

# Combining Local Appearance and Holistic View: Dual-Source Deep Neural Networks for Human Pose Estimation

Xiaochuan Fan, Kang Zheng, Yuewei Lin, Song Wang

Department of Computer Science & Engineering, University of South Carolina, Columbia, SC 29208, USA

{fan23, zheng37, lin59}@email.sc.edu, songwang@cec.sc.edu

## Abstract

We propose a new learning-based method for estimating 2D human pose from a single image, using Dual-Source Deep Convolutional Neural Networks (DS-CNN). Recently, many methods have been developed to estimate human pose by using pose priors that are estimated from physiologically inspired graphical models or learned from a holistic perspective. In this paper, we propose to integrate both the local (body) part appearance and the holistic view of each local part for more accurate human pose estimation. Specifically, the proposed DS-CNN takes a set of image patches (category-independent object proposals for training and multi-scale sliding windows for testing) as the input and then learns the appearance of each local part by considering their holistic views in the full body. Using DS-CNN, we achieve both joint detection, which determines whether an image patch contains a body joint, and joint localization, which finds the exact location of the joint in the image patch. Finally, we develop an algorithm to combine these joint detection/localization results from all the image patches for estimating the human pose. The experimental results show the effectiveness of the proposed method by comparing to the state-of-the-art human-pose estimation methods based on pose priors that are estimated from physiologically inspired graphical models or learned from a holistic perspective.

## 1. Introduction

By accurately locating the important body joints from 2D images, human pose estimation plays an essential role in computer vision. It has wide applications in intelligent surveillance, video-based action recognition, and human-computer interaction. However, human pose estimation from an 2D image is a well known challenging problem – too many degrees of freedom are introduced by the large variability of the human pose, different visual appearance of the human body and joints, different angles of camera

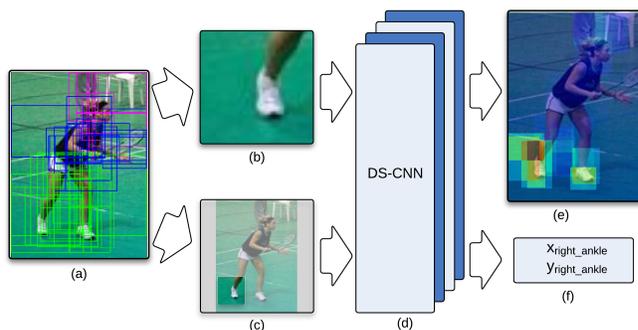


Figure 1. An illustration of the proposed method based on DS-CNN. (a) Input image and generated image patches. (b) DS-CNN input on an image patch (containing a local part – ankle). (c) DS-CNN input on full body and holistic view of the local part in the full body. (d) DS-CNN for learning. (e) DS-CNN output on joint detection. (f) DS-CNN output on joint localization.

view, and possible occlusions of body parts and joints.

Most of the previous works on human pose estimation are based on the two-layer part-based model [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. The first layer focuses on local (body) part appearance and the second layer imposes the contextual relations between local parts. One popular part-based approach is pictorial structures [1], which capture the pairwise geometric relations between adjacent parts using a tree model. However, these pose estimation methods using part-based models are usually sensitive to noise and the graphical model lacks expressiveness to model complex human poses [12]. Furthermore, most of these methods search for each local part independently and the local appearance may not be sufficiently discriminative for identifying each local part reliably.

Recently, deep neural network architectures, specifically deep convolutional neural networks (CNNs), have shown outstanding performance in many computer vision tasks. Due to CNNs’ large learning capacity and robustness to variations, there is a natural rise in the interest to directly learn high-level representations of human poses without using hand-crafted low-level features and even graphical mod-

els. Toshev et al. [16] present such a holistic-styled pose estimation method named DeepPose using DNN-based joint regressors. This method also uses a two-layer architecture: The first layer resolves ambiguity between body parts (e.g. left and right legs) in a holistic way and provides an initial pose estimation and the second layer refines the joint locations in a local neighborhood around the initial estimation. From the experiments in [16], DeepPose can achieve better performance on two widely used datasets, FLIC and LSP, than several recently developed human pose estimation methods. However, DeepPose does not consider local part appearance in initial pose estimation. As a result, it has difficulty in estimating complex human poses, even using the CNN architecture.

In this paper, we propose a dual-source CNN (DS-CNN) based method for human pose estimation, as illustrated in Fig. 1. This proposed method integrates both the local part appearance in image patches and the holistic view of each local part for more accurate human pose estimation. Following the region-CNN (R-CNN) that was developed for object detection [17], the proposed DS-CNN takes a set of category-independent object proposals detected from the input image for training. Compared to the sliding windows or the full image, that are used as the input in many previous human pose estimation methods, object proposals can capture the local body parts with better semantic meanings in multiple scales [17, 18]. In this paper, we extend the original single-source R-CNN to a dual-source model (DS-CNN) by including the full body and the holistic view of the local parts as a separate input, which provides a holistic view for human pose estimation. By taking both the local part object proposals and the full body as inputs in the training stage, the proposed DS-CNN performs a unified learning to achieve both joint detection, which determines whether an object proposal contains a body joint, and joint localization, which finds the exact location of the joint in the object proposal. In the testing stage, we use multi-scale sliding windows to provide local part information in order to avoid the performance degradation resulted by the uneven distribution of object proposals. Based on the DS-CNN outputs, we combine the joint detection results from all the sliding windows to construct a heatmap that reflects the joint location likelihood at each pixel and weightedly average the joint localization results at the high-likelihood regions of the heatmap to achieve the final estimation of each joint location.

