# Recurrent Human Pose Estimation

Vasileios Belagiannis and Andrew Zisserman
Visual Geometry Group
Department of Engineering Science
University of Oxford, UK
*vb,az@robots.ox.ac.uk*

***Abstract*— We propose a ConvNet model for predicting 2D human body poses in an image. The model regresses a heatmap representation for each body keypoint, and is able to learn and represent both the part appearances and the context of the part configuration.**

**We make the following three contributions: (i) an architecture combining a feed forward module with a recurrent module, where the recurrent module can be run iteratively to improve the performance; (ii) the model can be trained end-to-end and from scratch, with auxiliary losses incorporated to improve performance; (iii) we investigate whether keypoint visibility can also be predicted.**

**The model is evaluated on two benchmark datasets. The result is a simple architecture that achieves performance on par with the state of the art, but without the complexity of a graphical model stage (or layers).**

## I. Introduction

Estimating 2D human poses from images is a challenging task with many applications in computer vision, such as motion capture, sign language and activity recognition. For many years approaches have used variations on the pictorial structure model [14], [16] of a combination of part detectors and configuration constraints [2], [15], [30], [34], [44]. However, the advent of Convolutional Neural Networks (ConvNets), together with large scale training sets, has led to models that perform well in demanding scenarios with unconstrained body postures and large appearance variations [10], [18], [37], [39]. As the individual part detectors, e.g. the hand and limb detectors, and the pairwise part detectors have become stronger, so the importance of the configuration constraints has begun to wane, with quite recent methods not even including an explicit graphical model [4], [9], [27].

In this paper, we describe a new ConvNet model and training scheme for human pose estimation that makes the following contributions: (i) a model combining a feed-forward module with a recurrent module, where the recurrent module can be run iteratively to increase the effective receptive field of the network and thus improve the performance (see Fig. 2 and 5); (ii) the model can be trained end-to-end, and auxiliary losses can be incorporated to improve performance; and (iii) a preliminary investigation into improving occlusion prediction in human pose estimation.

Our model is mainly inspired by two recent papers: Pfister *et al.* [27] and Carreira *et al.* [9]. The first introduced the idea of 'fusion layers', convolutional layers that *implicitly* encode a configuration model and capture context. The second introduced an iterative update module which progressively makes incremental improvement to the pose estimate. We borrow the fusion layers idea from [27], but apply it multiple times as a recurrent network in the manner of [9]. However, unlike [9] our model is trained end-to-end and does not require a rendering function for combining the output with the input.

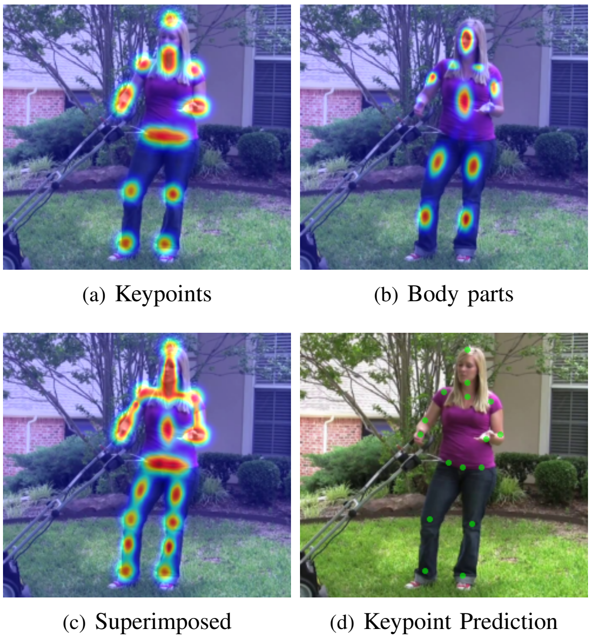

(a) Keypoints
(b) Body parts
(c) Superimposed
(d) Keypoint Prediction

Fig. 1: **Regressed Heatmaps**: The regressed keypoint (a) and body part (b) heatmaps are presented for a validation sample. In (c), both heatmaps are superimposed, resulting in a human skeletal shape. The final outcome is the keypoint prediction (d), while the body part heatmaps act as an auxiliary task.

In addition, our model shares with Convolutional Pose Machines [43] and the Hourglass model [25] the motivation of using large convolution kernels to capture more context (originally proposed by Pfister *et al.* [27]). Unlike these approaches, we use a recurrent convolutional neural network to increase the receptive fields which results in orders of magnitude less parameters in training. Including the recurrent module multiple times is similar to the stacking of more hourglass modules in [25].

