

Pose Proposal Networks

Taiki Sekii

Konica Minolta, Inc.

Abstract. We propose a novel method to detect an unknown number of articulated 2D poses in real time. To decouple the runtime complexity of pixel-wise body part detectors from their convolutional neural network (CNN) feature map resolutions, our approach, called *pose proposal networks*, introduces a state-of-the-art single-shot object detection paradigm using grid-wise image feature maps in a bottom-up pose detection scenario. Body part proposals, which are represented as region proposals, and limbs are detected directly via a single-shot CNN. Specialized to such detections, a bottom-up greedy parsing step is probabilistically redesigned to take into account the global context. Experimental results on the MPII Multi-Person benchmark confirm that our method achieves 72.8% mAP comparable to state-of-the-art bottom-up approaches while its total runtime using a GeForce GTX1080Ti card reaches up to 5.6 ms (180 FPS), which exceeds the bottleneck runtimes that are observed in state-of-the-art approaches.

Keywords: Human pose estimation, Object detection

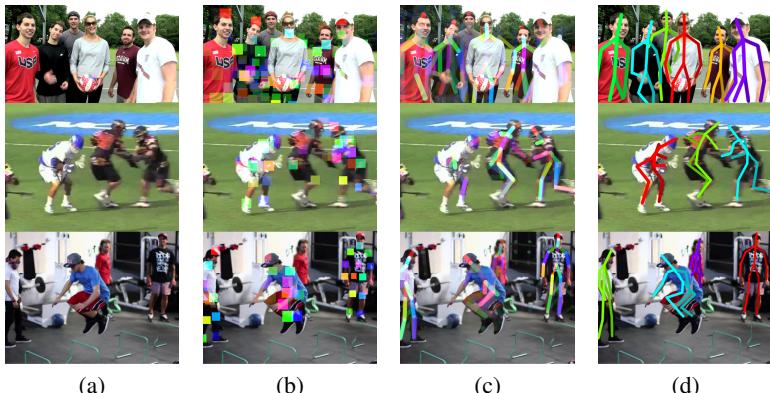


Fig. 1. Sample multi-person pose detection results by the ResNet-18-based *PPN*. Part bounding boxes (b) and limbs (c) are directly detected from input images (a) using single-shot CNNs and are parsed into individual people (d) (*cf.* § 3).

1 Introduction

The problem of detecting humans and simultaneously estimating their articulated poses (which we refer to as *poses*) as shown in Fig. 1 has become an important and highly practical task in computer vision thanks to recent advances in deep learning. While this task has broad applications in fields such as sports analysis and human-computer interaction, its test-time computational cost can still be a bottleneck in real-time systems. Human pose estimation is defined as the localization of anatomical keypoints or landmarks (which we refer to as *parts*) and is tackled using various methods, depending on the final goals and the assumptions made:

- The use of single or sequential images as input;
- The use (or not) of depth information as input;
- The localization of parts in a 2D or 3D space; and
- The estimation of single- or multi-person poses.

This paper focuses on multi-person 2D pose estimation from a 2D still image. In particular, we do not assume that the ground truth location and scale of the person instances are provided and, therefore, need to detect an unknown number of poses, *i.e.*, we need to achieve *human pose detection*. In this more challenging setting, referred to as “in the wild,” we pursue an end-to-end, detection framework that can perform in real-time.

Previous approaches [1–13] can be divided into the following two types: one detects person instances first and then applies single-person pose estimators to each detection and the other detects parts first and then parses them into each person instance. These are called as *top-down* and *bottom-up* approaches, respectively. Such state-of-the-art methods show competitive results in both runtime and accuracy. However, the runtime of top-down approaches is proportional to the number of people, making real-time performance a challenge, while bottom-up approaches require bottleneck parts association procedures that extract contextual cues between parts and parse part detections into individual people. In addition, most state-of-the-art techniques are designed to predict pixel-wise¹ part confidence maps in the image. These maps force convolutional neural networks (CNNs) to extract feature maps with higher resolutions, which are indispensable for maintaining robustness, and the acceleration of the architectures (*e.g.*, shrinking the architectures) is interfered depending on the applications.

In this paper, to decouple the runtime complexity of the human pose detection from the feature map resolution of the CNNs and improve the performance, we rely on a state-of-the-art single-shot object detection paradigm that roughly extracts grid-wise object confidence maps in the image using relatively smaller CNNs. We benefit from region proposal (RP) frameworks² [14–17] and reframe the human pose detection as an object detection problem, regressing from image pixels to RPs of person instances and parts, as shown in Fig. 2. In addition, instead of the previous parts association designed for pixel-wise part proposals, our framework directly detects limbs³ using single-shot

¹ We also use the term “pixel-wise” to refer to the downsampled part confidence maps.

² We use the term “RP frameworks” to refer broadly to CNN-based methods that predict a fixed set of bounding boxes depending on the input image sizes.

³ We refer to part pairs as limbs for clarity, despite the fact that some pairs are not human limbs (*e.g.*, faces).

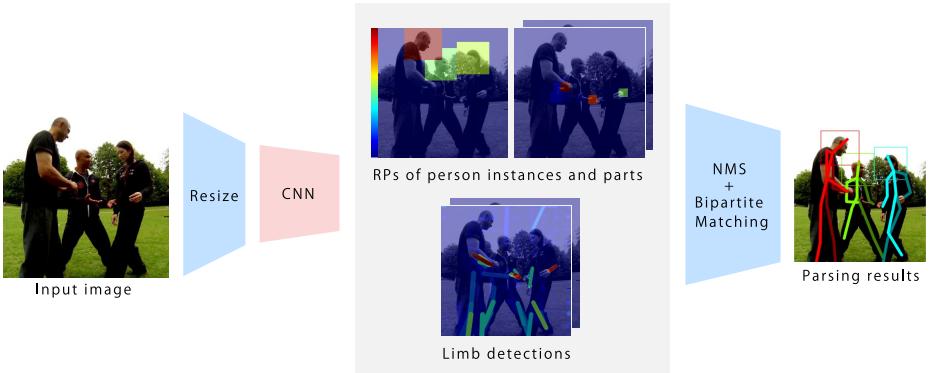


Fig. 2. Pipeline of our proposed approach. Pose proposals are generated by parsing RPs of person instances and parts into individual people (*cf.* § 3).

