Human Pose Estimation via Convolutional Part Heatmap Regression

Adrian Bulat(✉) and Georgios Tzimiropoulos

Computer Vision Laboratory, University of Nottingham, Nottingham, UK
{adrian.bulat,yorgos.tzimiropoulos}@nottingham.ac.uk

Abstract. This paper is on human pose estimation using Convolutional Neural Networks. Our main contribution is a CNN cascaded architecture specifically designed for learning part relationships and spatial context, and robustly inferring pose even for the case of severe part occlusions. To this end, we propose a detection-followed-by-regression CNN cascade. The first part of our cascade outputs part detection heatmaps and the second part performs regression on these heatmaps. The benefits of the proposed architecture are multi-fold: It guides the network where to focus in the image and effectively encodes part constraints and context. More importantly, it can effectively cope with occlusions because part detection heatmaps for occluded parts provide low confidence scores which subsequently guide the regression part of our network to rely on contextual information in order to predict the location of these parts. Additionally, we show that the proposed cascade is flexible enough to readily allow the integration of various CNN architectures for both detection and regression, including recent ones based on residual learning. Finally, we illustrate that our cascade achieves top performance on the MPII and LSP data sets. Code can be downloaded from http://www.cs.nott.ac.uk/~psxab5/.

Keywords: Human pose estimation · Part heatmap regression · Convolutional Neural Networks

1 Introduction

Articulated human pose estimation from images is a Computer Vision problem of extraordinary difficulty. Algorithms have to deal with the very large number of feasible human poses, large changes in human appearance (e.g. foreshortening, clothing), part occlusions (including self-occlusions) and the presence of multiple people within close proximity to each other. A key question for addressing these problems is how to extract strong low and mid-level appearance features capturing discriminative as well as relevant contextual information and how to model complex part relationships allowing for effective yet efficient pose inference. Being capable of performing these tasks in an end-to-end fashion, Convolutional Neural Networks (CNNs) have been recently shown to feature remarkably robust performance and high part localization accuracy. Yet, the accurate estimation of

B. Leibe et al. (Eds.): ECCV 2016, Part VII, LNCS 9911, pp. 717–732, 2016.
DOI: 10.1007/978-3-319-46478-7_44

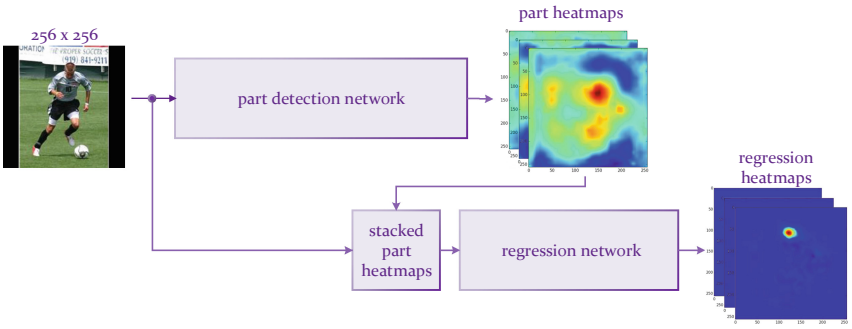

Fig. 1. Proposed architecture: Our CNN cascade consists of two connected deep subnetworks. The first one (upper part in the figure) is a part detection network trained to detect the individual body parts using a per-pixel sigmoid loss. Its output is a set of N part heatmaps. The second one is a regression subnetwork that jointly regresses the part heatmaps stacked alongside the input image to confidence maps representing the location of the body parts.

the locations of occluded body parts is still considered a difficult open problem. The main contribution of this paper is a CNN cascaded architecture specifically designed to alleviate this problem.

There is a very large amount of work on the problem of human pose estimation. Prior to the advent of neural networks most prior work was primarily based on pictorial structures [1] which model the human body as a collection of rigid templates and a set of pairwise potentials taking the form of a tree structure, thus allowing for efficient and exact inference at test time. Recent work includes sophisticated extensions like mixture, hierarchical, multimodal and strong appearance models [2–6], non-tree models [7,8] as well as cascaded/sequential prediction models like pose machines [9].

More recently methods based on Convolutional Neural Networks have been shown to produce remarkable performance for a variety of difficult Computer Vision tasks including recognition [10,11], detection [12] and semantic segmentation [13] outperforming prior work by a large margin. A key feature of these approaches is that they integrate non-linear hierarchical feature extraction with the classification or regression task in hand being also able to capitalize on very large data sets that are now readily available. In the context of human pose estimation, it is natural to formulate the problem as a regression one in which CNN features are regressed in order to provide joint prediction of the body parts [14–17]. For the case of non-visible parts though, learning the complex mapping from occluded part appearances to part locations is hard and the network has to rely on contextual information (provided from the other visible parts) to infer the occluded parts' location. In this paper, we show how to circumvent this problem by proposing a detection-followed-by-regression CNN cascade for articulated human pose estimation.

