This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

End-to-End Learning of Deformable Mixture of Parts and Deep Convolutional Neural Networks for Human Pose Estimation

Wei Yang Wanli Ouyang* Hongsheng Li Xiaogang Wang
Department of Electronic Engineering, The Chinese University of Hong Kong
{wyang, wlouyang, hsli, xgwang}@ee.cuhk.edu.hk

Abstract

Recently, Deep Convolutional Neural Networks (DCNNs) have been applied to the task of human pose estimation, and have shown its potential of learning better feature representations and capturing contextual relationships. However, it is difficult to incorporate domain prior knowledge such as geometric relationships among body parts into DCNNs. In addition, training DCNN-based body part detectors without consideration of global body joint consistency introduces ambiguities, which increases the complexity of training. In this paper, we propose a novel end-to-end framework for human pose estimation that combines DCNNs with the expressive deformable mixture of parts. We explicitly incorporate domain prior knowledge into the framework, which greatly regularizes the learning process and enables the flexibility of our framework for loopy models or tree-structured models. The effectiveness of jointly learning a DCNN with a deformable mixture of parts model is evaluated through intensive experiments on several widely used benchmarks. The proposed approach significantly improves the performance compared with state-of-the-art approaches, especially on benchmarks with challenging articulations.

1. Introduction

Articulated human pose estimation is one of the fundamental tasks in computer vision. It solves the problem of localizing human parts in images, and has many important applications such as action recognition [45], clothing parsing [49, 50], and human tracking [6]. The main challenges of this task are articulation, occlusion, cluttered background, and variations in clothing and lighting. Recently, state-of-the-art performance of human pose estimation has been achieved with Deep Convolutional Neural Networks (DCNNs) [42, 41, 40, 4, 3, 16, 15, 10, 7]. These approaches primarily fall into two categories: 1) regressing heat-maps of each body part location with DCNNs [16, 40]; 2) learning deep structured output to further model the relationships among body joints [15, 41].

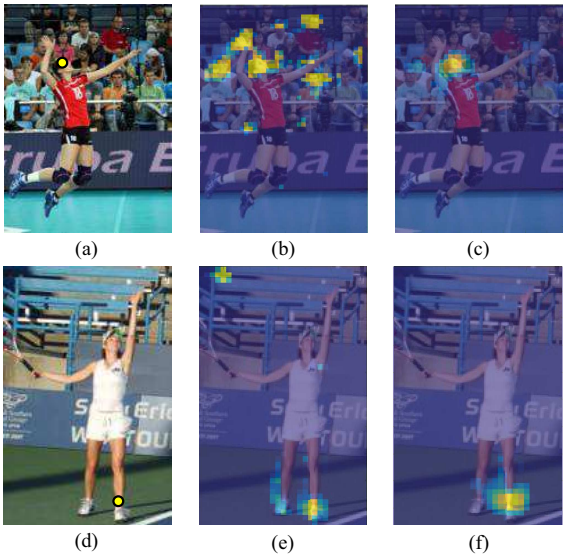

Figure 1. Motivation. Left: Ground-truth locations of *head* (a) and *ankle* (d). Middle: The noisy heat-maps predicted by conventional DCNN during the training stage. Right: With body joint consistency considered by the proposed framework, the heat-maps are better predicted.

DCNN-based heat-map regression models have shown the potential of learning better feature representations. However, geometric constraints among body parts, which are essential to ensure the joint consistency, are usually missed in training the DCNNs. As a consequence, during the training stage, these approaches may produce many imperfect results, as shown in Figure 1 (b, e). For example, regions with high response to *head* in Figure 1 (b) are heads of unannotated persons, which are reasonable but will be treated as false positives in learning the DCNN. Errors on these regions will be back propagated to penalize the features correspond to *head* detection, which is inappropriate. We observe that this problem could be addressed by considering global joint consistency during the training stage: the unannotated persons do not have their full bodies appearing in the image, hence can be suppressed when considering the

*Wanli Ouyang is the corresponding author.

full pose configuration, as shown in Figure 1 (c). Another example is shown in Figure 1 (e), where the false positive region for *ankle* at the background (top-left corner) will be treated as the hard negative for learning the DCNN. It is no longer a hard negative when the structure of full body is considered, as shown in Figure 1 (f).

Deep structured output learning has attracted considerable attention recently, and has shown promising results in tasks such as semantic segmentation [2], scene parsing [23], object detection [44], and depth estimation [24]. For human pose estimation, recent studies combine DCNNs with fully-connected Markov Random Field [41] or weakly spatial histogram over body part locations [15] to exploit structural constraints between body joint locations. However, the parameter space of learning spatial constraints with convolutional kernels [41] is too large, which makes the learning difficult. Additionally, for persons with a large range of possible poses, *e.g.*, the *head* is not always above the *shoulder*, these approaches will be less effective.

In vision community, domain knowledge has been proved effective in many tasks such as object recognition [11], detection [27, 14, 20, 47, 28], and person re-identification [48]. For pose estimation, the deformable mixture of parts model [51, 30] uses domain knowledge and designs a deformable model to constrain the spatial configuration between a pair of parts with multiple appearance mixtures. By using a DCNN for feature extraction together with deformable model for spatial constraints, Chen and Yuille [4] achieve a significant improvement. However, features and spatial constraints are still learned separately. Therefore, the problem in learning DCNNs as shown in Figure 1 still exists.

In this paper, we propose to incorporate the DCNN and the expressive mixture of parts model into an end-to-end framework. This enables us to predict the body part locations with the consideration of global pose configurations during the training stage, hence our framework is able to predict heat-maps with less false positives, as shown in Figure 1 (c), (f). Therefore, jointly learning the DCNN with the deformable model makes the feature learning more effective in handling the negative samples that are difficult when taking the full body pose into account. In addition, we explicitly incorporate human pose priors including body part mixture types and standard quadratic deformation constraints into our model. This greatly reduces the parameters to be learned compared with the use of convolution or histogram, and still keeps the flexibility of our framework in building loopy models or tree-structured models.

We show the efficiency of the proposed framework on three widely used pose estimation benchmarks: the LSP [18] dataset, the FLIC [35] dataset and the Image Parse [33] dataset. Our approach improves the state-of-the-art on all these datasets. The generalization ability of our framework is also validated by cross-dataset experiments on the Image Parse dataset.