In the experiments, we test the proposed method on two widely used datasets and compare its performance to several recently reported human pose estimation methods, including DeepPose. The results show the effectiveness of the proposed method which combines local appearance and holistic view.

## 2. Related Work

**Part-based models for human pose estimation.** In the part-based models, human body is represented by a collection of physiologically inspired parts assembled through a deformable configuration. Following the pictorial-structure model [19, 1], a variety of part-based methods have been developed for human pose estimation [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. While many early methods build appearance models for each local part independently, recent works [8, 20, 4, 12, 10, 15] attempt to design strong body part detectors by capturing the contextual relations between body parts. Johnson and Everingham [10] partition the pose space into a set of pose clusters and then apply nonlinear classifiers to learn pose-specific part appearance. In [12], independent regressors are trained for each joint and the results from these regressors are combined to estimate the likelihood of each joint at each pixel of the image. Based on the appearance models built for each part, these methods usually leverage tree-structured graphical models to further impose the pairwise geometric constraints between parts [2, 3, 8, 7, 21]. Due to the limited expressiveness [16], the tree-structured graphical models often suffer from the limb ambiguity, which affects the accuracy of human pose estimation. There have been several works that focus on designing richer graphical models to overcome the limitation of tree-structured graphical models. For example, in [10], mixture of pictorial structure models are learned to capture the ‘multi-modal’ appearance of each body part. Yang and Ramanan [21] introduce a flexible mixture-of-parts model to capture contextual co-occurrence relations between parts. In [2], the hierarchical structure is incorporated to model high-order spatial relation among parts. Loopy models [5, 22, 23, 24] allow to include additional part constraints, but require approximate inference. In the latter experiments, we include several above-mentioned part-based methods for performance comparison.

**Deep convolutional neural network (CNN) in computer vision.** As a popular deep learning approach, CNN [25] attempts to learn multiple levels of representation and abstraction and then use it to model complex nonlinear relations. It has been shown to be a useful tool in many computer vision applications. For example, it has demonstrated impressive performance for image classification [26, 27, 28, 29]. More recently, CNN architectures have been successfully applied to object localization and detection [30, 17, 31]. In [31], a single shared CNN named ‘Overfeat’ is used to simultaneously classify, locate and detect objects from an image by examining every sliding window. In this paper, we also integrate joint detection and localization using a single DS-CNN. But our problem is much more challenging than object detection – we need to find precise locations of a set of joints for human pose estimation. Girshick et al. [17] apply high-capacity R-CNNs to bottom-up

object proposals [32] for object localization and segmentation. It achieves 30% performance improvement on PASCAL VOC 2012 against the state of the art. Zhang et al. [18] adopt the R-CNN [17] to part localization and verify that the use of object proposals instead of sliding windows in CNN can help localize smaller parts. Based on this, R-CNN is shown to be effective for fine-grained category detection. However, this method does not consider the complex relations between different parts [18] and is not applicable to human pose estimation.

**CNN for human pose estimation.** In [16], a cascade of CNN-based joint regressors are applied to reason about pose in a holistic manner and the developed method was named ‘DeepPose’. The DeepPose networks take the full image as the input and output the ultimate human pose without using any explicit graphical model or part detectors.

In [33], Jain et al. introduce a CNN-based architecture and a learning technique that learns low-level features and a higher-level weak spatial model. Following [33], Tompson et al. show that the inclusion of a MRF-based graphical model into the CNN-based part detector can substantially increase the human pose estimation performance. Different from DeepPose and Tompson et al. [34], the proposed method takes both the object proposals and the full body as the input for training, instead of using the sliding-windowed patches, to capture the local body parts with better semantic meanings in multiple scales.

### 3. Problem Description and Notations

In this paper, we adopt the following notations. A human pose can be represented by a set of human joints  $\mathbf{J} = \{\mathbf{j}_i\}_{i=1}^L \in \mathbb{R}^{2L \times 1}$ , where  $\mathbf{j}_i = (x_i, y_i)^T$  denotes the 2D coordinate of the joint  $i$  and  $L$  is the number of human joints. In this paper, we are interested in estimating the 2D joint locations  $\mathbf{J}$  from a single image  $I$ . Since our detection and regression are applied to a set of image patches, in the form of rectangular bounding boxes, detected in  $I$ , it is necessary to convert absolute joint coordinates in image  $I$  to relative joint coordinates in an image patch. Furthermore, we introduce a normalization to make the locations invariant to size of different image patches. Specifically, given an image patch  $\mathbf{p}$ , the location of  $\mathbf{p}$  is represented by 4-element vector  $\mathbf{p} = (w(\mathbf{p}), h(\mathbf{p}), c(\mathbf{p}))^T$ , where  $w(\mathbf{p})$  and  $h(\mathbf{p})$  are the width and height of  $\mathbf{p}$ ,  $c(\mathbf{p}) = (x_c(\mathbf{p}), y_c(\mathbf{p}))^T$  is the center of  $\mathbf{p}$ . Then the normalized coordinate of joint  $\mathbf{j}_i$  relative to  $\mathbf{p}$  can be denoted as

$$\begin{aligned} \mathbf{j}_i(\mathbf{p}) &= (x_i(\mathbf{p}), y_i(\mathbf{p}))^T \\ &= \left( \frac{x_i - x_c(\mathbf{p})}{w(\mathbf{p})}, \frac{y_i - y_c(\mathbf{p})}{h(\mathbf{p})} \right)^T. \end{aligned} \quad (1)$$

Furthermore, the visibility of all the joints in  $\mathbf{p}$  is denoted



Figure 2. Extended part and body patches containing (a) right ankle, (b) left ankle, (c) right wrist, and (d) left wrist from the LSP training dataset. For each local part, the part patches are shown in the left while the corresponding body patches are shown in the right.