CNNs and generates pose proposals from such detections via a novel, probabilistic greedy parsing step in which the global context is taken into account. Part RPs are defined as bounding box detections whose sizes are proportional to the person scales and can be supervised using just the common keypoint annotations. The entire architecture is constructed from a single, fully CNN with relatively lower-resolution feature maps and is optimized end-to-end directly using a loss function designed for pose detection performance; we call this architecture the *pose proposal network* (PPN).

2 Related work

We will briefly review some of the recent progress in single- and multi-person pose estimations to put our contributions into context.

Single-person pose estimation. The majority of early classic approaches for single-person pose estimation [18–23] assumed that the person dominates the image content and that all limbs are visible. These approaches primarily pursued the modeling of structures together with the articulation of single-person body parts and their appearances in the image under various concepts such as pictorial structure models [18, 19], hierarchical models [22], and non-tree models [20, 21, 23]. Since the appearance of deep learning-based models [24–26] that make the problem tractable, the benchmark results have been successively updated by various base architectures, such as convolutional pose machines (CPMs) [27], residual networks (ResNets) [28, 11], and stacked hour-glass networks (SHNs) [29]. These models focus on strong part detectors that take into account the large, detailed spatial context and are used as fundamental part detectors in both state-of-the-art single- [30–33] and multi-person contexts [1, 2, 9].

Multi-person pose estimation. The performance of top-down approaches [2–4, 7, 10, 12] depends on human detectors and pose estimators; therefore, it has improved according to the performance of these detectors and estimators. More recently, to achieve efficiency and higher robustness, recent methods have tended to share convolutional

layers between the human detectors and pose estimators by introducing spatial transformer networks [2, 34] or RoIAlign [4].

Conversely, standard bottom-up approaches [1, 6, 8, 9, 11, 13] rely less on human detectors and instead detect poses by finding groups or pairs of part detections, which occur in consistent geometric configurations. Therefore, they are not affected by the limitations of human detectors. Recent bottom-up approaches use CNNs not only to detect parts but also to directly extract contextual cues between parts from the image, such as image-conditioned pairwise terms [6], part affinity fields (PAFs) [1], and associative embedding (AE) [9].

The state-of-the-art methods in both top-down and bottom-up approaches achieve real-time performance. Their “primitives” of the part proposals are pixel points. However, our method differs from such approaches in that our primitives are grid-wise bounding box detections in which the part scale information is encoded. Our reduced grid-wise part proposals allow shallow CNNs to directly detect limbs which can be represented with at most a few dozen patterns for each part proposal. Specialized for these detections, a greedy parsing step is probabilistically redesigned to encode the global context. Therefore, our method does not need time-consuming, pixel-wise feature extraction or parsing steps, and its total runtime, as a result, exceeds the bottleneck runtimes that are observed in state-of-the-art approaches.

3 Method

Human pose detection is achieved via the following steps.

1. Resize an input image to the input size of the CNN.
2. Run forward propagation of the CNN and obtain RPs of person instances and parts and limb detections.
3. Perform non-maximum suppression (NMS) for these RPs.
4. Parse the merged RPs into individual people and generate pose proposals.

Fig. 2 depicts the pipeline of our framework. §3.1 describes RP detections of person instances and parts and limb detections, which are used in steps 2 and 3. §3.2 describes step 4.

3.1 PPNs

We take advantage of YOLO [15, 16], one of the RP frameworks, and apply its concept to the human pose detection task. The PPNs are constructed from a single CNN and produce a fixed-size collection of RPs for each detection target (person instances or each part) over the input image. The CNN divides the input image into a $H \times W$ grid, each cell of which corresponds to an image block, and produces a set of RP detections $\{\mathcal{B}_k^i\}_{k \in \mathcal{K}}$ for each grid cell $i \in \mathcal{G} = \{1, \dots, H \times W\}$. Here, $\mathcal{K} = \{0, 1, \dots, K\}$ is the set of indices of the detection targets, and K is the number of parts. The index of the class representing the overall person instances (the person instance class) is given by $k = 0$ in \mathcal{K} .

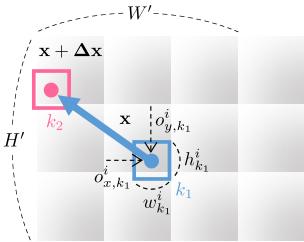


Fig.3. RP and limb detections by the PPN. The blue arrow indicates a limb (a directed connection) whose confidence score is encoded by $p(C|k_1, k_2, \mathbf{x}, \mathbf{x} + \Delta\mathbf{x})$.

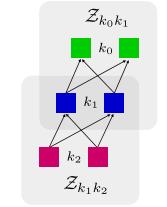
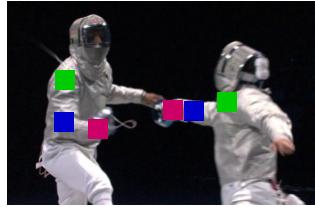


Fig.4. Parts association defined as bipartite matching sub-problems. Matchings are decomposed and solved for every pair of detection targets that constitute limbs (*e.g.*, they are separately computed for (k_0, k_1) and for (k_1, k_2)).

