Does Learning Specific Features for Related Parts Help Human Pose Estimation?

Wei Tang and Ying Wu
Northwestern University
2145 Sheridan Road, Evanston, IL 60208
{wtt450, yingwu}@eecs.northwestern.edu

Abstract

Human pose estimation (HPE) is inherently a homogeneous multi-task learning problem, with the localization of each body part as a different task. Recent HPE approaches universally learn a shared representation for all parts, from which their locations are linearly regressed. However, our statistical analysis indicates not all parts are related to each other. As a result, such a sharing mechanism can lead to negative transfer and deteriorate the performance. This potential issue drives us to raise an interesting question. Can we identify related parts and learn specific features for them to improve pose estimation? Since unrelated tasks no longer share a high-level representation, we expect to avoid the adverse effect of negative transfer. In addition, more explicit structural knowledge, e.g., ankles and knees are highly related, is incorporated into the model, which helps resolve ambiguities in HPE. To answer this question, we first propose a data-driven approach to group related parts based on how much information they share. Then a part-based branching network (PBN) is introduced to learn representations specific to each part group. We further present a multi-stage version of this network to repeatedly refine intermediate features and pose estimates. Ablation experiments indicate learning specific features significantly improves the localization of occluded parts and thus benefits HPE. Our approach also outperforms all state-of-the-art methods on two benchmark datasets, with an outstanding advantage when occlusion occurs.

1. Introduction

Human pose estimation (HPE) aims to locate body parts from input images[1]. It serves as a fundamental tool for several practical applications such as human-computer interaction [27], person re-identification [34] and action recognition [46]. Early work attempts to solve this problem via handcrafted features and graphical models [10, 30, 32, 36, 41, 53]. However, they failed to perform well in case of severe body deformation, occlusion, clutter backgrounds and varying viewpoints.

To tackle these difficulties, recent and state-of-the-art HPE systems [18, 38, 49, 50, 5, 6, 35, 3, 28, 48, 51, 45] are

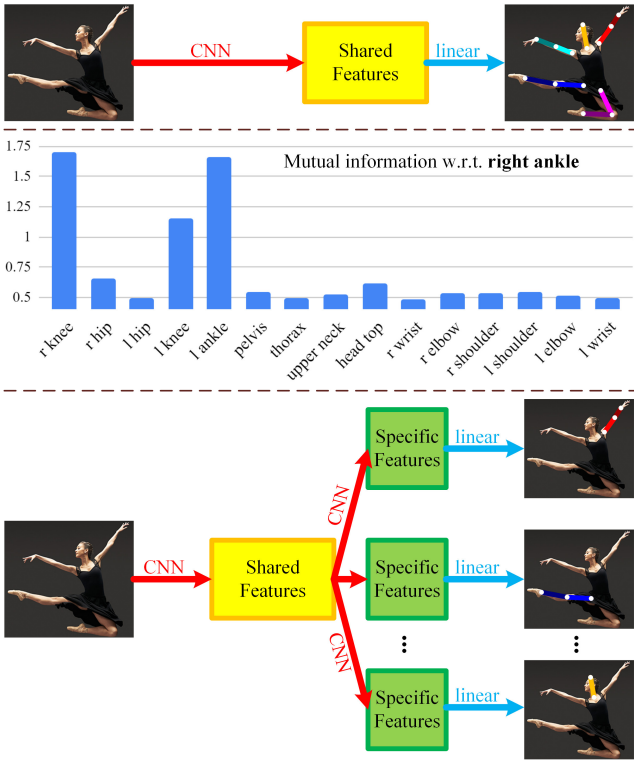

Figure 1. Top: Previous approaches exploit CNNs to learn fully shared features for all body parts, from which their locations, in the form of spatial coordinates or heat maps, are linearly regressed. Middle: Mutual information of each part's location w.r.t. the right ankle's location. Bottom: Our statistical analysis (Sec. 3.1) indicates not all parts are related to each other. Motivated by the fact that sharing a representation for unrelated tasks can deteriorate their performances, this paper tries to identify related parts and study whether learning specific features for them helps improve pose estimation.

[1]We focus on 2D single-person pose estimation from RGB images.

universally built on convolutional neural networks (CNNs) [11, 21, 20] due to their ability to learn robust feature representations for both images and spatial contexts directly from data. Toshev and Szegedy [45] use a cascade of CNNs to regress the spatial coordinates of body joints in a holistic fashion. Wei *et al*. [48] design a multi-stage network to recursively refine belief maps of part locations. Newell *et al*. [28] consolidate features across all scales via a novel *hourglass* network to capture various spatial relationships associated with the body. Tang *et al*. [38] exploit CNNs to learn the compositionality [39] of human bodies to resolve low-level ambiguities in high-level pose predictions.

One commonality of these approaches is that they learn a shared representation to linearly regress all part locations (in the form of spatial coordinates or heat maps), as shown in the upper part of Fig. 1. This is more effective and efficient than learning different networks for different parts because HPE is inherently a homogeneous multi-task learning (MTL) problem [33], with the localization of each part as a different task. Sharing a representation among related tasks can result in a more compact model and better generalization ability [4, 33]. Specifically, the first a few layers of CNNs learn low-level features such as Gabor filters and color blobs, which are general to many datasets and tasks [54]. Higher-level semantics, *e.g*., body parts, appears in deeper layers [56, 54]. Hints of some parts, *e.g*., knees, provide important information and constraints on locating other related parts, *e.g*., ankles, which are difficult to learn if the representations are not shared [4, 33].