as  $\mathbf{V}(\mathbf{p}) = \{v_i(\mathbf{p})\}_{i=1}^L \in \mathbb{R}^{L \times 1}$ , where

$$v_i(\mathbf{p}) = \begin{cases} 1, & |x_i(\mathbf{p})| \leq 0.5 \text{ and } |y_i(\mathbf{p})| \leq 0.5 \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

If  $v_i(\mathbf{p}) = 1$ , it indicates that the joint  $i$  is visible in  $\mathbf{p}$ , i.e., it is located inside the patch  $\mathbf{p}$ . On the contrary, if  $v_i(\mathbf{p}) = 0$ , it indicates that the joint  $i$  is invisible in  $\mathbf{p}$ , i.e., it is located outside of  $\mathbf{p}$ .

### 4. Model Inputs

As described earlier, to combine the local part appearance and the holistic view of each part, the proposed DS-CNN takes two inputs for training and testing – image patches and the full body. To make it clearer, we call the former input the *part patches*, denoted as  $\mathbf{p}_p$ , and the latter *body patches*, denoted as  $\mathbf{p}_b$ . So the dual-source input is  $\mathbf{p}_{p,b} = (\mathbf{p}_p, \mathbf{p}_b)$ . Randomly selected samples for these two kinds of inputs are shown in Fig. 2, where for each local part, the part patches are shown in the left while the corresponding body patches are shown in the right. From these samples, we can see that it is difficult to distinguish the left and right wrists, or some wrists and legs, based only on the local appearance in the part patches.

As we will see later, we use object proposals detected from an image for training and object proposals usually show different sizes and different aspect ratios. CNN requires the input to be of a fixed dimension. In [17], all the object proposals are non-uniformly scaled to a fixed-size square and it may need to vary the original aspect ratios. This may complicate the CNN training by artificially introducing unrealistic patterns into training samples. In particular, in our model we are only interested in the body joint that is closest to the center of a part patch (This will be elaborated in detail later). If the part patch is non-uniformly scaled, the joint of interest may be different after the change

of the aspect ratio. Thus, in this paper we keep the aspect ratio of image patches unchanged when unifying its size. Specifically, we extend the short side of the image patch to include additional rows or columns to make it a square. This extension is conducted in a way such that the center of each image patch keeps unchanged. After the extension, we can perform uniform scaling to make each patch a fixed-size square. This extension may not be preferred in object detection [17] by including undesired background information. However, in our problem this extension just includes more context information of the joint of interest. This will not introduce much negative effect to the part detection. The only minor effect is the a subtle reduction of the resolution of each patch (after the uniform scaling).

**Part Patches** In the training stage, we construct part patches in two steps. 1) Running an algorithm to construct a set of category-independent object proposals. Any existing object proposal algorithms can be used here. In our experiments, we use the algorithm developed in [35]. 2) Select a subset of the constructed proposals as the part patches. We consider two factors for Step 2). First, we only select object proposals with a size in certain range as part patches. If the size of an object proposal is too large, it may cover multiple body parts and its appearance lacks sufficient resolution (after the above-mentioned uniform scaling) for joint detection and localization. On the contrary, if the size of an object proposal is too small, its appearance may not provide sufficient features. To address this issue, we only select the object proposals  $\mathbf{p}_p$  with an area in a specified range as part patches, i.e.,

$$\mu_1 d^2(\mathbf{J}) \leq w(\mathbf{p}_p) \cdot h(\mathbf{p}_p) \leq \mu_2 d^2(\mathbf{J}) \quad (3)$$

where  $d(\mathbf{J})$  is the distance between two opposing joints on the human torso [16].  $\mu_1$  and  $\mu_2$  are two coefficients ( $\mu_1 < \mu_2$ ) that help define the lower bound and the upper bound for selecting an object proposal as a part patch.

Second, from the training perspective, we desire all the body joints are covered by sufficient number of part patches. In the ideal case, we expect the selected part patches cover all the joints in a balanced way – all the joints are covered by similar number of part patches. We empirically examine this issue and results are shown in Fig. 3 – on both FLIC and LSP datasets, this simple part-patch selection algorithm provides quite balanced coverage to all the joints. In this figure, the x-axis indicates the label of different joints – only upper-body joints are shown in FLIC dataset while all 14 body joints are shown in LSP dataset. The y-axis indicates the average number of part patches that covers the specified joint in each image. Here we count that a part patch covers a joint if this joint is visible to (i.e., located inside) this patch and this joint is the closest joint to the center of this patch. At each joint, we show three part-patch coverage numbers in three different colors. From left to right, they correspond

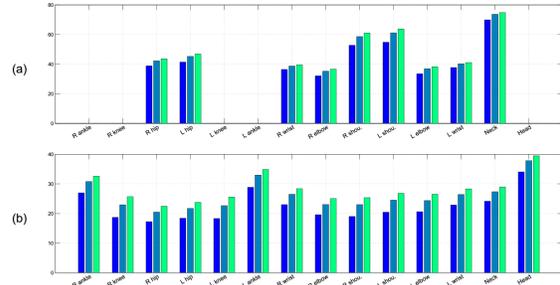


Figure 3. The average number of part patches that cover each joint in (a) FLIC and (b) LSP datasets. Three colors indicates the results by selecting different  $\mu_2$  values of 1.0, 1.5 and 2.0 respectively.

to three different  $\mu_2$  values of 1.0, 1.5 and 2.0 respectively. In this empirically study, we always set  $\mu_1 = 0.1$ .

