An Efficient Convolutional Network for Human Pose Estimation

Umer Rafi[1]
rafi@vision.rwth-aachen.de
Ilya Kostrikov[1]
ilya.kostrikov@rwth-aachen.de
Juergen Gall[2]
gall@iai.uni-bonn.de
Bastian Leibe[1]
leibe@vision.rwth-aachen.de

[1] Computer Vision Group, RWTH Aachen University, Germany
[2] Computer Vision Group, University of Bonn, Germany

Abstract

In recent years, human pose estimation has greatly benefited from deep learning and huge gains in performance have been achieved. The trend to maximise the accuracy on benchmarks, however, resulted in computationally expensive deep network architectures that require expensive hardware and pre-training on large datasets. This makes it difficult to compare different methods and to reproduce existing results. In this paper, we therefore propose an efficient deep network architecture that can be efficiently trained on mid-range GPUs without the need of any pre-training. Despite the low computational requirements of our network, it is on par with much more complex models on popular benchmarks for human pose estimation.

1 Introduction

Convolutional networks have raised the bar substantially for many computer vision benchmarks. Human pose estimation is such an example where methods based on convolutional networks dominate the leader boards [3, 13]. Despite the recent success in human pose estimation, a direct comparison between the architectures remains difficult. For architectures that do not provide the source code for training and testing, the reproducibility of the results can be very difficult due to small details that might be essential for the performance, such as the used image resolution or the exact form of data augmentation. Very often, pre-trained models are used that are fine-tuned on the benchmark datasets, making it difficult to compare them with methods that are trained from scratch on benchmarks and therefore on less training data. Another issue is the increasing complexity of the models for human pose estimation. Despite the impressive accuracy they achieve, computationally expensive network architectures with a large memory footprint can be impractical for many applications since they require high-end graphics cards.

In this work, we propose an efficient network architecture for human pose estimation that exploits the best current design choices for network architectures with a low memory

footprint and we train it using best-practice ingredients for efficient learning. An important design choice is to learn features in different layers at multiple scales. This has been recently exploited in the context of classification by Szegedy *et al*. [25] through the introduction of inception layers. Surprisingly, very little attention has been paid on exploiting this concept for human pose estimation. Similarly, learning features at multiple image resolutions has been shown to be very effective for human pose estimation [27], as it helps the network to use a larger context for difficult body joints like wrists and ankles. In our network, we combine both ideas to achieve maximum performance. Another important aspect is the use of features from the middle layers in addition to features from the last layer. While coarse features from the last layer are very good for classification but poor for localisation due to pooling, features from the middle layers are better for localisation. Long *et al*. [17] exploited this for semantic segmentation and [10] used it for object localisation. Similarly, using context around features from the last layer has shown to be very effective in the context of semantic segmentation [16]. Our network also exploits context around features from the last layer along with features from a middle layer. Other recent advances in deep learning, *e.g*., Adam optimiser [15], exponential learning rate decay, batch normalisation [12] and extensive data augmentation, also have shown to provide further benefits for the overall performance. We therefore exploit the above ingredients to add additional gains to the performance.

Based on the design choices, we propose a network architecture for human pose estimation that is efficient to train and has a low memory footprint such that a mid-range GPU is sufficient. Yet, our network architecture achieves state-of-the-art accuracy on the most popular benchmarks for human pose estimation, indicating that very complex architectures might not be needed for the task. For best comparison and reproducibility, we evaluate the network using the protocols of state-of-the-art benchmarks without any pre-training or post-processing. The learned models for all benchmarks and the source code for training and testing are publicly available[1] and serve as an up-to-date baseline for more complex models.

2 Related Work

Human pose estimation has been intensively studied in the last decades. The classical approaches are based on the pictorial structure model [1, 2, 7, 9, 13, 18, 19, 32] that uses a tree-structured graphical model to encode spatial dependencies between neighbouring joints. These approaches have shown to be successful for many applications but they can suffer from double counting, for instance, in case of occlusions. Another line of research is based on hierarchical models [24, 26] that first detect larger body parts in the image and then condition the detection of smaller body parts on the detected larger body parts. Complex non-tree models [23] have also been used that model spatial dependencies between unconnected body parts. Loopy belief propagation is then used for approximate inference to predict the joint positions in the image. Recently, sequential prediction machines [21] have been proposed that combine the benefit of modelling complex spatial dependencies between joints with an efficient inference procedure.