The main contributions of this work are three folds:

- We design a novel message passing layer, which is flexible to build tree-structured models or loopy models with appearance mixtures.
- An end-to-end deep CNN framework for human pose estimation is proposed. By jointly learning DCNNs with deformable mixture of parts models, global pose consistency is considered. Hence our framework is able to reduce the ambiguity and mine hard negatives effectively when learning features and part deformation.
- Domain knowledge is incorporated into our framework. Through quadratic deformation constraints, we reduce the parameter space in modeling the spatial and the appearance mixture relationships among parts.

2. Related Work

In literature, part-based models have been widely used to model the articulated relationships between rigid human body parts. Specifically, tree-structured pictorial structures [13] have been made tractable together with the development of general distance transform [11], and is popular in human pose estimation [39, 46, 9, 18, 29, 30, 18, 19, 35]. For example, Yang and Ramanan [51] proposed a flexible mixture model to capture contextual co-occurrence relations between parts. Johnson and Everingham [19] used a cascade of body parts detectors to obtain mixture models on the full model scale. Pishchulin *et al*. [30] extended part-based model based on rigid body parts with Poselet [1] priors. Despite efficient inference and impressive successes, tree-structured models suffer from the double-counting problem, which often happens to limbs.

To overcome the limited expressiveness of tree-structured models, there have been a lot of efforts that focused on constructing more expressive models [17, 34, 12, 46, 38]. For example, symmetry of appearance between limbs has been considered in [34, 38]. Ferrari *et al*. [12] proposed repulsive edges between opposite-sided arms to overcome double counting in upper-body pose estimation. These strong pose priors, however, may overfit to the statistics of some particular datasets [43]. To consider higher-order part relationships beyond primitive rigid parts, Wang *et al*. [46] incorporated hierarchical poselets for human parsing. In video pose estimation, Cherian *et al*. [5] designed temporal links between body parts to address inconsistency between parts that across the sequences. These methods achieved better expressiveness by loopy models. Inference on such models, however, requires approximate methods such as integer programs [17], integer quadratic programs [34], or loopy belief propagation [36]. Moreover, the above mentioned approaches are based on hand-crafted features (*e.g.*, HOG [8] and Shape Context [25]), and may

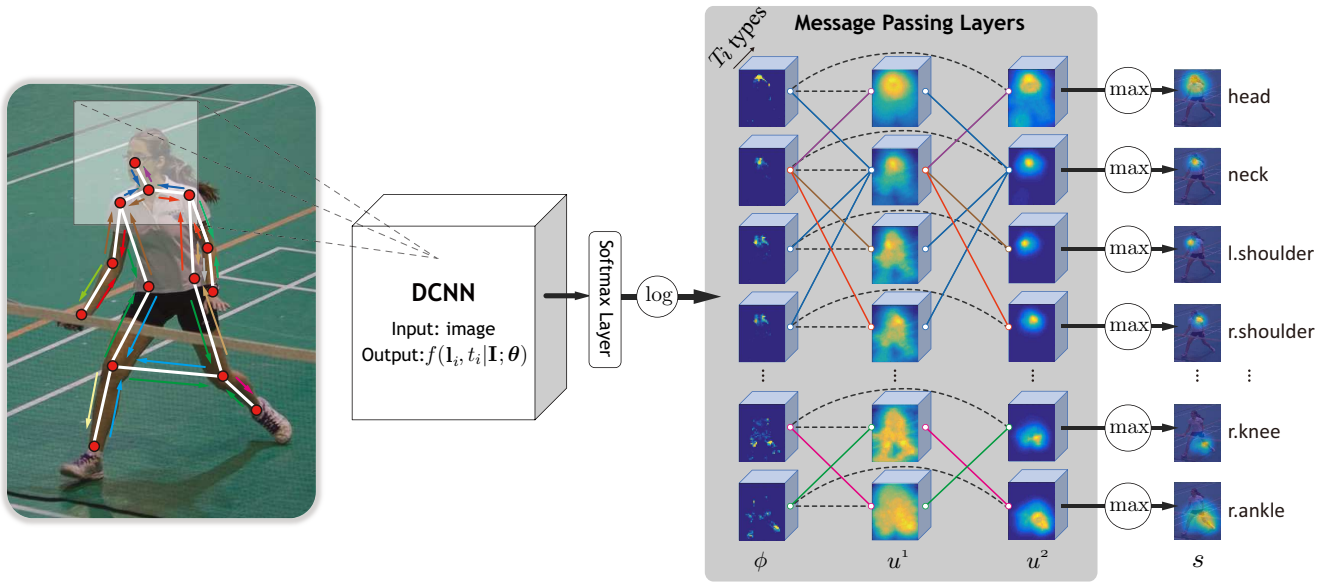

Figure 2. Illustration of the proposed framework. (a) visualizes a loopy model, where nodes (red circles) specify the positions and mixture types of body parts, and edges (white lines) indicate the relationships between parts. During inference, a node sends a message to each of its neighbors and receives messages from each neighbor (indicated by arrows). The proposed framework can be viewed as two components: (b) a front-end DCNN for learning feature representations of body parts and (c) message passing layers for conducting inference and learning on mixture of parts with deformation constraints between parts. Specifically, each message passing layer performs one iteration of message passing in a forward pass. (d) are predicted heat-maps for parts. Please refer to the text for the notations.

be limited by the representation ability.

Deep Models for Human Pose Estimation: Recently, deep models have been successfully applied in human pose estimation. Ouyang *et al*. [26] proposed a multi-source deep model for constructing the non-linear representation from multiple information sources. DeepPose [42] estimated body part locations by learning a regressor based on DCNNs in a holistic manner. However, this method suffered from inaccuracy in the high-precision regions. Jain *et al*. [15] used a multi-resolution DCNN and adopted motion features to improve the accuracy of body parts localization. Tompson *et al*. [40] proposed spatial pooling to overcome the reduced localization accuracy caused by pooling operations. Chen and Yuille [4] used a DCNN to learn the conditional probabilities for the presence of parts and their spatial relationships. They further proposed flexible compositions of object parts [3] to handle significant occlusions in images, and showed state-of-the-art results. However, part detection scores and detectors are fixed in [4, 3] but are not fixed in our model. The approaches in [4, 3] learned part detectors and spatial relationships independently, while we jointly learned them. Besides, our model learns global pose configuration and is not constrained to tree models while tree models were used in [4, 3].