However, due to the flexibility of an articulated body, not all parts are related to each other. For example, clues of the left or right wrist provide little *information* on the location of the right ankle, as illustrated in the middle part of Fig. 1. As studied in the literature of MTL [4, 33, 19, 17], sharing features for those unrelated or weakly related tasks can deteriorate their performances – a phenomenon called *negative transfer* [44]. While hints of related parts provide a reliable guide on locating an ambiguous or occluded part, regression from irrelevant features makes the model forcibly memorize them and leads to overfitting [8]. This line of analysis drives us to raise an interesting question. *Can we identify related parts and learn specific features for them to improve pose estimation?* The idea is illustrated in the lower part of Fig. 1. The representation learned in the shallower layers of a convolutional network is general [54, 56] and thus can be safely shared among all parts. Since unrelated tasks no longer share high-level features, we expect to avoid the adverse effect of negative transfer. In addition, more explicit structural knowledge, *e.g*., ankles and knees are highly related, is exposed, which encourages the model to exploit hints of related parts to resolve ambiguities in HPE.

The goal of this paper is to have a comprehensive study on this question. We start with two strategies to identify related parts. The first one is handcrafted and based on the human body structure [41, 38, 59]. Intuitively, parts connected in nature are related. The second strategy is data-driven and treats the location of each part as a random variable. We estimate their probability distributions from a public dataset [1] and group related parts based on their mutual information. Then a part-based branching network (PBN) is introduced. It consists of a *trunk* to learn a shared representation that is general to all body parts and some subsequent *branches* to learn high-level features that are specific to each group of related parts. Finally, we present a multi-stage version of this network to repeatedly refine intermediate features and pose estimates.

Our ablation study demonstrates that (1) the data-driven part grouping strategy generally works better than the handcrafted one and (2) learning those specific features significantly improves the localization of occluded parts and thus benefits HPE. Experimental results on two benchmarks show the proposed approach outperforms all state-of-the-art methods, with a clear advantage when occlusion occurs.

In sum, the contribution of this paper is as follows.

- All previous CNN-based HPE approaches take it for granted that features should be fully shared for all body parts. To the best of our knowledge, we are the first to identify the problem of this practice and address it via a simple and effective part-based branching network.
- This is the first attempt to exploit the probability distributions of part locations and their mutual information to group related parts. We show it is more effective than an alternative approach based on the human body structure.
- Our model has an outstanding advantage on locating occluded parts, which is the greatest challenge for existing methods. We also report new state-of-the-art results on two well-known benchmark datasets.

2. Related Work

CNN-based HPE. Different from all previous CNN-based HPE approaches [18, 38, 49, 50, 5, 6, 35, 3, 28, 48, 51, 45], which learn a fully shared representation for all body parts, this paper means to have a comprehensive study on whether learning specific features for related parts helps HPE. In addition, we propose different strategies to identify related parts and test their effectiveness.

MTL. By learning tasks in parallel while using a shared representation, MTL [4, 58, 13, 26] exploits the domain information contained in the training signals of related tasks as an inductive bias to improve generalization. It expects what is learned for each task can help other tasks be learned better.

Recently, MTL has been successfully applied to landmark detection. Zhang *et al*. [57] optimize facial landmark detection together with heterogeneous but subtly correlated tasks, *i.e*., head pose estimation and facial attribute inference. Ranjan *et al*. [31] design a unified deep MTL framework for simultaneous face detection, landmark localization, pose estimation and gender recognition. Li *et al*. [22] simultaneously learn a pose-joint regressor and a sliding-window body-part detector in a deep neural network. All these approaches treat the localization of all landmarks like a single task and introduce some auxiliary tasks for joint training. By contrast, we focus on HPE alone and explicitly treat the localization of each part as a different task. In addition, they share a representation for all landmarks while we learn specific features for related parts.

Some earlier work [19, 17] tries to tackle the negative transfer problem by imposing some structural prior, *e.g*., sparsity, on the model parameters. However, they focus on linear models with predefined features. Recently, Yang *et al*. [52] exploit tensor factorization to flexibly share knowledge in fully connected and convolutional layers. Lu *et al*. [24] propose a greedy and dynamic strategy to build MTL networks. However, they focus on network construction and limit their scope to classification tasks whose outputs are not structured. By contrast, we are the first to identify the problem of sharing a representation for all body parts in the context of HPE. This is also the first study on whether learning specific features for related parts improves pose estimation. In addition, we propose a novel and effective strategy to identify related parts by measuring their mutual information.

Related parts. Several lines of research have made use of related parts to build hierarchical graphical models [10, 16, 29, 30, 41, 47] or network architectures [38] for HPE. Our approach differs with them in that: (1) they use fully shared (handcrafted or learned) features for all body parts while we learn specific features for related parts; (2) they manually define related parts based on the body structure while we also consider a data-driven approach based on mutual information.

3. Our Approach

We first introduce two strategies to identify related body parts (Sec. 3.1). Then a part-based branching network is proposed to learn specific features for them (Sec. 3.2). Finally, we present a multi-stage version of this network to repeatedly refine intermediate features and part localizations (Sec. 3.3).