In the testing stage, part patches are selected from multi-scale sliding windows (this will be justified in Section 8).

**Body Patches** Similarly, in the training stage we construct body patches by selecting a subset of object proposals from the same pool of object proposals detected from the image. The only requirement is that the selected body patch should cover the whole body or all the joints, i.e.,

$$\sum_{i=1}^L v_i(\mathbf{p}_b) = L. \quad (4)$$

In the testing stage, the body patch can be generated by using a human detector. For the experiments in this paper, each testing image only contains one person and we simply take the whole testing image as the body patch.

For DS-CNN, each training sample is made up of a part patch  $\mathbf{p}_p$ , a body patch  $\mathbf{p}_b$ , and the binary mask that specifies the location of  $\mathbf{p}_p$  in  $\mathbf{p}_b$ , as shown in Fig. 2, where both  $\mathbf{p}_p$  and  $\mathbf{p}_b$  are extended and normalized to a fixed-size square image. For part patch  $\mathbf{p}_p$ , we directly take the RGB values at all the pixels as the first source of input to DS-CNN. For body patch  $\mathbf{p}_b$ , we take the binary mask as an additional alpha channel and concatenate the RGB values of  $\mathbf{p}_b$  and the alpha values as the second source of input to DS-CNN. Given that we extend and normalize all the patches to a  $N \times N$  square, the first source of input is of dimension  $3N^2$  and the second source of input is of size  $4N^2$ . In the training stage, based on the constructed part patches and body patches, we randomly select one from each as a training sample. For both training and testing, if the selected part patch is not fully contained in the selected body patch, we crop the part patch by removing the portion located outside the body patch before constructing the training or testing sample.

## 5. Multi-Task Learning

We combine two tasks in a single DS-CNN – joint detection, which determines whether a part patch contains a body

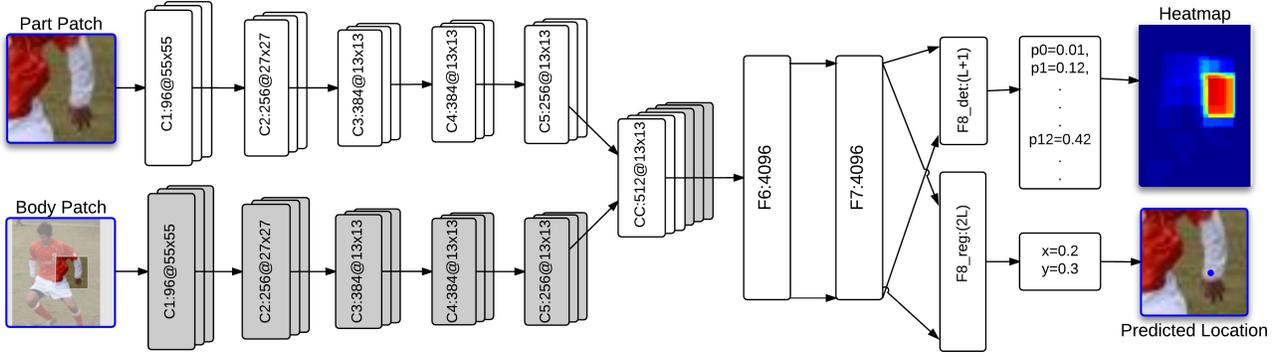


Figure 4. The structure of DS-CNN.

joint, and joint localization, which finds the exact location of the joint in the part patch. Each task is associated with a loss function.

**Joint detection** For joint detection, we label a patch-pair  $\mathbf{p}_{p,b}$  to joint  $i^*$ , where

$$i^*(\mathbf{p}_p) = \begin{cases} \arg \min_{1 \leq i \leq L} \|\mathbf{j}_i(\mathbf{p}_p)\|^2 & \text{if } \sum_{k=1}^L v_k(\mathbf{p}_p) > 0 \\ 0 & \text{otherwise,} \end{cases} \quad (5)$$

and this is taken as the ground truth for training.

Let the DS-CNN output for joint detection be  $(\ell_0(\mathbf{p}_{p,b}), \ell_1(\mathbf{p}_{p,b}), \dots, \ell_L(\mathbf{p}_{p,b}))^T$ , where  $\ell_0$  indicates the likelihood of no body joint visible in  $\mathbf{p}_p$  and  $\ell_i, i = 1, \dots, L$  represents the likelihood that joint  $i$  is visible in  $\mathbf{p}_p$  and is the closest joint to the center of  $\mathbf{p}_p$ . We use a softmax classifier where the loss function is

$$C_d(\mathbf{p}_{p,b}) = - \sum_{i=0}^L 1(i^*(\mathbf{p}_p) = i) \log(\ell_i(\mathbf{p}_{p,b})), \quad (6)$$

where  $1(\cdot)$  is the indicator function.

**Joint localization** Joint localization is formulated as a regression problem. In DS-CNN training, the ground-truth joint location for a patch-pair  $\mathbf{p}_{p,b}$  is  $\mathbf{j}_{i^*(\mathbf{p}_p)}(\mathbf{p}_p) = (x_{i^*(\mathbf{p}_p)}(\mathbf{p}_p), y_{i^*(\mathbf{p}_p)}(\mathbf{p}_p))^T$ , where  $i^*(\mathbf{p}_p)$  is defined in Eq. (5). Let the DS-CNN output on joint localization be  $\{\mathbf{z}_i(\mathbf{p}_{p,b})\}_{i=1}^L \in \mathbb{R}^{2L \times 1}$ , where  $\mathbf{z}_i(\mathbf{p}_{p,b}) = (\hat{x}_i, \hat{y}_i)^T$  denotes the predicted location of the  $i$ -th joint in  $\mathbf{p}_p$ . We use the mean squared error as the loss function,

$$C_r(\mathbf{p}_{p,b}) = \begin{cases} \|\mathbf{z}_{i^*(\mathbf{p}_p)}(\mathbf{p}_{p,b}) - \mathbf{j}_{i^*(\mathbf{p}_p)}(\mathbf{p}_p)\|^2 & \text{if } i^* > 0 \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

Combining the joint detection and joint localization, the loss function for DS-CNN is

$$C = \sum_{\mathbf{p}_{p,b}} \{\lambda_d C_d(\mathbf{p}_{p,b}) + C_r(\mathbf{p}_{p,b})\}, \quad (8)$$

where the summation is over all the training samples (patch pairs) and  $\lambda_d > 0$  is a factor that balances the two loss functions.