To capture contextual relationships directly within DCNNs, some recent studies explored to combine DCNNs with Conditional Random Fields (CRFs), Markov Random Fields (MRFs), or Deformable Part Models, and showed promising results on several applications, such as depth estimation [24], semantic segmentation [22], and object detection [14, 44, 27]. For pose estimation, Tompson *et al*. [41] jointly trained a multi-scale DCNN with an approximate MRF, which is to model the spatial relationships between body parts. Our approach is different from this approach in the following aspects. First, their model has difficulty in learning effective spatial relationships on datasets with large pose variations. Our method addresses this problem by using appearance mixtures. Second, to cover the largest body joint displacement, very large 128×128 convolution kernels were used in [41]. Hence its parameter space is very large, and it is hard to learn when body parts are with large variations in relative locations. We take the body part articulation property into account and model spatial relations by mixture of deformation constraints, which are only 4 parameters for each pair of mixture-of-parts. Therefore, our model is better in handling large range of possible poses.

Parameterized deformation constraints are also jointly learned with DCNN for pedestrian detection in [27] and object detection in [14]. However, these approaches do not consider the appearance mixtures, hence are limited to body part variances, while we learn deformation constraints taking the appearance mixture into account to handle the variation. In addition, only the star model is considered in [27, 14] while our approach is flexible for star models, tree-structured models or loopy models.

3. The Model

We formulate the human pose estimation problem by using a graph. Let $G = (V, E)$ denote a graph with vertices V specifying the positions as well as the mixture types of body parts, and edges $E \subseteq V \times V$ indicating the spatial relationships between parts. Let $K = |V|$ be the number of parts, and $i \in \{1, \cdots, K\}$ be the ith part. Given an image $\mathbf{I}$, we denote the pixel locations of parts by $\mathbf{l} = \{\mathbf{l}_i\}_{i=1}^K = \{(x_i, y_i)\}_{i=1}^K$, and denote the mixture type of different spatial relationships by $\mathbf{t} = \{t_i\}_{i=1}^K$, where $t_i \in \{1, \cdots, T_i\}$. The full score of a pose configuration given an input image $\mathbf{I}$ is as follows:

$$F(\mathbf{l}, \mathbf{t}|\mathbf{I}; \boldsymbol{\theta}, \boldsymbol{w}) = \sum_{i \in V} \phi(\mathbf{l}_i, t_i|\mathbf{I}; \boldsymbol{\theta}) + \sum_{(i,j) \in E} \psi(\mathbf{l}_i, \mathbf{l}_j, t_i, t_j|\mathbf{I}; \boldsymbol{w}_{i,j}^{t_i,t_j}), \quad (1)$$

where $\boldsymbol{\theta}$ and $\boldsymbol{w} = \{\boldsymbol{w}_{i,j}^{t_i,t_j}\}$ are parameters of the model.

Part Appearance Terms: Given an image patch located at $\mathbf{l}_i$, the unary terms $\phi(\mathbf{l}_i, t_i|\mathbf{I}; \boldsymbol{\theta})$ provide local confidence of the appearance of part i with mixture type t_i, which are defined as the log probability,

$$\phi(\mathbf{l}_i, t_i|\mathbf{I}; \boldsymbol{\theta}) = \log p(\mathbf{l}_i, t_i|\mathbf{I}; \boldsymbol{\theta}) = \log \sigma(f(\mathbf{l}_i, t_i|\mathbf{I}; \boldsymbol{\theta})). \quad (2)$$

The probability $p(\mathbf{l}_i, t_i|\mathbf{I}; \boldsymbol{\theta})$ is given by the softmax function $\sigma(\cdot)$, which is to predict the probability of the ith part at location $\mathbf{l}_i$ with type t_i in image $\mathbf{I}$. $f(\mathbf{l}_i, t_i|\mathbf{I}; \boldsymbol{\theta})$ is modeled by the front-end DCNN to predict a score for part i located at $\mathbf{l}_i$ with type t_i, where $\boldsymbol{\theta}$ are its parameters. Appearance terms $\phi(\mathbf{l}_i, t_i|\mathbf{I}; \boldsymbol{\theta})$ are obtained from the DCNN through a classification layer as shown in Figure 2 (b).

Spatial Relationship Terms: The pairwise terms model the spatial compatibility of two neighboring parts i and j. We define the pairwise terms as follows:

$$\psi(\mathbf{l}_i, \mathbf{l}_j, t_i, t_j|\mathbf{I}; \boldsymbol{w}_{i,j}^{t_i,t_j}) = < \boldsymbol{w}_{i,j}^{t_i,t_j}, d(\mathbf{l}_i - \mathbf{l}_j) > . \quad (3)$$

Here we incorporate standard quadratic deformation constraints into our model, where $d(\mathbf{l}_i - \mathbf{l}_j)$ is deformation feature defined as $d(\mathbf{l}_i - \mathbf{l}_j) = [\Delta x \quad \Delta x^2 \quad \Delta y \quad \Delta y^2]^T$, and $\Delta x = x_i - x_j$ and $\Delta y = y_i - y_j$ are the relative locations of part i with respect to part j, and $\boldsymbol{w}_{i,j}^{t_i,t_j}$ are the 4-dimensional deformation weights to encode pairwise terms for mixture types (t_i, t_j).

4. Inference

Inference is to find the optimal part locations $\mathbf{l}^*$ and mixture types $\mathbf{t}^*$ that maximize the score function $F(\mathbf{l}, \mathbf{t}|\mathbf{I}; \boldsymbol{\theta}, \boldsymbol{w})$ as follows:

$$(\mathbf{l}^*, \mathbf{t}^*) = \arg\max_{\mathbf{l}, \mathbf{t}} F(\mathbf{l}, \mathbf{t}|\mathbf{I}; \boldsymbol{\theta}, \boldsymbol{w}). \quad (4)$$

An overview of the inference procedure is demonstrated in Figure 2 (b-d). Given an image, the heat-maps $f(\mathbf{l}_i, t_i|\mathbf{I}; \boldsymbol{\theta})$ of each part are computed by a forward pass through the DCNN. Then the log probability $\phi(\mathbf{l}_i, t_i)$ of each part with each type is obtained from $f(\mathbf{l}_i, t_i|\mathbf{I}; \boldsymbol{\theta})$ through a softmax layer and a logarithm layer. Taking $\phi(\mathbf{l}_i, t_i)$ as input, we propose to pass messages in neural networks by designing a novel message passing layer, which is flexible to build tree-structured models or loopy models.

4.1. Message Passing

We first give a brief review of message passing on the proposed model. Max-sum algorithm has been widely used for inferring the best configuration in graphical models. Although the max-sum algorithm is only an approximation and the convergence cannot be guaranteed on loopy structures, it still provided excellent experimental results [36].

