Human Pose Estimation Using Deep Consensus Voting

Ita Lifshitz, Ethan Fetaya(✉), and Shimon Ullman

Weizmann Institute of Science, Rehovot, Israel
ethan.fetaya@weizmann.ac.il

Abstract. In this paper we consider the problem of human pose estimation from a single still image. We propose a novel approach where each location in the image votes for the position of each keypoint using a convolutional neural net. The voting scheme allows us to utilize information from the whole image, rather than rely on a sparse set of keypoint locations. Using dense, multi-target votes, not only produces good keypoint predictions, but also enables us to compute image-dependent joint keypoint probabilities by looking at consensus voting. This differs from most previous methods where joint probabilities are learned from relative keypoint locations and are independent of the image. We finally combine the keypoints votes and joint probabilities in order to identify the optimal pose configuration. We show our competitive performance on the MPII Human Pose and Leeds Sports Pose datasets.

1 Introduction

In recent years, with the resurgence of deep learning techniques, the accuracy of human pose estimation from a single image has improved dramatically. Yet despite this recent progress, it is still a challenging computer vision task and state-of-the-art results are far from human performance.

The general approach in previous works, such as [22,26], is to train a deep neural net as a keypoint detector for all keypoints. Given an image $\mathcal{I}$, the net is fed a patch of the image $\mathcal{I}_y \subset \mathcal{I}$ centered around pixel y and predicts if y is one of the M keypoints of the model. This process is repeated in a sliding window approach, using a fully convolutional implementation, to produce M heat maps, one for each keypoint. Structured prediction, usually by a graphical model, is then used to combine these heat maps into a single pose prediction. This approach has several drawbacks. First, most pixels belonging to the person are not themselves any of the keypoints and therefore contribute only limited information to the pose estimation process. Information from the entire person can be used to get more reliable predictions, particularly in the face of partial

I. Lifshitz and E. Fetaya—Equal contribution.

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46475-6_16) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part II, LNCS 9906, pp. 246–260, 2016.
DOI: 10.1007/978-3-319-46475-6_16

occlusion where the keypoint itself is not visible. Another drawback is that while the individual keypoint predictors use state-of-the-art classification methods to produce high quality results, the binary terms in the graphical model, enforcing global pose consistency, are based only on relative keypoint location statistics gathered from the training data and are independent of the input image.

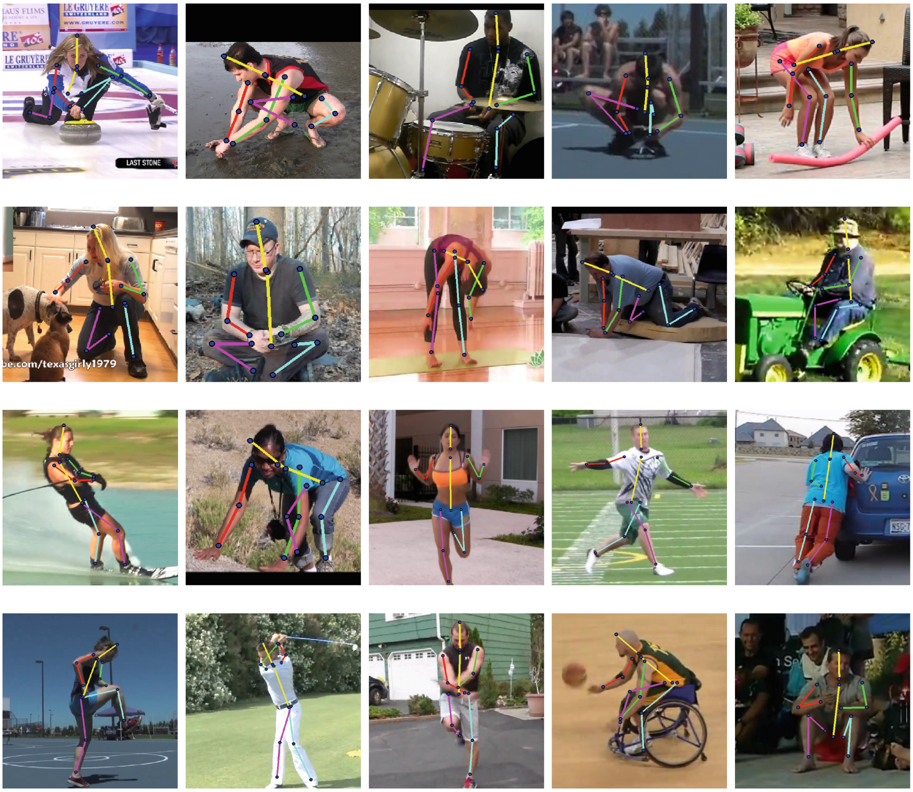


Fig. 1. Our model's predicted pose estimation on the MPII-human-pose database test-set [1]. Each pose is represented as a stick figure, inferred from predicted joints. Different limbs in the same image are colored differently, same limb across different images has the same color. (Color figure online)

To overcome these limitations, we propose a novel approach in which for every patch center y we predict the location of all keypoints relative to y, instead of classifying y as one of the keypoints. This enables us to use 'wisdom of the crowd' by aggregating many votes to produce accurate keypoint detections. In addition, by looking at agreements between votes, we infer informative image-dependent binary terms between keypoints. Our binary terms are generated by consensus voting - we look at a set of keypoints pairs, and for each possible pair of locations, we aggregate all votes for this combination. The total vote will be high if both keypoint locations get strong votes from the same voters. We show that this approach produces competitive results on the challenging MPII human-pose [21] and the Leeds sports pose [12] datasets.