Approaches based on CNNs became popular in the last two years. Toshev *et al*. [29] first used CNNs to directly regress the positions of body joints. Tompson *et al*. [27] have shown that predicting belief maps as opposed to point estimates of joints improves accuracy. In their later work, Tompson *et al*. [28] further improved performance by introducing a cascade

[1] https://web-info8.informatik.rwth-aachen.de/software/pose-cnn

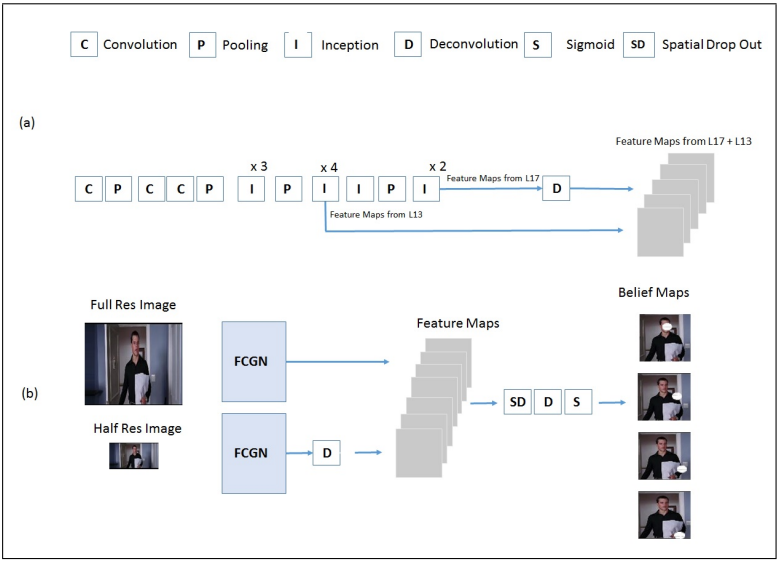

Figure 1: (a) Proposed fully convolutional GoogLeNet (FCGN), which is an adaptation of the batch-normalised inception network [12]. (b) The proposed multi-resolution network combines two FCGNs. One takes the full resolution image as input and one takes a half resolution image as input.

architecture that compensates for the negative effect of pooling. In [4] the joint positions are not directly predicted, but an iterative procedure is used to refine the pose step by step until it converges to a pose configuration. While the majority of approaches estimates the pose of a single person, multi-person pose estimation is addressed in [20]. Very recently, convolutional pose machines have been proposed that stack multiple CNNs where each CNN refines the pose [30]. The method achieves very accurate pose estimates, but it is very expensive to train and requires 6GB of GPU memory. In contrast, our network is fast to train and requires only 3GB which makes our network also suitable for mid-range GPUs like GTX980.

3 Fully Convolutional Deep Network for Human Pose Estimation

For 2D human pose estimation, the positions of all body joints in an image need to be predicted. Recent approaches [27, 29] have shown that regressing point estimates for body joints may be sub-optimal and a better strategy is to use fully convolutional deep architectures to predict dense belief maps for each body joint. If not well designed, fully convolutional networks, however, can be very inefficient in terms of memory usage and training time. We therefore propose an efficient fully convolutional network for predicting belief maps for body joints. To this end, we adapt the batch-normalised inception network [12], which was proposed for image classification and is based on the GoogLeNet architecture.

3.1 Network Architecture

Our adaptation of [12] is illustrated in Figure 1(a), which we refer in this work as Fully Convolutional GoogLeNet (FCGN). We use the first 17 layers of [12] and remove the average pooling, drop-out, linear and soft-max layers from the last stages of the network. We add a skip connection to combine feature maps from layer 13 with feature maps from layer 17. We upsample the feature maps from layer 17 to the resolution of the feature maps from layer 13 by a deconvolution filter with size 2×2 and stride 2. The output of the FCGN consists of feature maps from layer 13 and 17 that have 16 times lower resolution than the input image due to pooling.