At each iteration, a vertex sends a message to its neighbors and receives messages from its neighbors. We denote $m_{ij}(\mathbf{l}_j, t_j)$ as the message sent from part i to part j, and $u_i(\mathbf{l}_i, t_i)$ as the belief of part i, then the max-sum algorithm updates the messages and beliefs as follows:

$$m_{ij}(\mathbf{l}_j, t_j) \leftarrow \alpha_m \max_{\mathbf{l}_i, t_i} \left(u_i(\mathbf{l}_i, t_i) + \psi(\mathbf{l}_i, \mathbf{l}_j, t_i, t_j)\right), \quad (5)$$

$$u_i(\mathbf{l}_i, t_i) \leftarrow \alpha_u(\phi(\mathbf{l}_i, t_i) + \sum\nolimits_{k \in \mathbb{N}(i)} m_{ki}(\mathbf{l}_i, t_i)), \quad (6)$$

where α_m and α_u are normalization terms, and $\mathbb{N}(i)$ denotes the set of neighbors of part i. To simplify the notation, we omit model parameters here. Figure 2 (a) gives a visualization of this message passing procedure.

The algorithm starts with all message vectors initialized to constant functions. The normalization terms in Eq.(5-6) are not necessary. However, we find that they help to make the inference more stable in practice.

Maximum Score Assignment: Suppose the algorithm converges at the Nth iteration, then the belief for each location and each type $(\mathbf{l}_i, t_i)$ is the approximation of the maximum score function. Hence we can obtain the max-sum assignment $(\mathbf{l}_i^*, t_i^*)$ by

$$(\mathbf{l}_i^*, t_i^*) = \arg\max_{\mathbf{l}_i, t_i} u_i^*(\mathbf{l}_i, t_i), \quad \forall i \in \{1, \cdots, K\}, \quad (7)$$

where $u_i^*(\mathbf{l}_i, t_i)$ is the belief computed in the last iteration, and $(\mathbf{l}_i^*, t_i^*)$ is the solution for the maximum score in Eq.(4).

Special Case: Tree-Structured Model: For tree structures, exact inference can be performed efficiently by one pass of dynamic programming, which is a special case of max-sum algorithm by passing messages from leaves to a chosen root node. By keeping track of indexes of $\arg\max_{\mathbf{l}_k^*, t_k^*}$ for each pass, the maximum score assignment can be obtained by backtracking from the root node to the leaves. This procedure is also known as *Verberti decoding*, and has been widely used in previous pose estimation works with tree-structured models [51, 52, 4, 3].

4.2. The Message Passing Layer in Neural Networks

In literature, there are mainly two possible ways to organize the message passing schedule. The *flooding schedule*

Figure 3. From left to right, we show the estimated poses generated by the first, the second, and the third message passing layer, respectively. Intuitively, a part could receive messages from further parts as the number of message passing layers increases, which may result in better results.

simultaneously passes messages across every link in both directions at each time step, while the *serial schedule* passes one message at each time. By following the flooding schedule, we integrate the procedure introduced in Eq.(5-6) into the network by designing a novel message passing layer.

As shown in Figure 2 (c), each node sends a message to each of its neighbors simultaneously (*solid lines* in Figure 2 (c)), and the belief of each part is updated by summing its unary potential $\phi(\mathbf{l}_i, t_i)$ (*dashed lines*) and the incoming messages. The belief $u_i(*, t_i)$ corresponds to a feature map with mixture type t_i for the ith part in the message passing layer. After convergence, the optimum pose estimation is obtained by selecting the location and type with maximum belief for each part, as in Eq. (7).

Although message passing may need several iterations to converge, we observe that a cascade of three message passing layers is enough to produce satisfactory results in practice. Examples of estimated poses from different message passing layers are visualized in Figure 3. Intuitively, more message passing layers (*i.e.*, more iterations in the max-sum algorithm) lead to better results. We take the tree-structured model shown in Figure 2 (a) as an example: In the first round, *neck* only receives messages from its neighbors *head* and *shoulders*. In the second round, however, *neck* could receive messages from parts a step further such as *elbows* and *hips*.

Computation: The computational complexity of message passing is $O(L^2T^2)$ for L possible part locations and T mixture types. Since our pairwise terms are quadratic functions of location $\mathbf{l}_i$ and $\mathbf{l}_j$, we can accelerate the maximization over $\mathbf{l}_i$ by employing the generalized distance transforms [11], and the computational complexity of updating one message is reduced to $O(LT^2)$.

5. Learning

Several recent works produced heat-maps of body parts by using fully convolutional networks [41, 40, 4, 3]. Some approaches train the fully convolutional networks with full images [41, 40]. Others first train a DCNN from local image patches, then the learned DCNNs are fixed [4, 3]. We initialize the DCNN by pretraining from local image patches with mixture type labels, and then jointly learn the DCNN and the deformable model by finetuning from full images.

Pretraining with Part Mixture Types: Pretraining utilizes part mixture types as supervision to train the front-end DCNN that serves as part detectors. Existing human pose datasets are annotated with body part locations $\mathbf{l}$, but without part mixture type labels $\mathbf{t}$. We define the part types as the different relative locations clusters of a part with respect to its neighboring parts. Let $\mathbf{r}_{ij}$ be the relative position from part i to its neighboring part j, we cluster the relative position $\mathbf{r}_{ij}$ over the training set into T_i clusters. Each cluster corresponds to a set of part instances that share with similar relative locations. The type label for each part can be derived by cluster membership, and serves as an extra supervision for pretraining the front-end DCNN. The mixture types obtained from locations are strongly correlated to appearance of parts. For example, *horizontal arm* is one part type and *vertical arm* is another type – they are different in pose configuration and appearance. We tried to remove pretraining, but the net failed to converge to satisfactory training loss.

Finetuning of the Full Model: We finetune the unified model by the hinge loss function. Suppose there are N message passing layers, the final heat-map for each part is obtained as follows:

$$s_i(\mathbf{l}_i) = \max_{t_i}(u_i^N(\mathbf{l}_i, t_i) + b_i), \tag{8}$$

where b_i is the bias. Denote the ground-truth location of part i by $\tilde{\mathbf{l}}_i$. The ground-truth heat-map for part i is

$$\tilde{s}_i(\mathbf{l}_i) = \begin{cases} +1, & \text{if } ||\mathbf{l}_i - \tilde{\mathbf{l}}_i||_\infty \leq \delta; \\ -1, & \text{otherwise.} \end{cases} \tag{9}$$

where δ is a constant threshold. This produces the ground-truth heat-map with a box centered at location $\tilde{\mathbf{l}}_i$: the ground-truth heat-map has value 1 inside the box, and value -1 outside the box.