2 Related Work

Human body pose estimation from a single image is a challenging task. The need to cope with a variety of poses in addition to a large range of appearances due to different clothing, scales and light conditions gave rise to many approaches to dealing with the various aspects of this task. One common approach is using part detectors combined with a pictorial structure for capturing relations between parts [8,28]. In addition to the standard pictorial structure [9], some methods incorporate higher-order part dependencies such as poselet-based features [21]. Alternative methods, like the chains-model [13] replace the pictorial structure with a set of voting chains, starting at the head keypoint and ending at the hand.

With the reappearance of convolutional neural nets, part detectors became more reliable, leading to a significant improvement in accuracy [3,5,22,25–27]. The works of [25,26] focus on multi-scale representation of body parts in order to infer the location of keypoints. In [22], the authors deal with simultaneous detection and pose estimation of multiple people. Recent works [3,27] use an iterative scheme to refine pose estimation. We use a neural net to generate votes on location of different keypoints. As opposed to [5], where a neural net is used to explicitly generate an image dependent binary term, in our model it arises naturally from the voting scheme. The use of voting in detecting targets of interest has been commonly used in various computer vision tasks, for example, in shape detection [2,16], object detection [10,18] and body parts detection [20].

3 Overview of the Method

We now describe the main parts of our algorithm, which will be explained in detail in next sections.

At test time, we first use a deep neural net, described in Sect. 4.2, to predict for each image patch $\mathcal{I}_y$ centered around pixel y, and for each keypoint j, the location of the keypoint relative to y. From this we can compute the probability of keypoint location K_j being equal to a possible location x as seen from $\mathcal{I}_y$, $P_y(K_j = x)$. We aggregate these votes over all image patches to get the probability distribution for each keypoint location $\{P(K_j = x)\}_{j=1}^{M}$. Examples of $P_y(K_j)$ and $P(K_j)$ are shown in Figs. 3(a)–(d).

Next we compute our consensus voting binary term. The voting net above was trained using a separate loss per keypoint, which is equivalent to an independence assumption, i.e. for each y,

$$P_y(K_i = x_i, K_j = x_j) = P_y(K_i = x_i) \cdot P_y(K_j = x_j). \tag{1}$$

If we now average over all locations y we get a joint distribution

$$P(K_i = x_i, K_j = x_j) \propto \sum_y P_y(K_i = x_i) \cdot P_y(K_j = x_j) \qquad (2)$$

in which the keypoints are no longer independent. Because of the multiplication, the joint distribution is high when both locations get strong votes from the *same voters*. More details on the consensus voting can be found in Sect. 5.

Finally, we estimate the pose by minimizing an energy function over the unary and binary terms generated by the voting scheme. We do this sequentially, focusing at each step on a subset of keypoints. We start with the most reliable keypoints until making the full pose estimation. This process is presented in more details in Sect. 6.

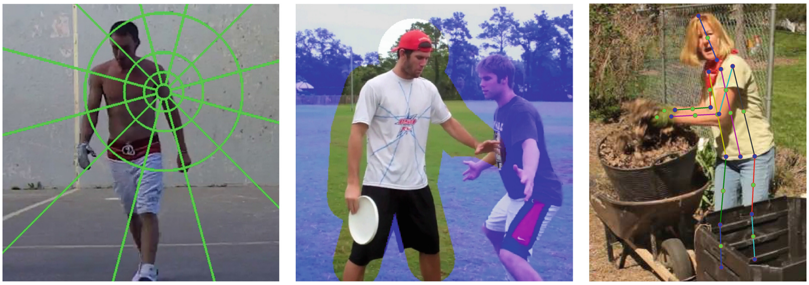
4 Keypoint Voting

4.1 Voting Representation

The first stage in our pose-estimation method is a keypoint detector learned by a deep neural net, which we apply to the image patches in a fully convolutional fashion [17]. This is an accelerated way to run the net in a sliding window manner with the stride determined by pooling layers of the network. At each patch center y, the net predicts the location of each keypoint $K_1, ..., K_M$ relative to y. This differs from previous methods [25,26] where the net only needed to classify if the center y is any of the M keypoints.