The outcome of our approach is a simple recurrent model

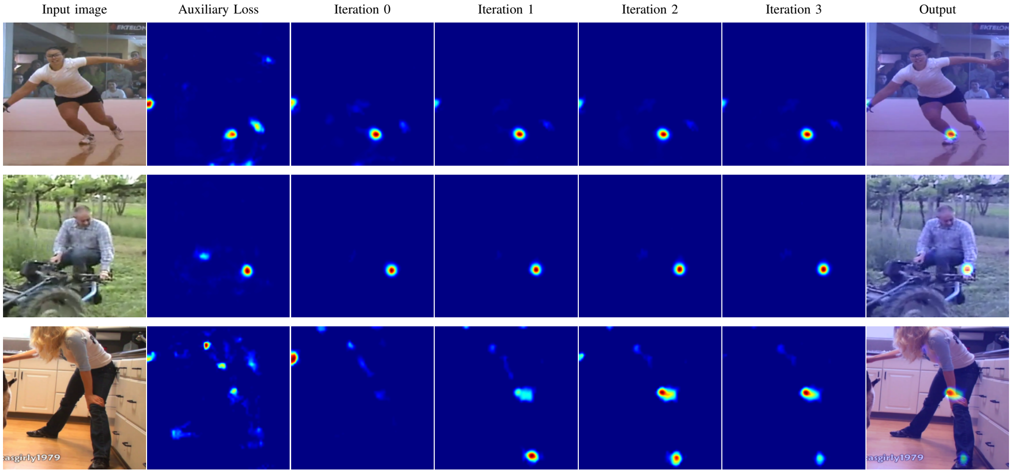


Fig. 2: **Results of the Recurrent Human Model**: The predicted heatmaps (MPII Human Pose dataset [1]) are visualized after every iteration of the recurrent module for the right ankle (*first row*), left wrist (*second row*) and left wrist (*third row*). The model progressively suppresses false positive detections that occur at the first iterations.

that reaches state-of-the-art performance on different standard benchmark datasets, but does not employ an explicit configuration model [10] nor a complicated network architecture [37].

### A. Related Work

For many years the 'workhorse' in human pose estimation has been a tree structured graphical model, often based on the efficient pictorial structure methods of Felzenszwalb and Huttenlocher [14]. This supported a host of methods, including [7], [12], [34], [44]. An alternative approach, which also included configuration constraints, was based around the poselet idea [6], [17].

Early methods using ConvNets predicted pose coordinates of human keypoints directly (as 2D coordinates) [39]. An alternative, which it turns out might be better suited to ConvNets, is an indirect prediction by first regressing a heatmap over the image for each keypoint, and then obtaining the keypoint position as a mode in this heatmap [20], [27], [38], [37]. The advantage of the heatmap over direct prediction is threefold: it mostly avoids problems with ConvNets predicting real values; it can handle multiple instances in the image (e.g. if there are several hands present and consequently several corresponding hand keypoints); and it can represent uncertainty by multiple modes.

Furthermore, combining heatmaps with large convolutional kernels and deeper models [8], [23], [25], [27], [43] improves performance – since the effective receptive fields, and consequently the context captured, is increased. For example, Pfister *et al.* [27] added several large convolutions (e.g. $13 \times 13$ kernels). However, a disadvantage is that this increases the number of parameters and makes the optimization more difficult. In our model, we employ a recurrent module that essentially increases the effective receptive fields without introducing additional parameters.

The method of Carreira *et al.* [9] is an interesting hybrid that switches between regressing direct pose coordinates (as the output of the iterated module) and using a heatmap as the input (to the iterated module). In this respect it is similar to the architecture of [26] which also switches between direct pose coordinates and an image representation in an iterated module. In other related work, the iterated implicit configuration module of our model bears similarities to auto-context [40] and the message-passing inference machines of [32].

## II. Recurrent Human Pose Model Architecture

Our aim is to predict 2D human body pose from a single image, represented as a set of keypoints. In this section, we describe our ConvNet model that takes the image as input, and learns to regress a heatmap for each keypoint, where the location of the keypoint is obtained as a mode of the heatmap.

The architecture of the ConvNet is overviewed in Fig. 3. It consists of two modules: a *Feed-Forward module* that is run once, and a *Recurrent module* that can be run multiple times. Both modules output heatmaps, and can be trained with auxiliary losses. However, the key design idea of the architecture is how context is apportioned in training and inference. The Feed-Forward module mainly acts as an independent 'part' detector, regressing the keypoint heatmaps, but largely unaware of context from the configuration of other parts, due to the smaller effective receptive fields. In contrast, the recurrent module progressively brings in more context each time it is run, in part because the effective receptive field is increased with each iteration (Fig. 2).

In the following we describe the architecture of the two modules and the loss function used for training. The entire network can be trained end-to-end, but we also describe the use of auxiliary losses that can be employed to speed up the training and improve performance. We also investigate two other aspects: the benefit of including additional supervision in the form of hallucinated annotation; and the benefit of training that is occlusion-aware.

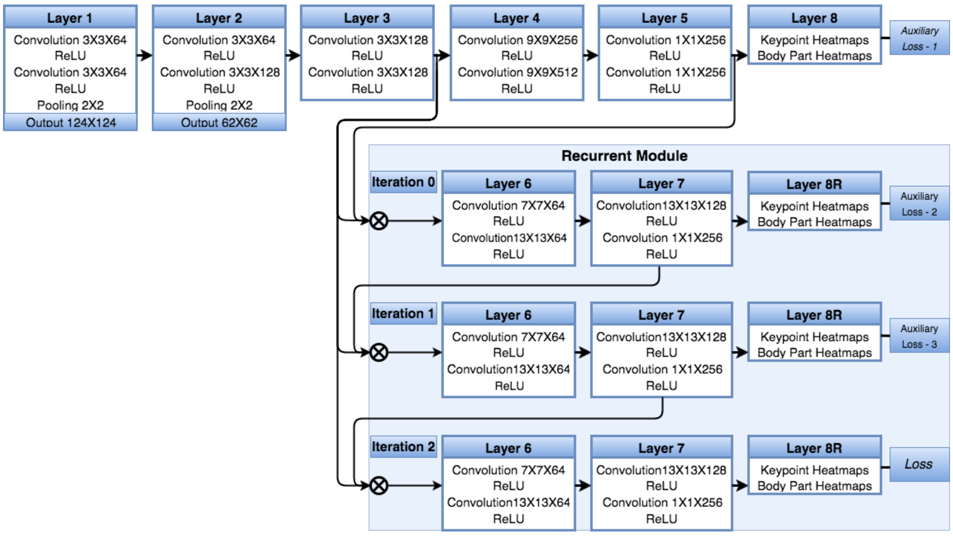


Fig. 3: **ConvNet with Recurrent Module**: Our network is composed of 7 layers. The recurrent model is introduced for Layer 6 and 7. In the current example, a network with 2 iterations is visualized. Note that all loss functions are auxiliary for facilitating the optimization and the final outcome comes from Layer 8D. Moreover, the body part heatmaps is an auxiliary task for additional data augmentation. In our graph, the symbol $\otimes$ corresponds to the concatenation operation.