Ideally, we not only hope the predicted part locations to be close to ground-truth locations, but also hope the maximum response of each part in Eq.(8) to be higher than a threshold. This motivates us to train our model in a max-margin manner.

Given the ground-truth heat-map $\tilde{s}_i(\mathbf{l}_i)$ and the predicted heat-map $s_i(\mathbf{l}_i)$ of part i, the loss function is

$$J(\mathbf{l}, \mathbf{t}) = \frac{1}{KL}\sum_{i=1}^{K}\sum_{\mathbf{l}_i=1}^{L}\max(0, 1 - \tilde{s}_i(\mathbf{l}_i)\cdot s_i(\mathbf{l}_i)), \tag{10}$$

where $\max(0, 1 - \tilde{s}_i(\mathbf{l}_i)\cdot s_i(\mathbf{l}_i))$ is the hinge loss at location $\mathbf{l}_i$ and J is the overall loss for all parts.

We apply stochastic gradient descent to learn the parameters. First, we compute the subgradients w.r.t. the final heat-map for each part as,

$$\frac{\partial J}{\partial s_i(\mathbf{l}_i)} = \begin{cases} -\tilde{s}_i(\mathbf{l}_i), & \text{if } \tilde{s}_i(\mathbf{l}_i)\cdot s_i(\mathbf{l}_i) < 1; \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Then the partial derivatives w.r.t. each layer can be computed by using the standard backpropagation algorithm. For example, the partial derivative of the deformation weights $\boldsymbol{w}$ are computed as,

$$\frac{\partial J}{\partial \boldsymbol{w}_{k,i}^{t_k,t_i}} \propto \sum_{n=1}^{N} \sum_{\mathbf{l}_i,t_i} \frac{\partial J}{\partial u_i^n(\mathbf{l}_i,t_i)} d(\mathbf{l}_k - \mathbf{l}_i), \tag{12}$$

where $k \in \mathbb{N}(i)$. Recall that $d(\mathbf{l}_k - \mathbf{l}_i)$ is the standard quadratic deformation features defined in Eq.(3).

6. Experiments

In this section, we present experimental settings, experimental results, and diagnostic analysis.

6.1. Experimental Settings

Datasets: We evaluate the proposed methods on three well known public pose estimation benchmarks: The Leeds Sports Poses (LSP) [18] dataset, the Frames Labeled in Cinema (FLIC) [35] dataset, and the Image Parse (PARSE) [33] dataset. (i) LSP contains 1000 training and 1000 testing images from sports activities with challenging articulations. Each person is roughly 150 pixels in height with 14 joints full-body annotations. (ii) FLIC consists of 3987 training and 1016 testing images collected from popular Hollywood movies with diverse appearances and poses. Each person has 10 upper-body joints annotated. (iii) PARSE contains 305 images of highly articulated human poses with full body annotations. The PARSE dataset is only used for the evaluation of cross-dataset generalization: we directly apply the model trained on the LSP dataset to the 205 test images of the PARSE dataset. To compare with previous methods, we use Observer-Centric annotations on both the LSP dataset and the FLIC dataset, and Person-Centric annotations on the PARSE dataset.

Data Augmentation: To reduce overfitting, we augment the training data by rotating through 360 degrees for every 9 degrees. Then we mirror the images horizontally. Note that this also increases the training patches of body parts with different mixture types. The negative samples are randomly cropped from the negative images of the INRIA Person dataset [8]. We randomly select 5% of the training data as validation set when we pretrain the front-end DCNN, and these data are further used to finetune the full model.

Previous works [52, 4] observed that adding midway parts between neighboring annotated parts helps to reduce foreshortening and improves overall performance. Hence we interpolate midway parts on both the LSP and the FLIC datasets, which results in $K = 26$ and 18 parts respectively.

Evaluation Measure: Two widely used evaluation metrics, *i.e.*, Percentage of Correct Parts (PCP) and Percentage of Detected Joints (PDJ), are used for comparison. PCP measures the rate of correctly detected limbs: a limb is considered as correctly detected if the distances between detected limb endpoints and groundtruth limb endpoints are within half of the limb length. However, different interpretations of PCP lead to different results. Hence we adopt the strict PCP as discussed in [52, 4] for fair comparison[1].

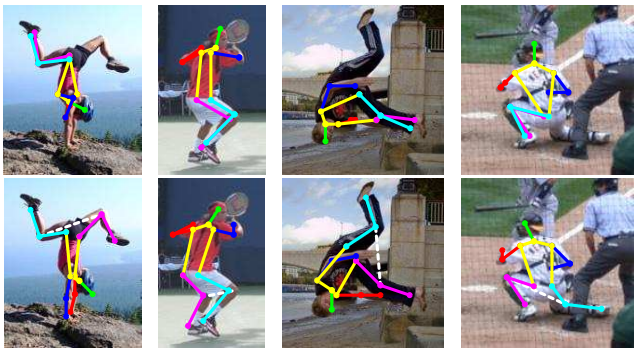

Figure 4. The double-counting problem in tree-structured models (the 1st row), could be reduced by introducing additional pairwise constraints (indicated by white dashed lines in the 2nd row).

PDJ is introduced in [35] as a complementary evaluation metric of PCP, as PCP penalizes short limbs. PDJ measures the detection rate of joints, where a joint is considered as detected if the distance between the predicted joint and the ground-truth joint is less than a fraction of torso diameter. The torso diameter is defined as the distance between the left shoulder and the right hip of each ground-truth pose.

Front-End DCNN Architecture: We investigate two DCNN architectures in this paper. The first one (ChenNet) is based on [4], which consists of five convolution layers, two max-pooling layers and three fully-connected layers, and is trained from random initialization. The second one is the 16-layer VGG architecture pretrained on the ImageNet dataset [37]. To reduce computation, we resize the original input of VGG from 224×224 to 112×112, and remove the last pooling layer. This also improves the spatial localization accuracy with fewer pooling operations. The number of mixture types is set as $T_i = 13$ for all parts $i \in \{1, \cdots, K\}$. Both architectures produce $\sum_{i=1}^{K} T_i + 1$ heat-maps as the input of the message passing layers, which include one *background* heat-map. If the mixture number T_i is reduced to 11, PCP drops by 1%. The stride size is 4 for ChenNet and is 16 for VGG, hence the heat-maps size is $1/4$ of the input image size for ChenNet and $1/16$ for VGG.