The problem of predicting the relative displacement vectors $\{K_1 - y, ..., K_M - y\}$ is a regression problem which is usually solved by minimizing an L_2 loss function. However, for the current task the L_2 loss has shortcomings, as the net produces only a single prediction as output. Consequently, in cases of ambiguity, e.g. difficulty to distinguish between left and right hand, the optimal L_2 loss would be the mean location instead of "splitting" the vote between both locations. Indeed, when trained using this approach, we found that the net performance to be lackluster. To better address this issue, we modify the prediction task into a classification problem by discretizing the image into log-polar bins centered around y, as seen in Fig. 2(a). Using log-polar binning allows for more precise predictions for keypoints near y and a rough estimate for far away keypoints. We classify into 50 classes, one for the central location, one for background i.e. non-person, and the rest are four log-polar rings around the center with each ring divided into 12 angular bins. Since not all people in the training set are labeled, we are unable to use background locations for training the non-person label. For this reason, we ignore image locations far from the person of interest, as seen in Fig. 2(b), and use non-person images from the PASCAL dataset [6] for non-person label training.

We augmented the 16 humanly annotated keypoints supplied in the dataset with additional 12 keypoints generated from the original ones, by taking the middle point between two neighboring skeleton keypoints, e.g. the middle point



(a) Log-polar bins (b) Ignore mask (c) Keypoints

Fig. 2. (a) Log-polar bins, used for keypoints locations voting, centered around the left upper arm; (b) Patch centers outside the person of interest, marked in blue, are not used for training; (c) Our model makes use of 30 keypoints: 16 annotated body joints, supplied in the dataset, 12 synthetically generated mid-section keypoints and estimated hands locations. Original keypoints marked in blue, synthetically generated keypoints in green (Color figure online)

between the shoulder and elbow. We also obtained estimated location of the hands by extrapolating the elbow-wrist vector by 30 %. This produces a total of 30 keypoints and allows us to have a more dense coverage of the body. All keypoints can be seen in Fig. 2(c).

4.2 Net Architecture

The net architecture we use is based on the VGG-16 network [24]. We use the first 13 convolutional layers parameters of the VGG-16 net (denoted by columns 1–9 in Table 1(a)), pre-trained on the imagenet dataset [23]. On top of these layers we employ a max pooling with stride 1 and use convolution with holes [4] in order to increase resolution and generate a descriptor of size 2048 every 8 pixels. To further increase resolution we learn a deconvolution layer [17] and get a probability distribution over 50 classes, indicating the relative keypoint location, every 4 pixels. The last two layers are distinct per keypoint resulting in 30 distributions on 50 bins every 4 pixels. More details about the structure of the net can be found in Table 1. Since the size of each classification bin is considerably different, we minimize a weighted softmax loss function, where each class is weighted inversely proportional to the size of its bin.

4.3 The Voting Scheme

At each patch center y and for each keypoint j the network returns a softmax probability distribution over the log-polar bins $s_y^j \in \mathbb{R}^{1\times 1\times C}$. At inference we use deconvolution, with a predefined fixed kernel $w \in \mathbb{R}^{H_k\times W_k\times C}$, to transform the log-polar representation s_y^j back to the image space and get the probability distribution of keypoint location over pixels $P_y(K_j = x)$. The deconvolution kernel maps each channel, representing a log-polar bin, to the corresponding image

Table 1. Comparisons between the network architectures of VGG_{16} and *deep voting* (our model), as shown in (a) and (b). Each table contains five rows, representing the 'name of layer', 'receptive field of filter - stride', 'number of output feature maps', 'activation function' and 'size of output feature maps', respectively. The terms 'conv', 'max', 'fc', 'hconv' and 'deconv' represent the convolution, max pooling, fully connection, convolution with holes [4] and deconvolution [17], respectively. The terms 'relu', 'idn', 'soft' and 'w-soft' represent the activation functions, rectified linear unit, identity, softmax and weighted-softmax, respectively. The last two layers in (b) are distinct per keypoint.

(a) VGG_{16}: 244x244x3 input image; 1x1000 output labels

	1	2	3	4	5	6	7	8	9	10	11	12	13	
layer	2 x conv	max	2 x conv	max	3 x conv	max	3 x conv	max	3 x conv	max	fc	fc	fc	
filter-stride	3-1	2-2	3-1	2-2	3-1	2-2	3-1	2-2	3-1	2-2	-	-	-	
channels	64	64	128	128	256	256	512	512	512	512	4096	4096	1000	
activation	relu	idn	relu	idn	relu	idn	relu	idn	relu	idn	relu	relu	soft	
size	224	112	112	56	56	28	28	14	14	7	1	1	1	

