g}) - \sum_{\mathbf{g}_i \in \mathbf{g}} \Bigg(\sum_{\mathbf{p}_v \in \mathbf{g}_i} \psi(\mathbf{p}_v, u_v) + \sum_{\mathbf{p}_v, \mathbf{p}_w \in \mathbf{g}_i} \phi(\mathbf{p}_v, u_v, \mathbf{p}_w, u_w) \Bigg). \tag{5}$$

Here, $\varphi(\mathbf{p}, \mathbf{g})$ scores the quality of joint partitions $\mathbf{g}$ generated from joint candidates $\mathbf{p}$ for the input image $\mathbf{I}$, $\psi(\mathbf{p}_v, u_v)$ scores how the position $\mathbf{p}_v$ is compatible with label u_v, and $\phi(\mathbf{p}_v, u_v, \mathbf{p}_w, u_w)$ represents how likely the positions $\mathbf{p}_v$ with label u_v and $\mathbf{p}_w$ with label u_w belong to the same person, *i.e.*, characterizing the proximity $\mathbf{b}_{(v,w)}$. In the following subsections, we will give details for detecting joint candidates $\mathbf{p}$, generating optimal joint partitions $\mathbf{g}$, inferring joint configurations $\mathbf{u}$ and $\mathbf{b}$ along with the algorithm to optimize the energy function.

3.2 Joint Candidate Detection

To reliably detect human body joints, we use confidence maps to encode probabilities of joints presenting at each position in the image. The joint confidence maps are constructed by modeling the joint locations as Gaussian peaks, as shown in Fig. 2(a). We use $\mathbf{C}_j$ to denote the confidence map for the jth joint with $\mathbf{C}_j^i$ being the confidence map of the jth joint for the ith person. For a position $\mathbf{p}_v$ in the given image, $\mathbf{C}_j^i(\mathbf{p}_v)$ is calculated by $\mathbf{C}_j^i(\mathbf{p}_v) = \exp\left(-\|\mathbf{p}_v - \mathbf{p}_j^i\|_2^2/\sigma^2\right)$, where $\mathbf{p}_j^i$ denotes the groundtruth position of the jth joint of the ith person, and σ is an empirically chosen constant to control variance of the Gaussian distribution and set as 7 in the experiments. The target confidence map, which the proposed PPN model learns to predict, is an aggregation of peaks of all the persons in a single map. Here, we choose to take the maximum of confidence maps rather than average to remain distinctions between close-by peaks [3], *i.e.* $\mathbf{C}_j(\mathbf{p}_v) = \max_i \mathbf{C}_j^i(\mathbf{p}_v)$. During testing, we first find peaks with confidence scores greater than a given threshold τ (set as 0.1) on predicted confidence maps $\tilde{\mathbf{C}}$ for all types of joints. Then we perform Non-Maximum Suppression (NMS) to find the joint candidate set $\tilde{\mathbf{p}}=\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\}$.

3.3 Pose Partition via Dense Regression

Our proposed pose partition model performs dense regression over all the joint candidates to localize centroids of multiple persons and partitions joints into different person instances accordingly, as shown in Fig. 2(b) and (c). It learns to transform all the pixels belonging to a specific person to an identical single point in a carefully designed embedding space, where they are easy to cluster into corresponding persons. Such a dense regression framework enables partitioning joints by one single feed-forward pass, reducing joint detection complexity that troubles top-down solutions.

To this end, we propose to parameterize the joint candidate embedding space by the human body centroids, as they are stable and reliable to discriminate difference person instances even in presence of some extreme poses. We denote the constructed embedding space as $\mathcal{H}$. In $\mathcal{H}$, each person corresponds to a single point (*i.e.*, the centroid), and each point $\mathbf{h}_* \in \mathcal{H}$ represents a hypothesis about centroid location of a specific person instance. An example is given in the left image of Fig. 3(a).