## 6. DS-CNN Structure

The structure of the proposed DS-CNN is based on the CNN described in [29], which is made up of five convolutional layers, three fully-connected layers, and a final 1000-way softmax, in sequence. The convolutional layers 1, 2 and 5 are followed by max pooling. In the proposed DS-CNN, we include two separate sequence of convolutional layers. As shown in Fig. 4, one sequence of 5 convolutional layers takes the part-patch input as defined in Section 4 and extracts the features from local appearance. The other sequence of 5 convolutional layers takes the body-patch input and extracts the holistic features of each part. The output from these two sequences of convolutional layers are then concatenated together, which are then fed to a sequence of three fully connected layers. We replace the final 1000-way softmax by a  $(L + 1)$ -way softmax and a regressor for joint detection and joint localization, respectively. In DS-CNN, all the convolutional layers and the fully-connected layers are shared by both the joint detection and the joint localization.

In Fig. 4, the convolutional layer and the following pooling layer is labeled  $C_i$  and the fully-connected layer are labeled as  $F_i$  where  $i$  is the index of layer. The size of a convolutional layer is described as depth@width  $\times$  height, where depth is the number of convolutional filters, width and height denote the spatial dimension.

## 7. Human Pose Estimation

Given a testing image, we construct a set of patch-pairs using multi-scale sliding windows as discussed in Section 4. We then run the trained DS-CNN on each patch-pair  $\mathbf{p}_{p,b}$  to obtain both joint detection and localization. In this section, we propose an algorithm for estimating the final human pose on the testing image by combining the joint detection and localization results from all the patch pairs.

At first, we construct a heatmap  $H_i$  for each joint  $i$  – the heatmap is of the same size as the original image and  $H_i(\mathbf{x})$ , the heatmap value at a pixel  $\mathbf{x}$ , reflects the likelihood that joint  $i$  is located at  $\mathbf{x}$ . Specifically, for each patch-pair  $\mathbf{p}_{p,b}$ , we uniformly allocate its joint-detection likelihood to all the pixels in  $\mathbf{p}_p$ , i.e.,

$$h_i(\mathbf{x}, \mathbf{p}_{p,b}) = \begin{cases} \ell_i(\mathbf{p}_{p,b}) / (w(\mathbf{p}_p) \cdot h(\mathbf{p}_p)), \\ \text{if } \mathbf{x} \in \mathbf{p}_p \text{ and } \ell_i(\mathbf{p}_{p,b}) > \ell_j(\mathbf{p}_{p,b}), \forall j \neq i \\ 0 \text{ otherwise.} \end{cases} \quad (9)$$

We then sum up the allocated joint-detection likelihood over all the patch-pairs in a testing image as

$$H_i(\mathbf{x}) = \sum_{\mathbf{p}_{p,b}} h_i(\mathbf{x}, \mathbf{p}_{p,b}). \quad (10)$$

Figure 4 shows an example of the heatmap for the left wrist. We can see that, by incorporating the body patches, the constructed heatmap resolves the limb ambiguity. However, while the heatmap provides a rough estimation of the joint location, it is insufficient to accurately localize the body joints.

To find the accurate location of a joint, we take the DS-CNN joint-localization outputs from a selected subset of patch-pairs, where the joint is visible with high likelihood. We then take the weighted average of these selected outputs as the final location of the joint. More specifically, we only select patch pairs  $\mathbf{p}_{p,b}$  that satisfy the following conditions when finding the location of joint  $i$  in the testing image.

1. The likelihood that no body joint is visible in  $\mathbf{p}_p$  is smaller than the likelihood that joint  $i$  is visible in the part patch, i.e.

$$\ell_0(\mathbf{p}_{p,b}) < \ell_i(\mathbf{p}_{p,b}). \quad (11)$$

2. The likelihood that joint  $i$  is visible in  $\mathbf{p}_p$  should be among the  $k$  largest ones over all  $L$  joints. In a special case, if we set  $k=1$ , this condition requires  $\ell_i(\mathbf{p}_{p,b}) > \ell_j(\mathbf{p}_{p,b}), \forall j \neq i$ .
3. The maximum heatmap value (for joint  $i$ ) in  $\mathbf{p}_p$  is close to the maximum heatmap value over the body patch (full testing image in our experiments). Specifically, let

$$H_i^p = \max_{\mathbf{x} \in \mathbf{p}_p} H_i(\mathbf{x}), \quad (12)$$

and

$$H_i^b = \max_{\mathbf{x} \in \mathbf{p}_b} H_i(\mathbf{x}). \quad (13)$$

We require

$$H_i^p > \lambda_h H_i^b, \quad (14)$$

where  $\lambda_h$  is a scaling factor between 0 and 1. In our experiments, we set  $\lambda_h = 0.9$ .