The proposed multi-resolution network for pose estimation is illustrated in Figure 1(b). It uses two FCGNs with shared weights, where each FCGN takes the same image at a different resolution and produces feature maps as previously described. The feature maps obtained from the half resolution image are upsampled to the resolution of the feature maps extracted from the full resolution image by a deconvolution filter with size 2×2 and stride 2. The feature maps of the half resolution and full resolution FCGN are then directly upsampled to obtain belief maps for different body joints by using a larger deconvolution filter of size 32×32 and stride 16. Due to the large deconvolution filter, we implicitly exploit the context of neighbouring pixels in the feature maps for predicting belief maps for joints. The belief maps are then normalised by using a sigmoid function. We also use spatial drop out [28] before upsampling to further regularise our network.

3.2 Training

We denote a training example as $(I, \{B_j\})$. While I denotes the image, which is in our experiments of size 256×256, B_j denotes the ground-truth 2D belief map for a joint j. Each belief map has the same size as the image and is created by setting all pixels with distance larger than 8 pixels to the joint j to 0 and all other pixels to 1, as shown in Figure 2(a). Given the training samples $N = \{(I, \{B_j\})\}$, we minimise the binary cross entropy between the ground-truth and predicted belief maps for k joints in each training image I as follows:

$$\arg\min_{w} - \sum_{j=1}^{k} \sum_{x,y} B_j(x,y) \log(\hat{B}_j^w(x,y)) + (1 - B_j(x,y)) \log(1 - \hat{B}_j^w(x,y)), \tag{1}$$

where w and $\hat{B}_j^w$ are the parameters of our network and the predicted belief maps, respectively. The weights of the network are then learned using back propagation and the Adam optimiser [15]. At inference, we take the maximum scoring location in each predicted belief map as the final joint position for each joint.

3.3 Data Augmentation

Data augmentation is an essential ingredient for deep networks and has a significant impact on performance. In contrast to image classification, we do not only transform the image but also the joint annotations. We apply the following transformation to the training images:

$$W = \begin{pmatrix} 1 & 0 & c_x \\ 0 & 1 & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} rscos\theta & -ssin\theta & 0 \\ ssin\theta & scos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -x_p + t_x \\ 0 & 1 & -y_p + t_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{2}$$

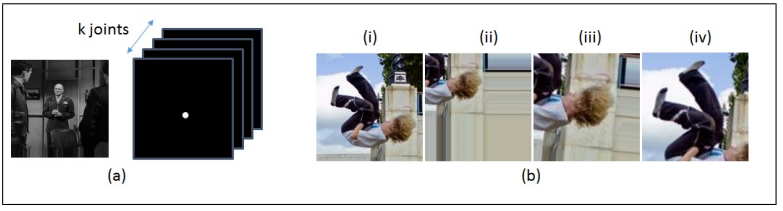

Figure 2: (a) Image with ground truth binary belief maps for all joints. (b) Illustration of the data augmentation procedure: (i) original image (ii) person shifted to the top left (iii) warped image (iv) person shifted back to centre of the cropped image.

The first transformation on the right hand side moves the person centred at (x_p, y_p) to the origin and applies a random translation (t_x, t_y) to it. The second transformation applies random scaling s, rotation θ and reflection r and the last transformation shifts the warped person back to the centre (c_x, c_y) of the cropped image. Figure 2(b) illustrates the procedure. For each training image, we generate and apply random transformations W. If one or more joints of the person are outside the image, we discard the transformation and replace it with another random transformation.

4 Experiments

We evaluate our network on standard benchmarks for human pose estimation, namely the MPII Human Pose (MPII) dataset [3], the Leeds Sports Pose (LSP) dataset [13, 14] and the Frames Labelled In Cinema (FLIC) datasets [22]. Some qualitative results are shown in Figure 3.