\mathcal{B}_k^i encodes the two probabilities taking into consideration the confidence of the bounding box and the coordinates, width, and height of the bounding box, as shown in Fig. 3, and is given by

$$\mathcal{B}_k^i = \{p(R|k, i), p(I|R, k, i), o_{x,k}^i, o_{y,k}^i, w_k^i, h_k^i\}, \quad (1)$$

where R and I are binary random variables. Here, $p(R|k, i)$ is a probability that represents the grid cell i “responsible” for detections of k . If the center of a ground truth bounding box of k falls into a grid cell, that grid cell is “responsible” for detections of k . $p(I|R, k, i)$ is a conditional probability that represents how well the bounding box predicted in i fits k and is supervised by the intersection over union (IoU) between the predicted bounding box and the ground truth bounding box.

The $(o_{x,k}^i, o_{y,k}^i)$ coordinates represent the center of the bounding box relative to the bounds of the grid cell with the scale normalized by the length of the cells. w_k^i and h_k^i are normalized by the image width and height, respectively. The bounding boxes of person instances can be represented as rectangles around the entire body or the head. Unlike previous pixel-wise part detectors, parts are grid-wise detected in our method and the box sizes are supervised proportional to the person scales, *e.g.*, one-fifth of the length of the upper body or half the head segment length. The ground truth boxes supervise these predictions regarding the bounding boxes.

Conversely, for each grid cell i located at \mathbf{x} , the CNN also produces a set of limb detections, $\{\mathcal{C}_{k_1 k_2}\}_{(k_1, k_2) \in \mathcal{L}}$, where \mathcal{L} is a set of pairs of indices of detection targets that constitute limbs. $\mathcal{C}_{k_1 k_2}$ encodes a set of probabilities that represents the presence of each limb and is given by

$$\mathcal{C}_{k_1 k_2} = \{p(C|k_1, k_2, \mathbf{x}, \mathbf{x} + \Delta\mathbf{x})\}_{\Delta\mathbf{x} \in \mathcal{X}}, \quad (2)$$

where C is a binary random variable. $p(C|k_1, k_2, \mathbf{x}, \mathbf{x} + \Delta\mathbf{x})$ encodes the presence of a limb represented as a directed connection from the bounding box of k_1 predicted in \mathbf{x} to that of k_2 predicted in $\mathbf{x} + \Delta\mathbf{x}$, as shown in Fig. 3. Here, we assume that all the

limbs from \mathbf{x} reach only the local $H' \times W'$ area centered on \mathbf{x} and define \mathcal{X} as a set of finite displacements from \mathbf{x} , which is given by

$$\mathcal{X} = \{\Delta\mathbf{x} = (\Delta x, \Delta y) \mid |\Delta x| \leq W' \wedge |\Delta y| \leq H'\}. \quad (3)$$

Here, $\Delta\mathbf{x}$ is a position relative to \mathbf{x} and, therefore, $p(C|k_1, k_2, \mathbf{x}, \mathbf{x} + \Delta\mathbf{x})$ can be independently estimated at each grid cell using CNNs thanks to their characteristic of translation invariance.

Each of the above mentioned predictions corresponds to each channel in the depth of the output 3D tensor produced by the CNN. Finally, the CNN outputs an $H \times W \times \{6(K+1) + H'W'|\mathcal{L}|\}$ tensor. During training, we optimize the following, multi-part loss function:

$$\begin{aligned} & \lambda_{\text{resp.}} \sum_{i \in \mathcal{G}} \sum_{k \in \mathcal{K}} \{\delta_k^i - \hat{p}(R|k, i)\}^2 \\ & + \lambda_{\text{IoU}} \sum_{i \in \mathcal{G}} \sum_{k \in \mathcal{K}} \delta_k^i \{(p(I|R, k, i) - \hat{p}(I|R, k, i))\}^2 \\ & + \lambda_{\text{coor.}} \sum_{i \in \mathcal{G}} \sum_{k \in \mathcal{K}} \delta_k^i \{(o_{x,k}^i - \hat{o}_{x,k}^i)^2 + (o_{y,k}^i - \hat{o}_{y,k}^i)^2\} \\ & + \lambda_{\text{size}} \sum_{i \in \mathcal{G}} \sum_{k \in \mathcal{K}} \delta_k^i \left\{ \left(\sqrt{w_k^i} - \sqrt{\hat{w}_k^i} \right)^2 + \left(\sqrt{h_k^i} - \sqrt{\hat{h}_k^i} \right)^2 \right\} \\ & + \lambda_{\text{limb}} \sum_{i \in \mathcal{G}} \sum_{\Delta\mathbf{x} \in \mathcal{X}} \sum_{(k_1, k_2) \in \mathcal{L}} \max(\delta_{k_1}^i, \delta_{k_2}^j) \left\{ \delta_{k_1}^i \delta_{k_2}^j - \hat{p}(C|k_1, k_2, \mathbf{x}, \mathbf{x} + \Delta\mathbf{x}) \right\}^2, \end{aligned} \quad (4)$$

where $\delta_k^i \in \{1, 0\}$ is a variable that indicates if i is responsible for the k of only a single person, j is the index of a grid cell located at $\mathbf{x} + \Delta\mathbf{x}$, and $(\lambda_{\text{resp.}}, \lambda_{\text{IoU}}, \lambda_{\text{coor.}}, \lambda_{\text{size}}, \lambda_{\text{limb}})$ are the weights for each loss.

3.2 Pose proposal generation

Overview. Applying standard NMS using an IoU threshold for the RPs of each detection target, we can obtain the fixed-size, merged RP subsets. Then, in the condition where both true and false positives of multiple people are contained in these RPs, pose proposals are generated by matching and associating the RPs between the detection targets that constitute limbs. This parsing step corresponds to a K -dimensional matching problem that is known to be NP hard [35], and many relaxations exist.