(b) *deep voting*: 504x504x3 input image; 102x102x50x30 output label maps

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
layer	$2 \times conv$	max	$2 \times conv$	max	$3 \times conv$	max	$3 \times conv$	max	$3 \times hconv$	max	$hconv$	$conv$	$conv_{K_i}$	$deconv_{K_i}$
filter-stride	3-1	2-2	3-1	2-2	3-1	2-2	3-1	2-1	3-1	3-1	7-1	1-1	1-1	6-2
channels	64	64	128	128	256	256	512	512	512	512	2048	2048	50	50
activation	relu	idn	relu	idn	relu	idn	relu	idn	relu	idn	relu	relu	idn	w-soft
size	504	252	252	126	126	63	63	62	62	60	50	50	50	102

locations. We use a deconvolution kernel of size $(H_k \times W_k \times C) = (65 \times 65 \times 50)$ to efficiently aggregate votes from a neighborhood of 65 × 65 locations. Most of the kernel weights are zeros, shown as black in Fig. 3(e). At the top of the figure we show an illustration of weights for a specific bin. Since this bin corresponds to the upper left log-polar segment it is zero at all locations except for the pixels of that segment which are set to $\frac{1}{|bin|}$.

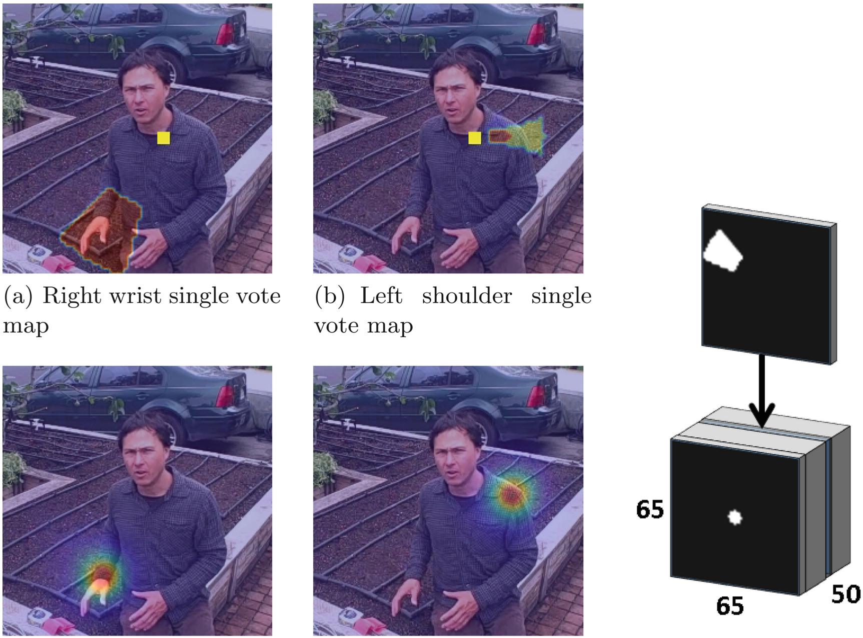
$$\hat{P}_y(K_j = x) = deconv(s_y^j, w) \tag{3}$$

Then $P_y(K_j)$ is simply $\hat{P}_y(K_j)$ translated by y. Examples for $P_y(K_j)$ are shown in Figs. 3(a) and (b). We aggregate these votes over all patch centers to get the final probability of each keypoint at each location.

$$P(K_j = x) = \sum_{y \in \mathcal{Y}} P_y(K_j = x) = deconv(s^j, w) \tag{4}$$

The term $P(K_j) \in \mathbb{R}^{(\frac{H}{4}+H_k-1)\times(\frac{W}{4}+W_k-1)}$ is the aggregated votes of keypoint j, and $s^j \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C}$ is the softmax distribution output of keypoint j. Examples for $P(K_j)$ are shown in Fig. 3(c) and (d). Note that the size of the aggregated distribution is larger than $\frac{H}{4} \times \frac{W}{4}$, the image size at the output resolution. This enables the algorithm to generate votes for locations outside the visible image.

During inference, we make use of the Caffe [11] implementation for deconvolution by adding an additional deconvolution layer with fixed weights w on



(a) Right wrist single vote map

(b) Left shoulder single vote map

(c) Right wrist probability

(d) Left shoulder probability

(e) Vote aggregation deconvolution kernel

Fig. 3. (a) Voting map from location y (yellow rectangle) for Right Wrist $P_y(K_{RightWrist})$; (b) Voting map from location y (yellow rectangle) for Left Shoulder $P_y(K_{LeftShoulder})$; (c) Right wrist probability $P(K_{RightWrist})$, generated by aggregating voting maps for right wrist; (d) Left shoulder probability $P(K_{LeftShoulder})$, generated by aggregating voting maps for left shoulder; (e) The vote aggregation deconvolution kernel w. The weights of a specific channel are illustrated at the top (Color figure online)

top of the net softmax output. This layer generates the aggregated probability distribution of each keypoint in a fast and efficient way.

5 Consensus Voting

The voting scheme described in Sect. 4 produces a heat map per keypoint, which represents the probability of a keypoint being at each location. The next challenge is to combine all the estimated distributions into a single pose prediction. What makes this task especially challenging is that beyond just having to handle false keypoint detections, we need to handle true detections but of other individuals in the image as well. Other challenges include self-occlusion, and confusion between left and right body parts.