Our experimental settings are as follows: We crop the images in all datasets to a resolution of 256×256 pixels. For the training images, we crop around the person's centre, computed as the midpoint between maximum and minimum ground-truth joint positions in x, y directions. For the test images, we crop around the provided rough location when available and around the centre of the image otherwise. We train the network from scratch without any pre-training with a learning rate of 0.00092 using a stair case decay of 0.95 applied after 73 epochs. We use a batch size of 8. For Adam, we use $\beta_1 = 0.9$ and $\varepsilon = 0.1$. The ground-truth belief maps are created by setting all pixels within an 8 pixel distance to the annotated joint to 1. We train the network for 120 epochs for each dataset. For data augmentation, we transform each training image 120 times. The scaling parameter $s \in [0.5, 1.5]$, the translation parameters $t_{x,y} \in [-20, 20]$ and the rotation parameter $\theta \in [-20°, 20°]$ are randomly selected with uniform probability. Horizontal flipping r is applied with probability 0.5.

We use the Torch 7 [6] framework for our experiments. Unless otherwise stated, we report results for the above-mentioned settings. For evaluation, we use the PCK measure [22] for the LSP and FLIC datasets and PCKh [3] for the MPII dataset.

4.1 FLIC dataset

The FLIC [22] dataset consists of 3,987 training images and 1,016 test images. Our model takes 10 hours to train on the FLIC dataset using a GTX 980 GPU. Following the standard practice, we report PCK @ 0.2 only for wrists and elbows. Our quantitative results are given

Method	Elbows	Wrists
Tompson *et al.* [28]	93.1	89.0
Toshev *et al.* [29]	92.3	82.0
Chen *et al.* [5]	95.3	92.4
Wei *et al.* (with scale normalisation) [30]	97.6	95.0
Ours	96.1	89.7
Ours with scale normalisation	98.5	96.5
Ours with reduced augmentation	92.4	81.9
Ours without learning rate decay	94.5	86.5

Table 1: Comparison with the state-of-the-art on the FLIC datasets using PCK @ 0.2.

in Table 1. We outperform the other methods for elbows and wrists except for [5] when we do not exploit scale information. However, our model does not use any image dependent explicit prior model as compared to [5]. We also study the impact of scale normalisation, data augmentation and learning rate decay on this dataset.

Impact of scale normalisation. For the FLIC dataset, rough torso detections are available for the training and testing images. We thus normalise all training and test images to the same scale by re-normalising the height of the detected torso in each image to 200 pixels. The results reported in Table 1 show that using scale information, when available, can provide significant gains in accuracy, especially for wrists from 89.7 to 96.5. Our network outperforms the convolutional pose machines [30], which also use scale information. This is remarkable, since our model has a low memory footprint of 3 GB and runs on a mid-range GPU, while convolutional pose machines require high-end GPUs with more than 6 GB memory.

Impact of data augmentation. We evaluate the impact of data augmentation by reducing the ranges for scaling $s \in [0.7, 1.3]$, translation $t_{x,y} \in \{-5, 5\}$ and rotation $\theta \in \{-5°, 5°\}$ on the FLIC dataset. The results reported in Table 1 show that the accuracy drops from 96.1 to 92.4 for elbows and from 89.7 to 81.9 for wrists. This shows that extensive data augmentation is indeed important for achieving high accuracy and that the exact details of data augmentation are important for reproducibility.

Effect of exponential learning rate decay. We also compare an exponential decay of the learning rate with a constant learning rate using the same number of epochs. Without the learning rate decay, accuracy drops from 96.1 to 94.5 for elbows and from 89.7 to 86.5 for wrists as reported in Table 1.

4.2 Leeds Sports Pose (LSP) dataset

The LSP dataset [13] consists of 1,000 training and 1,000 test images. The extended LSP dataset [14] consists of an additional 10,000 training images. For our experiments, we use the 11,000 training images from LSP and extended LSP together. Our model takes 2 days for training using a GTX 980 GPU. Quantitative results for PCK @ 0.2 are shown in Table 2. Our network achieves a higher accuracy than most of the other methods and gets very