### A. *Feed-forward Module*

The module is based on the heatmap regression architecture of [27] with modifications. We use smaller filters (i.e. $3 \times 3$) for the initial convolutional layers, combined with non-linear activations (Layer $1-3$ in Fig. 3). This idea from [35] allows more non-linearities to be included in the architecture, and leads to better performance. Pooling is applied only twice in order to retain the output heatmap resolution sufficient large. The activation function is ReLU after every convolution and the prediction layers (Layer 8) are also convolutions, followed by ReLU, that output the predicted heatmaps.

From Layer 4 to 6, larger convolutions filters are employed to learn more of the body structure, followed by convolutions with $1 \times 1$ filters (Layer 5 and 7). The skip layer concatenates the output from Layer 3 and 5, which composes the input for the fusion layer [27] (Layer 6 and 7).

### B. *Recurrent Module*

Our objective is to combine intermediate feature representations for learning context information and improving the final heatmap predictions. To that end, we introduce the recurrent module for the Layer 6 and 7 of our network. The input to the recurrent module is the concatenated output of Layer 3 and Layer 7. At every iteration, the input from Layer 3 is fixed, while Layer 7 is updated (see Fig. 3). Note that by using intermediate network layers for the recurrent module, we do not blend the predictions with the input, as in [9]. Finally, our network can be trained in an end-to-end fashion.

### C. *Body Part Heatmaps as Supplementary Supervision*

Inspired by the idea of part-based models [2], [3], we additionally propose body part heatmaps which are constructed by pairs of keypoints. In practice, we define the body part heatmap by taking the midpoint between the two keypoint as the center of the Gaussian distribution and define the variance based on the Euclidean distance between the two keypoints. Eventually, we model heatmaps for the body limbs, as it is depicted by Fig. 1. The keypoint heatmaps mostly represent body joints and the body part heatmap mainly capture limbs. Although, our main objective is to predict keypoints, modelling pairs of keypoints helps to capture additional body constraints and mainly acts as data augmentation, in terms of labels.

### D. *Target Heatmaps and Loss Function*

At training time, the ground-truth labels are heatmaps synthesised for each keypoint separately by placing a Gaussian with fixed variance at the ground truth keypoint position. We then use the mean squared error, which penalises the squared pixel-wise differences between the predicted heatmap and the synthesised ground-truth heatmap.

The same loss is also used for the feed-forward part and the recurent module of the network. At every loss layer, we equally weights the keypoint and body part heatmaps. Finally, the training of the ConvNet is accomplished using backpropagation [33] and stochastic gradient descent [5].

During training, Layer 8A is used as an auxiliary loss to comfort the optimization [36]. In addition, we propose to use an auxiliary loss function at the end of every iteration of the recurrent module, other than the last iteration, to boost the gradients' magnitude during backpropagation. As a result, Layer 8B and 8C are auxiliary tasks and the actual prediction is the outcome of Layer 8D, given a network of 2 iterations as in Fig. 3. Finally, the cost function of our model for a set of $S$ training samples is defined as:

$$E = \sum_{s=1}^{S} \|\mathbf{h}^s - f(\mathbf{x}, t; \boldsymbol{\theta})^s\|^2, \tag{1}$$

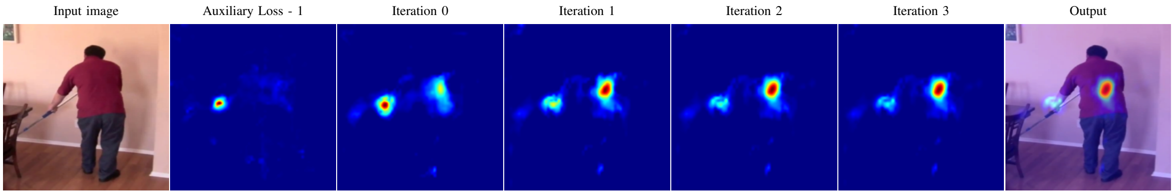


Fig. 4: **Prediction of Occluded Keypoints**: The right wrist is erroneously predicted even though it is not visible because the model learns to capture context and it accordingly predicts. Note that occluded keypoints are not provided during training as ground-truth labels, instead they are ignored or penalized based on the occlusion scenario.

where $\mathbf{h}^s$ is the synthesised ground-truth heatmap and $f(.)$ represents the ConvNet with learned parameters $\boldsymbol{\theta}$ and the recurrent module at the $t$ iteration.

### *E. Occluded Keypoints*

One of the most challenging aspects of predicting human body parts in images is dealing with the problem of occlusion – both self-occlusion and occlusion by other entities.

With the context carried by the recurrent module, we have a new way of approaching this problem. A body keypoint, such as a wrist, generates a strong response in a heatmap for two reasons: because the keypoint is visible and because it can be inferred from the configuration of the other keypoints of the body. The latter can potentially be a problem if configuration dominates over visibility, and a keypoint is predicted even though it is occluded (Fig. 4).