3.1. Related body parts

The most straightforward way to identify related parts is to exploit the human body structure. Intuitively, parts connected in nature are related. Following [38, 41], sixteen

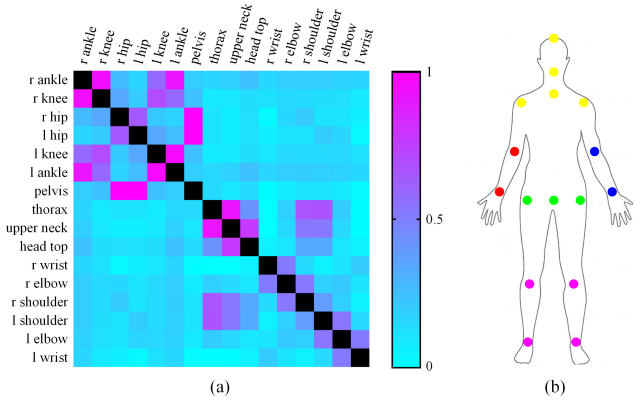

Figure 2. (a) Normalized mutual information between each pair of body parts. (b) Five groups of related parts obtained by applying spectral clustering to the matrix in (a).

body parts can be divided into six groups: (1) head top, upper neck and thorax, (2) left wrist, left elbow and left shoulder, (3) right wrist, right elbow and right shoulder, (4) left knee and left ankle, (5) right knee and right ankle, (6) left hip, right hip and pelvis.

The second strategy treats the location of each part as a random variable $l_m \in \mathcal{L}, m \in \{1, ..., M\}$, where $\mathcal{L}$ is the spatial domain and M is the total number of interesting body parts. A natural way to measure the *relatedness* or mutual dependency between two random variables is to calculate their mutual information [25]:

$$I(l_m, l_n) = \sum_{l_m \in \mathcal{L}} \sum_{l_n \in \mathcal{L}} p(l_m, l_n) \log \left(\frac{p(l_m, l_n)}{p(l_m)p(l_n)} \right) \quad (1)$$

where $p(\cdot)$ and $p(\cdot, \cdot)$ respectively represent marginal and joint probability distributions. It quantifies the amount of information obtained about one random variable through observing the other random variable. A high value of $I(l_m, l_n)$ indicates that features strongly relevant to part m also provide informative clues of part n, and vice versa. Thus, sharing a high-level representation for them should be beneficial. Compared with the Pearson correlation, which measures the strength of a linear association between two random variables, the mutual information is a more suitable metric here because it accounts for both linear and nonlinear associations and is zero if and only if two random variables are independent.

We estimate distributions of part locations from data in a nonparametric fashion. The MPII human pose dataset [1] is adopted here because (1) it has 25k training samples with high-quality annotations, *e.g*., human poses, scales and centers and (2) it covers a wide range of everyday human activities and a great variety of full-body poses. We scale the poses and center them in a normalized spatial domain, *i.e*., a 16×16 lattice. A low resolution is necessary because (1) it makes the statistical estimation robust to small pose

perturbations and (2) the total number of samples is limited. Then we use histograms to estimate $p(l_m, l_n)$ where $m, n \in \{1, ..., M\}$.

Fig. 2(a) visualizes the mutual information computed between each pair of body parts. To focus on the relatedness between different parts, we have removed the diagonal elements and linearly normalized all the remaining entries to be within $[0, 1]$. Obviously, some parts, *e.g.*, right ankle and left ankle, are more related than the others, *e.g.*, right ankle and left wrist.

Finally, we treat $\{I(l_m, l_n)\}_{m,n \in \{1,...,M\}}$ as an affinity matrix and use spectral clustering [9] to group related parts. For example, setting the cluster number to five will result in a part grouping shown in Fig. 2(b). We can see most parts within the same group are connected in the body skeleton, which agrees to our intuition. The only exception is the group of ankles and knees denoted by the purple dots. This result is easy to understand from Fig. 2(a): all of them share high values of mutual information between each other. These four parts will still be in the same group even if the cluster number is increased by one. Instead, the head and neck will be detached from the shoulders and thorax.

3.2. Part-based branching network (PBN)

As illustrated in Fig. 3, a part-based branching network (PBN) is a CNN architecture consisting of two sequential stages: a *trunk* to learn a shared representation that is general to all body parts and some *branches* to learn high-level features that are specific to each group of related parts. Following the standard protocol of single-person pose estimation [6, 18, 28, 38, 50], its input is a RGB image cropped around a target person and scaled to a fixed size, *e.g.*, 256×256.

The network first uses convolutions and max-poolings to produce feature maps with decreasing spatial dimensions but increasing channel numbers, a practice adopted in recent CNN architectures [14, 28, 38, 50]. Specifically, it starts with three 3×3 convolutional layers (64 channels) and one 2×2 pooling layer (after the first convolution), followed by a residual block[2] (128 channels) and another round of pooling to bring the resolution down from 256×256 to 64×64. After two subsequent residual blocks (128 and 256 channels), we get a 256-channel feature map of resolution 64×64, *i.e.*, the first yellow rectangle in Fig. 3.

Next is an hourglass network [28] to strengthen the shared representation. It uses residual blocks and max-poolings to process input features down to a very low resolution, *i.e.*, 4×4. At each max-pooling step, the network

[2]The bottleneck residual block [14] is used throughout our network. It consists of three layers, *i.e.*, 1×1, 3×3, and 1×1 convolutions. The 1×1 layers are responsible for reducing and then increasing (restoring) dimensions, leaving the 3×3 layer a bottleneck with smaller input/output dimensions.