Connections Among parts: The tree-structured models are visualized in Figure 7. Based on the tree structure, the loopy structured models add edges between knees on the LSP dataset, the structure of which is visualized in the second row of Figure 4. On the FLIC dataset, we only conduct experiments with tree-structured model. We perform exact inference with tree-structured models. If not specified, three message passing layers are used for loopy models.

[1] We use a widely used implementation of strict PCP available at http://human-pose.mpi-inf.mpg.de/ to evaluate our results.

Parameter Settings: During the pretraining stage, each image is normalized to 150 pixels in body height. Patch size is set to 36×36, which is able to cover sufficient context. By changing the patch size to 0.8 and 1.2 times of the original scale, PCP is reduced by 6.2% and 0.5% on the LSP dataset with VGG architecture. We keep the batch size as 512, and the learning rates are initialized as 0.005 and 0.001 for ChenNet and VGG, respectively. The dropout rate is set as 0.5. We drop the learning rate by a factor of 10 for every 5 epochs, and the front-end DCNN is trained for 15 epochs. δ in Eq. (9) is set as $1/5$ of the patch size. The change of δ from $1/5$ to $1/3$ results in less than 1% strict PCP variation.

During the joint finetuning stage, the batch size is 5, and the learning rate is relatively low at 0.0001 for both ChenNet and VGG. The dropout rate is increased to 0.6 to avoid overfitting. Since the parameters of the DCNN are well initialized during pretrianing, and the deformation weights are shared across different message passing layers and are relatively few, finetuning the model for 1 epoch already provides satisfactory results.

6.2. Experimental Results

Method	Torso	Head	U. arms	L. arms	U. legs	L. legs	Mean
Yang&Ramanan [51]	84.1	77.1	52.5	35.9	69.5	65.6	60.8
Pishchulin *et al*. [29]	87.4	77.4	54.4	33.7	75.7	68.0	62.8
Eichner&Ferrari [9]	86.2	80.1	56.5	37.4	74.3	69.3	64.3
Kiefel&Gehler [21]	84.3	78.3	54.1	28.3	74.5	67.6	61.2
Pose Machines [32]	88.1	80.4	62.8	39.5	79.0	73.6	67.8
Ouyang *et al*. [26]	88.6	84.3	61.9	45.4	77.8	71.9	68.7
Pishchulin *et al*. [30]	88.7	85.1	61.8	45.0	78.9	73.2	69.2
DeepPose [42]	-	-	56	38	77	71	-
Chen&Yuille [4]	92.7	**87.8**	69.2	55.4	82.9	77.0	75.0
Ours-ChenNet-Unary	62.1	62.3	35.8	18.2	48.5	38.2	40.6
Ours-ChenNet-T	94.8	82.4	75.0	62.4	85.3	79.2	78.1
Ours-ChenNet-LG-Ind	93.0	82.1	70.6	55.4	82.1	75.3	74.2
Ours-ChenNet-LG	95.0	83.5	75.0	61.9	86.9	79.8	78.6
Ours-VGG-Unary	83.4	69.0	53.5	34.9	72.2	63.5	60.1
Ours-VGG-T	96.2	83.4	78.7	65.8	87.9	81.1	80.7
Ours-VGG-LG-MP1	96.3	84.3	78.4	66.3	87.9	80.7	80.7
Ours-VGG-LG-MP2	**96.7**	83.6	78.2	66.3	88.3	81.2	80.9
Ours-VGG-LG	96.5	83.1	**78.8**	**66.7**	**88.7**	**81.7**	**81.1**

Table 1. Comparison of strict PCP on the LSP dataset. We investigate our method with different network architectures (ChenNet and VGG), as well as different graph structures (tree-structured model (T) and loopy graph (LG)). We also investigate the performance of invividual part detectors (Unary), joint training vs. independent training (Ind), and different number of message passing layers (MP1, MP2). Note that DeepPose [42] uses Person-Centric annotations.

Method	U.arms	L.arms	Mean
MODEC [35]	84.4	52.1	68.3
Tompson *et al*. [41]	93.7	80.9	87.3
Chen&Yuille [4]	97.0	86.8	91.9
Ours-ChenNet-T	97.9	88.3	93.1
Ours-VGG-T	**98.1**	**89.5**	**93.8**

Table 2. Strict PCP results on the FLIC dataset. We investigate our method with different network architectures (ChenNet and VGG) with tree-structured model (T).

Method	Torso	Head	U. arms	L. arms	U. legs	L. legs	Mean
Yang&Ramanan [51]	82.9	77.6	55.1	35.4	69.0	63.9	60.7
Johnson&Everingham [19]	87.6	76.8	67.3	45.8	74.7	67.1	67.4
Pishchulin *et al*. [31]	88.8	73.7	53.7	36.1	77.3	67.1	63.1
Pishchulin *et al*. [29]	92.2	70.7	54.9	39.8	74.6	63.7	62.9
Pishchulin *et al*. [30]	93.2	86.3	63.4	48.8	77.1	68.0	69.4
Yang&Ramanan [52]	85.9	86.8	63.4	42.7	74.9	68.3	67.1
Ouyang *et al*. [26]	89.3	**89.3**	67.8	47.8	78.0	72.0	71.0
Ours-ChenNet-LG	96.6	87.3	80.0	65.9	83.7	74.1	79.1
Ours-VGG-LG	**97.1**	86.8	**80.2**	**69.3**	**84.9**	**78.5**	**81.0**

Table 3. Strict PCP results on PARSE dataset. Note that our model is trained on the LSP dataset to demonstrate its generalization ability.

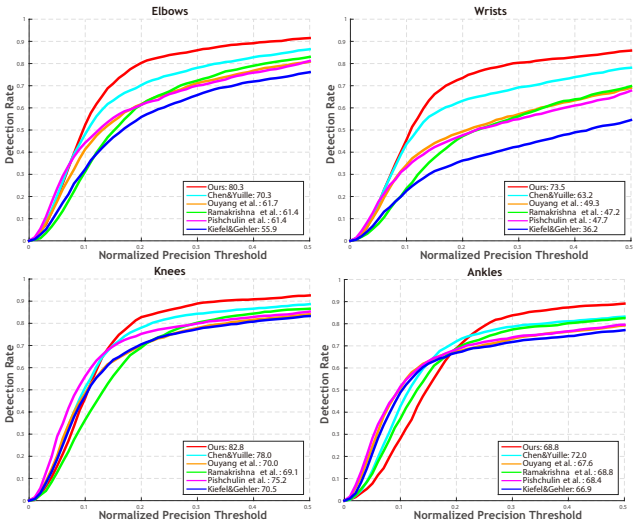

Figure 5. PDJ results for *elbows, wrists, knees* and *ankles* on the LSP dataset. We compare our method (VGG using loopy model) with Chen and Yuille [4], Ouyang *et al*. [26], Ramakrishna *et al*. [32], Pishchulin *et al*. [30], and Kiefel and Gehler [21]. We report the PDJ rate at the threshold of 0.2 in the legend.