Joint candidates are densely transformed into $\mathcal{H}$ and can collectively determine the centroid hypotheses of their corresponding person instances, since they are tightly related with articulated kinematics, as shown in Fig. 2(c). For instance, a candidate of the head joint would add votes for the presence of a person's centroid to the location just below it. A single candidate does not necessarily provide sufficient evidence for the exact centroid of a person instance, but the population of joint candidates can vote for the correct centroid with large probability and determine the joint partitions correctly. In particular, the probability of generating joint partition $\mathbf{g}_*$ at location $\mathbf{h}_*$ is calculated by summing the votes from different joint candidates together, *i.e.*

$$\mathbb{P}(\mathbf{g}_*|\mathbf{h}_*) \propto \sum_j w_j \Big(\sum_{\mathbf{p}_v \in \tilde{\mathbf{p}}} \mathbb{1}[\tilde{\mathbf{C}}_j(\mathbf{p}_v) \geq \tau] \exp\{-\|f_j(\mathbf{p}_v) - \mathbf{h}_*\|_2^2\} \Big), \qquad (6)$$

where $\mathbb{1}[\cdot]$ is the indicator function and w_j is the weight for the votes from jth joint category. We set w_j=1 for all joints assuming all kinds of joints equally contribute to the localization of person instances in view of unconstrained shapes of human body and uncertainties of presence of different joints. The function $f_j : \mathbf{p} \rightarrow \mathcal{H}$ learns to densely transform *every* pixel in the image to the embedding space $\mathcal{H}$. For learning f_j, we build the target regression map $\mathbf{T}_j^i$ for the jth joint of the ith person as follows:

$$\mathbf{T}_j^i(\mathbf{p}_v) = \begin{cases} \mathbf{o}_{j,v}^i / Z & \text{if } \mathbf{p}_v \in \mathcal{N}_j^i \\ 0 & \text{otherwise,} \end{cases}, \quad \mathbf{o}_{j,v}^i = (\mathbf{p}_{\mathrm{c}}^i - \mathbf{p}_v) = (x_{\mathrm{c}}^i - x_v, y_{\mathrm{c}}^i - y_v), \qquad (7)$$

where $\mathbf{p}_{\mathrm{c}}^i$ denotes the centroid position of the ith person, $Z=\sqrt{H^2+W^2}$ is the normalization factor, H and W denote the height and width of image $\mathbf{I}$,

$\mathcal{N}_j^i=\{\mathbf{p}_v| \ \|\mathbf{p}_v-\mathbf{p}_j^i\|_2\le r\}$ denotes the neighbor positions of the jth joint of the ith person, and r is a constant to define the neighborhood size, set as 7 in our experiments. An example is shown in right image of Fig. 3(a) for construction of a regression target of a pixel in a given image. Then, we define the target regression map $\mathbf{T}_j$ for the jth joint as the average for all persons by

$$\mathbf{T}_j(\mathbf{p}_v) = \frac{1}{N_v}\sum_i \mathbf{T}_j^i(\mathbf{p}_v), \tag{8}$$

where N_v is the number of non-zero vectors at position $\mathbf{p}_v$ across all persons. During testing, after predicting the regression map $\tilde{\mathbf{T}}_j$, we define transformation function f_j for position $\mathbf{p}_v$ as $f_j(\mathbf{p}_v)=\mathbf{p}_v + Z\tilde{\mathbf{T}}_j(\mathbf{p}_v)$. After generating $\mathbb{P}(\mathbf{g}_*|\mathbf{h}_*)$ for each point in the embedding space, we calculate the score $\varphi(\mathbf{p},\mathbf{g})$ as $\varphi(\mathbf{p},\mathbf{g})=\sum_i \log\mathbb{P}(\mathbf{g}_i|\mathbf{h}_i)$.

Then the problem of joint partition generation is converted to finding peaks in the embedding space $\mathcal{H}$. As there are no priors on the number of persons in the image, we adopt the Agglomerative Clustering [2] to find peaks by clustering the votes, which can automatically determine the number of clusters. We denote the vote set as $\mathbf{h}=\{\mathbf{h}_v|\mathbf{h}_v=f_j(\mathbf{p}_v), \tilde{\mathbf{C}}_j(\mathbf{p}_v)\ge\tau, \mathbf{p}_v\in\tilde{\mathbf{p}}\}$, and use $\mathcal{C}=\{\mathcal{C}_1,\ldots,\mathcal{C}_M\}$ to denote the clustering result on $\mathbf{h}$, where $\mathcal{C}_i$ represents the ith cluster and M is the number of clusters. We assume the set of joint candidates casting votes in each cluster corresponds to a joint partition $\mathbf{g}_i$, defined by

$$\mathbf{g}_i = \{\mathbf{p}_v|\mathbf{p}_v\in\tilde{\mathbf{p}}, \tilde{\mathbf{C}}_j(\mathbf{p}_v)\ge\tau, f_j(\mathbf{p}_v)\in\mathcal{C}_i\}. \tag{9}$$