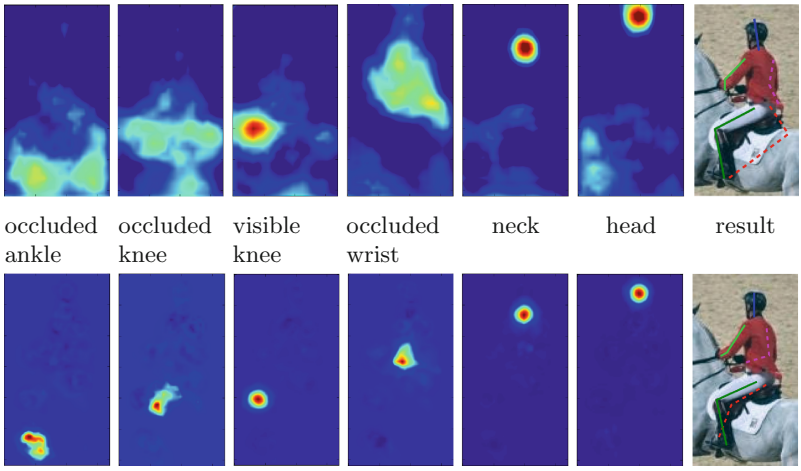

Fig. 2. Paper's main idea: The first row shows the produced part detection heatmaps for both visible (neck, head, left knee) and occluded (ankle, wrist, right knee) parts (drawn with a dashed line). Observe that the confidence for the occluded parts is much lower than that of the non-occluded parts but still higher than that of the background providing useful context about their rough location. The second row shows the output of our regression subnetwork. Observe that the confidence for the visible parts is higher and more localized and clearly the network is able to provide high confidence for the correct location of the occluded parts. **Note**: image taken from LSP test set.

1.1 Main Contribution

The proposed architecture is a CNN cascade consisting of two components (see Fig. 1): the first component (part detection network) is a deep network for part detection that produces detection heatmaps, one for each part of the human body. We train part detectors jointly using pixelwise sigmoid cross entropy loss function [18]. The second component is a deep regression subnetwork that jointly regresses the location of all parts (both visible and occluded), trained via confidence map regression [16]. Besides the two subnetworks, the key feature of the proposed architecture is the input to the regression subnetwork: we propose to use a stacked representation comprising the part heatmaps produced by the detection network. The proposed representation guides the network where to focus and encodes structural part relationships. Additionally, our cascade does not suffer from the problem of regressing occluded part appearances: because the part heatmaps for the occluded parts provide low confidence scores, they subsequently guide the regression part of our network to rely on contextual information (provided by the remaining parts) in order to predict the location of these parts. See Fig. 2 for a graphical representation of our paper's main idea. The proposed cascade is very simple, can be trained end-to-end, and is flexible enough to readily allow the integration of various CNN architectures for both our detection and regression subnetworks. To this end, we illustrate two

instances of our cascade, one based on the more traditional VGG converted to fully convolutional (FCN) [11,13] and one based on residual learning [10,19]. Both architectures achieve top performance on both MPII [20] and LSP [21] data sets.

2 Closely Related Work

Overview of prior work. Recently proposed methods for articulated human pose estimation using CNNs can be classified as detection-based [22–26] or regression-based [14–17,27,28]. Detection-based methods are relying on powerful CNN-based part detectors which are then combined using a graphical model [22,23] or refined using regression [24,25]. Regression-based methods try to learn a mapping from image and CNN features to part locations. A notable development has been the replacement of the standard L2 loss between the predicted and ground truth part locations with the so-called confidence map regression which defines an L2 loss between predicted and ground truth confidence maps encoded as 2D Gaussians centered at the part locations [16,23] (these regression confidence maps are not to be confused with the part detection heatmaps proposed in our work). As a mapping from CNN features to part locations might be difficult to learn in one shot, regression-based methods can be also applied sequentially (i.e. in a cascaded manner) [14,27,28]. Our CNN cascade is based on a two-step detection-followed-by-regression approach (see Fig. 1) and as such is related to both detection-based [24,25] and regression-based methods [16,27,28].

Relation to regression-based methods. Our detection-followed-by-regression cascade is related to [16] which can be seen as a two-step regression-followed-by-regression approach. As a first step [16] performs confidence map regression (based on an L2 loss) as opposed to our part detection step which is learnt via pixelwise sigmoid cross entropy loss. Then, in [16] pre-confidence maps are used as input to a subsequent regression network. We empirically found that such maps are too localised providing small spatial support. In contrast, our part heatmaps can provide large spatial context for regression. For comparison purposes, we implemented the idea of [16] using two different architectures, one based on VGG-FCN and one on residual learning, and show that the proposed detection-followed-by-regression cascade outperforms it for both cases (see Sect. 4.2). In order to improve performance, regression methods applied in a sequential, cascaded fashion have been recently proposed in [27,28]. In particular, [28] has recently reported outstanding results on both LSP [21] and MPII [20] data sets using a six-stage CNN cascade.

Relation to detection-based methods. Regarding detection-based methods, [25] has produced state-of-the-art results on both MPII and LSP data sets using a VGG-FCN network [11,13] to detect the body parts along with an L2 loss for regression that refines the part prediction. Hence, [25] does not include a subsequent part heatmap regression network as our method does. The work of [24] uses a part detection network as a first step in order to provide crude

estimates for the part locations. Subsequently, CNN features are cropped around these estimates and used for refinement using regression. Hence, [24] does not include a subsequent part heatmap regression network as our method does, and hence does not account for contextual information but allows only for local refinement.