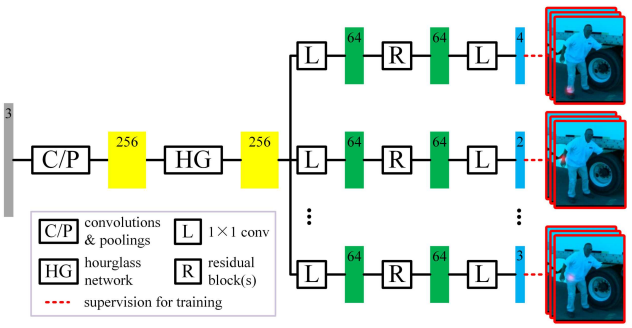

Figure 3. Illustration of a part-based branching network (PBN) for HPE. The gray and blue rectangles respectively denote an input image and predicted heat maps. The yellow and green rectangles respectively represent shared and specific features. The channel number is included in each colored rectangle. An MSE loss is applied to compare the predicted heat map to a ground truth one.

branches off and applies one more residual block at the original pre-pooled resolution. When reaching the lowest resolution, it begins a top-down sequence of upsamplings and elementwise additions to combine features across scales. After a subsequent residual block at the output resolution, the network outputs a feature map with the same size as its input. All residual blocks here output 256-channel features.

The hourglass network is adopted here for two reasons. First, by processing and consolidating features across multiple scales, it captures various spatial relationships and contexts within the input feature maps. Second, the eight-stack hourglass network and its recent variants [6, 50, 18, 38] have achieved state-of-the-art results on standard benchmarks. Thus, it serves as a suitable baseline to test whether learning specific features for related parts can help improve pose estimation.

Finally, the network uses a set of branches to learn specific features for related parts, as shown in Fig. 3. For each part group, we first apply a 1×1 convolution to reduce the feature dimension from 256 to W, *e.g.*, $W = 64$. After D subsequent residual blocks, *e.g.*, $D = 1$, another 1×1 convolution is used to regress the heat map of each part in this group. Each pixel of a heat map represents the probability of a part's presence at the corresponding coordinate. Here W and D are two hyperparameters respectively controlling the *width* and *depth* [14] of specific feature layers. We will use ablation experiments to study how they affect the HPE performance. In the training phase, a mean squared error (MSE) loss is applied to compare the predicted heat map to a ground truth one consisting of a 2D Gaussian (std=1 pixel) centered on the part location.

Since unrelated tasks are no longer learned using a fully shared representation, a PBN can reduce the adverse effect of negative transfer. Compared with using different branches for different parts or one branch for all parts, our approach incorporates more explicit structural knowledge,

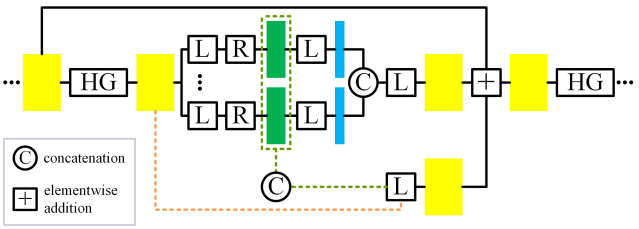

Figure 4. Illustration of stacking multiple PBNs. The symbols in Fig. 3 are reused here. Each PBN predicts a set of heat maps, *i.e.*, the blue rectangles. An MSE loss is applied to them using the same ground truth (omitted in the figure). The orange and green dashed lines denote two options to propagate shared or specific features to the next PBN. In practice, we find the former works better.

e.g., ankles and knees are highly related, into the network and guides it to exploit hints of related parts to resolve ambiguities in pose estimation.

3.3. Stacked PBNs

Recent study [28, 50, 6, 38] shows that sequentially stacking multiple CNN modules end-to-end, feeding the output of one as input into the next, can repeatedly refine initial estimates and intermediate features across the whole image. This motivates us to extend our network to a multi-stage version as illustrated in Fig. 4. Following Newell *et al*. [28], three feature maps are fused via elementwise additions: (1) an identity mapping from the input of the current hourglass, (2) heat map predictions remapped by a 1×1 convolution to match the channel number of the intermediate features and (3) shared features after the hourglass, denoted by the orange dashed line in Fig. 4. The fusion result then serves directly as the input for the next PBN, which generates another set of predictions. An MSE loss is applied to the predictions of all stacked PBNs using the same ground truth.

We have also considered propagating specific features instead of shared ones to the subsequent PBN, denoted by the green dashed line in Fig. 4. However, we practically find this will bring difficulty to the learning process, increasing both training and validation losses, and degrade the HPE performance.

4. Experiments

Our approach is evaluated on two HPE benchmark datasets: MPII Human Pose [1] and Leeds Sports Pose (LSP) [16]. The MPII dataset consists of around 25k images with 40k annotated samples (28k for training, 11k for testing). Following [43, 28, 38], 3k samples are taken as a validation set to tune hyperparameters and conduct ablation study. The LSP dataset and its extended training set contain 11k training images and 1k testing images from sports activities. As a common practice [48, 6, 38], we train the network by including the MPII training samples.

	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	Mean
Bulat, ECCV'16 [3]	97.9	95.1	89.9	85.3	89.4	85.7	81.7	89.7
Gkioxary, ECCV'16 [12]	96.2	93.1	86.7	82.1	85.2	81.4	74.1	86.1
Insafutdinov, ECCV'16 [15]	96.8	95.2	89.3	84.4	88.4	83.4	78.0	88.5
Lifshitz, ECCV'16 [23]	97.8	93.3	85.7	80.4	85.3	76.6	70.2	85.0
Newell, ECCV'16 [28]	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9
Belagiannis, FG'17 [2]	97.7	95.0	88.2	83.0	87.9	82.6	78.4	88.1
Chu, CVPR'17 [6]	98.5	96.3	91.9	88.1	90.6	88.0	85.0	91.5
Chen, ICCV'17 [5]	98.1	96.5	92.5	88.5	90.2	89.6	86.0	91.9
Sun, ICCV'17 [37]	97.5	94.3	87.0	81.2	86.5	78.5	75.4	86.4
Sun, ICCV'17 [35]	98.1	96.2	91.2	87.2	89.8	87.4	84.1	91.0
Yang, ICCV'17 [50]	98.5	96.7	92.5	88.7	91.1	88.6	86.0	92.0
Ke, ECCV'18 [18]	98.5	96.8	92.7	88.4	90.6	89.4	86.3	92.1
Tang, ECCV'18 [38]	98.4	96.9	92.6	88.7	91.8	89.4	86.2	92.3
Ours	**98.7**	**97.1**	**93.1**	**89.4**	**91.9**	**90.1**	**86.7**	**92.7**

Table 1. Comparisons of PCKh@0.5 scores on the MPII testing set.