3.4 Local Greedy Inference for Pose Estimation

According to Eq. (4), we maximize the conditional probability $\mathbb{P}(\mathbf{p},\mathbf{u},\mathbf{b},\mathbf{g}|\mathbf{I},\Theta)$ by minimizing energy function $E(\mathbf{p},\mathbf{u},\mathbf{b},\mathbf{g})$ in Eq. (5). We optimize $E(\mathbf{p},\mathbf{u},\mathbf{b},\mathbf{g})$ in two sequential steps: (1) generate joint partition set based on joint candidates; (2) conduct joint configuration inference in each joint partition locally, which reduces the joint configuration complexity and overcomes the drawback of bottom-up approaches.

After getting joint partition according to Eq. (9), the score $\varphi(\mathbf{p},\mathbf{g})$ becomes a constant. Let $\tilde{\mathbf{g}}$ denote the generated partition set. The optimization is then simplified as

$$\tilde{\mathbf{u}},\tilde{\mathbf{b}} = \arg\min_{\mathbf{u},\mathbf{b}}\Bigg(-\sum_{\mathbf{g}_i\in\tilde{\mathbf{g}}}\Big(\sum_{\mathbf{p}_v\in\mathbf{g}_i}\psi(\mathbf{p}_v,u_v)+\sum_{\mathbf{p}_v,\mathbf{p}_w\in\mathbf{g}_i}\phi(\mathbf{p}_v,u_v,\mathbf{p}_w,u_w)\Big)\Bigg). \tag{10}$$

Pose estimation in each joint partition is independent, thus inference over different joint partitions becomes separate. We propose the following local greedy inference algorithm to solve Eq. (10) for multi-person pose estimation. Given a joint partition $\mathbf{g}_i$, the unary term $\psi(\mathbf{p}_v,u_v)$ is the confidence score at $\mathbf{p}_v$ from the u_vth joint detector: $\psi(\mathbf{p}_v,u_v)=\tilde{\mathbf{C}}_{u_v}(\mathbf{p}_v)$. The binary term $\phi(\mathbf{p}_v,u_v,\mathbf{p}_w,u_w)$

Algorithm 1. Local greedy inference for multi-person pose estimation

```
input : joint candidates $\tilde{\mathbf{p}}$, joint partitions $\tilde{\mathbf{g}}$, joint confidence maps $\tilde{\mathbf{C}}$, dense
        regression maps $\tilde{\mathbf{T}}$, $\tau$.
output: multi-person pose estimation $\mathcal{R}$
initialization: $\mathcal{R} \leftarrow \varnothing$
for $\mathbf{g}_i \in \tilde{\mathbf{g}}$ do
    while $\mathbf{g}_i \neq \varnothing$ do
        Initialize single-person pose estimation $\mathcal{P} \leftarrow \varnothing$
        for $j$th joint category, $j = 1$ to $K$ do
            if $\mathcal{P} = \varnothing$ then
                Find root joint candidate in $\mathbf{g}_i$ for $\mathcal{P}$ by:
                $\mathbf{p}_* \leftarrow \arg\max_{\mathbf{p}_v \in \mathbf{g}_i} \tilde{\mathbf{C}}_j(\mathbf{p}_v)$
            else
                Find joint candidate closest to centroid $\mathbf{c}$:
                $\mathbf{p}_* \leftarrow \arg\max_{\mathbf{p}_v \in \mathbf{g}_i} \frac{\mathbb{1}[\tilde{\mathbf{C}}_j(\mathbf{p}_v) \geq \tau]}{\exp\{\|f_j(\mathbf{p}_v) - \mathbf{c}\|_2^2\}}$
            end
            if $\tilde{\mathbf{C}}_j(\mathbf{p}_*) \geq \tau$ then
                Update $\mathcal{P} \leftarrow \mathcal{P} \cup \{(\mathbf{p}_*, j)\}$, $\mathbf{g}_i \leftarrow \mathbf{g}_i \backslash \{\mathbf{p}_*\}$ Update $\mathbf{c}$ by averaging
                the person centroid hypotheses: $\mathbf{c} \leftarrow \sum_{(\mathbf{p}_v, n) \in \mathcal{P}} f_n(\mathbf{p}_v)/|\mathcal{P}|$
            end
        end
        Update $\mathcal{R} \leftarrow \mathcal{R} \cup \{\mathcal{P}\}$
    end
end
```