The standard approach to solve this problem is to minimize an objective of the following form:

$$\sum_{i=1}^{N} \phi_i(x_i) + \sum_{(i,j)\in E} \phi_{(i,j)}(x_i, x_j) \tag{5}$$

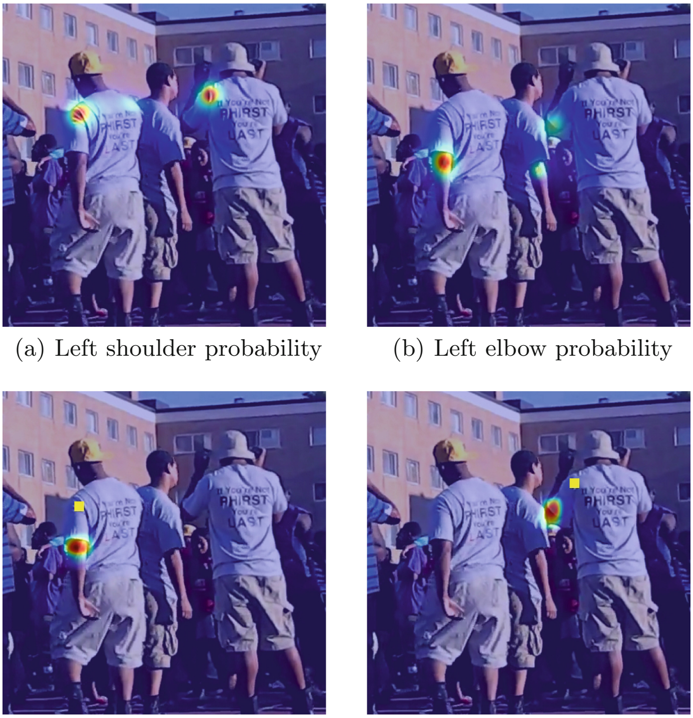
The $\phi_i(x_i)$ unary term is the score of having keypoint i at location x_i. The $\phi_{(i,j)}(x_i, x_j)$ binary term measures compatibility and is the score of having keypoint i at location x_i and keypoint j at location x_j. The edge set E is some preselected subset of pairs of keypoints.

Usually in pose estimation, the binary term is image independent and comes from a prior on relative keypoint location. For example, given the location of the shoulder, it produces a set of possible elbow displacements based on training samples pose statistics. While this can be helpful in excluding anatomically implausible configurations, an image independent binary term has limitations. For example, the location of the left shoulder relative to the head is strongly dependent on whether the person is facing forwards or backwards and this information is not incorporated into the standard binary term. One main advantage of our keypoint voting scheme is that we can compute from it an image-based "consensus voting" binary term, which has more expressive power. This is especially important for less common poses, where the image-independent prior gives a low probability to the right pose.

At each image location y, we compute from the net output the distribution $P_y(K_i = x)$ for each keypoint. These probabilities were trained under a (naive) independence assumption, i.e. a separate softmax loss for each keypoint. If we now average over all locations y, we get a joint distribution which is no longer independent (see Eq. 2). By having each center y vote for a combination of keypoints, the probabilities become dependent through the voters. For $P(k_i = x_i, k_j = x_j)$ to be high, it is not enough for x_i and x_j to receive strong votes separately, the combination needs to get strong votes from many common voters. For this reason we call this the consensus voting term. In our algorithm, we use the unary term of the form $\phi_i(x_i) = -\log(P(K_i = x_i))$. The binary term $\phi_{(i,j)}(x_i, x_j)$ we use is a weighted combination of the joint distribution $-\log(P(k_i = x_i, k_j = x_j))$ just described and the commonly used binary term based on relative keypoint location statistics.

In Fig. 4 we show an example of the conditional probability $P(K_i = x_i|K_j = x_j)$ calculated from the previously described joint probability. As can be seen in Fig. 4(b), the left elbow of the person to the right has a weak response. In addition, there is a misleading weak response to the elbow of another individual nearby. After conditioning on the left shoulder location, we see a strong response in Fig. 4(d), but only in the correct location.

Computing the consensus voting is challenging, since calculating Eq. 2 in a naive way is computationally expensive. Each keypoint has N^2 possible locations, where the image is of size $N \times N$. Considering all possible combinations yields N^4 pairs of locations. For each pair of possible locations, we sum over all N^2 voters, resulting in $\mathcal{O}(N^6)$ running time. In order to reduce running time, we use the



(a) Left shoulder probability

(b) Left elbow probability

(c) Conditional probability

(d) Conditional probability

Fig. 4. (a) Left shoulder probability $P(K_{LeftShoulder})$; (b) Left elbow probability $P(K_{LeftElbow})$; (c) Conditional probability of left elbow given left shoulder of person on the left (yellow rectangle); (d) Conditional probability of left elbow given left shoulder of person on the right (yellow rectangle) (Color figure online)

observation that Eq. 2 is in fact a convolution and use the highly optimized Caffe GPU implementation [11] to calculate the binary tables. In addition we calculate the binary term over a coarse scale of 1/12 of the image scale, using only the first two log-polar rings. This reduces the running time of a single keypoint pair to $\sim$100 ms. The restriction to the first two rings limits the maximal distance between keypoints, which we overcome by using the augmented keypoints shown in Fig. 2(c).