Residual learning. Notably, all the aforementioned methods were developed prior to the advent of residual learning [10]. Very recently, residual learning was applied for the problem of human pose estimation in [26] and [19]. Residual learning was used for part detection in the system of [26]. The "stacked hourglass network" of [19] elegantly extends FCN [13] and deconvolution nets [29] within residual learning, also allowing for a more sophisticated and heavy processing during top-down processing. We explore residual learning within the proposed CNN cascade; notably for our residual regression subnetwork, we used a single hourglass network [19].

3 Method

The proposed part heatmap regression is a CNN cascade illustrated in Fig. 1. Our cascade consists of two connected subnetworks. The first subnetwork is a part detection network trained to detect the individual body parts using a per-pixel softmax loss. The output of this network is a set of N part detection heatmaps. The second subnetwork is a regression subnetwork that jointly regresses the part detection heatmaps stacked with the image/CNN features to confidence maps representing the location of the body parts.

We implemented two instances of part heatmap regression: in the first one, both subnetworks are based on VGG-FCN [11,13] and in the second one, on residual learning [10,19]. For both cases, the subnetworks and their training are described in detail in the following subsections. The following paragraphs outline important details about the training of the cascade, and are actually independent of the architecture used (VGG-FCN or residual).

Part detection subnetwork. While [13] uses a per-pixel softmax loss encoding different classes with different numeric levels, in practice, for the human body this is suboptimal because the parts are usually within close proximity to each other, having high chance of overlapping. Therefore, we follow an approach similar to [18] and encode part label information as a set of N binary maps, one for each part, in which the values within a certain radius around the provided ground truth location are set to 1 and the values for the remaining background are set to 0. This way, we thus tackle the problem of having multiple parts in the very same region. Note that the detection network is trained using visible parts only, which is fundamentally different from the previous regression-based approaches [16,23,24].

The radius defining "correct location" was selected so that the targeted body part is fully included inside. Empirically, we determined that a value of 10 px to be optimal for a body size of 200 px of an upright standing person.

We train our body part detectors jointly using pixelwise sigmoid cross entropy loss function:

$$l_1 = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{W}\sum_{j=1}^{H}[p_{ij}^n \log \hat{p_{ij}^n} + (1 - p_{ij}^n)\log(1 - \hat{p_{ij}^n})], \tag{1}$$

where p_{ij}^n denotes the ground truth map of the nth part at pixel location (i, j) (constructed as described above) and $\hat{p_{ij}^n}$ is the corresponding sigmoid output at the same location.

Regression subnetwork. While the detectors alone provide good performance, they lack a strong relationship model that is required to improve (a) accuracy and (b) robustness particularly required in situations where specific parts are occluded. To this end, we propose an additional subnetwork that jointly regresses the location of all parts (both visible and occluded). The input of this subnetwork is a multi-channel representation produced by stacking the N heatmaps produced by the part detection subnetwork, along with the input image (see Fig. 1). This multichannel representation guides the network where to focus and encodes structural part relationships. Additionally, it ensures that our network does not suffer from the problem of regressing occluded part appearances: because the part detection heatmaps for the occluded parts provide low confidence scores, they subsequently guide the regression part of our network to rely on contextual information (provided by the remaining parts) in order to predict the location of these parts.

The goal of our regression subnetwork is to predict the points' location via regression. However, direct regression of the points is a difficult and highly non-linear problem caused mainly by the fact that only one single correct value needs to be predicted. We address this by following a simpler alternative route [16,23], regressing a set of confidence maps located in the immediate vicinity of the correct location (instead of regressing a single value). The ground truth consists of a set of N layers, one for each part, in which the correct location of each part, be it visible or not is represented by Gaussian with a standard deviation of 5px.

We train our subnetwork to regress the location of all parts jointly using the following L2 loss:

$$l_2 = \frac{1}{N}\sum_{n=1}^{N}\sum_{ij}\left\|\widetilde{M}_n(i,j) - M_n(i,j)\right\|^2, \tag{2}$$

where $\widetilde{M}_n(i,j)$ and $M_n(i,j)$ represent the predicted and the ground truth confidence maps at pixel location (i, j) for the nth part, respectively.

3.1 VGG-FCN Part Heatmap Regression

Part detection subnetwork. We based our part detection network architecture on the VGG-16 network [11] converted to fully convolutional by replacing the fully connected layers with convolutional layers of kernel size of 1 [13].

Because the localization accuracy offered by the 32 px stride is insufficient, we make use of the entire algorithm as in [13] by combining the earlier level CNN features, thus reducing the stride to 8 px. For convenience, the network is shown in Fig. 3 and Table 1.