is the similarity score of votes of two joint candidates based on the global affinity cues in the embedding space:

$$\phi(\mathbf{p}_v, u_v, \mathbf{p}_w, u_w) = \mathbb{1}[\tilde{\mathbf{C}}_{u_v}(\mathbf{p}_v) \geq \tau]\mathbb{1}[\tilde{\mathbf{C}}_{u_w}(\mathbf{p}_w) \geq \tau] \exp\{-\|\mathbf{h}_v - \mathbf{h}_w\|_2^2\}, \quad (11)$$

where $\mathbf{h}_v = \mathbf{p}_v + Z\tilde{\mathbf{T}}_{u_v}(\mathbf{p}_v)$ and $\mathbf{h}_w = \mathbf{p}_w + Z\tilde{\mathbf{T}}_{u_w}(\mathbf{p}_w)$.

For efficient inference in Eq. (10), we adopt a greedy strategy which guarantees the energy monotonically decreases and eventually converges to a lower bound. Specifically, we iterate through each joint one by one, first considering joints around torso and moving out to limb. We start the inference with neck. For a neck candidate, we use its embedding point in $\mathcal{H}$ to initialize the centroid of its person instance. Then, we select the head top candidate closest to the person centroid and associate it with the same person as the neck candidate. After that, we update person centroid by averaging the derived hypotheses. We loop through all other joint candidates similarly. Finally, we get a person instance and its associated joints. After utilizing neck as root for inferring joint configurations of person instances, if some candidates remain unassigned, we utilize joints from torso, then from limbs, as the root to infer the person instance. After all candidates find their associations to persons, the inference terminates. See details in Algorithm 1.

4 Learning Joint Detector and Dense Regressor with CNNs

PPN is a generic model and compatible with various CNN architectures. Extensive architecture engineering is out of the scope of this work. We simply choose the state-of-the-art Hourglass network [18] as the backbone of PPN. Hourglass network consists of a sequence of Hourglass modules. As shown in Fig. 3(b), each Hourglass module first learns down-sized feature maps from the input image, and then recovers full-resolution feature maps through up-sampling for precise joint localization. In particular, each Hourglass module is implemented as a fully convolutional network. Skipping connections are added between feature maps with the same resolution symmetrically to capture information at every scale. Multiple Hourglass modules are stacked sequentially for gradually refining the predictions via reintegrating the previous estimation results. Intermediate supervision is applied at each Hourglass module.

Hourglass network was proposed for *single*-person pose estimation. PPN extends it to *multi*-person cases. PPN introduces modules enabling simultaneous joint detection (Sect. 3.2) and dense joint-centroid regression (Sect. 3.3), as shown in Fig. 3(b). In particular, PPN utilizes the Hourglass module to learn image representations and then separates into two branches: one produces the dense regression maps for detecting person centroids, via one 3 × 3 convolution on feature maps from the Hourglass module and another 1 × 1 convolution for classification; the other branch produces joint detection confidence maps. With this design, PPN obtains joint detection and partition in one feed-forward pass. When using multi-stage Hourglass modules, PPN feeds the predicted dense regression maps at every stage into the next one through 1 × 1 convolution, and then combines intermediate features with features from the previous stage.