In this paper, inspired by [1], we introduce two relaxations capable of real-time generation of consistent matches. First, a minimal number of edges are chosen to obtain a spanning tree skeleton of articulated poses, whose nodes and edges represent the merged RP subsets of the detection targets and the limb detections between them, respectively, rather than using the complete graph. This tree consists of directed edges and, its root nodes belong to the person instance class. Second, the matching problem is further decomposed into a set of bipartite matching sub-problems, and the matching in adjacent tree nodes is determined independently, as shown in Fig. 4. Cao *et al.* [1]

demonstrated that such a minimal greedy inference well approximates the global solution at a fraction of the computational cost and concluded that the relationship between nonadjacent tree nodes can be implicitly modeled in their pairwise part association scores, which the CNN estimates. In contrast to their approach, in order to use relatively shallow CNNs whose receptive fields are narrow and reduce the computational cost, we propose a probabilistic, greedy parsing algorithm that takes into account the relationship between nonadjacent tree nodes.

Confidence scores. Given the merged RPs of the detection targets, we define a confidence score for the detection of the n -th RP of k as follows:

$$D_k^n = p(R|k, n)p(I|R, k, n). \quad (5)$$

Each probability on the right-hand side of Eq. (5) is encoded by \mathcal{B}_k^i in Eq. (1). $n \in \mathcal{N} = \{1, \dots, N\}$, where N is the number of merged RPs of each detection target. In addition, the confidence score of the limb, *i.e.*, the directed connection from the n_1 -th RP of k_1 predicted at \mathbf{x} to the n_2 -th RP of k_2 predicted at $\mathbf{x} + \Delta\mathbf{x}$, is defined by making use of Eq. (2) as follows:

$$E_{k_1 k_2}^{n_1 n_2} = p(C|k_1, k_2, \mathbf{x}, \mathbf{x} + \Delta\mathbf{x}). \quad (6)$$

Parts association. Parts association, which uses pairwise part association scores, can be generally defined as an optimal assignment problem for the set of all the possible connections,

$$\mathcal{Z} = \left\{ Z_{k_1 k_2}^{n_1 n_2} | (k_1, k_2) \in \mathcal{L}, n_1 \in \mathcal{N}_1, n_2 \in \mathcal{N}_2 \right\}, \quad (7)$$

which maximizes the confidence score that approximates the joint probability over all possible limb detections,

$$F = \prod_{\mathcal{L}} \prod_{\mathcal{N}_1} \prod_{\mathcal{N}_2} (E_{k_1 k_2}^{n_1 n_2})^{Z_{k_1 k_2}^{n_1 n_2}}. \quad (8)$$

Here, $Z_{k_1 k_2}^{n_1 n_2}$ is a binary variable that indicates whether the n_1 -th RP of k_1 and the n_2 -th RP of k_2 are connected and satisfies

$$\sum_{\mathcal{N}_1} Z_{k_1 k_2}^{n_1 n_2} = 1 \wedge \sum_{\mathcal{N}_2} Z_{k_1 k_2}^{n_1 n_2} = 1, \quad (9)$$

$$\forall n_1 \in \mathcal{N}_1, \forall n_2 \in \mathcal{N}_2.$$

Using Eq. (9) ensures that no multiple edges share a node, *i.e.*, that an RP is not connected to different multiple RPs. In this graph-matching problem, the nodes of the graph are all the merged RPs of the detection targets, the edges are all the possible connections between the RPs, which constitute the limbs, and the confidence scores of the limb detections give the weights for the edges. Our goal is to find a matching in the bipartite graph as a subset of the edges chosen with maximum weight.

In our improved parts association with the abovementioned two relaxations, person instances are used as a root part, and the proposals of each part are assigned to person

instance proposals along the route on the pose graph. Bipartite matching sub-problems are defined for each respective pair (k_1, k_2) of detection targets that constitute the limbs so as to find the optimal assignment for the set of connections between k_1 and k_2 ,

$$\mathcal{Z}_{k_1 k_2} = \left\{ Z_{k_1 k_2}^{n_1 n_2} \mid n_1 \in \mathcal{N}_1, n_2 \in \mathcal{N}_2 \right\}, \quad (10)$$

where

$$\{\mathcal{Z}_{k_1 k_2}\}_{(k_1, k_2) \in \mathcal{L}} = \mathcal{Z}. \quad (11)$$

We obtain the optimal assignment $\hat{\mathcal{Z}}_{k_1 k_2}$ as follows:

$$\hat{\mathcal{Z}}_{k_1 k_2} = \arg \max_{\mathcal{Z}_{k_1 k_2}} F_{k_1 k_2}, \quad (12)$$

where

$$F_{k_1 k_2} = \prod_{\mathcal{N}_1} \prod_{\mathcal{N}_2} (S_{k_1 k_2}^{n_1 n_2})^{Z_{k_1 k_2}^{n_1 n_2}}. \quad (13)$$

Here, the nodes of k_1 are closer to those of the person instances on the route of the graph than those of k_2 and

$$S_{k_1 k_2}^{n_1 n_2} = \begin{cases} D_{k_1}^{n_1} E_{k_2 k_1}^{n_2 n_1} D_{k_2}^{n_2} & \text{if } k_1 = 0, \\ S_{k_0 k_1}^{\hat{n}_0 n_1} E_{k_2 k_1}^{n_2 n_1} D_{k_2}^{n_2} & \text{otherwise.} \end{cases} \quad (14)$$

$k_0 \neq k_2$ indicates that another detection target is connected to k_1 . \hat{n}_0 is the index of the RPs of k_0 , which is connected to the n_1 -th RP of k_1 and satisfies

$$Z_{k_0 k_1}^{\hat{n}_0 n_1} = 1. \quad (15)$$

This optimization using Eq. (14) needs to be calculated from the parts connected to the person instances. We can use the Hungarian algorithm [36] to obtain the optimal matching. Finally, with all the optimal assignments, we can assemble the connections that share the same RPs into full-body poses of multiple people.