	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	Mean
Yang, ICCV'17 [50]	97.4	96.2	91.1	86.9	90.1	86.0	83.9	91.0
Tang, ECCV'18 [38]	97.4	96.2	91.0	86.9	90.6	86.8	84.5	91.2
Ours	**97.5**	**96.5**	**91.7**	**87.7**	**91.1**	**87.7**	**85.2**	**91.8**

Table 2. Comparisons of PCKh@0.5 scores on the MPII validation set.

Following previous work, we use the Percentage of Correct Keypoints (PCK) [1] as the evaluation metric. It calculates the percentage of part localizations that fall within a normalized distance of the ground truth. For LSP, the distance is normalized by the torso size, and for MPII, by a fraction of the head size (referred to as PCKh).

4.1. Implementation details

Each input image is cropped around the target person according to the annotated body position and scale. They are then resized to 256×256 pixels before being fed into the network. Training data are augmented by random scaling (+/- 0.25), rotation (+/- 30 degrees), shearing (+/- 0.5), horizontal flipping and color jittering. Our implementation is based on Torch [7]. We optimize the network via RMSProp [42] with a batch size 16 for 250 epochs. The learning rate is initialized as 2.5×10^{-4} and then dropped by a factor of 10 at the 170th and 220th epochs. The final prediction is the maximum activating location of each heat map estimated by the last PBN.

4.2. Benchmark results

We use an eight-stack PBN for benchmark evaluation. One residual block with 64 input/output channels, *i.e.*, $D = 1$ and $W = 64$, is used to learn specific features for each part group shown in Fig. 2(b). Testing is conducted on six-scale image pyramids with flipping [50, 38].

MPII. Tab. 1 compares the performances of our network and the most recent HPE methods on the MPII testing set.

	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	Mean
Yang, ICCV'17 [50]	-	90.6	74.7	63.8	83.1	67.8	**63.5**	76.8
Tang, ECCV'18 [38]	-	90.5	74.5	62.9	84.2	68.8	62.2	76.7
Ours	-	**92.0**	**76.2**	**64.6**	**86.1**	**70.3**	63.2	**78.2**

Table 3. Comparisons of PCKh@0.5 scores on the invisible parts in the MPII validation set.

	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	Mean
Bulat, ECCV'16 [3]	97.2	92.1	88.1	85.2	92.2	91.4	88.7	90.7
Insafutdinov, ECCV'16 [15]	97.4	92.7	87.5	84.4	91.5	89.9	87.2	90.1
Lifshitz, ECCV'16 [23]	96.8	89.0	82.7	79.1	90.9	86.0	82.5	86.7
Yu, ECCV'16 [55]	87.2	88.2	82.4	76.3	91.4	85.8	78.7	84.3
Chu, CVPR'17 [6]	98.1	93.7	89.3	86.9	93.4	94.0	92.5	92.6
Chen, ICCV'17 [5]	98.5	94.0	89.8	87.5	93.9	94.1	93.0	93.1
Sun, ICCV'17 [35]	97.9	93.6	89.0	85.8	92.9	91.2	90.5	91.6
Yang, ICCV'17 [50]	98.3	94.5	92.2	88.9	**94.4**	95.0	93.7	93.9
Tang, ECCV'18 [40]	97.5	95.0	92.5	**90.1**	93.7	95.2	94.2	94.0
Ours	**98.6**	**95.4**	**93.3**	89.8	94.3	**95.7**	**94.4**	**94.5**

Table 4. Comparisons of PCK@0.2 scores on the LSP testing set.

	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	Mean
Tang, ECCV'18 [38]	98.3	95.9	93.5	**90.7**	95.0	96.6	**95.7**	95.1
Ours	**98.7**	**96.4**	**94.3**	90.6	**95.2**	**97.2**	**95.7**	**95.4**

Table 5. Comparisons of PCK@0.2 scores on the corrected LSP testing set.

Our approach achieves an overall PCKh@0.5 score 92.7%, which is a new state-of-the-art result. It also outperforms all previous methods on each individual body part.

Tab. 2 compares the performance of our model on the MPII validation set with those of two state-of-the-art methods[3]. Our network achieves the highest scores on all parts.

The MPII dataset also provides visibility annotations for each part (except for the head). This enables us to evaluate different models on the subset of invisible parts and study their robustness to occlusion. The results are shown in Tab. 3. Note none of these three networks has exploited the visibility labels for training. Comparing Tabs. 2 and 3, we can observe occlusion significantly deteriorates the performances of all approaches. It is still a great challenge for high-accuracy pose estimation. Nevertheless, the specific features learned for related parts provide informative hints and constraints on the locations of occluded parts, which results in a much better performance than the state-of-the-art methods. Specially, our model respectively achieves 1.5%, 1.7%, 1.7%, 1.9%, 1.5% and 1.0% improvements on shoulders, elbows, wrists, hips, knees and ankles compared to the top-performing method [38] on the MPII dataset.