For training PPN, we use ℓ_2 loss to learn both joint detection and dense regression branches with supervision at each stage. The losses are defined as

$$\begin{aligned} L^t_{\text{joint}} &\triangleq \sum_j \sum_v \|\tilde{\mathbf{C}}^t_j(\mathbf{p}_v) - \mathbf{C}_j(\mathbf{p}_v)\|^2_2 \\ L^t_{\text{regression}} &\triangleq \sum_j \sum_v \|\tilde{\mathbf{T}}^t_j(\mathbf{p}_v) - \mathbf{T}_j(\mathbf{p}_v)\|^2_2, \end{aligned} \tag{12}$$

where $\tilde{\mathbf{C}}^t_j$ and $\tilde{\mathbf{T}}^t_j$ represent predicted joint confidence maps and dense regression maps at the tth stage, respectively. The groundtruth $\mathbf{C}_j(\mathbf{p}_v)$ and $\mathbf{T}_j(\mathbf{p}_v)$ are constructed as in Sects. 3.2 and 3.3 respectively. The total loss is given by $L = \sum_{t=1}^{T}(L^t_{\text{joint}} + \alpha L^t_{\text{regression}})$, where $T = 8$ is the number of Hourglass modules (stages) used in implementation and weighting factor α is empirically set as 1.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate the proposed PPN on three widely adopted benchmarks: MPII Human Pose Multi-Person (MPII) dataset [1], extended PASCAL-Person-Part dataset [28], and "We Are Family" (WAF) dataset [7]. The MPII dataset

Table 1. Comparison with state-of-the-arts on the full testing set of MPII Human Pose Multi-Person dataset (AP)

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total	Time [s]
Iqbal and Gall [13]	58.4	53.9	44.5	35.0	42.2	36.7	31.1	43.1	10
Insafutdinov *et al.* [11]	78.4	72.5	60.2	51.0	57.2	52.0	45.4	59.5	485
Levinkov *et al.* [16]	89.8	85.2	71.8	59.6	71.1	63.0	53.5	70.6	-
Insafutdinov *et al.* [12]	88.8	87.0	75.9	64.9	74.2	68.8	60.5	74.3	-
Cao *et al.* [3]	91.2	87.6	77.7	66.8	75.4	68.9	61.7	75.6	1.24
Fang *et al.* [8]	88.4	86.5	78.6	70.4	74.4	73.0	65.8	76.7	1.5
Newell and Deng [19]	92.1	89.3	78.9	69.8	76.2	71.6	64.7	77.5	-
PPN (Ours)	**92.2**	**89.7**	**82.1**	**74.4**	**78.6**	**76.4**	**69.3**	**80.4**	**0.77**

consists of 3,844 and 1,758 groups of multiple interacting persons for training and testing respectively. Each person in the image is annotated for 16 body joints. It also provides more than 28,000 training samples for single-person pose estimation. The extended PASCAL-Person-Part dataset contains 3,533 challenging images from the original PASCAL-Person-Part dataset [4], which are split into 1,716 for training and 1,817 for testing. Each person is annotated with 14 body joints shared with MPII dataset, without pelvis and thorax. The WAF dataset contains 525 web images (350 for training and 175 for testing). Each person is annotated with 6 line segments for the upper-body.

Data Augmentation. We follow conventional ways to augment training samples by cropping original images based on the person center. In particular, we augment each training sample with rotation degrees sampled in $[-40°, 40°]$, scaling factors in $[0.7, 1.3]$, translational offset in $[-40\text{px}, 40\text{px}]$ and horizontally mirror. We resize each training sample to 256×256 pixels with padding.

Implementation. For MPII dataset, we reserve 350 images randomly selected from the training set for validation. We use the rest training images and all the provided single-person samples to train the PPN for 250 epochs. For evaluation on the other two datasets, we follow the common practice and finetune the PPN model pretrained on MPII for 30 epochs. To deal with some extreme cases where centroids of persons are overlapped, we slightly perturb the centroids by adding small offset to separate them. We implement our model with PyTorch [21] and adopt the RMSProp [26] for optimization. The initial learning rate is 0.0025 and decreased by multiplying 0.5 at the 150th, 170th, 200th, 230th epoch. In testing, we follow conventions to crop image patches using the given position and average person scale of test images, and resize and pad the cropped samples to 384×384 as input to PPN. We search for suitable image scales over 5 different choices. Specially, when testing on MPII, following previous works [3,19], we apply a single-person model [18] trained on MPII to refine the estimations. We use the standard Average Precision (AP) as performance metric on all the datasets, as suggested by [11,28]. Our codes and pre-trained models will be made available.