The difference between F in Eq. (8) and $F_{k_1 k_2}$ in Eq. (13) is that the confidence scores for the RPs and the limb detections on the route from the nodes of the person instances on the graph are considered in the matching using Eq. (12). This leads to a global context for wider image regions than the receptive fields of the CNN is taken into account in the parsing. In §4, we show detailed comparison results, demonstrating that our improved parsing approximates the global solution well when using shallow CNNs.

4 Experiments

4.1 Dataset

We evaluated our approach on the challenging, public “MPII Human Pose” dataset [37], which includes approximately 25K images containing over 40K annotated people (three-quarters of which are available for training). For a fair comparison, we followed the

official evaluation protocol and used the publicly available evaluation scripts⁴ for self-comparison on the validation set used in [1].

First, the “Single-Person” subset, containing only sufficiently separated people, was used to evaluate the pure performance of the proposed part RP representation. This subset contains a set of 6908 people, and the approximate locations and scales of each person are available. For the evaluation on this subset, we used the standard “Percentage of Correct Keypoints” evaluation metric (PCKh) whose matching threshold is defined as half the head segment length.

Second, to evaluate the full performance of the PPN for human pose detection in the wild, we used the “Multi-Person” subset, which contains a set of 1758 groups of multiple overlapping people in highly articulated poses with a variable number of parts. These groups are taken from the test set as outlined in [11]. In this subset, even though the regions that each group occupies and the mean scales of all the people in each group are available, no information is provided concerning the number of people or the scales of the individual people. For the evaluation on this subset, we used the evaluation metric outlined by Pishchulin *et al.* [11], calculating the average precision (AP) of the part detections.

4.2 Implementation

Setting of the RPs. As shown in Fig. 2, the RPs of person instances and those of parts are defined as square detections centered on the head and on each part, respectively. These lengths are defined as twice the head segment length for person instances and as half the head segment length for parts. Therefore, all ground truth boxes can be computed from two given head keypoints. For limb detections, the two head keypoints are defined as being connected to person instances and the other connections are defined similar to those in [1]. Therefore, $|\mathcal{L}|$ is set to 15.

Architecture. As the base architecture, we use an 18-layer standard ResNet pre-trained on the ImageNet 1000-class competition dataset [38]. The average-pooling layer and the fully connected layer in this architecture are replaced with three additional new convolutional layers. In this setting, the output grid cell size of the CNN on the image, which is described in § 3.1, corresponds to $32 \times 32 \text{ px}^2$ and $(H, W) = (12, 12)$ for the normalized 384×384 input size of the CNN used in the training. This grid cell size on the image is fairly large compared to those of previous pixel-wise part detectors (usually $4 \times 4 \text{ px}^2$ or 8×8).

The last added convolutional layer uses a linear activation function and the other added layers use the following leaky rectified linear activation:

$$\phi(u) = \begin{cases} u & \text{if } u > 0, \\ 0.1u & \text{otherwise.} \end{cases} \quad (16)$$

All the added layers use a 1-px stride, and the weights are all randomly initialized. The first layer in the added layers uses batch normalization. The filter sizes and the number of filters of the added layers other than the last layer are set to 3×3 and 512,

⁴ <http://human-pose.mpi-inf.mpg.de>

respectively. In the last layer, the filter size is set to 1×1 , and, as described in § 3.1, the number of filters is set to $6(K + 1) + H'W'|\mathcal{L}| = 1311$, where (H', W') is set to $(9, 9)$. K is set to 15, which is similar to the values used in [1].

Training. During training, in order to have normalized 384×384 input samples, we first resized the images to make the samples roughly the same scale (*w.r.t.* 200 px person height) and cropped or padded the image according to the center positions and the rough scale estimations provided in the dataset. Then, we randomly augmented the data with rotation degrees in $[-40, 40]$, an offset perturbation, and horizontal flipping in addition to scaling with factors in $[0.35, 2.5]$ for the multi-person task and in $[1.0, 2.0]$ for the single-person task.

$(\lambda_{\text{resp}}, \lambda_{\text{IoU}}, \lambda_{\text{coor}}, \lambda_{\text{size}})$ in Eq. (4) are set to $(0.25, 1, 5, 5)$, and λ_{limb} is set to 0.5 in the multi-person task and to 0 in the single-person task. The entire network is trained using SGD for 260K iterations in the multi-person task and for 130K iterations in the single-person task with a batch size of 22, a momentum of 0.9, and a weight decay of 0.0005 on two GPUs. 260K iterations on two GPUs roughly correspond to 422 epochs of the training set. The learning rate l is linearly decreased depending on the number of iterations, m , calculated as follows:

$$l = 0.007(1 - m/260,000). \quad (17)$$

Training takes approximately 1.8 days using a machine with two GeForce GTX1080Ti cards, a 3.4 GHz Intel CPU, and 64 GB RAM.

Testing. During the testing of our method, the images were resized such that the mean scales for the target people corresponded to 1.43 in the multi-person task and to 1.3 in the single-person task. Then, they were cropped around the target people. The accuracies of previous approaches are taken from the original papers or are reproduced using their publicly available evaluation codes. During the timings of all the approaches including the baselines, the images were resized with each of the mean resolutions used when they were evaluated. The timings are reported using the same single GPU card and deep learning framework (Caffe [39]) on the machine described above averaged over the batch sizes with which each method performs the fastest. Our detection steps other than forward propagation by CNNs are run on the CPU.

4.3 Human part detection

We compare part detections by the PPN with several, pixel-wise part detectors used by state-of-the-art methods in both single-person and multi-person contexts. Predictions with pixel-wise detectors and those of the PPN are the maximum activating locations of the heatmap for a given part and the locations of the maximum activating RPs of each part, respectively.