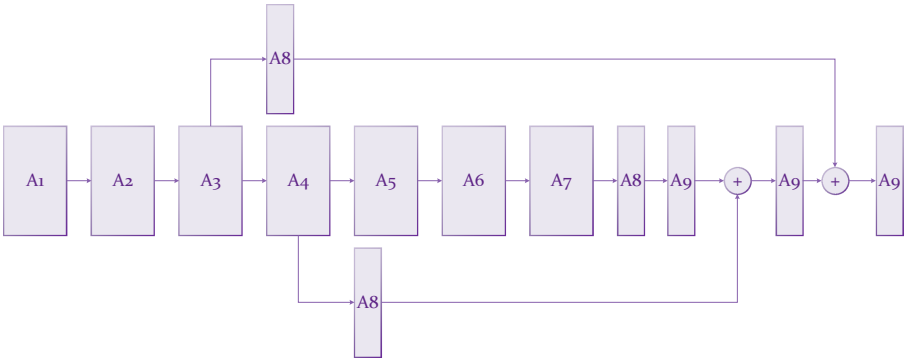

Fig. 3. The VGG-FCN subnetwork used for body part detection. The blocks A1–A9 are defined in Table 1.

Table 1. Block specification for the VGG-FCN part detection subnetwork. Torch notations (channels, kernel, stride) and (kernel, stride) are used to define the conv and pooling layers.

A1	A2	A3	A4	A5	A6	A7	A8	A9
2x conv layer (64, 3×3, 1×1), pooling	2x conv layer (128, 3×3, 1×1), pooling	3x conv layer (256, 3×3, 1×1), pooling	3x conv layer (512, 3×3, 1×1), pooling	3X conv layer (512, 1×1, 1×1), pooling	conv layer (4096, 7×7, 1×1)	conv layer (4096, 1×1, 1×1)	conv layer(16, 1×1, 1×1)	bilinear upsample

Regression subnetwork. We have chosen a very simple architecture for our regression sub-network, consisting of 7 convolutional layers. The network is shown in Fig. 4 and Table 2. The first 4 of these layers use a large kernel size that varies from 7 to 15, in order to capture a sufficient local context and to increase the receptive field size which is crucial for learning long-term relationships. The last 3 layers have a kernel size equal to 1.

Training. For training on MPII, all images were cropped after centering on the person and then scaled such that a standing-up human has height 300 px. All images were resized to a resolution of 380×380 px. To avoid overfitting, we performed image flipping, scaling (between 0.7 and 1.3) and rotation (between -40 and $40°$). Both rotation and scaling were applied using a set of predefined step sizes. Training the network is a straightforward process. We started by first

training the body part detection network, fine-tuning from VGG-16 [11,13] pre-trained on ImageNet [30]. The detectors were then trained for about 20 epochs using a learning rate progressively decreasing from $1e-8$ to $1e-9$. For the regression subnetwork, all layers were initialized with a Gaussian distribution (std = 0.01). To accelerate the training and avoid early divergence we froze the training of the detector layers, training only the subnetwork. We let this train for 20 epochs with a learning rate of 0.00001 and then 0.000001. We then trained jointly both networks for 10 epochs. We found that one can train both the part detection network and the regression subnetwork jointly, right from the beginning, however, the aforementioned approach results in faster training.

For LSP, we fine-tuned the network for 10 epochs on the 1000 images of the training set. Because LSP provides the ground truth for only 14 key points, during fine-tuning we experimented with two different strategies: (i) generating the points artificially and (ii) stopping the backpropagation for the missing points. The later approach produced better results overall. The training was done using the caffe [31] bindings for Torch7 [32].

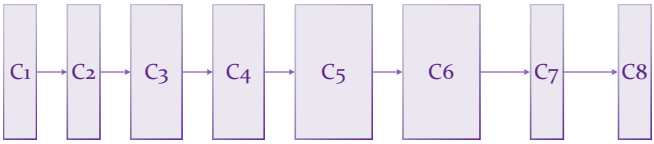

Fig. 4. The VGG-based subnetwork used for regression. The blocks C1–C8 are defined in Table 2.

Table 2. Block specification for the VGG-based regression subnetwork. Torch notations (channels, kernel, stride) and (kernel, stride) are used to define the conv and pooling layers.

C1	C2	C3	C4	C5	C6	C7	C8
conv layer (64, 9×9, 1×1)	conv layer (64, 13×13, 1×1)	conv layer (128, 13×13, 1×1)	conv layer (256, 15×15, 1×1)	conv layer (512, 1×1, 1×1)	conv layer (512, 1×1, 1×1)	conv layer (16, 1×1, 1×1)	deconv layer (16, 8×8, 4×4)

3.2 Residual Part Heatmap Regression

Part detection subnetwork. Motivated by recent developments in image recognition [10], we used ResNet-152 as a base network for part detection. Doing so requires making the network able to make predictions at pixel level which is a relative straightforward process (similar ways to do this are described in [26,33,34]). The network is shown in Fig. 5 and Table 3. Blocks B1-B4 are the same as the ones in the original ResNet, and B5 was slightly modified. We firstly removed both the fully connected layer after B5 and then the preceding average

pooling layer. Then, we added a scoring convolutional layer B6 with N outputs, one for each part. Next, to address the extremely low output resolution, we firstly modified B5 by changing the stride of its convolutional layers from 2 px to 1 px and then added (after B6) a deconvolution [29] layer B7 with a kernel size and stride of 4, that upsamples the output layers to match the resolution of the input. We argue that for our detection subnetwork, knowing the exact part location is not needed. All added layers were initialized with 0 and trained using rmsprop [35].