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	PCK
Tompson *et al.* [27]	90.6	79.2	67.9	63.4	69.5	71.0	64.2	72.0
Fan *et al.* [8]	92.4	75.2	65.3	64.0	75.7	68.3	70.4	73.0
Carreira *et al.* [3]	90.5	81.8	65.8	59.8	81.6	70.6	62.0	73.1
Chen *et al.* [5]	91.8	78.2	71.8	65.5	73.3	70.2	63.4	73.4
Yang *et al.* [31]	90.6	78.1	73.8	68.8	74.8	69.9	58.9	73.6
Wei *et al.* [30]	.	.	.	.	.	.	.	84.3
Pishchulin *et al.* [20] + MPII	97.0	91.0	83.8	78.1	91.0	86.7	82.0	87.1
Wei *et al.* [30] + MPII	97.8	92.5	87.0	83.9	91.5	90.8	89.9	90.5
Ours	95.8	86.2	79.3	75.0	86.6	83.8	79.8	83.8
Ours with single resolution only	96.0	85.8	79.6	73.7	86.3	80.8	77.7	82.8
Ours with feature maps from last layer only	95.4	83.7	74.8	70.4	84.4	78.2	74.2	80.2

Table 2: Comparison with the state-of-the-art on the LSP datasets using PCK @ 0.2.

close to [30] when the same training data is used, which is consistent with the results on the FLIC dataset. We also study the impact of two design choices of our network illustrated in Figure 1, namely using the features from layer 13 and a multi-resolution architecture.

Impact of middle layer features. To evaluate the impact of using feature maps from the middle layer on the accuracy, we remove the skip layer connection from layer 13. The accuracy drops for all joints and in average from 83.8 to 80.2 as shown in Table 2.

Impact of multi-resolution architecture. We compare our model that combines two FCGNs, one applied to the full resolution image and the second one to the half resolution image. If we use only one FCGN with the full resolution image for training and testing, the average accuracy decreases slightly from 83.8 to 82.8 as shown in Table 2. The slight decrease can be explained by the fact that the half resolution FCGN introduces more context for the prediction. As a result, the additional context only improves the prediction of the joints that are far away from the head, namely wrist, knee and ankle.

4.3 MPII Human Pose dataset

The MPII Human Pose [3] dataset is a challenging dataset and consists of around 40,000 images of people. We use 25,925 images for training and use 2,958 images for validation according to the train/validation split from [28]. We evaluate on the 7,247 single person test images with withheld annotations. The dataset provides a rough scale and person location for both training and test images. We crop test images around the given rough person location. We normalise both training and test images to the same scale by using the provided rough scale information. Our model takes 3 days to train on the MPII dataset using a GTX 980 GPU.

Our quantitative results are shown in Table 3. Our network outperforms most of the recent state-of-the-art methods and is competitive with [30]. [30], however, used additional training data from the LSP dataset to boost the accuracy. The fact that complex models like [30] do not significantly outperform the proposed network shows that increasing the parameters and complexity of the models and thus the memory consumption and training time might not be the best way to increase the accuracy for human pose estimation.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	PCKh
Hu *et al.* [11]	95.0	91.6	83	76.6	81.9	74.5	69.5	82.4
Carreira *et al.* [4]	95.7	91.7	81.7	72.4	82.8	73.2	66.4	81.3
Tompson *et al.* [28]	96.1	91.9	83.9	77.8	80.9	72.3	64.8	82.0
Pishchulin *et al.* [20]	94.1	90.2	83.4	77.3	82.6	75.7	68.6	82.4
Wei *et al.* [30] + LSP	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.5
Ours	97.2	93.9	86.4	81.3	86.8	80.6	73.4	86.3

Table 3: Comparison with the state-of-the-art on the MPII datasets using PCKh @ 0.5.