Tables 1 and 2 compare the PCKh performance and the speeds of the PPN and other detectors on the single-person test set and lists the properties of the networks used in each approach. Note that [6] proposes part detectors while dealing with multi-person pose estimation. They use the same ResNet-based architecture as the PPN, which is several times deeper (152 layers) than ours and is different from ours only in that the network is massive to produce pixel-wise part proposals. We found that the speed

and FLOP count (multiply-adds) of our detector overwhelm all others and are at least 11 times faster even when considering its slightly (several percent) lower PCKh. In particular, the fact that the PPN achieves a comparable PCKh to that of the ResNet-based part detector [6] using the same architecture as ours demonstrates that the part RPs effectively work as the part primitives when exploring the speed/accuracy trade-off.

4.4 Human pose detection

Tables 3 and 4 compare the mean AP (mAP) performance between the full implementation of the PPN and previous approaches on the same subset of 288 testing images, as in [11], and on the entire multi-person test set. An illustration of the predictions made by our method can be seen in Fig. 7. Note that [1] is trained using unofficial masks of unlabeled persons (reported as *w/ or w/o masks* in Fig. 6) and ranks with a favorable margin of a few percent mAP according to the original paper and that our method can be adjusted by replacing the base architecture with the 50- and 101-layer ResNets. Despite rough part detections, the deepest mode of our method (reported as *w/ ResNet-101*) achieves the top performance for upper body parts. The total runtime of this fast PPN reaches up to 5.6 ms (180 FPS) that exceeds the state-of-the-art bottleneck runtime described below. The runtime of the forward propagation with the CNN and the parsing step are 4 ms and 0.3 ms, respectively. The remaining runtime (1.3 ms) is mostly consumed by part proposal NMS.

Fig. 5 is a scatterplot that visualizes the mAP performances and speeds of our method and the top-3 approaches reported using their publicly available implementation or from the original papers. The colored dot lines, each of which corresponds to one of previous approaches, denote limits of speed in total processing as speed for processing other than the forward propagation of CNNs such as resizing of CNN feature maps [1, 2, 9], grouping of parts [1, 9], and NMS in human detection [2] or for part proposals [1, 9] (The colors represent each method). Such bottleneck steps were optimized or were accelerated by GPUs to a certain extent. Improving the base architectures without the loss of accuracy will not help each state-of-the-art approach exceed their speed

Table 1. Pose estimation results on the MPII Single-Person test set.

Method	Architecture	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	PCKh
Ours	ResNet-18	97.9	95.3	89.1	83.5	87.9	82.7	76.2	88.1
SHN [29]	Custom	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9
DeeperCut [6]	ResNet-152	96.8	95.2	89.3	84.4	88.4	83.4	78.0	88.5
CPM [27]	Custom	97.7	94.5	88.3	83.4	87.9	81.9	78.3	87.9

Table 2. The properties of the networks on the MPII Single-Person test set.

Method	PCKh	Architecture	Input size	Output size	FLOPs	Num. param.	FPS
Ours	88.1	ResNet-18	384 × 384	12 × 12	6G	16M	388
SHN [29]	90.9	Custom	256 × 256	64 × 64	30G	34M	19
DeeperCut [6]	88.5	ResNet-152	344 × 344	43 × 43	37G	66M	34
CPM [27]	87.9	Custom	368 × 368	46 × 46	175G	31M	9

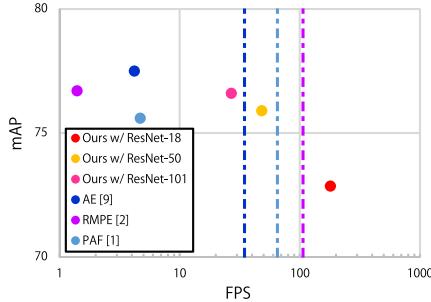


Fig. 5. Accuracy versus speed on the MPII Multi-Person test set. See text for details.

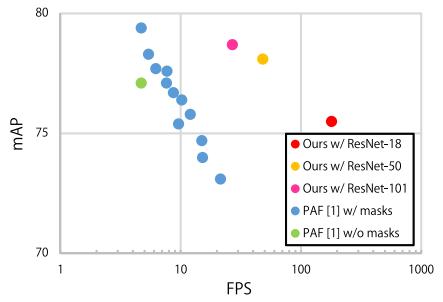


Fig. 6. Accuracy versus speed on the MPII Multi-Person validation set.

limits without leaving redundant pixel-wise or top-down strategies. It is also clear that all CNN-based methods significantly degrade when their accelerated speed reaches the speed limits. Our method is more than an order of magnitude faster compared with the state-of-the-art methods on average and can pass through the abovementioned bottleneck speed limits.

In addition, to compare our method with state-of-the-art methods in more detail, we reproduced the state-of-the-art bottom-up approach [1] based on its publicly available evaluation code and accelerated it by adjusting the number of multi-stage convolutions

Table 3. Pose estimation results of a subset of 288 images on the MPII Multi-Person test set.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	U.Body	L.Body	mAP
Ours w/ ResNet-18	94.0	91.6	80.7	68.1	75.0	65.5	61.3	83.6	67.8	76.6
Ours w/ ResNet-50	95.6	92.5	82.4	73.6	76.2	71.1	64.1	86.0	71.5	79.4
Ours w/ ResNet-101	95.2	92.2	83.2	73.8	74.8	71.3	63.4	86.1	71.3	79.1
ArtTrack [5]	92.2	91.3	80.8	71.4	79.1	72.6	67.8	83.9	73.2	79.3
PAF [1]	92.9	91.3	82.3	72.6	76.0	70.9	66.8	84.8	72.2	79.0
RMPE [2]	89.4	88.5	81.0	75.4	73.7	75.4	66.5	83.6	73.5	78.6
DeeperCut [6]	92.1	88.5	76.4	67.8	73.6	68.7	62.3	81.2	68.9	75.6
AE [9]	91.5	87.2	75.9	65.4	72.2	67.0	62.1	80.0	67.9	74.5