6 Pose Prediction

In the previous sections, we described the unary and binary terms which are the basic building blocks of our algorithm. We now present additional steps that we employ to improve performance on top of these basic building blocks. First, we add geometrical constraints on our augmented keypoints. Second, we perform the inference in parts, starting from the reliable parts proceeding to the less certain ones.

6.1 Local Geometric Constraints

We generate additional keypoints, as seen in Fig. 2(c) by taking the mid-point between keypoints, e.g. shoulder and elbow. While we could simply minimize Eq. 5 with these added variables as well, this fails to take into account the fact that these new points are determined by other points. We can enforce these constraints by removing the new synthetic variables and rewriting our binary constraints. Assume our two original keypoints had indexes i and j and the middle point had index ℓ. Focusing on the relevant terms, instead of solving

$$\min_{x_i, x_j, x_\ell} \phi_i(x_i) + \phi_j(x_j) + \phi_\ell(x_\ell) + \phi_{(i,\ell)}(x_i, x_\ell) + \phi_{(\ell,j)}(x_\ell, x_j) \tag{6}$$

we add the geometric constraint by solving

$$\min_{x_i, x_j} \phi_i(x_i) + \phi_j(x_j) + \tilde{\phi}_{(i,j)}(x_i, x_j). \tag{7}$$

Where we define

$$\tilde{\phi}_{(i,j)}(x_i, x_j) = \phi_\ell(f(x_i, x_j)) + \phi_{(i,\ell)}(x_i, f(x_i, x_j)) + \phi_{(\ell,j)}(f(x_i, x_j), x_j)$$

and $f(x, y) = \frac{1}{2}(x + y)$.

This is equivalent to adding the constraint that x_ℓ is the mid-point between x_i and x_j, but faster to optimize. By adding this mid-point constraint, and using it as a linking feature [14], we get a more reliable binary term which also looks at the appearance of the space between the two respective keypoints.

6.2 Sequential Prediction

An issue that arises when optimizing Eq. 5 over all keypoint is that not all keypoints are detected with equal accuracy. Some, like the head, are detected with high accuracy while others, like the wrist, are more difficult to locate. In some cases, occluded or unclear keypoints can distort predictions of more visible keypoints. In order to have the more certain keypoints influence the prediction of less certain ones, but not vice versa, we predict the keypoints in stages. We start with the most reliable keypoints, and at each stage use the previously predicted keypoints as an "anchor" to predict the other parts. Once "anchor" points are set, we allow them only small refinement movements in subsequent stages. Optimizing Eq. 5 is performed at three stages. At the first stage we predict the location of the head, neck, thorax and pelvis. At the second stage we predict the shoulders and hips given all previously predicted keypoints. Last, we locate all remaining keypoints.

The optimization is done by the TRW-S algorithm [15] at each of the previously discussed stages. The hyper-parameters, i.e. the relative weights of the unary and binary terms, where tuned by using a grid search on a held out validation set containing 3330 annotated poses.

7 Results

7.1 MPII

We tested our method on the MPII human pose dataset [1], which consists of 19,185 training and 7,247 testing images of various activities containing over $40K$ annotated people. The dataset is highly challenging and has people in a wide array of poses. At test time we are given an image with a rough location and scale of the person of interest and need to return the location of 16 keypoints: head-top, upper-neck, thorax, pelvis, shoulders, elbows, wrist, hips, knees and ankles. The standard evaluation is made on a subset of test images, named “single person”, where the person of interest is well separated from other people. We note that several images in the “single person” dataset still have another person nearby.

We trained the net described in Sect. 4 using Caffe [11] in a cascaded fashion. First, training the added layers, denoted by columns 11–14, in Table 1(b), while keeping the first 13 convolutional layers with learning rate 0. Then training layers denoted by columns 9–14 in Table 1(b), and finally fine tuning the entire network. We start with learning rate of 0.001 and no weight decay and continue with learning rate of 0.0001 and 0.0002 weight decay.

At test time, in order to restrict our algorithm to the person in question, we crop a window of size 504 × 504 around the person using the given position and scale, adding zero padding if needed. In addition, we multiply our mid-body keypoint heatmap (the synthetic point between thorax and pelvis) with a mask centered around the given person position, which is part of the standard input in the “single person” challenge. This zeros responses far from the given center and insures that we return the pose of the correct person. Examples of our model’s predictions are shown in Fig. 1.