Table 2. Comparison with state-of-the-arts on the testing set of the extended PASCAL-Person-Part dataset (AP)

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Chen and Yuille [5]	45.3	34.6	24.8	21.7	9.8	8.6	7.7	21.8
Insafutdinov et al. [11]	41.5	39.3	34.0	27.5	16.3	21.3	20.6	28.6
Xia et at. [28]	58.0	52.1	43.1	37.2	22.1	30.8	31.1	39.2
PPN (Ours)	**66.9**	**60.0**	**51.4**	**48.9**	**29.2**	**36.4**	**33.5**	**46.6**

Table 3. Comparison with state-of-the-arts on testing set of WAF dataset (AP)

Method	Head	Shoulder	Elbow	Wrist	Total
Chen and Yuille [5]	83.3	56.1	46.3	35.5	55.3
Pishchulin *et al.* [22]	76.6	80.8	73.7	73.6	76.2
Insafutdinov *et al.* [11]	92.6	81.1	75.7	78.8	82.0
PPN (Ours)	**93.1**	**82.9**	**83.5**	**79.9**	**84.8**

5.2 Results and Analysis

MPII. Table 1 shows the evaluation results on the full testing set of MPII. We can see that the proposed PPN achieves overall 80.4% AP and significantly outperforms previous state-of-the-art achieving 77.5% AP [19]. In addition, the proposed PPN improves the performance for localizing all the joints consistently. In particular, it brings remarkable improvement over rather difficult joints mainly caused by occlusion and high degrees of freedom, including wrists (74.4% vs 69.8% AP), ankles (69.3% vs 64.7% AP), and knees (with absolute 4.8% AP increase over [19]), confirming the robustness of the proposed pose partition model and global affinity cues to these distracting factors. These results clearly show PPN is outstandingly effective for multi-person pose estimation. We also report the computational speed of PPN[1] in Table 1. PPN is about 2 times faster than the bottom-up approach [3] with state-of-the-art speed for multi-person pose estimation. This demonstrates the efficiency of performing joint detection and partition simultaneously in our model.

PASCAL-Person-Part. Table 2 shows the evaluation results. PPN provides absolute 7.4% AP improvement (46.6% vs 39.2% AP) over the state-of-the-art [28]. Moreover, the proposed PPN brings significant improvement on difficult joints, such as wrist (48.9% vs 37.2% AP). These results further demonstrate the effectiveness and robustness of our model for multi-person pose estimation.

WAF. As shown in Table 3, PPN achieves overall 84.8% AP, bringing 3.4% improvement over the best bottom-up approach [11]. PPN achieves the best

[1] The runtime time is measured on CPU Intel I7-5820K 3.3GHz and GPU TITAN X (Pascal). The time is counted with 5 scale testing, not including the refinement time by single-person pose estimation.

Table 4. Ablation experiments on MPII validation set (AP)

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total	InferTime [ms]
PPN-Full	94.4	90.0	81.3	72.1	77.8	72.7	64.7	79.0	1.9
PPN-w/o-Partition	93.2	89.3	79.9	70.1	78.8	73.1	65.7	78.6	3.4
PPN-w/o-LGI	93.1	89.1	79.5	68.5	79.0	71.4	64.4	77.8	-
PPN-w/o-Refinement	90.4	86.8	79.3	69.8	77.5	69.3	61.9	76.4	-
PPN-256 × 256	91.0	87.1	78.6	70.2	76.7	70.5	60.0	76.3	-
PPN-Vanilla	90.5	86.4	77.1	69.4	72.2	67.7	60.2	74.8	-

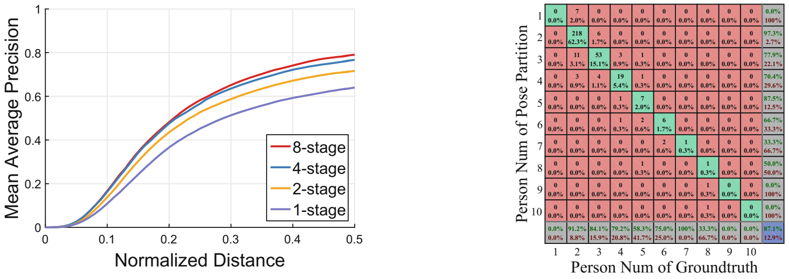
performance for all upper-body joints. In particular, it gives the most significant performance improvement on the elbow, about 10.3% higher than previous best results. These results verify the effectiveness of the proposed PPN for tackling the multi-person pose estimation problem.