Table 4. Pose estimation results on the entire MPII Multi-Person test set.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	U.Body	L.Body	mAP
Ours w/ ResNet-18	93.2	89.0	74.9	62.4	72.2	62.6	55.4	79.9	63.6	72.8
Ours w/ ResNet-50	93.7	90.1	78.0	68.0	74.9	67.2	59.3	82.5	67.5	75.9
Ours w/ ResNet-101	93.9	90.2	79.0	68.7	74.8	68.7	60.5	83.0	68.6	76.6
AE [9]	92.1	89.3	78.9	69.8	76.2	71.6	64.7	82.5	71.3	77.5
RMPE [2]	88.4	86.5	78.6	70.4	74.4	73.0	65.8	81.0	71.8	76.7
PAF [1]	91.2	87.6	77.7	66.8	75.4	68.9	61.7	80.8	68.7	75.6
ArtTrack [5]	88.8	87.0	75.9	64.9	74.2	68.8	60.5	79.2	67.9	74.3
KLj*r [8]	89.8	85.2	71.8	59.6	71.1	63.0	53.5	76.6	62.4	70.6
DeeperCut [6]	89.4	84.5	70.4	59.3	68.9	62.7	54.6	75.9	62.4	70.0

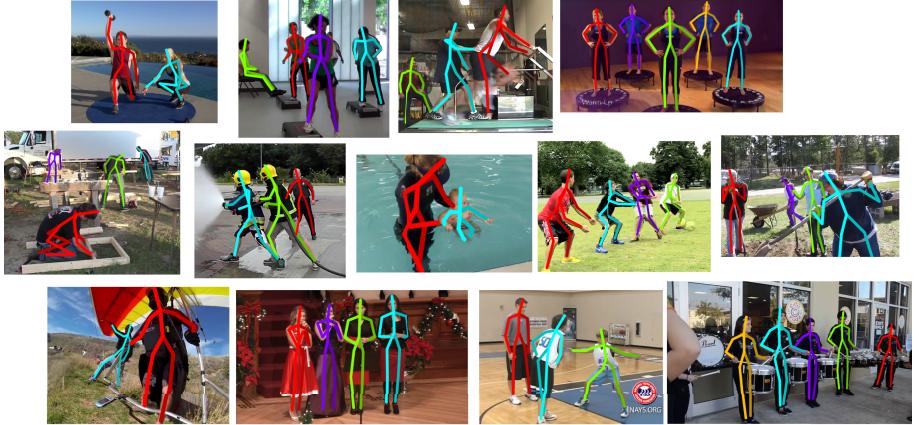


Fig. 7. Qualitative pose estimation results by the ResNet-18-based PPN on MPII test images.

and scale search. Fig. 6 is a scatterplot that visualizes the mAP performance and speeds of both our method and the method proposed in [1], which is adjusted with several patterns. In general, we observe that our method achieves faster and more accurate predictions on an average. The above comparisons with previous approaches indicate that our method can minimize the computational cost of the overall algorithm when exploring the speed/accuracy trade-off.

Table 5 lists the mAP performances of several different versions of our method. First, when $p(I|R, k, i)$ in Eq. (1), which is not estimated by the pixel-wise part detectors, is ignored in our approach (*i.e.*, when $p(I|R, k, i)$ is replaced by 1), and when our NMS follows the previous pixel-wise scheme that finds the maxima on part confidence maps (reported as *w/o scale*), the performance deteriorates from that of the full implementation (reported as *Full.*). This indicates that the speed/accuracy trade-off is improved by additional information regarding the part scales obtained from the fact that the part proposals are bounding boxes. Second, when only the local context is taken into account in parts association (reported as *w/o glob.*), *i.e.*, $S_{k_0 k_1}^{\hat{n}_0 n_1}$ is replaced by $D_{k_1}^{n_1}$ in

Table 5. Quantitative comparison for different versions of the proposed method on the MPII Multi-Person validation set.

Method	Architecture	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	mAP
Full.	ResNet-18	92.8	90.7	78.8	66.9	77.0	63.5	58.4	75.5
w/o scale		88.6	88.0	75.6	64.6	74.2	60.8	55.3	72.4
w/o glob.		91.8	90.1	77.7	63.9	76.7	61.5	51.7	73.3
Full.	ResNet-50	93.8	91.9	81.4	71.5	77.6	69.8	60.3	78.1
w/o scale		91.1	89.4	79.4	68.9	75.8	67.1	59.7	75.9
w/o glob.		93.3	92.2	81.4	69.7	77.8	70.2	58.1	77.5
Full.	ResNet-101	93.4	91.2	81.8	72.2	78.8	70.5	62.9	78.7
w/o scale		91.6	90.0	80.1	70.4	78.5	68.8	62.9	77.5
w/o glob.		93.2	91.6	81.8	71.0	79.6	70.5	61.4	78.4

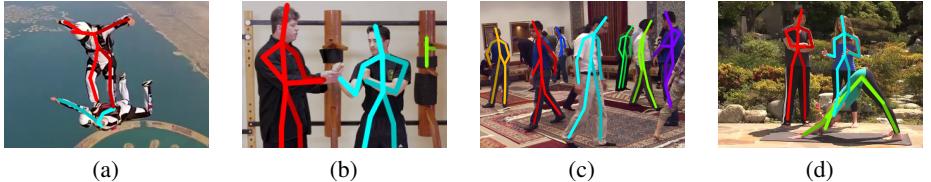


Fig. 8. Common failure cases: (a) rare pose or appearance, (b) false parts detection, (c) missing parts detection in crowded scenes, and (d) wrong connection associating parts from two persons.

Eq. (14), the performance of our shallowest architecture, *i.e.*, ResNet-18, deteriorates further than the deepest one, *i.e.*, ResNet-101 (-2.2% vs -0.3%). This indicates that our context-aware parse works effectively for shallow CNNs.