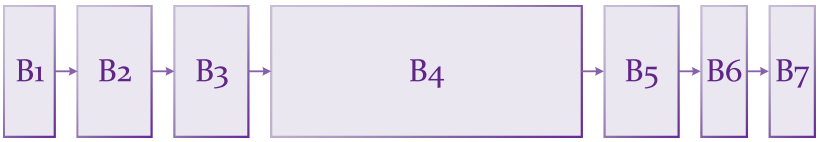

Fig. 5. The architecture of the residual part detection subnetwork. The network is based on ResNet-152 and its composing blocks. The blocks B1–B7 are defined in Table 3. See also text.

Table 3. Block specification for the residual part detection network. Torch notations (channels, kernel, stride) and (kernel, stride) are used to define the conv and pooling layers. The bottleneck modules are defined as in [10].

B1	B2	B3	B4	B5	B6	B7
1x conv layer (64,7 × 7,2 × 2) 1x pooling (3 × 3, 2 × 2)	3x bottleneck modules [(64,1 × 1), (64,3 × 3), (256,1 × 1)]	8x bottleneck modules [(128,1 × 1), (128,3 × 3), (512,1 × 1)]	38x bottleneck modules [(256,1 × 1), (256,3 × 3), (1024,1 × 1)]	3x bottleneck modules [(512,1 × 1), (512,3 × 3), (2048,1 × 1)]	1x conv layer (16,1 × 1, 1 × 1)	1x deconv layer (16,4 × 4, 4 × 4)

Regression subnetwork. For the residual regression subnetwork, we used a (slightly) modified "hourglass network" [19], which is a recently proposed state-of-the-art architecture for bottom-up, top-down inference. The network is shown in Fig. 6 and Table 4. Briefly, the network builds on top of the concepts described in [13], improving a few fundamental aspects. The first one is that extends [13] within residual learning. The second one is that instead of passing the lower level futures through a convolution layer with the same number of channels as the final scoring layer, the network passes the features through a set of 3 convolutional blocks that allow the network to re-analyze and learn how to combine features extracted at different resolutions. See [19] for more details. Our modification was in the introduction of deconvolution layers D5 for recovering the lost spatial resolution (as opposed to nearest neighbour upsampling used in [19]). Also, as in the detection network, the output is brought back to the input's resolution using another trained deconvolutional layer D5.

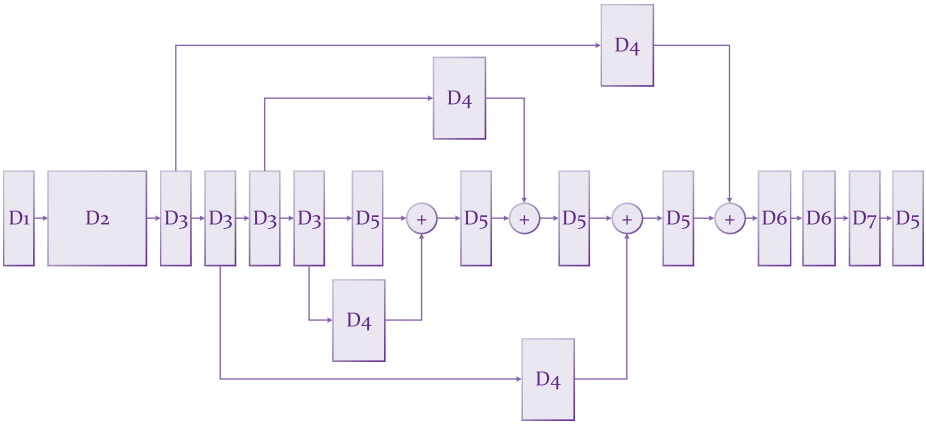

Fig. 6. The "hourglass network" [19] used as the residual regression network. The Blocks D1-D7 are defined in Table 4. See also text.

Table 4. Block specification for the "hourglass network". Torch notations (channels, kernel, stride) and (kernel, stride) are used to define the conv and pooling layers. The bottleneck modules are defined as in [36].

D1	D2	D3	D4	D5	D6	D7
1x conv layer (64, 7×7, 2×2), 1x pooling (2×2,2×2)	3x bottleneck modules	1x maxpooling (2×2, 2×2), 3x bottleneck modules	3x bottleneck modules	1x deconv. layer (256, 2×2, 2×2)	1x conv layer (512, 1×1, 1×1)	1x conv scoring layer (16, 1×1, 1×1)

Training. For training on MPII, we applied similar augmentations as before, with the difference being that augmentations were applied randomly. Also, due to memory issues, the input image was rescaled to 256×256 px. Again, we started by first training the body part detection network, fine-tuning from ResNet-152 [10] pre-trained on ImageNet [30]. The detectors were then trained for about 50 epochs using a learning rate progressively decreasing from $1e-3$ to $2.5e-5$. For the regression "hourglass" subnetwork, we froze the learning for the detector layers, training only the regression subnetwork. We let this train for 40 epochs using a learning rate of $1e-4$ and then $2.5e-5$. In the end, the networks were trained jointly for 50 more epochs. While we experimented with different initialization strategies, all of them seemed to produce similar results. For the final model, all layers from the regression subnetwork were zero-initialized, except for the deconvolution layers, which were initialized using bilinear upsampling filters, as in [13]. The network made use of batch normalization, and was trained with a batch size of 8. For LSP, we follow the same procedure as the one for VGG-FCN, changing only the number of epochs to 30. The network was implemented and trained using Torch7 [32]. The code, along with the pretrained models will be published on our webpage.