LSP. Tab. 4 compares the performances of our model and the most recent HPE methods on the LSP testing set. Our approach achieves an overall PCK@0.2 score 94.5% and outperforms all state-of-the-art methods. Tang *et al.* [38] found a few annotations in the LSP dataset are on the wrong side and manually corrected them. Tab. 5 compares their approach with ours on the corrected testing set and shows that our network has an overall better performance.

[3]In Tabs. 2 and 3, the predictions of [50, 38] on the MPII validation set were released by their respective authors.

4.3. Ablation study

We conduct ablation experiments on the MPII validation set. Mean PCKh@0.5 over ten hard joints, *i.e.*, ankles, knees, hips, wrists and elbows, is used as the evaluation metric. We use single-scale testing in all the experiments.

Depth and width of specific feature layers. Fig. 5(a) compares the performances of using $D = 1$ and $D = 2$ residual blocks to learn specific features for each group of related parts. We can see that using more residual blocks generally worsens pose estimation regardless of the channel numbers. We have also tried $D = 3$ and got the same observation. This is likely due to overfitting because increasing D always results in a lower training loss. Thus, we set $D = 1$ in the remaining ablation experiments.

Fig. 5(b) shows how the width of specific feature layers affects the performance. Using more feature channels does not always lead to a gain in performance. $W = 64$ turns out to be a good balance between accuracy and complexity.

Do specific feature layers help? We try to have a rigorous study on whether learning specific features for related parts helps improve HPE. We first build a baseline by removing the branches from our network and adding a linear layer to predict the heat maps of all parts. Fig. 5(c) shows that it performs much worse than our original model. In order to rule out the advantage brought by a larger model capacity, we also consider a deep baseline (denoted as *Deep BS*). It is constructed by replacing our branches with a 256-channel residual block, followed by a linear layer for heat map regression. Fig. 5(c) shows that our network, having fewer parameters and a lower computational complexity, clearly outperforms this deep baseline.

We further compare an eight-stack PBN, which has been used for benchmark evaluation, with its deep baseline. Tab. 6 shows that our approach can achieve an overall better performance with a smaller model capacity. Learning specific features instead of a fully shared representation leads to an overall 1.3% improvement on the occluded parts while retaining the high accuracy for visible parts. Tab. 7 shows that our model respectively achieves 2.02%, 1.85% and 1.91% improvements on occluded wrists, hips and ankles, which are considered as the most challenging parts to be detected.

Part grouping strategy. We have considered two strategies to identify related parts in Sec. 3.1. They are respectively based on the human body structure (denoted as *Body*) and statistical analysis (denoted as *Stat*), *i.e.*, mutual information among parts. Fig. 5(d) shows that the proposed data-driven approach always outperforms the handcrafted one regardless of the width of the specific feature layers.

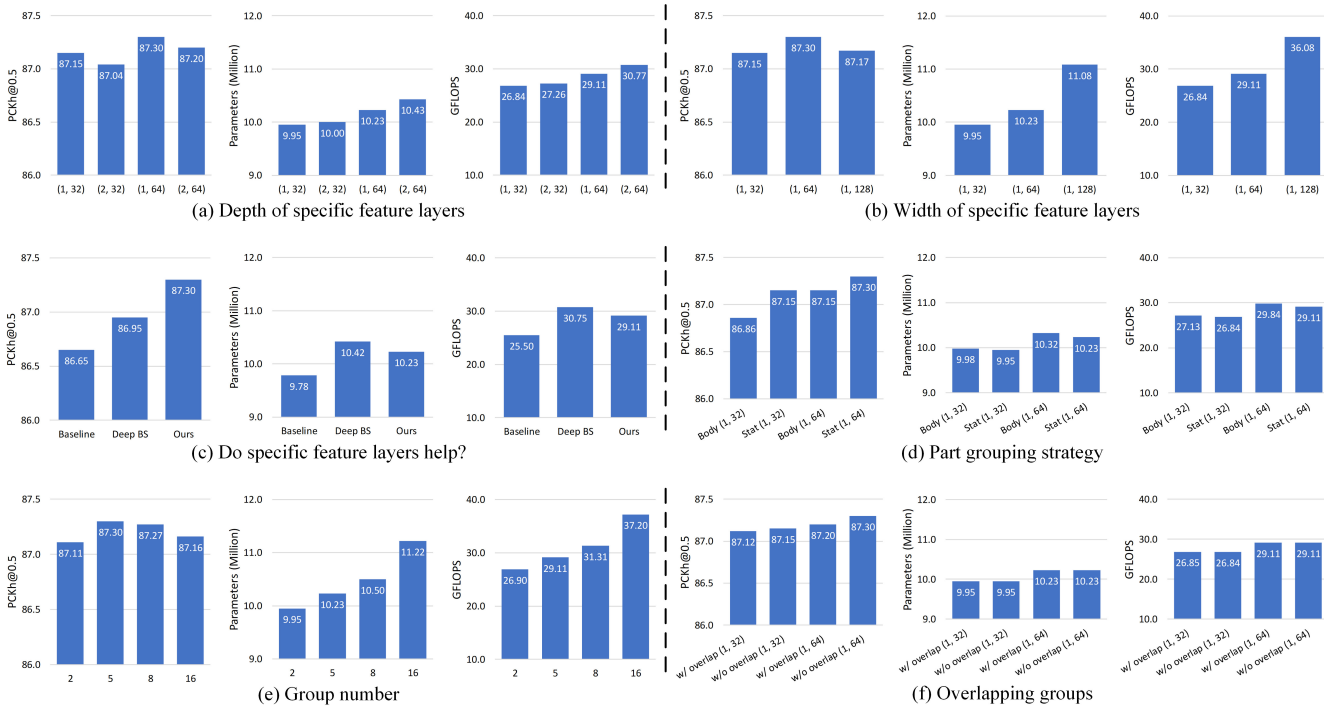

Figure 5. Ablation study using variants of three-stack PBNs. (D, W) denotes the depth and width of specific feature layers. Unless otherwise stated, one residual block with 64 input/output channels, *i.e.*, $D = 1$ and $W = 64$, is used to learn specific features for each of the five part groups shown in Fig. 2(b). See Sec. 4.3 for detailed analysis.