Table 1 and Table 2 report strict PCP results on the LSP dataset and the FLIC dataset respectively. Our best performance on the LSP is achieved by using VGG together with loopy model (Ours-VGG-LG), which improves the mean strict PCP by 6.1% when compared with [4]. The best performance on FLIC is achieved by using VGG with tree-structured model (Ours-VGG-T), and improves the mean strict PCP by 1.9% when compared with [4]. On the LSP dataset with many challenging articulations, our method has significant improvements on limbs, *i.e*. *arms* and *legs*, which are the most difficult body parts to locate.

Figure 5 shows PDJ results on the LSP dataset. By comparing the PDJ value at the threshold 0.2, our method outperforms state-of-the-art methods by a significant margin on all body parts except *ankles*.

PDJ results on the FLIC dataset is reported in Figure 6. Our method achieves the best performance on both *elbows* and *wrists* compared with state-of-the-art methods.

Generalization Evaluation: To investigate the generalization ability of our method, we apply the model trained on the LSP dataset directly to the official test set of the PARSE

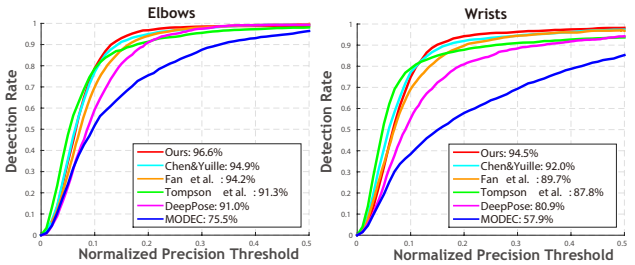

Figure 6. PDJ comparison of *elbows* and *wrists* on the FLIC dataset. We compare our method (VGG with tree-structured models) with Chen and Yuille [4], Fan *et al*. [10], Tompson *et al*. [40], DeepPose [42], and MODEC [35]. We report the PDJ rate at the threhold of 0.2 in the legend.

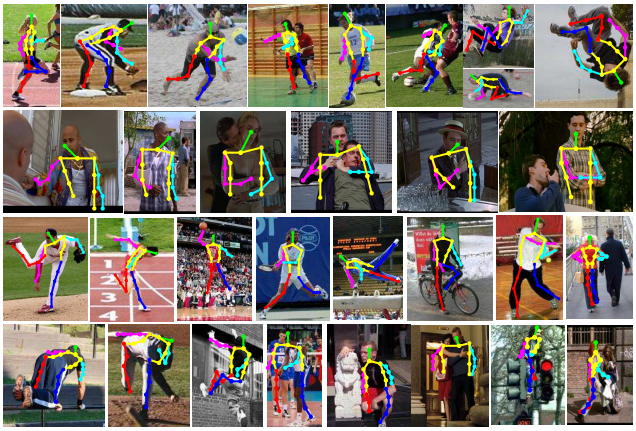

Figure 7. Qualitative results on the LSP dataset (the 1st row), the FLIC dataset (the 2nd row), and the PARSE dataset (the 3rd row). We visualize the joint locations together with the connections among parts used in this paper (for simplicity, we only show tree-structure), and the same limb across different images has the same color. Some failure cases are showed in the last row. Our method may lead to wrong estimations due to significant occlusions, ambiguous background, or heavily overlapping persons.

dataset. As shown in Table 3, our method outperforms the state-of-the-art methods with a large margin, which implies that our method has good generalization ability.

Joint Training vs. Independent Training: To investigate the efficiency of joint training of part detectors and deformable mixture of parts, we train a model whose architecture is the same as *Ours-ChenNet-LG* on the LSP dataset. However, we first train the part detectors, then we fix the part detectors to train the message passing layers. In this scenario, the mean PCP is 74.2%, as reported in Table 1 (Ours-ChenNet-LG-Ind). In comparison, our proposed joint learning (Ours-ChenNet-LG) has 4.4% gain.

Number of the Message Passing Layers: The mean PCPs obtained by the first, the second and the third message passing layer of *Ours-VGG-LT* are 80.7% (Ours-VGG-LT-MP1), 80.9% (Ours-VGG-LT-MP2), and 81.1% (Ours-VGG-LT), respectively. As discussed in Section 4.2, we observe that a cascade of three message passing layers is enough to produce satisfactory results in practice, as shown in Figure 3.

Components Investigation: We first evaluate the performance of individual part detectors. Without spatial constraints, our method obtains 40.6% and 60.1% strict PCPs on the LSP dataset with ChenNet (Ours-ChenNet-Unary) and VGG (Ours-VGG-Unary) respectively, as reported in Table 1.

We conduct four experiments to analyze the influence of different components on the LSP dataset, and report the results in Table 1. First, we use ChenNet with tree-structured model (Ours-ChenNet-T), which outperforms the best previously published result [4] by 3.1% on average. This proves the effectiveness of jointly training DCNNs and deformable mixture of parts. Part detector and message passing are jointly learned in Ours-ChenNet-T but separately learned in [4]. Second, we build a loopy model based on tree-structured model by adding an edge between knees (Ours-ChenNet-LG), and get 0.5% improvement. We observe that this improvement is mainly due to the reduction of double-counting problem, as shown in Figure 4. Next, we evaluate the ImageNet pretrained VGG with tree-structured model (Ours-VGG-T). This gives an improvement over the ChenNet (Ours-ChenNet-T in Table 1) by 2.6%, which shows the expressive power of deeper DCNN and the robustness of the ImageNet pretrained feature representation. Finally, by combining VGG with loopy model, the Ours-ChenNet-LG in Table 1 achieve the best result on the LSP dataset.

Qualitative Evaluation: Figure 7 shows some pose estimation results on all the three datasets. Our method is robust to highly articulated poses with variant orientation, foreshortening, cluttered background, occlusion, and overlapping people. Some failure cases are also showed in the last row of Figure 7. Our method may lead to wrong estimations due to significant occlusions, ambiguous background, or heavily overlapping persons. Please refer to the captions for detailed discussion.