Method	Arm		Leg		Torso	Head	Avg.
	Upper	Lower	Upper	Lower			
DS-CNN	<b>0.80</b>	<b>0.63</b>	<b>0.90</b>	<b>0.88</b>	<b>0.98</b>	0.85	<b>0.84</b>
DeepPose [16]	0.56	0.38	0.77	0.71	-	-	-
Dantone et al. [12]	0.45	0.25	0.68	0.61	0.82	0.79	0.60
Tian et al. [2]	0.52	0.33	0.70	0.60	0.96	<b>0.88</b>	0.66
Johnson et al. [36]	0.54	0.38	0.75	0.67	0.88	0.75	0.66
Wang et al. [3]	0.49	0.32	0.74	0.70	0.92	0.86	0.67
Pishchulin et al. [7]	0.54	0.34	0.76	0.68	0.88	0.78	0.66
Pishchulin et al. [37]	0.62	0.45	0.79	0.73	0.89	0.86	0.72

Table 1. PCP comparison on LSP. Note that DS-CNN, DeepPose [16] and Johnson et al. [36] are trained with both the LSP and its extension, while the other methods use only LSP.

Let  $\mathbf{P}^i$  be the set of the selected patch-pairs that satisfy these three conditions. We estimate the location of joint  $i$  by

$$\mathbf{j}'_i = \frac{\sum_{\mathbf{p}_{p,b} \in \mathbf{P}^i} \mathbf{z}'_i(\mathbf{p}_{p,b}) \ell_i(\mathbf{p}_{p,b})}{\sum_{\mathbf{p}_{p,b} \in \mathbf{P}^i} \ell_i(\mathbf{p}_{p,b})}, \quad (15)$$

where  $\mathbf{z}'_i(\mathbf{p}_{p,b})$  is the DS-CNN estimated joint- $i$  location in the coordinates of the body patch  $\mathbf{p}_b$ . As mentioned earlier, in our experiments, each testing image only contains one person and we simply take the whole image as the body patch  $\mathbf{p}_b$ , so  $\mathbf{z}'_i(\mathbf{p}_{p,b})$  can be derived from the DS-CNN joint localization output  $\mathbf{z}_i(\mathbf{p}_p)$  by applying the inverse transform of Eq. (1).

## 8. Experiments

In this paper, we evaluate the proposed method on Leeds Sports Pose (LSP) dataset [10], the extended LSP dataset [36], and Frames Labeled in Cinema (FLIC) dataset [15]. LSP and its extension contains 11,000 training and 1,000 testing images of sports people gathered from Flickr with 14 full body joints annotated. These images are challenging because of different appearances and strong articulation. The images in LSP dataset have been scaled so that the most prominent person is about 150 pixels high. FLIC dataset contains 3,987 training and 1,016 testing images from Hollywood movies with 10 upper body joints annotated. The images in FLIC dataset contain people with diverse poses and appearances and are biased towards front-facing poses.

Most LSP images only contain a single person. While each image in FLIC may contain multiple people, similar to [16], a standard preprocessing of body detection has been conducted to extract individual persons. As in previous works, we take the subimages of these detected individual persons as training and testing samples. This way, the training and testing data only contain *a single person* and as mentioned earlier, in the testing stage, we simply take the whole image (for FLIC dataset, this means a whole subimage for an individual person) as the body patch.

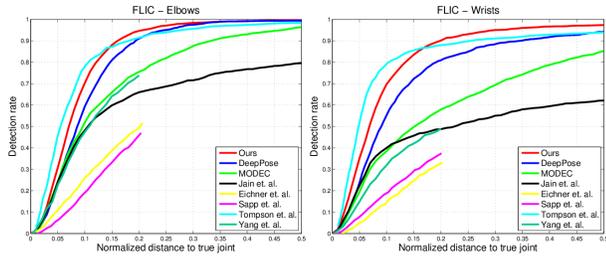


Figure 5. PDJ comparison on FLIC.

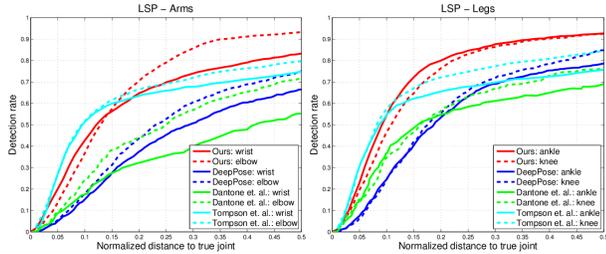


Figure 6. PDJ comparison on LSP.

It has been verified that, in the training stage, the use of object proposals can help train better CNNs for object detection and part localization [17, 18]. However, in the testing stage, object proposals detected on an image may be unevenly distributed. As a result, an image region covered by dense low-likelihood object proposals may undesirably show higher values in the resulting heatmap than a region covered by sparser high-likelihood object proposals. To avoid this issue, in our experiments we use multi-scale sliding windows (with sizes of  $0.5d(\mathbf{J})$  and  $d(\mathbf{J})$ , stride 2) to provide part patches in the testing stage.

To compare with previous works, we evaluate the performance of human pose estimation using two popular metrics: Percentage of Corrected Parts (PCP) [38] and Percentage of Detected Joints (PDJ) [15, 16]. PCP measures the rate of correct limb detection – a limb is detected if the distances between detected limb joints and true limb joints are no more than half of the limb length. Since PCP penalizes short limbs, PDJ is introduced to measure the detection rate of joints, where a joint is considered to be detected if the distance between detected joint and the true joint is less than a fraction of the torso diameter  $d(\mathbf{J})$  as described in Section 4. For PDJ, we can obtain different detection rate by varying the fraction and generate a PDJ curve in terms of the normalized distance to true joint [16].