	Invisible parts	Visible parts	Overall	Parameters
Deep BS (8 stacks)	70.29	93.22	87.74	27.22M
Ours (8 stacks)	**71.59**	**93.31**	**88.14**	**26.69M**

Table 6. Comparisons of an eight-stack PBN and its deep baseline on the MPII validation set. Mean PCKh@0.5 scores on the ten hard joints are reported.

	Elb.	Wri.	Hip	Knee	Ank.	Mean
Deep BS (8 stacks)	75.05	62.10	83.43	69.30	61.57	70.29
Ours (8 stacks)	**75.14**	**64.12**	**85.28**	**69.92**	**63.48**	**71.59**

Table 7. Comparisons of PCKh@0.5 scores obtained by an eight-stack PBN and its deep baseline on the invisible parts in the MPII validation set. Results on the ten hard joints are reported.

Number of part groups. We study the affect of group numbers on the HPE performance by setting the cluster number in Sec. 3.1 to 2, 5, 8 or 16. The results are shown in Fig. 5(e). While increasing the group number from 2 to 5 boosts the performance, using more than 5 groups can hardly result in further improvement.

Overlapping groups. The part groups we identify in Sec. 3.1 are disjoint sets, *i.e.*, having no element in common. Having a close look at Fig. 2(a), we find there exist parts which share significant mutual information but are not in the same group, *e.g.*, left (right) elbow and left (right) shoulder. This motivates us to learn specific features for overlapping part groups. For the related parts within each group, we still use a residual block and a linear layer to regress their heat maps. If multiple heat maps from different branches correspond to a same part, we use their average as the final prediction. Fig. 5(f) reports the results obtained using overlapping groups. Here the left (right) shoulder belongs to two groups: its original group represented by the yellow dots in Fig. 2(b) and the group of the left (right) elbow and the left (right) wrist. We can see using overlapping groups does not improve the performance.

Feature fusion in stacked PBNs. We find propagating shared features along the trunk of the network, *i.e.*, the orange dashed line in Fig. 4, generally works better than fusing specific features with the shared ones, *i.e.*, the green dashed line in Fig. 4. For a three-stack network, the PCKh@0.5 scores achieved by the former and the latter are respectively 87.30% and 87.21%. Their training losses are respectively 1.95×10^{-3} and 1.99×10^{-3}. The gap is more significant for an eight-stack PBN: 88.14% versus 87.68% for PCKh@0.5 scores and 4.99×10^{-3} versus 5.11×10^{-3} for training losses.

Negative transfer. We find removing ankles from the tasks of an hourglass network generally improves the localization of upper body parts (0.30%) but degrades the results of lower body parts (0.45%). This indicates that (1) learning related body parts is beneficial and (2) sharing features among unrelated parts can be harmful.

Figure 6. Human pose estimation results obtained by our approach on the MPII dataset (top two rows) and LSP dataset (bottom two rows).

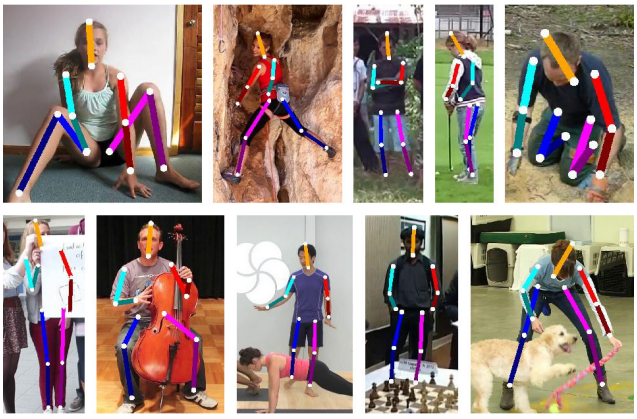

Figure 7. Examples showing our approach can handle both self-occlusion (top row) and other-occlusion (bottom row).

4.4. Qualitative results

Fig. 6 shows some pose estimation results obtained by our approach on the MPII dataset and LSP dataset. Fig. 7 provides some examples showing our approach can handle both self-occlusion and other-occlusion. Fig. 8 shows our approach is able to correct some wrong part localizations obtained by a state-of-the-art method [38] due to occlusion.

Figure 8. Examples showing our approach (bottom row) is able to correct some wrong part localizations (highlighted by green circles) obtained by a state-of-the-art method [38] due to occlusion (top row).

5. Conclusion

With substantial benchmark experiments and ablation study, we conclude that learning specific features for related body parts significantly improves the localization of occluded parts and thus benefits human pose estimation.