To this end, we investigate three different training scenarios for the network: one *ignoring* occluded keypoints and body parts in the loss function (since they are not visible), two *including* occluded keypoints and body parts in the loss function to increase the amount of training data, and three *excluding* occluded keypoints and body parts by considering them as background and consequently penalizing a corresponding heatmap response. In the first scenario, heatmaps are synthesized, but the gradient values of occluded keypoints or body parts are ignored at the respective heatmap regions during backpropagation (i.e. zero gradient). Thus, the network is trained without including occluded keypoints and at the same time without penalizing them. In the second scenario, the occluded keypoints are included and we learn to infer them. In the third scenario, the gradient values of occluded keypoints or body parts encourage the heatmap areas of the occluded parts to converge to zero. This penalizes heatmaps that erroneously infer occluded parts at points predicted by context. Fortunately, the MPII Human Pose dataset [1] and LSP [21] datasets, which are used for training, provide occluded keypoint annotation within the context of predicted positions that can be used for this purpose.

## III. Implementation Details

The network takes as input an RGB image with resolution $248 \times 248$ and outputs heatmaps with resolution a quarter of the input that is $62 \times 62$. The input image is normalized by mean subtraction at each channel. Furthermore, data augmentation is performed by rotating, scaling, flipping and cropping the input image. Regarding the network parameters, the learning rate is set to $10^{-5}$ and gradually decreased to $10^{-6}$, while we found that no more than $40$ training epochs are required for obtaining a stable solution. Note that we train the model from scratch and the training time is less than 2 days. The momentum is set to $0.95$ and the batch size to 20 samples. Also, batch normalization [19] is used for every convolutional layer other than the layers with $1 \times 1$ filters and the output layers.

The generated target heatmaps have $\sigma$ variance set to $1.3$ for the keypoint Gaussian distributions, while the body part heatmaps have different variance for the $x$ and $y$ direction based on the Euclidean distance between the two keypoints that form a part. In particular, we set $\sigma_x$ and $\sigma_y$ equal to $0.15$ and $0.1$ of the Euclidean distance. Moreover, we found it crucial to weight the gradients of the heatmaps, since the heatmap data is unbalanced. A heatmap has most of its area equal to zero (background) and only a small portion of it corresponds to the Gaussian distribution (foreground). For that reason, it is important to weight the gradient responses so that there is an equal contribution to the parameter update between the foreground and background heatmap pixels. Otherwise, there is a prior towards the background that forces the network to converge to zero. In addition, we magnify the Gaussian distributions so that their mode is around to 12 and consequently enlarge the difference between foreground and background pixels.

Furthermore a heatmap can include multiple individuals (e.g. MPII Human Pose dataset [1]). For our experiments, it is assumed that one is the active individual and the predictions of the rest are ignored during backpropagation. As a result the network learns to predict a single body configuration.

The implementation of our model is in MatConvNet [41] and our code is publicly available[1]. In the next section, we evaluate the components of the recurrent human model, examine how well the regressed heatmaps address the problem of occlusion detection and compare our results with related approaches.

## IV. Experiments

We evaluate the components of our model and compare with related methods for the task of 2D human pose estimation from a single image. The evaluation is based on the MPII Human Pose [1] and LSP [21] datasets. On the MPII Human

[1] `http://www.robots.ox.ac.uk/~vgg/software/keypoint_detection/`

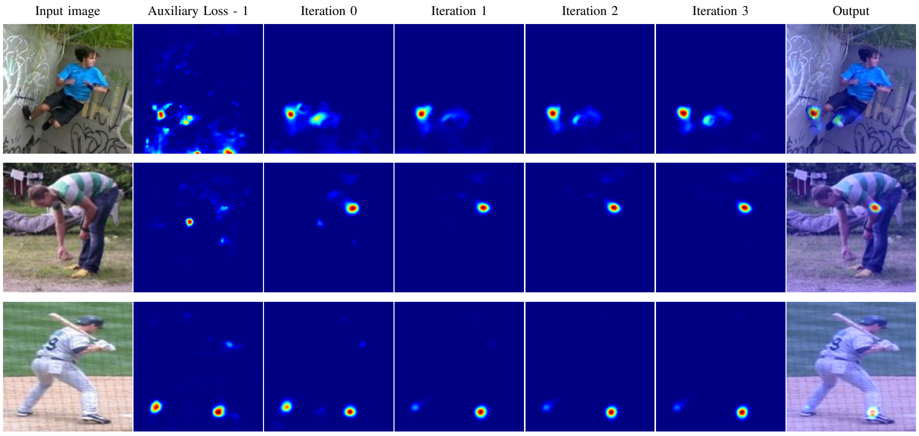


Fig. 5: **More Results from the LSP dataset**: We visualize the predicted heatmaps after every iteration of the recurrent module for the right ankle (*first row*), left elbow (*second row*) and right ankle (*third row*).

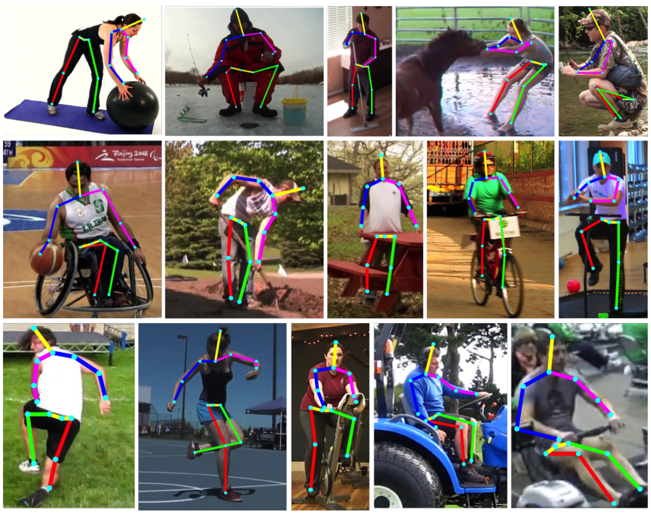


Fig. 6: **Pose Results on MPII Human Pose**: the predictions are from two iterations of the recurrent module.