Qualitative Results. Visualization examples of pose partition, local inference, and multi-person pose estimation by the proposed PPN on these three datasets are provided in the supplemental materials.

5.3 Ablation Analysis

We conduct ablation analysis for the proposed PPN model using the MPII validation set. We evaluate multiple variants of our proposed PPN model by removing certain components from the full model ("PPN-Full"). "PPN-w/o-Partition" performs inference on the whole image without using obtained joint partition information, which is similar to the pure bottom-up approaches. "PPN-w/o-LGI" removes the local greedy inference phase. It allocates joint candidates to persons through finding the most activated position for each joint in each joint partition. This is similar to the top-down approaches. "PPN-w/o-Refinement" does not perform refinement by using single-person pose estimator. We use "PPN-256 × 256" to denote testing over 256 × 256 images and "PPN-Vanilla" to denote single scale testing without refinement.

From Table 4, "PPN-Full" achieves 79.0% AP and the joint partition inference only costs 1.9ms, which is very efficient. "PPN-w/o-Partition" achieves slightly lower AP (78.6%) with slower inference speed (3.4ms). The results confirm effectiveness of generating joint partitions by PPN—inference within each joint partition individually reduces complexity and improves pose estimation over multi-persons. Removing the local greedy inference phase as in "PPN-w/o-LGI" decreases the performance to 77.8% AP, showing local greedy inference is beneficial to pose estimation by effectively handling false alarms of joint candidate detection based on global affinity cues in the embedding space. Comparison of "PPN-w/o-Refinement"(76.4% AP) with the full model demonstrates that single-person pose estimation can refine joint localization. "PPN-Vanilla" achieves 74.8% AP, verifying the stableness of our approach for multi-person pose estimation even in the case of removing refinement and multi-scale testing.



(a)

	1	2	3	4	5	6	7	8	9	10	
1	0 0.0%	7 2.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0.0% 100%
2	0 0.0%	218 62.3%	6 1.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	97.3% 2.7%
3	0 0.0%	11 3.1%	53 15.1%	3 0.9%	1 0.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	77.9% 22.1%
4	0 0.0%	3 0.9%	4 1.1%	19 5.4%	1 0.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	70.4% 29.6%
5	0 0.0%	0 0.0%	0 0.0%	1 0.3%	7 2.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	87.5% 12.5%
6	0 0.0%	0 0.0%	0 0.0%	1 0.3%	2 0.6%	6 1.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	66.7% 33.3%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.6%	1 0.3%	0 0.0%	0 0.0%	0 0.0%	33.3% 66.7%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.3%	0 0.0%	0 0.0%	1 0.3%	0 0.0%	0 0.0%	50.0% 50.0%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.3%	0 0.0%	0 0.0%	0.0% 100%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.3%	0 0.0%	0 0.0%	0.0% 100%
	0.0% 0.0%	91.2% 8.8%	84.1% 15.9%	79.2% 20.8%	58.3% 41.7%	75.0% 25.0%	100% 0.0%	33.3% 66.7%	0.0% 0.0%	0.0% 0.0%	87.1% 12.9%

(b)

Fig. 4. (a) Ablation study on multi-stage Hourglass network. (b) Confusion matrix on person number inferred from pose partition (Sect. 3.3) with groundtruth. Mean square error is 0.203. Best viewed in color and 2× zoom

We also evaluate the pose estimation results from 4 different stages of the PPN model and plot the results in Fig. 4(a). The performance increases monotonically when traversing more stages. The final results achieved at the 8th stage give about 23.4% improvement comparing with the first stage (79.0% vs 64.0% AP). This is because the proposed PPN can recurrently correct errors on the dense regression maps along with the joint confidence maps conditioned on previous estimations in the multi-stage design, yielding gradual improvement on the joint detections and partitions for multi-person pose estimation.