7. Conclusion

This paper has proposed to incorporate the DCNN and the deformable mixture of parts model into an end-to-end framework. Our framework is able to mine hard negatives by considering the spatial and appearance consistency among body parts. Therefore, the DCNN can be trained more effectively. The joint learning of DCNN and deformable mixture of parts improves the performance on several widely used benchmarks, which demonstrates the effectiveness of our method. In the future work, we plan to investigate learning graph structures with deep models.

Acknowledgments: This work is partially supported by SenseTime Group Limited, and the General Research Fund sponsored by the Research Grants Council of Hong Kong (Project Nos. CUHK14206114, CUHK14205615, CUHK14203015, CUHK14207814).

References

[1] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009. 2

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 2

[3] X. Chen and A. Yuille. Parsing occluded people by flexible compositions. In *CVPR*, 2015. 1, 3, 4, 5

[4] X. Chen and A. L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*, 2014. 1, 2, 3, 4, 5, 6, 7, 8

[5] A. Cherian, J. Mairal, K. Alahari, and C. Schmid. Mixing body-part sequences for human pose estimation. In *CVPR*, 2014. 2

[6] N.-G. Cho, A. L. Yuille, and S.-W. Lee. Adaptive occlusion state estimation for human pose tracking under self-occlusions. *Pattern Recognition*, 46(3):649–661, 2013. 1

[7] X. Chu, W. Ouyang, H. Li, and X. Wang. Structured feature learning for pose estimation. In *CVPR*, 2016. 1

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2, 6

[9] M. Eichner and V. Ferrari. Appearance sharing for collective human pose estimation. In *ACCV*, 2013. 2, 7

[10] X. Fan, K. Zheng, Y. Lin, and S. Wang. Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. In *CVPR*, 2015. 1, 8

[11] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005. 2, 5

[12] V. Ferrari, M. Marín-Jiménez, and A. Zisserman. 2d human pose estimation in tv shows. In *Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 128–147. Springer, 2009. 2

[13] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92, 1973. 2

[14] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. In *CVPR*, 2015. 2, 3

[15] A. Jain, J. Tompson, M. Andriluka, G. W. Taylor, and C. Bregler. Learning human pose estimation features with convolutional networks. In *ICLR*, 2014. 1, 2, 3

[16] A. Jain, J. Tompson, Y. LeCun, and C. Bregler. Modeep: A deep learning framework using motion features for human pose estimation. In *ACCV*, 2014. 1

[17] H. Jiang and D. R. Martin. Global pose estimation using non-tree models. In *CVPR*, 2008. 2

[18] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010. 2, 6

[19] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *CVPR*, 2011. 2, 7

[20] K. Kang, W. Ouyang, H. Li, and X. Wang. Object detection from video tubelets with convolutional neural networks. In *CVPR*, 2016. 2

[21] M. Kiefel and P. V. Gehler. Human pose estimation with fields of parts. In *ECCV*, 2014. 7

[22] G. Lin, C. Shen, I. Reid, and A. v. d. Hengel. Deeply learning the messages in message passing inference. In *NIPS*, 2015. 3

[23] L. Lin, G. Wang, R. Zhang, R. Zhang, X. Liang, and W. Zuo. Deep structured scene parsing by learning with image descriptions. In *CVPR*, 2015. 2

[24] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, 2015. 2, 3

[25] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *ECCV*, 2002. 2

[26] W. Ouyang, X. Chu, and X. Wang. Multi-source deep learning for human pose estimation. In *CVPR*, 2014. 3, 7

[27] W. Ouyang and X. Wang. Joint deep learning for pedestrian detection. In *ICCV*, 2013. 2, 3

[28] W. Ouyang, X. Wang, C. Zhang, and X. Yang. Factors in finetuning deep model for object detection. In *CVPR*, 2016. 2

[29] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *CVPR*, 2013. 2, 7

[30] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Strong appearance and expressive spatial models for human pose estimation. In *ICCV*, 2013. 2, 7

[31] L. Pishchulin, A. Jain, M. Andriluka, T. Thormaehlen, and B. Schiele. Articulated people detection and pose estimation: Reshaping the future. In *CVPR*, 2012. 7

[32] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh. Pose machines: Articulated pose estimation via inference machines. In *ECCV*, 2014. 7

[33] D. Ramanan. Learning to parse images of articulated objects. In *NIPS*, 2006. 2, 6

[34] X. Ren, A. C. Berg, and J. Malik. Recovering human body configurations using pairwise constraints between parts. In *ICCV*, 2005. 2

[35] B. Sapp and B. Taskar. Modec: Multimodal decomposable models for human pose estimation. In *CVPR*, 2013. 2, 6, 7, 8

[36] L. Sigal and M. J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *CVPR*, 2006. 2, 4

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[38] T.-P. Tian and S. Sclaroff. Fast globally optimal 2d human detection with loopy graph models. In *CVPR*, 2010. 2

[39] Y. Tian, C. L. Zitnick, and S. G. Narasimhan. Exploring the spatial hierarchy of mixture models for human pose estimation. In *ECCV*, 2012. 2

[40] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *CVPR*, 2015. 1, 3, 5, 8

[41] J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014. 1, 2, 3, 5, 7

[42] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014. 1, 3, 7, 8

[43] D. Tran and D. Forsyth. Improved human parsing with a full relational model. In *ECCV*, 2010. 2

[44] L. Wan, D. Eigen, and R. Fergus. End-to-end integration of a convolutional network, deformable parts model and non-maximum suppression. In *CVPR*, 2015. 2, 3

[45] C. Wang, Y. Wang, and A. L. Yuille. An approach to pose-based action recognition. In *CVPR*, 2013. 1

[46] F. Wang and Y. Li. Beyond physical connections: Tree models in human pose estimation. In *CVPR*, 2013. 2

[47] L. Wang, H. Lu, X. Ruan, and M.-H. Yang. Deep networks for saliency detection via local estimation and global search. In *CVPR*, 2015. 2

[48] T. Xiao, H. Li, W. Ouyang, and X. Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *CVPR*, 2016. 2

[49] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *CVPR*, 2012. 1

[50] W. Yang, P. Luo, and L. Lin. Clothing co-parsing by joint image segmentation and labeling. In *CVPR*, 2014. 1

[51] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011. 2, 4, 7

[52] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *TPAMI*, 35(12):2878–2890, 2013. 4, 6, 7