Pose [1] dataset, we evaluate for single human pose estimation, while the LSP [21] dataset includes labels only for single human evaluation. Keypoint annotation is provided for both datasets, 16 keypoints in MPII Human Pose (Fig. 1) and 14 in LSP, which we use for generating the target ground-truth keypoint and body part heatmaps for training. The parameters of the model, such as the learning rate and number of training epochs, are defined based on the validation dataset of MPII Human Pose, as proposed by [37], and remain the same for all evaluations. Moreover, the validation dataset of [37] is used for all the baseline evaluations. Our network architecture is significantly different from the common recognition models [22], [35] and thus we choose to train from scratch instead of fine-tuning a pre-learnt model.

The evaluation of the recurrent human model is divided into three parts: the component evaluation, occlusion evaluation and comparison with related methods. The different parts of our model are examined in the model components evaluation. In the occlusion part, we evaluate the potential of our model to predict whether a keypoint is visible. Finally, we compare our model with related methods, mainly deep learning approaches. The main performance metric for the evaluations is the PCKh measure [1]. Based on the PCKh definition, a keypoint is correctly localized if the distance between the predicted and ground-truth keypoint is smaller than $50\%$ of the head length.

### A. Component Evaluation

The proposed model is composed of different objectives and the recurrent module, where the recurrent module can include several iterations. In this evaluation, we investigate the contribution of each component to the final performance. For this purpose, we rely on the MPII Human Pose [1] dataset with the validation dataset from [37]. The results of the component evaluation are summarized in Table I.

TABLE I: **Model Components**: We evaluate the components of the model for the body keypoints on MPII Human Pose dataset using the PCKh metric and the validation dataset from [37]. First, we evaluate our model by training under different occlusion settings (including, ignoring or excluding occluded keypoints). Next, different number of recurrent iterations is examined and finally the body part heatmaps are added to the model. We also report results by training an equivalent model (2 iter. and body parts) in MSCOCO dataset [24] and then fine-tuning on the MPII dataset. At the end, we investigate the model's performance by adding scale augmentation during testing.

| Heatmaps | Head | Shoulder | Elbow | Wirst | Hip | Knee | Ankle | PCKh |
|---|---|---|---|---|---|---|---|---|
| Keypoint (exclude occl.) | 95.4 | 89.6 | 79.1 | 74.3 | 78.9 | 73.0 | 66.7 | 80.3 |
| Keypoint (ignore occl.) | 95.3 | 91.4 | 81.9 | 75.2 | 80.5 | 73.1 | 67.4 | 81.4 |
| Keypoint (include occl.) | 95.2 | 92.2 | 82.9 | 77.0 | 82.5 | 75.7 | 69.6 | 82.8 |
| + 1 Iteration | 96.1 | 93.0 | 84.0 | 77.5 | 83.5 | 76.2 | 69.7 | 83.6 |
| + 2 Iterations | 96.1 | 93.0 | 84.1 | 77.4 | 83.4 | 76.1 | 70.0 | 83.6 |
| + Body Part | 96.1 | 93.1 | 84.9 | 78.3 | 84.6 | 78.5 | 72.6 | 84.6 |
| + Fine-tune | 96.3 | 94.0 | 85.9 | 80.2 | 86.0 | 80.0 | 75.7 | 86.0 |
| + Scale Aug. | 96.4 | 94.0 | 86.3 | 80.2 | 86.4 | 80.5 | 75.9 | 86.3 |

At first, we evaluate the objective of the keypoint heatmaps, w/o occluded keypoints (first three rows, Table I). Including occluded points during training (i.e. constructing heatmaps using them) gives the best performance because of the larger amount of training data. This evaluation composes the baseline of the proposed model. Next, recurrent iterations are added to the keypoint model. Table I shows that one iteration is sufficient and adding more does not improve the final result. To gain more from the recurrent model, we add the objective of the body parts (sixth row, Table I). We do not aim to predict body parts, but observe that this additional objective is helpful for capturing additional body constraints, propagate back more gradients and thus it brings a boost to the model's performance. We notice that after two recurrent iterations, there is not significant improvement of the final result. In general, the parts that are already well predicted using the feed-forward model benefit less from the recurrent module (e.g. head). Note that the model with one iteration that also includes body parts has the same performance as the model with two iterations and only keypoints objective. To examine how additional amount of training data affects model's behaviour, we included the MSCOCO dataset [24] during training, but there was not improvement. On the hand, training first our full model (i.e. 2 iterations and body parts) on MSCOCO dataset and then fine-tuning it on MPII dataset improved our final performance (seventh row, Table I). Finally, scale augmentation is added at test time to gain another small boost in our performance.

### B. Occlusion Prediction

In this experiment, we analyse the potential of the heatmaps to predict the visibility of a keypoint. Empirically, the heatmaps of occluded keypoints tend to have low responses (in terms of magnitude); and, as a result, the visibility of a keypoint can be inferred from the heatmap responses. We evaluate to what extent the response magnitude can be used to predict keypoint visibility on the validation set of the MPII Human Pose dataset, since this dataset provides occlusion labels. However, the distribution of visible and occluded keypoints is unbalanced (only $\sim 23\%$ of annotated keypoints are tagged as occluded), so there is bias towards the visible keypoints.