The parameters that need to be set in the proposed method are

1. Lower bound coefficient  $\mu_1$  and the upper bound coefficient  $\mu_2$  in Eq.(3).
2. Balance factor  $\lambda_d$  in the loss function in Eq. (8).
3.  $k$  and  $\lambda_h$  that are used for selecting patch-pairs for joint localization in Section 7.

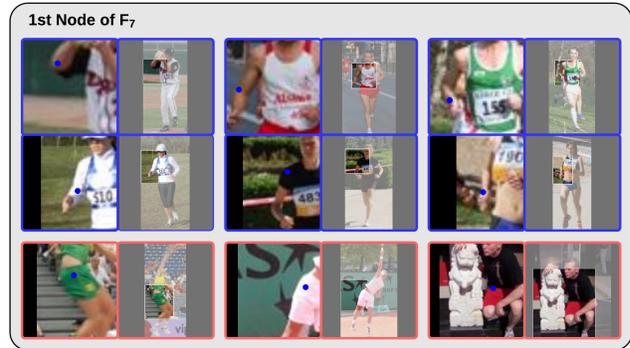


Figure 7. Visualization of the features extracted by layer F7 in DS-CNN.

In our experiments, we set  $\mu_1 = 0.1$ ,  $\mu_2 = 1.0$ ,  $\lambda_d = 4$ ,  $k = 3$ , and  $\lambda_h = 0.9$ .

In this paper, we use the open-source CNN library Caffe [39] for implementing DS-CNN. We finetune a CNN network pretrained on ImageNet [29] for training the proposed DS-CNN. Following [17], the learning rate is initialized to a tenth of the initial ImageNet learning rate and is decreased by a factor of ten after every certain number of iterations.

We first evaluate our method on LSP dataset. The PCP of the proposed method, DeepPose and six other comparison methods for head, torso, and four limbs (upper/lower arms and upper/lower legs) is shown in Table 1. Except for ‘head’, our method outperforms all the comparison methods including DeepPose at all body parts. The improvement on average PCP is over 15% against the best results obtained by the comparison methods.

Figure 5 shows the PDJ curves of the proposed method and seven comparison methods at the elbows and wrists on the FLIC dataset [16, 15, 33, 4, 11, 34, 21]. We can see that the proposed method outperforms all the comparison methods except for Tompson et al. Tompson et al’s PDJ performance is higher than the proposed method, when normalized distance to true joint, or for brevity, normalized distance, is less than a threshold  $t$ , but a little lower than the proposed method when normalized distance is larger than  $t$ . As shown in Fig. 5, the value of  $t$  is 0.15 and 0.18 for elbows and wrists respectively.

As a further note, Tompson et al. [34] combines an MRF-based graphic model into CNN-based part detection. It shows that the inclusion of the graphical model can substantially improve the PDJ performance. In this paper, we focus on developing a new CNN-based method to detect local parts without using any high-level graphical models. We believe the PDJ performance of the proposed method can be further improved if we combine it to a graphical model as in Tompson et al.

Performance comparison on LSP dataset using PDJ metric is shown in Fig. 6. Similar to PDJ comparison on FLIC, the PDJ of the proposed method is better than all the com-

LSP	ankle	knee	hip	wrist	elbow	shoulder	neck	head	mAP	FLIC	hip	wrist	elbow	shoulder	Head	mAP
$\mathbf{p}_p$	35.7	25.5	27.3	20.7	17.1	35.0	47.9	70.3	31.5	$\mathbf{p}_p$	61.2	56.0	71.2	88.8	93.8	72.0
$\mathbf{p}_b$	39.7	39.6	37.5	21.3	29.3	40.7	44.4	70.4	37.9	$\mathbf{p}_b$	72.8	59.3	77.7	91.0	94.0	77.2
$\mathbf{p}_{p,b}$	<b>44.6</b>	<b>41.9</b>	<b>41.8</b>	<b>30.4</b>	<b>34.2</b>	<b>48.7</b>	<b>58.9</b>	<b>79.6</b>	<b>44.4</b>	$\mathbf{p}_{p,b}$	<b>74.3</b>	<b>68.1</b>	<b>82.0</b>	<b>93.5</b>	<b>96.4</b>	<b>81.4</b>

Table 2. Average precision (%) of joint detection on LSP and FLIC testing datasets when CNN takes different types of patches as input.



Figure 8. Human pose estimation on sample images from FLIC and LSP testing datasets.

parison methods except for Tompson et al. When compared with Tompson et al, the proposed method performs substantially better when normalized distance is large and performs worse when normalized distance is small. One observation is that the PDJ gain of the proposed method over Tompson et al. at large normalized distance in LSP is more significant than the same gain in FLIC.

We also conduct an experiment to verify the effectiveness of using dual sources of inputs:  $\mathbf{p}_p$  and  $\mathbf{p}_b$ . In this experiment, we compute the average precision (AP) of the joint detection when taking either 1) only part patches  $\mathbf{p}_p$ , 2) only body patches  $\mathbf{p}_b$ , or 3) the proposed patch pairs  $\mathbf{p}_{p,b}$  as the input to CNN. The results are shown in Table 2. On both LSP and FLIC testing datasets, the use of the dual-source patch-pairs achieves better AP at all joints, and the best mAP, the average AP over all the joints. Note that the body patch  $\mathbf{p}_b$  in this paper actually include part patch information, in the form of a binary mask as discussed in Section 4. That’s why the use of only  $\mathbf{p}_b$  can lead significantly better AP than the use of only  $\mathbf{p}_p$  on both LSP and FLIC testing datasets. However, the binary mask is usually of very low resolution because we normalize the body patch to a fixed dimension. As a result, we still need to combine  $\mathbf{p}_p$  and  $\mathbf{p}_b$  and construct a dual-source CNN for pose estimation.