4.5 Limitations

Our method can predict one RP for each detection target for every grid cell, and therefore this spatial constraint limits the number of nearby people that our model can predict within each grid cell. This causes our method to struggle with groups of people, such as crowded scenes, as shown in Fig. 8(c). Specifically, we observe that our approach will perform poorly on the “COCO” dataset [40] that contains large scale variations such as small people in close proximity. Even though a solution to this problem is to enlarge the input size of the CNN, this in turn causes the speed/accuracy trade-off to degrade, depending on its applications.

5 Conclusions

We proposed a method to detect people and simultaneously estimate their 2D articulated poses from a 2D still image. Our principal innovations to improve speed/accuracy trade-offs are to introduce a state-of-the-art single-shot object detection paradigm to a bottom-up pose detection scenario and to represent part proposals as RPs. In addition, limbs are detected directly with CNNs, and a greedy parsing step is probabilistically redesigned for such detections to encode the global context. Experimental results on the MPII Human Pose dataset confirm that our method has comparable accuracy to state-of-the-art bottom-up approaches and is much faster, while providing an end-to-end training framework⁵. In future studies, to improve the performance for the spatial constraints caused by rough grid-wise predictions, we plan to explore an algorithm to harmonize the high-level and low-level features obtained from state-of-the-art architectures in both part detection and parts association.

⁵ For the supplementary material and videos, please visit: <http://taikisekii.com>

References

1. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2D pose estimation using part affinity fields. In: CVPR. (2017)
2. Fang, H.S., Xie, S., Tai, Y.W., Lu, C.: RMPE: Regional multi-person pose estimation. In: ICCV. (2017)
3. Gkioxari, G., Hariharan, B., Girshick, R., Malik, J.: Using k -poselets for detecting people and localizing their keypoints. In: CVPR. (2014)
4. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV. (2017)
5. Insafutdinov, E., Andriluka, M., Pishchulin, L., Tang, S., Levinkov, E., Andres, B., Schiele, B.: ArtTrack: Articulated multi-person tracking in the wild. In: CVPR. (2017)
6. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., Schiele, B.: DeeperCut: A deeper, stronger, and faster multi-person pose estimation model. In: ECCV. (2016)
7. Iqbal, U., Gall, J.: Multi-person pose estimation with local joint-to-person associations. In: *ECCV Workshops, Crowd Understanding*. (2016)
8. Levinkov, E., Uhrig, J., Tang, S., Omran, M., Insafutdinov, E., Kirillov, A., Rother, C., Brox, T., Schiele, B., Andres, B.: Joint graph decomposition and node labeling: Problem, algorithms, applications. In: CVPR. (2017)
9. Newell, A., Huang, Z., Deng, J.: Associative embedding: End-to-end learning for joint detection and grouping. In: NIPS. (2017)
10. Papandreou, G., Zhu, T., Kanazawa, N., Toshev, A., Tompson, J., Bregler, C., Murphy, K.: Towards accurate multi-person pose estimation in the wild. In: CVPR. (2017)
11. Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P., Schiele, B.: DeepCut: Joint subset partition and labeling for multi person pose estimation. In: CVPR. (2016)
12. Pishchulin, L., Jain, A., Andriluka, M., Thormählen, T., Schiele, B.: Articulated people detection and pose estimation: Reshaping the future. In: CVPR. (2012)
13. Varadarajan, S., Datta, P., Tickoo, O.: A greedy part assignment algorithm for real-time multi-person 2D pose estimation. arXiv preprint arXiv:1708.09182 (2017)
14. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single shot multibox detector. In: ECCV. (2016)
15. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. (2016)
16. Redmon, J., Farhadi, A.: YOLO9000: Better, faster, stronger. In: CVPR. (2017)
17. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. *PAMI* **39**(6) (2017)
18. Andriluka, M., Roth, S., Schiele, B.: Pictorial structures revisited: People detection and articulated pose estimation. In: CVPR. (2009)
19. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. *IJCV* **61**(1) (2005)
20. Lan, X., Huttenlocher, D.P.: Beyond trees: Common-factor models for 2D human pose recovery. In: ICCV. (2005)
21. Sigal, L., Black, M.J.: Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In: CVPR. (2006)
22. Tian, Y., Zitnick, C.L., Narasimhan, S.G.: Exploring the spatial hierarchy of mixture models for human pose estimation. In: ECCV. (2012)
23. Wang, Y., Mori, G.: Multiple tree models for occlusion and spatial constraints in human pose estimation. In: ECCV. (2008)
24. Chen, X., Yuille, A.L.: Articulated pose estimation by a graphical model with image dependent pairwise relations. In: NIPS. (2014)

25. Toshev, A., Szegedy, C.: DeepPose: Human pose estimation via deep neural networks. In: CVPR. (2014)
26. Tompson, J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. In: NIPS. (2014)
27. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: CVPR. (2016)
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)
29. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: ECCV. (2016)
30. Bulat, A., Tzimiropoulos, G.: Human pose estimation via convolutional part heatmap regression. In: ECCV. (2016)
31. Chen, Y., Shen, C., Wei, X.S., Liu, L., Yang, J.: Adversarial PoseNet: A structure-aware convolutional network for human pose estimation. In: ICCV. (2017)
32. Chu, X., Yang, W., Ouyang, W., Ma, C., Yuille, A.L., Wang, X.: Multi-context attention for human pose estimation. In: CVPR. (2017)
33. Yang, W., Li, S., Ouyang, W., Li, H., Wang, X.: Learning feature pyramids for human pose estimation. In: ICCV. (2017)
34. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: NIPS. (2015)
35. West, D.B.: Introduction to Graph Theory. Featured Titles for Graph Theory Series. Prentice Hall (2001)
36. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistic Quarterly (1955)
37. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2D human pose estimation: New benchmark and state of the art analysis. In: CVPR. (2014)
38. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. IJCV **115**(3) (2015)
39. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: MM, ACM. (2014)
40. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft COCO: Common objects in context. arXiv preprint arXiv:1405.0312 (2014)