Performance is measured by the PCKh metric [1], where a keypoint location is considered correct if its distance to the ground truth location is no more than half the head segment length. In Table 2 we compare our results to the leading methods on the MPII human pose leaderboard. We show competitive results with mean PCKh score of 85.0 % and state-of-the-art performance on the head

Table 2. PCKh results on the MPII single person dataset.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Mean
Tompson et al. [26]	95.8	90.3	80.5	74.3	77.6	69.7	62.8	79.6
Carreira et al. [3]	95.7	91.7	81.7	72.4	82.8	73.2	66.4	81.3
Tompson et al. [25]	96.1	91.9	83.9	77.8	80.9	72.3	64.8	82.0
Pishchulin et al. [22]	94.1	90.2	83.4	77.3	82.6	75.7	68.6	82.4
Wei et al. [27]	**97.8**	95.0	88.7	84.0	88.4	82.8	79.4	88.5
Newell et al. [19]	97.6	**95.4**	**90.0**	**85.2**	**88.7**	**85.0**	**80.6**	**89.4**
Our model	**97.8**	93.3	85.7	80.4	85.3	76.6	70.2	85.0

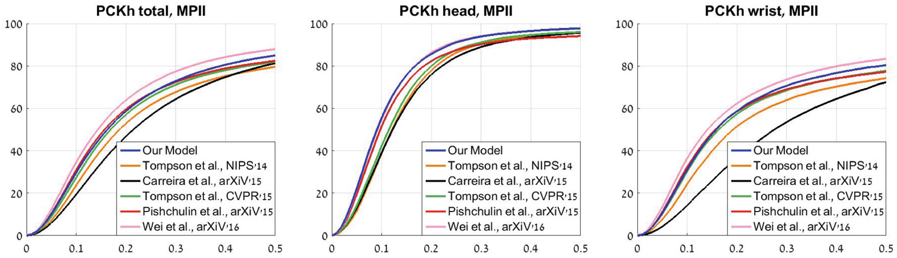


Fig. 5. Pose estimation results on the MPII dataset for varying PCKh thresholds.

keypoints. A detailed ablation study and further discussion of the results appear in the supplementary material.

7.2 Leeds Sports

The Leeds sports dataset [12] (LSP) contains 2,000 images of people in various sport activities, 1,000 for training and 1,000 for testing. The task is to return 14 keypoints, the same keypoints as in the MPII dataset except for the thorax and pelvis.

The LSP dataset has two evaluation settings, person-centric (PC) and observer-centric (OC). We use the person-centric settings where right/left labels are according to body parts (same as the MPII annotation) and not according to relative image location. The standard performance measure for the LSP dataset is strict percentage of correct parts (PCP) metric. The PCP score measures limb detection: a limb is considered to be correctly detected if the distances between the detected limb endpoints and groundtruth limb endpoints are within half of the limb length.

We use the model trained on the MPII human pose dataset and fine-tune it on the LSP training set. Examples of our model's predictions are shown in Fig. 6. At test time we run our algorithm twice, once with the input image and once with it flipped up-side-down, and pick the pose with the optimal score. This is done in order to handle up-side-down people which are more common in the LSP dataset than the MPII dataset, and are therefore under-represented at training time. Additional results on the LSP dataset, without flipping images up-side-down, can be found in the supplementary material.

We compare the performance of our approach to that of other leading pose estimation methods in Table 3. Our performance is comparable to that of Pishchulin et al. [22], and superior to other methods. We note that in [22] the authors use the LSP-Extended dataset, containing additional 10,000 annotated poses, not used in our model.

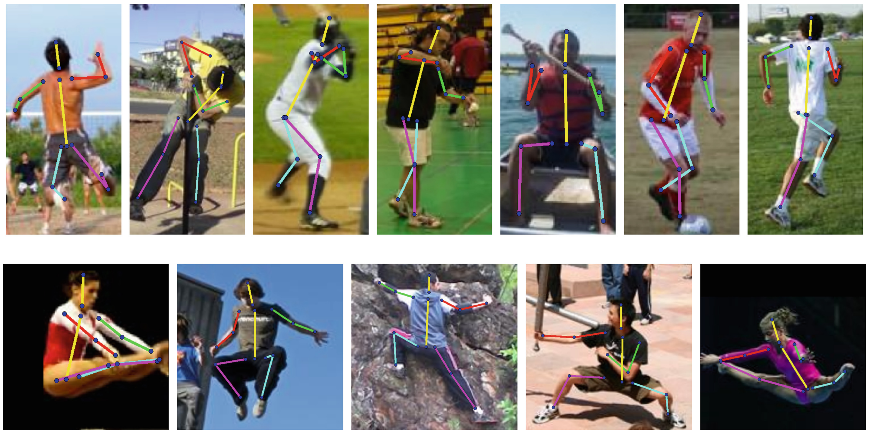


Fig. 6. Additional results of our model's predicted joint positions on the LSP database test-set [12]

Table 3. PCP results on the LSP dataset (PC).