Following [17, 40], we visualize the patterns extracted by DS-CNN. We compute the activations in each hidden node in layer  $F_7$  on a set of patch-pairs and Figure 7 shows several patch pairs with the largest activations in the first node of  $F_7$ . We can see that this node fires for two pose patterns – the bent right elbow and the right hip. For each pattern, the corresponding full-body pose also show high similarity because of the inclusion of both part and body patches in DS-CNN.

Finally, sample human pose estimation results on FLIC and LSP testing datasets are shown in Fig. 8. In general, upper-body poses in FLIC are usually front-facing, while the full-body pose in LSP contains many complex poses. As a result, human pose estimation in LSP is less accurate than that in FLIC. By including holistic views of part patches, the proposed method can estimate the human pose even if some joints are occluded, as shown in Fig. 8.

## 9. Conclusion

In this paper, we developed a new human pose estimation method based on a dual-source convolutional neural network (DS-CNN), which takes two kinds of patches – part patches and body patches – as inputs to combine both local and contextual information for more reliable pose estimation. In addition, the output of DS-CNN is designed for both joint detection and joint localization, which are combined for estimating the human pose. By testing on the FLIC and LSP datasets, we found that the proposed method can produce superior performance against several existing methods. When compared with Tompson et al [34], the proposed method performs better when normalized distance is large and performs worse when normalized distance is small. The proposed method is implemented using the open-source CNN library Caffe and therefore has good expandability.

**Acknowledgement:** This work was supported in part by AFOSR FA9550-11-1-0327 and NSF IIS-1017199.

## References

- [1] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *IJCV*, 2005.
- [2] Y. Tian, C. L. Zitnick, and S. G. Narasimhan, "Exploring the spatial hierarchy of mixture models for human pose estimation," in *ECCV*, 2012.
- [3] F. Wang and Y. Li, "Beyond physical connections: Tree models in human pose estimation," in *CVPR*, 2013.
- [4] M. Eichner and V. Ferrari, "Appearance sharing for collective human pose estimation," in *ACCV*, 2012.
- [5] K. Duan, D. Batra, and D. J. Crandall, "A multi-layer composite model for human pose estimation," in *BMVC*, 2012.
- [6] M. Sun and S. Savarese, "Articulated part-based model for joint object detection and pose estimation," in *ICCV*, 2011.
- [7] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, "Poselet conditioned pictorial structures," in *CVPR*, 2013.
- [8] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *CVPR*, 2009.
- [9] L. Pishchulin, A. Jain, M. Andriluka, T. Thormahlen, and B. Schiele, "Articulated people detection and pose estimation: Reshaping the future," in *CVPR*, 2012.
- [10] S. Johnson and M. Everingham, "Clustered pose and non-linear appearance models for human pose estimation," in *BMVC*, 2010.
- [11] B. Sapp, A. Toshev, and B. Taskar, "Cascaded models for articulated pose estimation," in *ECCV*, 2010.
- [12] M. Dantone, J. Gall, C. Leistner, and L. Van Gool, "Human pose estimation using body parts dependent joint regressors," in *CVPR*, 2013.
- [13] V. K. Singh, R. Nevatia, and C. Huang, "Efficient inference with multiple heterogeneous part detectors for human pose estimation," in *ECCV*, 2010.
- [14] Y. Wang and G. Mori, "Multiple tree models for occlusion and spatial constraints in human pose estimation," in *ECCV*, 2008.
- [15] B. Sapp and B. Taskar, "Modec: Multimodal decomposable models for human pose estimation," in *CVPR*, 2013.
- [16] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," *CVPR*, 2014.
- [17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.
- [18] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for fine-grained category detection," in *ECCV*, 2014.
- [19] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on Computers*, 1973.
- [20] M. Eichner and V. Ferrari, "Better appearance models for pictorial structures," in *BMVC*, 2009.
- [21] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *CVPR*, 2011.
- [22] H. Jiang and D. R. Martin, "Global pose estimation using non-tree models," in *CVPR*, 2008.
- [23] T.-P. Tian and S. Sclaroff, "Fast globally optimal 2D human detection with loopy graph models," in *CVPR*, 2010.
- [24] X. Ren, A. C. Berg, and J. Malik, "Recovering human body configurations using pairwise constraints between parts," in *ICCV*, 2005.
- [25] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *NIPS*, 1990.
- [26] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *ICCV*, 2009.
- [27] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *CVPR*, 2004.
- [28] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *ICML*, 2009.
- [29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.
- [30] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *NIPS*, 2013.
- [31] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," in *ICLR*, 2014.
- [32] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *IJCV*, 2013.
- [33] A. Jain, J. Tompson, M. Andriluka, G. W. Taylor, and C. Bregler, "Learning human pose estimation features with convolutional networks," *ICLR*, 2014.
- [34] J. 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.
- [35] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *ECCV*, 2014.
- [36] S. Johnson and M. Everingham, "Learning effective human pose estimation from inaccurate annotation," in *CVPR*, 2011.
- [37] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, "Strong appearance and expressive spatial models for human pose estimation," in *ICCV*, 2013.
- [38] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari, "2D articulated human pose estimation and retrieval in (almost) unconstrained still images," *IJCV*, 2012.
- [39] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACMMM*, 2014.
- [40] W. Ouyang, X. Chu, and X. Wang, "Multi-source deep learning for human pose estimation," in *CVPR*, 2014.