Acknowledgement. This work was supported in part by National Science Foundation grant IIS-1619078, IIS-1815561, and the Army Research Ofice ARO W911NF-16-1-0138.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 2, 3, 5

[2] Vasileios Belagiannis and Andrew Zisserman. Recurrent human pose estimation. In *FG*, 2017. 5

[3] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *ECCV*, 2016. 1, 2, 5, 6

[4] Rich Caruana. Multitask learning. *Machine learning*, 1997. 2

[5] Yu Chen, Chunhua Shen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. In *ICCV*, 2017. 1, 2, 5, 6

[6] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *CVPR*, 2017. 1, 2, 4, 5, 6

[7] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7: A matlab-like environment for machine learning. In *NIPS Workshop*, 2011. 5

[8] Kan Deng. *Omega: On-line memory-based general purpose system classifier*. PhD thesis, Carnegie Mellon University, 1998. 2

[9] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *SIGKDD*, 2004. 4

[10] Kun Duan, Dhruv Batra, and David J Crandall. A multi-layer composite model for human pose estimation. In *BMVC*, 2012. 1, 3

[11] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 1980. 2

[12] Georgia Gkioxari, Alexander Toshev, and Navdeep Jaitly. Chained predictions using convolutional neural networks. In *ECCV*, 2016. 5

[13] Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In *ECCV*, 2018. 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[15] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, 2016. 5, 6

[16] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010. 3, 5

[17] Zhuoliang Kang, Kristen Grauman, and Fei Sha. Learning with whom to share in multi-task feature learning. In *ICML*, 2011. 2, 3

[18] Lipeng Ke, Ming-Ching Chang, Honggang Qi, and Siwei Lyu. Multi-scale structure-aware network for human pose estimation. In *ECCV*, 2018. 1, 2, 4, 5

[19] Abhishek Kumar and Hal Daume III. Learning task grouping and overlap in multi-task learning. *arXiv preprint arXiv:1206.6417*, 2012. 2, 3

[20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 2015. 2

[21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. 2

[22] Sijin Li, Zhi-Qiang Liu, and Antoni B Chan. Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network. In *CVPR workshop*, 2014. 3

[23] Ita Lifshitz, Ethan Fetaya, and Shimon Ullman. Human pose estimation using deep consensus voting. In *ECCV*, 2016. 5, 6

[24] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogério Schmidt Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *CVPR*, 2017. 3

[25] David JC MacKay and David JC Mac Kay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003. 3

[26] Youssef A Mejjati, Darren Cosker, and Kwang In Kim. Multi-task learning by maximizing statistical dependence. In *CVPR*, 2018. 2

[27] Thomas B Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer vision and image understanding*, 2001. 1

[28] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 1, 2, 4, 5

[29] Seyoung Park, Xiaohan Nie, and Song-Chun Zhu. Attribute and-or grammar for joint parsing of human pose, parts and attributes. *PAMI*, 2017. 3

[30] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *CVPR*, 2013. 1, 3

[31] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *PAMI*, 2017. 3

[32] Brandon Rothrock, Seyoung Park, and Song-Chun Zhu. Integrating grammar and segmentation for human pose estimation. In *CVPR*, 2013. 1

[33] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 2

[34] Chi Su, Jianing Li, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. Pose-driven deep convolutional model for person re-identification. In *ICCV*, 2017. 1

[35] Ke Sun, Cuiling Lan, Junliang Xing, Wenjun Zeng, Dong Liu, and Jingdong Wang. Human pose estimation using global and local normalization. In *ICCV*, 2017. 1, 2, 5, 6

[36] Min Sun and Silvio Savarese. Articulated part-based model for joint object detection and pose estimation. In *ICCV*, 2011. 1

[37] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *ICCV*, 2017. 5

[38] Wei Tang, Pei Yu, and Ying Wu. Deeply learned compositional models for human pose estimation. In *ECCV*, 2018. 1, 2, 3, 4, 5, 6, 8

[39] Wei Tang, Pei Yu, Jiahuan Zhou, and Ying Wu. Towards a unified compositional model for visual pattern modeling. In *ICCV*, 2017. 2

[40] Zhiqiang Tang, Xi Peng, Shijie Geng, Lingfei Wu, Shaoting Zhang, and Dimitris Metaxas. Quantized densely connected u-nets for efficient landmark localization. In *ECCV*, 2018. 6

[41] Yuandong Tian, C Lawrence Zitnick, and Srinivasa G Narasimhan. Exploring the spatial hierarchy of mixture models for human pose estimation. In *ECCV*, 2012. 1, 2, 3

[42] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 2012. 5

[43] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *CVPR*, 2015. 5

[44] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. 2010. 2

[45] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014. 1, 2

[46] Chunyu Wang, Yizhou Wang, and Alan L Yuille. An approach to pose-based action recognition. In *CVPR*, 2013. 1

[47] Yang Wang, Duan Tran, and Zicheng Liao. Learning hierarchical poselets for human parsing. In *CVPR*, 2011. 3

[48] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016. 1, 2, 5

[49] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018. 1, 2

[50] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *ICCV*, 2017. 1, 2, 4, 5, 6

[51] Wei Yang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *CVPR*, 2016. 1, 2

[52] Yongxin Yang and Timothy Hospedales. Deep multi-task representation learning: A tensor factorisation approach. *arXiv preprint arXiv:1605.06391*, 2016. 3

[53] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011. 1

[54] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014. 2

[55] Xiang Yu, Feng Zhou, and Manmohan Chandraker. Deep deformation network for object landmark localization. In *ECCV*, 2016. 6

[56] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 2

[57] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, 2014. 3

[58] Xiangyun Zhao, Haoxiang Li, Xiaohui Shen, Xiaodan Liang, and Ying Wu. A modulation module for multi-task learning with applications in image retrieval. *arXiv preprint arXiv:1807.06708*, 2018. 2

[59] Long Leo Zhu, Yuanhao Chen, and Alan Yuille. Recursive compositional models for vision: Description and review of recent work. *Journal of Mathematical Imaging and Vision*, 2011. 2