4 Results

4.1 Overview

We report results for two sets of experiments on the two most challenging datasets for human pose estimation, namely LSP [21] and MPII [20]. A summary of our results is as follows:

- We show the benefit of the proposed detection-followed-by-regression cascade over a two-step regression approach, similar to the idea described in [16], when implemented with both VGG-FCN and residual architectures.
- We provide an analysis of the different components of our network illustrating their importance on overall performance. We show that stacking the part heatmaps as proposed in our work is necessary for achieving high performance, and that this performance is significantly better than that of the part detection network alone.
- We show the benefit of using a residual architecture over VGG-FCN.
- We compare the performance of our method with that of recently published methods illustrating that both versions of our cascade achieve top performance on both the MPII and LSP data sets.

4.2 Analysis

We carried out a series of experiments in order to investigate the impact of the various components of our architecture on performance. In all cases, training and testing was done on MPII training and validation set, respectively. The results are summarized in Table 5. In particular, we report the performance of

i. the overall part heatmap regression (which is equivalent to "Detection+regression") for both residual and VGG-FCN architectures.
ii. the residual part detection network alone (Detection only).
iii. the residual detection network but trained to perform direct regression (Regression only).
iv. a two-step regression approach as in [16] (Regression+regression), but implemented with both residual and VGG-FCN architectures.

We first observe that there is a large performance gap between residual part heatmap regression and the same cascade but implemented with a VGG-FCN. Residual detection alone works well, but the regression subnetwork provides a large boost in performance showing that using the stacked part heatmaps as input to residual regression is necessary for achieving high performance.

Furthermore, we observe that direct regression alone (case iii above) performs better than detection alone, but overall our detection-followed-by-regression cascade significantly outperforms the two-step regression approach. Notably, we found that the proposed part heatmap regression is also considerably easier to train. Not surprisingly, the gap between detection-followed-by-regression and two-step regression when both are implemented with VGG-FCN is much bigger. Overall, these results clearly verify the importance of using (a) part detection heatmaps to guide the regression subnetwork and (b) a residual architecture.

Table 5. Comparison between different variants of the proposed architecture on MPII validation set, using PCKh metric. The overall part heatmap regression architecture is equivalent to "Detection+regression".

	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Part heatmap regression (Res)	**97.3**	**95.2**	**89.9**	**85.3**	**89.4**	**85.7**	**81.9**	**88.2**
Part heatmap regression (VGG)	95.6	92.2	83.5	78.3	84.5	77.3	70.0	83.2
Detection only (Res)	96.2	91.3	83.4	74.5	83.1	76.6	71.3	82.6
Regression only (Res)	96.4	92.8	84.5	77.3	84.5	79.9	74.0	84.2
Regression+regression (Res)	96.7	93.6	86.1	80.1	88.1	80.5	76.7	85.7
Regression+regression (VGG)	92.8	85.6	77.5	70.4	73.5	69.3	66.5	76.7

Table 6. PCKh-based comparison with state-of-the-art on MPII

	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Part heatmap regression (Res)	**97.9**	95.1	89.9	**85.3**	**89.4**	**85.7**	**81.9**	**89.7**
Part heatmap regression (VGG)	96.8	91.3	82.9	77.5	83.2	74.4	67.5	82.7
Newell et al., arXiv'16 [19]	97.6	**95.4**	**90.0**	85.2	88.7	85.0	80.6	89.4
Wei et al., CVPR'16 [28]	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.5
Insafutdinov et al., arXiv'16 [26]	96.6	94.6	88.5	84.4	87.6	83.9	79.4	88.3
Gkioxary et al., arXiv'16 [37]	96.2	93.1	86.7	82.1	85.2	81.4	74.1	86.1
Lifshitz et al., arXiv'16 [38]	97.8	93.3	85.7	80.4	85.3	76.6	70.2	85.0
Pishchulin et al., CVPR'16 [25]	94.1	90.2	83.4	77.3	82.6	75.7	68.6	82.4
Hu & Ramanan., CVPR'16 [39]	95.0	91.6	83.0	76.6	81.9	74.25	69.5	82.4
Carreira et al., CVPR'16 [40]	95.7	91.7	81.7	72.4	82.8	73.2	66.4	81.3
Tompson et al., NIPS'14 [23]	95.8	90.3	80.5	74.3	77.6	69.7	62.8	79.6
Tompson et al., CVPR'15 [24]	96.1	91.9	83.9	77.8	80.9	72.3	64.8	82.0