For the evaluation we make the assumption that there should only be a single response for each heat map for a visible point (and no or a low response if the point is occluded) since the data set has only one of each keypoint (e.g. left elbow) for each test sample. We then pick the maximum response in each heatmap, and order all these maximum responses (over all images and all heatmaps) by their strength. A Precision-Recall curve is then computed where the positives are visible points (and negatives are the occluded points). Note, since the evaluation examines only the heatmap responses, and not their positions, we are not determining whether the predicted visible point is at the correct position or not.

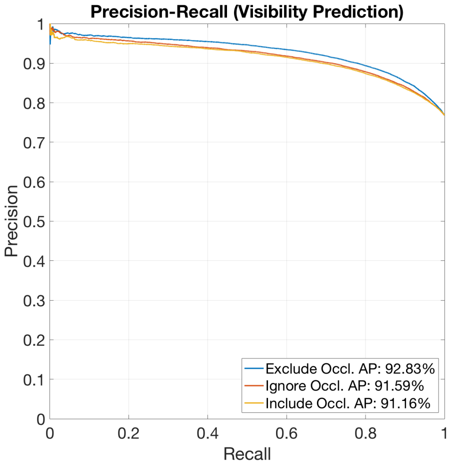


Fig. 7: **Visibility Prediction Precision-Recall**: The evaluation is performed using three different models: the first model is trained by ignoring occluded keypoints and body parts (*Ignore Occl.*) during training, the second by including them (*Include Occl.*) to increase the amount positive training data and the last by excluding (*Exclude Occl.*) them and thus treating them as background. The average precision (AP) is reported for all training scenarios.

This experiment is performed using the three training scenarios that were defined in Sec. II-E using the model with 1 iteration, including body parts. In the first case, the occluded keypoints and body parts are ignored from the ground-truth labels. In the second case, the occluded keypoints and body parts are included to the training, while in the third case, the heatmap regions of the occluded keypoints are penalized (i.e. considered as background). Our results are summarized in Fig. 7. One can see that the model with penalized occluded keypoints (in the training process) performs better than the models that ignore or include occluded keypoints and body parts. In practice, we observe that the network learns to treat areas of occluded keypoints as background (see Fig. 8). Nevertheless, we find that the overall performance of the network that penalizes occluded keypoints during training is around $3\%$ worse than the network that includes the occluded keypoints (Table I) – this is a consequence of the fact that the MPII evaluation ignores occluded keypoints, so there is no disadvantage in predicting them, and they clearly provide some context in training. Our average precision (AP) is more than $90\%$ for the case of the visibility prediction. We do not compare with another approach since we are not aware of any related method that performs occlusion detection on this dataset. Our results are not directly comparable to the $35\%$ of average detection accuracy of occluded joint from [31] or the $85\%$ of accuracy of occlusion prediction from [11]; but these evaluations are indicative that our performance is good for this problem.

### C. Comparison with other Methods

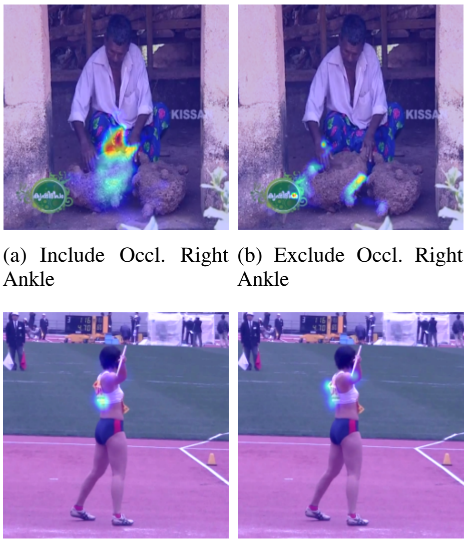
In our last experiment, we compare our results with related methods on the MPI Human Pose [1] and LSP [21] datasets. In both evaluations, our model is executed for two iterations. We do not use any ground-truth information for the localization of the individuals in the LSP dataset, while rough localization is provided for the MPI Human Pose dataset.

TABLE II: **MPII Human Pose Evaluation**. The PCKh measure is used for the evaluation. The scores are reported for each keypoint separately and for the whole body. The area under the curve (AUC) is also reported. In addition, we include the results by training first our model on MSCOCO and then fine-tuning on the MPII dataset.

| | Head | Shoulder | Elbow | Wrist | Hip | Knee | Ankle | Total | AUC |
|---|---|---|---|---|---|---|---|---|---|
| Ours | 97.5 | 94.3 | 86.9 | 80.8 | 86.7 | 80.7 | 76.0 | 86.7 | 56.8 |
| Ours (fine-tuned) | 97.7 | 95.0 | 88.2 | 83.0 | 87.9 | 82.6 | 78.4 | 88.1 | 58.8 |
| Tompson et al. [38] | 95.8 | 90.3 | 80.5 | 74.3 | 77.6 | 69.7 | 62.8 | 79.6 | 51.8 |
| Carreira et al. [9] | 95.7 | 91.7 | 81.7 | 72.4 | 82.8 | 73.2 | 66.4 | 81.3 | 49.1 |
| Tompson et al. [37] | 96.1 | 91.9 | 83.9 | 77.8 | 80.9 | 72.3 | 64.8 | 82.0 | 54.9 |
| Pishchulin et al. [29] | 94.1 | 90.2 | 83.4 | 77.3 | 82.6 | 75.7 | 68.6 | 82.4 | 56.5 |
| Insafutdinov et al. [18] | 96.8 | 95.2 | 89.3 | 84.4 | 88.4 | 83.4 | 78.0 | 88.5 | 60.8 |
| Wei et al. [43] | 97.8 | 95.0 | 88.7 | 84.0 | 88.4 | 82.8 | 79.4 | 88.5 | 61.4 |
| Bulat et al. [8] | 97.9 | 95.1 | 89.9 | 85.3 | 89.4 | 85.7 | 81.7 | 89.7 | 59.6 |
| Newell et al. [25] | 98.2 | 96.3 | 91.2 | 87.1 | 90.1 | 87.4 | 83.6 | 90.9 | 62.9 |