Method	Torso	Upper leg	Lower leg	Upper arm	Forearm	Head	Mean
Tompson et al. [26]	90.3	70.4	61.1	63.0	51.2	83.7	66.6
Carreira et al. [3]	95.3	81.8	73.3	66.7	51.0	84.4	72.5
Chen and Yuille [5]	96.0	77.2	72.2	69.7	58.1	85.6	73.6
Fan et al. [7]	95.4	77.7	69.8	62.8	49.1	86.6	70.1
Pishchulin et al. [22]	97.0	88.8	82.0	82.4	71.8	95.8	84.3
Wei et al. [27]	**98.0**	82.2	**89.1**	**85.8**	**77.9**	**95.0**	**88.3**
Our model	97.3	**88.8**	84.5	80.6	71.4	94.8	84.3

8 Discussion

In this work, we proposed a method for dealing with the challenging task of human pose estimation in still images. We presented a novel approach of using a deep convolutional neural net for keypoint voting rather than keypoint detection. The keypoint voting scheme has several useful properties compared to keypoint detection. First, all image regions of the evaluated person participate in the voting, utilizing the 'wisdom of the crowd' to produce reliable keypoint predictions. Second, any informative location can contribute to multiple keypoints, not just to a single one. This allows us to use consensus voting in order to compute expressive image-dependent joint keypoint probabilities. Empirical results on the diverse MPII human pose and Leeds sports pose datasets show competitive results, improving the state-of-the-art on a subset of evaluated keypoints. We showed that our model generalized well from the MPII dataset to the LSP dataset, using only 1000 samples for fine tuning. Models and code will be made publicly available. Additional contributions of the current scheme are the use of log-polar bins for location prediction rather than estimating L2 translations, and the use of convolutions for fast aggregation of votes from multiple locations.

The voting scheme is a natural tool for estimating the locations of unobserved body parts. In future work, we plan to harness this property for dealing with multiple people and occlusions, resulting from closely interacting people, which are difficult to handle by existing schemes. In addition, we plan to combine our voting scheme with iterative methods, refining predictions by feeding the output of previous iterations as input to the network.

Acknowledgment. This work was supported by ERC Advanced Grant 269627 Digital Baby and in part by the Center for Brains, Minds and Machines, funded by National Science Foundation Science and Technology Centers Award CCF-1231216. We thank the NVIDIA Corporation for providing GPUs through their academic program.

References

1. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2D human pose estimation: new benchmark and state of the art analysis. In: CVPR (2014)
2. Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. Pattern Recogn. **13**(2), 111–122 (1981)
3. Carreira, J., Agrawal, P., Fragkiadaki, K., Malik, J.: Human pose estimation with iterative error feedback (2015). arXiv preprint: arXiv:1507.06550
4. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets, fully connected CRFs (2014). arXiv preprint: arXiv:1412.7062
5. Chen, X., Yuille, A.L.: Articulated pose estimation by a graphical model with image dependent pairwise relations. In: NIPS (2014)
6. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
7. Fan, X., Zheng, K., Lin, Y., Wang, S.: Combining local appearance and holistic view: dual-source deep neural networks for human pose estimation. In: CVPR (2015)
8. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. Int. J. Comput. Vis. **61**, 55–79 (2005)
9. Fischler, M.A., Elschlager, R.A.: The representation and matching of pictorial structures. IEEE Trans. Comput. **22**, 67–92 (1973)
10. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: Criminisi, A., Shotton, J. (eds.) Decision Forests for Computer Vision and Medical Image Analysis, pp. 143–157. Springer, London (2013)
11. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding (2014). arXiv preprint: arXiv:1408.5093
12. Johnson, S., Everingham, M.: Clustered pose and nonlinear appearance models for human pose estimation. In: BMVC (2010)
13. Karlinsky, L., Dinerstein, M., Harari, D., Ullman, S.: The chains model for detecting parts by their context. In: CVPR, pp. 25–32 (2010)

14. Karlinsky, L., Ullman, S.: Using linking features in learning non-parametric part models. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 326–339. Springer, Heidelberg (2012)
15. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. IEEE Trans. Pattern Anal. Mach. Intell. **28**, 1568–1583 (2006)
16. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: Workshop on Statistical Learning in Computer Vision, ECCV, vol. 2, p. 7 (2004)
17. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
18. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 1038–1045. IEEE (2009)
19. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. Arxiv preprint (2016)
20. Okoda, R.: Discriminative generalized hough transform for object detection. In: ICCV, pp. 2000–2005 (2009)
21. Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Poselet conditioned pictorial structures. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 588–595 (2013)
22. Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P., Schiele, B.: DeepCut: joint subset partition and labeling for multi person pose estimation (2015). arXiv preprint: arXiv:1511.06645
23. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Li, F.-F.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**, 211–252 (2015)
24. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
25. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional networks. In: CVPR (2015)
26. Tompson, J., Jain, A., LeCun, Y., Bregler, C.: Join training of a convolutional network and a graphical model for human pose estimation. In: NIPS (2014)
27. Wei, S.-E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: CVPR (2016)
28. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: CVPR, pp. 1385–1392 (2011)