Table 7. PCK-based comparison with the state-of-the-art on LSP

	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Part heatmap regression (Res)	96.3	92.2	**88.2**	**85.2**	**92.2**	**91.5**	88.6	**90.7**
Part heatmap regression (VGG)	94.8	86.6	79.5	73.5	88.1	83.2	78.5	83.5
Wei et al., CVPR'16 [28]	**97.8**	92.5	87.0	83.9	91.5	90.8	**89.9**	90.5
Insafutdinov et al., arXiv'16 [26]	97.4	**92.7**	87.5	84.4	91.5	89.9	87.2	90.1
Pishchulin et al.CVPR'16 [25]	97.0	91.0	83.8	78.1	91.0	86.7	82.0	87.1
Lifshitz et al., arXiv'16 [38]	96.8	89.0	82.7	79.1	90.9	86.0	82.5	86.7
Yang et al., CVPR'16 [41]	90.6	78.1	73.8	68.8	74.8	69.9	58.9	73.6
Carreira et al., CVPR'16 [40]	90.5	81.8	65.8	59.8	81.6	70.6	62.0	73.1
Tompson et al., NIPS'14 [23]	90.6	79.2	67.9	63.4	69.5	71.0	64.2	72.3
Fan et al., CVPR'15 [42]	92.4	75.2	65.3	64.0	75.7	68.3	70.4	73.0
Chen & Yuille, NIPS'14 [22]	91.8	78.2	71.8	65.5	73.3	70.2	63.4	73.4

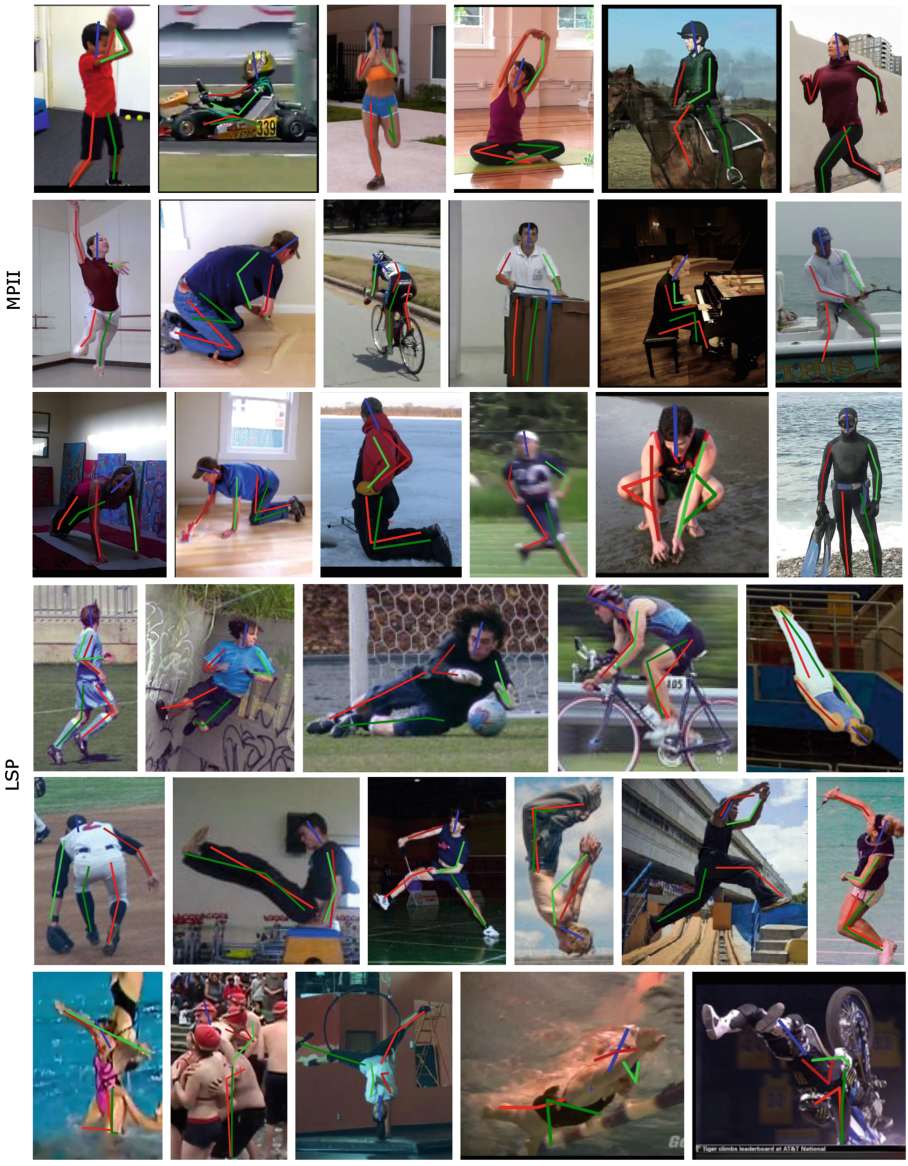

Fig. 7. Examples of poses obtained using our method on MPII (first 3 rows), and LSP (4th and 5th row). Observe that our method copes well with both occlusions and difficult poses. The last row shows some fail cases caused by combinations of extreme occlusion and rare poses.

4.3 Comparison with State-of-the-Art

In this section, we compare the performance of our method with that of published methods currently representing the state-of-the-art. Tables 6 and 7 summarize our results on MPII and LSP, respectively. Our results show that both VGG-based and residual part heatmap regression are very competitive with the latter, along with the other two residual-based architectures [19,26], being top performers on both datasets. Notably, very close in performance is the method of [28] which is not based on residual learning but performs a sequence of 6 CNN regressions, being also much more challenging to train [28]. Examples of fitting results from MPII and LSP for the case of residual part heatmap regression can be seen in Fig. 7.