TABLE III: **Model Size**. The number of convolutional parameters for Wei *et al.* [43] and Carreira *et al.* [9] is calculated from the online models; while the authors of [25] have communicated the approximate number of convolutional and deconvolutional parameters in the Hourglass model. All examined model configurations are used for the MPII dataset evaluation.

| | Model Parameters |
|---|---|
| Newell *et al.* [25] | $23.7 \times 10^6$ |
| Wei *et al.* [43] | 342042 |
| Ours | 100670 |
| Carreira *et al.* [9] | 48528 |



(a) Include Occl. Right Ankle (b) Exclude Occl. Right Ankle

(c) Include Occl. Left Elbow (d) Exclude Occl. Left Elbow

Fig. 8: **Visibility Heatmaps**: On 8(a), the predicted *right ankle* heatmap is visualized for the model that includes the occluded keypoints during training, while on 8(b) the same heatmap is presented for the model that excludes (i.e. penalizes) the occluded keypoints (during training). Similarly on 8(c), the self-occluded *left elbow* is recovered due to the context information, while the response at the correct area is low in8(d) for the model that has been trained by excluding during training occluded keypoints.

*a) MPII Human Pose Dataset:* We use the same training and validation protocol as [37]. Our results are summarized in Table II and also samples are visualized in Fig. 6. In all cases, we achieve on par performance with other methods. It is worth noting that our model architecture is significantly simpler than [37] and it does not depend on a graphical model inference as [38]. One should also observe that our model performs better than the iterative method of Carreira *et al.* [9] which relies on a pre-trained model and training in stages. In terms of the number of model parameters, as can be seen from Table III, our model has two orders of magnitudes fewer parameters, and thus smaller capacity, than the Hourglass model [25], and a third of the parameters of the Convolutional Pose Machines [43], though it has comparable performance in the evaluation.

*b) LSP Dataset:* The dataset is composed of 2000 images, where half of the images are used for training (Fig 9). There is also the extension of LSP [21] with 10000 training samples which we use for this experiment. However, we observe that the training data is not sufficient for training our model from scratch, and thus we merge the training data of the extended LSP with the MPII Human Pose dataset. We also report results using a model trained or fined-tuned on the MPII Human Pose dataset. Our results are presented in Table IV. The evaluation is accomplished using the PCK measure (threshold at 0.2) that is similar to PCKh, but it has as reference part the length of the torso instead of the head. It is clear that we achieve promising performance for all keypoints. In particular, our recurrent human model performs better than the iterative method of Carreira *et al.* [9], as well as, the graph-based model with deep body part detectors of Chen & Yuille [10].

## V. Conclusion

We have introduced a recurrent human model for 2D human pose estimation that is able to capture context iteratively, resulting in improved localization performance. We demonstrate performance comparable to the state-of-the-art on two challenging human pose estimation datasets, training the model from scratch. Finally, the regressed heatmaps can be useful for predicting occlusion of keypoints.

Future work will investigate whether the heat map obtained by combining the keypoints and body parts (shown in Fig. 1(c)) can be used to avoid erroneous keypoint predictions (such as left/right hand swopping).

## VI. Acknowledgements

We are grateful for the help provided by Tomas Pfister. Funding for this research was provided by the EPSRC Programme Grant Seebibyte EP/M013774/1.

## References

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proc. CVPR*, pages 3686–3693, 2014.

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proc. CVPR*, 2009.

[3] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic. 3D pictorial structures for multiple human pose estimation. In *Proc. CVPR*, June 2014.

[4] V. Belagiannis, C. Rupprecht, G. Carneiro, and N. Navab. Robust optimization for deep regression. In *Proc. ICCV*, December 2015.

TABLE IV: **LSP Evaluation**. The PCK measure is used for the evaluation. The scores are reported for each keypoint separately and for the whole body. We report results using the trained model from MPII Human Pose, the MPII model fined tuned on the extended LSP training data, and also training a new model by combining the training data of the MPII Human Pose with the extended LSP dataset.