Finally, we evaluate the effectiveness of pose partition model for partition person instances. In particular, we evaluate how its produced partitions match the real number of persons. The confusion matrix is shown in Fig. 4(b). We can observe the proposed pose partition model can predict very close number of persons with the groundtruth, with mean square error as small as 0.203.

6 Conclusion

We presented the Pose Partition Network (PPN) to efficiently and effectively address the challenging multi-person pose estimation problem. PPN solves the problem by simultaneously detecting and partitioning joints for multiple persons. It introduces a new approach to generate partitions through inferring over joint candidates in the embedding space parameterized by person centroids. Moreover, PPN introduces a local greedy inference approach to estimate poses for person instances by utilizing the partition information. We demonstrate that PPN can provide appealing efficiency for both joint detection and partition, and it can significantly overcome limitations of pure top-down and bottom-up solutions on three benchmarks multi-person pose estimation datasets.

Acknowledgement. Jiashi Feng was partially supported by NUS IDS R-263-000-C67-646, ECRA R-263-000-C87-133 and MOE Tier-II R-263-000-D17-112.

References

1. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2D human pose estimation: New benchmark and state of the art analysis. In: CVPR (2014)
2. Bourdev, L., Malik, J.: Poselets: body part detectors trained using 3D human pose annotations. In: ICCV (2009)
3. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2D pose estimation using part affinity fields. In: CVPR (2017)
4. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.L.: Detect what you can: detecting and representing objects using holistic models and body parts. In: CVPR (2014)
5. Chen, X., Yuille, A.L.: Parsing occluded people by flexible compositions. In: CVPR (2015)
6. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc. B. **39**(1), 1–38 (1977)
7. Eichner, M., Ferrari, V.: We are family: joint pose estimation of multiple persons. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 228–242. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15549-9_17
8. Fang, H., Xie, S., Tai, Y., Lu, C.: RMPE: regional multi-person pose estimation. In: ICCV (2017)
9. Gkioxari, G., Hariharan, B., Girshick, R., Malik, J.: Using k-poselets for detecting people and localizing their keypoints. In: CVPR (2014)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
11. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., Schiele, B.: DeeperCut: a deeper, stronger, and faster multi-person pose estimation model. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 34–50. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_3
12. Insafutdinov, E., Andriluka, M., Pishchulin, L., Tang, S., Andres, B., Schiele, B.: Articulated multi-person tracking in the wild. In: CVPR (2017)
13. Iqbal, U., Gall, J.: Multi-person pose estimation with local joint-to-person associations. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 627–642. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_44
14. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: NIPS (2015)
15. Ladicky, L., Torr, P.H., Zisserman, A.: Human pose estimation using a joint pixel-wise and part-wise formulation. In: CVPR (2013)
16. Levinkov, E., et al.: Joint graph decomposition & node labeling: problem, algorithms, applications. In: CVPR (2017)
17. Liu, W., et al.: SSD: single shot MultiBox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
18. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 483–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_29
19. Newell, A., Deng, J.: Associative embedding: end-to-end learning for joint detection and grouping. In: NIPS (2017)

20. Papandreou, G., et al.: Towards accurate multi-person pose estimation in the wild. In: CVPR (2017)
21. Paszke, A., Gross, S., Chintala, S.: PyTorch (2017)
22. Pishchulin, L., et al.: DeepCut: joint subset partition and labeling for multi person pose estimation. In: CVPR (2016)
23. Pishchulin, L., Jain, A., Andriluka, M., Thormählen, T., Schiele, B.: Articulated people detection and pose estimation: Reshaping the future. In: CVPR (2012)
24. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015)
25. Sun, M., Savarese, S.: Articulated part-based model for joint object detection and pose estimation. In: ICCV (2011)
26. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning (2012)
27. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: CVPR (2016)
28. Xia, F., Wang, P., Chen, X., Yuille, A.L.: Joint multi-person pose estimation and semantic part segmentation. In: CVPR (2017)