5 Conclusions

We proposed a CNN cascaded architecture for human pose estimation particularly suitable for learning part relationships and spatial context, and robustly inferring pose even for the case of severe part occlusions. Key feature of our network is the joint regression of part detection heatmaps. The proposed architecture is very simple and can be trained end-to-end, achieving top performance on the MPII and LSP data sets.

Acknowledgement. We would like to thank Leonid Pishchulin for graciously producing our results on MPII with unprecedented quickness.

References

1. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. IJCV **61**(1), 55–79 (2005)
2. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: CVPR (2011)
3. Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Poselet conditioned pictorial structures. In: CVPR (2013)
4. Tian, Y., Zitnick, C.L., Narasimhan, S.G.: Exploring the spatial hierarchy of mixture models for human pose estimation. In: ECCV (2012)
5. Sapp, B., Taskar, B.: Modec: Multimodal decomposable models for human pose estimation. In: CVPR (2013)
6. Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Strong appearance and expressive spatial models for human pose estimation. In: CVPR (2013)
7. Karlinsky, L., Ullman, S.: Using linking features in learning non-parametric part-models. In: ECCV (2012)
8. Dantone, M., Gall, J., Leistner, C., Gool, L.: Human pose estimation using body parts dependent joint regressors. In: CVPR (2013)
9. Ramakrishna, V., Munoz, D., Hebert, M., Bagnell, J.A., Sheikh, Y.: Pose machines: articulated pose estimation via inference machines. In: ECCV (2014)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)

11. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
12. Girshick, R.: Fast R-CNN. In: ICCV (2015)
13. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
14. Toshev, A., Szegedy, C.: DeepPose: Human pose estimation via deep neural networks. In: CVPR (2014)
15. Pfister, T., Simonyan, K., Charles, J., Zisserman, A.: Deep convolutional neural networks for efficient pose estimation ingesture videos. In: ACCV (2014)
16. Pfister, T., Charles, J., Zisserman, A.: Flowing convnets for human pose estimation in videos. In: ICCV (2015)
17. Belagiannis, V., Rupprecht, C., Carneiro, G., Navab, N.: Robust optimization for deep regression. In: ICCV (2015)
18. Zhang, N., Shelhamer, E., Gao, Y., Darrell, T.: Fine-grained pose prediction, normalization, and recognition. arXiv preprint arXiv:1511.07063 (2015)
19. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. arXiv preprint arXiv:1603.06937 (2016)
20. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2D human pose estimation: new benchmark and state of the art analysis. In: CVPR (2014)
21. Johnson, S., Everingham, M.: Clustered pose and nonlinear appearance models for human pose estimation. In: BMVC (2010)
22. Chen, X., Yuille, A.L.: Articulated pose estimation by a graphical model with image dependent pairwise relations. In: NIPS (2014)
23. Tompson, J.J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. In: NIPS (2014)
24. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional networks. In: CVPR (2015)
25. 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 (2015)
26. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., Schiele, B.: DeeperCut: a deeper, stronger, and faster multi-person pose estimation model. arXiv preprint arXiv:1605.03170 (2016)
27. Carreira, J., Agrawal, P., Fragkiadaki, K., Malik, J.: Human pose estimation with iterative error feedback. arXiv preprint arXiv:1507.06550 (2015)
28. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: CVPR (2016)
29. Zeiler, M.D., Taylor, G.W., Fergus, R.: Adaptive deconvolutional networks for mid and high level feature learning. In: 2011 International Conference on Computer Vision, pp. 2018–2025. IEEE (2011)
30. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR (2009)
31. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
32. Collobert, R., Kavukcuoglu, K., Farabet, C.: Torch7: a matlab-like environment for machine learning. In: BigLearn, NIPS Workshop, Number EPFL-CONF-192376 (2011)
33. Wu, Z., Shen, C., Hengel, A.V.D.: High-performance semantic segmentation using very deep fully convolutional networks. arXiv preprint arXiv:1604.04339 (2016)

34. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. arXiv preprint arXiv:1605.06409 (2016)
35. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA: Neural Netw. Mach. Learn. **4**(2) (2012)
36. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. arXiv preprint arXiv:1603.05027 (2016)
37. Gkioxari, G., Toshev, A., Jaitly, N.: Chained predictions using convolutional neural networks. arXiv preprint arXiv:1605.02346 (2016)
38. Lifshitz, I., Fetaya, E., Ullman, S.: Human pose estimation using deep consensus voting. arXiv preprint arXiv:1603.08212 (2016)
39. Hu, P., Ramanan, D.: Bottom-up and top-down reasoning with convolutional latent-variable models. arXiv preprint arXiv:1507.05699 (2015)
40. Carreira, J., Agrawal, P., Fragkiadaki, K., Malik, J.: Human pose estimation with iterative error feedback. arXiv preprint arXiv:1507.06550 (2015)
41. Yang, W., Ouyang, W., Li, H., Wang, X.: End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In: CVPR (2016)
42. Fan, X., Zheng, K., Lin, Y., Wang, S.: Combining local appearance and holistic view: dual-source deep neural networks for human pose estimation. In: CVPR (2015)