| | Head | Shoulder | Elbow | Wrist | Hip | Knee | Ankle | Total |
|---|---|---|---|---|---|---|---|---|
| Ours (MPII model) | 90.8 | 84.4 | 76.3 | 70.4 | 81.5 | 81.9 | 77.8 | 80.5 |
| Ours (MPII fine-tuned) | 94.3 | 87.1 | 78.6 | 72.0 | 78.1 | 83.2 | 77.1 | 81.5 |
| Ours (MPII & LSP, 1 it.) | 95.6 | 88.8 | 80.7 | 75.5 | 83.0 | 86.2 | 80.6 | 84.3 |
| Ours (MPII & LSP, 2 it.) | 95.2 | 89.0 | 81.5 | 77.0 | 83.7 | 87.0 | 82.8 | 85.2 |
| Pishchulin et al. [28] | 87.2 | 56.7 | 46.7 | 38.0 | 61.0 | 57.5 | 52.7 | 57.1 |
| Wang Li et al. [42] | 84.7 | 57.1 | 43.7 | 36.7 | 56.7 | 52.4 | 50.8 | 54.6 |
| Carreira et al. [9] | 90.5 | 81.8 | 65.8 | 59.8 | 81.6 | 70.6 | 62.0 | 73.1 |
| Chen & Yuille [10] | 91.8 | 78.2 | 71.8 | 65.5 | 73.3 | 70.2 | 63.4 | 73.4 |
| Fan et al. [13] | 92.4 | 75.2 | 65.3 | 64.0 | 75.7 | 68.3 | 70.4 | 73.0 |
| Pishchulin et al. [29] | 97.0 | 91.0 | 83.8 | 78.1 | 91.0 | 86.7 | 82.0 | 87.1 |
| Wei et al. [43] | 97.8 | 92.5 | 87.0 | 83.9 | 91.5 | 90.8 | 89.9 | 90.5 |
| Bulat et al. [8] | 97.2 | 92.1 | 88.1 | 85.2 | 92.2 | 91.4 | 88.7 | 90.7 |

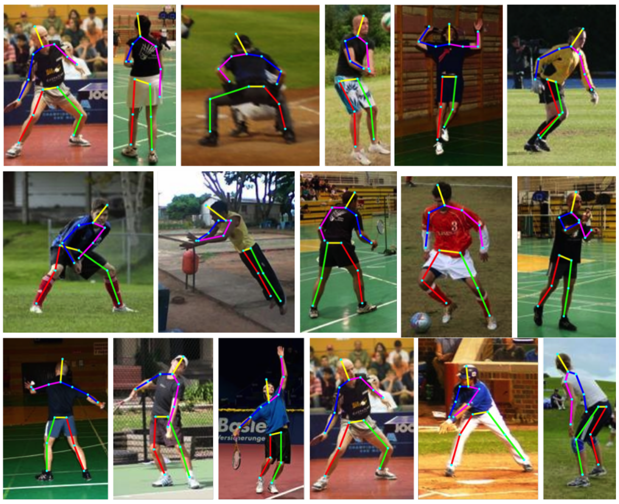


Fig. 9: **Pose Results on LSP Dataset**: the result after two iterations in the recurrent module.

[5] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186, 2010.

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

[7] P. Buehler, M. Everingham, D. P. Huttenlocher, and A. Zisserman. Upper body detection and tracking in extended signing sequences. *IJCV*, 95(2):180–197, 2011.

[8] A. Bulat and G. Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *Proc. ECCV*, pages 717–732, 2016.

[9] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In *Proc. CVPR*, pages 4733–4742, 2016.

[10] X. Chen and A. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*, pages 1736–1744, 2014.

[11] X. Chen and A. Yuille. Parsing occluded people by flexible compositions. In *Proc. CVPR*, pages 3945–3954, 2015.

[12] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari. 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. *IJCV*, 99:190–214, 2012.

[13] 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 *Proc. CVPR*, pages 1347–1355, 2015.

[14] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1), 2005.

[15] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Proc. CVPR*, 2008.

[16] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computer*, c-22(1):67–92, Jan 1973.

[17] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik. Using k-poselets for detecting people and localizing their keypoints. In *Proc. CVPR*, pages 3582–3589, 2014.

[18] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *Proc. ECCV*, pages 34–50, 2016.

[19] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, 2015.

[20] A. Jain, J. Tompson, Y. LeCun, and C. Bregler. Modeep: A deep learning framework using motion features for human pose estimation. In *Proc. ACCV*, pages 302–315, 2014.

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

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012.

[23] I. Lifshitz, E. Fetaya, and S. Ullman. Human pose estimation using deep consensus voting. In *Proc. ECCV*, pages 246–260, 2016.

[24] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proc. ECCV*, pages 740–755, 2014.

[25] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *Proc. ECCV*, pages 483–499, 2016.

[26] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a feedback loop for hand pose estimation. In *Proc. ICCV*, 2015.

[27] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *Proc. ICCV*, 2015.

[28] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Strong appearance and expressive spatial models for human pose estimation. In *Proc. ICCV*, pages 3487–3494, 2013.

[29] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proc. CVPR*, pages 4929–4937, 2016.

[30] L. Pishchulin, A. Jain, M. Andriluka, T. Thormählen, and B. Schiele. Articulated people detection and pose estimation: Reshaping the future. In *Proc. CVPR*, pages 3178–3185, 2012.

[31] U. Rafi, J. Gall, and B. Leibe. A semantic occlusion model for human pose estimation from a single depth image. In *Proc. of the IEEE CVPR Workshops*, pages 67–74, 2015.

[32] S. Ross, D. Munoz, M. Hebert, and J. A. Bagnell. Learning message-passing inference machines for structured prediction. In *Proc. CVPR*, pages 2737–2744, 2011.

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[34] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *Proc. ECCV*, pages 406–420, 2010.

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, 2015.

[37] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *Proc. CVPR*, pages 648–656, 2015.

[38] 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*, pages 1799–1807, 2014.

[39] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proc. CVPR*, pages 1653–1660, 2014.

[40] Z. Tu and X. Bai. Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *IEEE PAMI*, 32(10):1744–1757, 2010.

[41] A. Vedaldi and K. Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proc. ACMM*, 2015.

[42] F. Wang and Y. Li. Beyond physical connections: Tree models in human pose estimation. In *Proc. CVPR*, pages 596–603, 2013.

[43] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proc. CVPR*, pages 4724–4732, 2016.

[44] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proc. CVPR*, 2011.