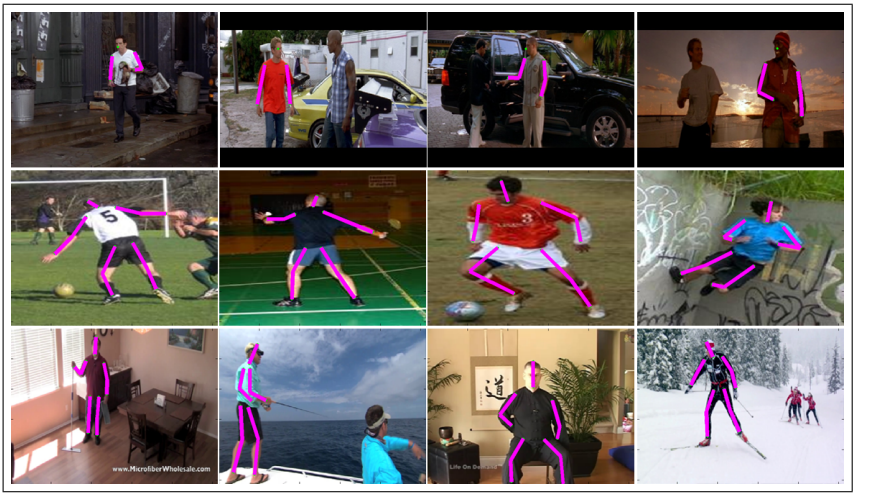

Figure 3: Qualitative results for the Frames Labelled In Cinema (FLIC) dataset [22], the Leeds Sports Pose (LSP) dataset [13, 14] and the MPII Human Pose (MPII) dataset [3].

5 Conclusion

In this work, we have proposed a deep network with a low memory footprint for human pose estimation that can be trained efficiently on a mid-range GPU. It achieves competitive results on popular benchmarks for human pose estimation, which is impressive since the model does not require any pre-training on large datasets as other models and can be trained from scratch also on small datasets like FLIC. The proposed network, which is publicly available, can serve as a baseline for more complex models in the future.

Acknowledgement: The work in this paper was funded, in parts, by the EU projects STRANDS (ICT-2011-600623) and SPENCER (ICT-2011-600877) and the ERC Starting Grant CV-SUPER (ERC-2012-StG-307432). Juergen Gall was supported by the ERC Starting Grant ARCA.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial Structures revisited: People Detection and articulated pose estimation. In *IEEE Conference on Computer Vision and Pattern*

Recognition, 2009.

[2] M. Andriluka, S. Roth, and B. Schiele. Monocular 3D pose estimation and tracking by detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[3] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2D Human Pose Estimation: New BenchMark. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[4] J. Carreira, P. Agarwal, K. Fragkiadaki, and J. Malik. Human Pose Estimation with Iterative Error Feedback. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[5] X. Chen and A. Yuille. Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations. In *Neural Information Processing Systems*, 2014.

[6] R. Collobert, K. Kavukcuoglu, and C. Farabe. Torch7: A matlab-like environment for machine learning. 2011.

[7] M Dantone, J Gall, C Leistner, and L. van Gool. Body Parts Dependent Joint Regressors for Human Pose Estimation in Still Images. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 36, pages 2131–2143, 2014.

[8] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[9] P. Felzenswalb and D. Huttenlocher. Pictorial Structures for object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[10] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik. Hypercolumns for Object Segmentation and Fine-grained Localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[11] P. Hu and D. Ramanan. Bottom Up and Top Down Reasoning with Hierarchical Rectified Gaussians. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[12] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

[13] S. Johnson and M. Everingham. Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. In *British Machine Vision Conference*, 2010.

[14] S. Johnson and M. Everingham. Learning Effective Human Pose Estimation from Inaccurate Annotation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[15] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

[16] G. Lin, C. Shen, A. van dan Hengel, and I. Reid. Efficient piece wise training of deep strucutred models for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[17] J. Long, E. Shelhamer, and T. Darrel. Fully Convolutional Netowrks for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[18] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet Conditioned Pictorial Structures. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[19] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Strong appearance and expressive spatial models for human pose estimation. In *International Conference on Computer Vision*, 2013.

[20] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[21] V. Ramakrishna, D. Munoz, M. Hebert, J. Bagnell, and Y. Sheikh. Articulated Pose Estimation via Inference Machines. In *European Conference on Computer Vision*, 2014.

[22] B. Sapp and B. Taskar. MODEC : Multimodel Decomposable Models for Human Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[23] L. Sigal and M. Black. Measure locally, reason globally:Occlusion-sensitive articulated pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[24] M. Sun and S. Savarese. Articulated part-based model for joint object detetion and pose estimation. In *International Conference on Computer Vision*, 2011.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[26] Y. Tian, L. Zitnick, and S. Narasimhan. Exploring the spatial hierarchy of mixture models for human pose estimation. In *European Conference on Computer Vision*, 2012.

[27] J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. In *Neural Information Processing Systems*, 2014.

[28] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient Object Localization Using Convolutional Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[29] A. Toshev and C. Szegedy. DeepPose: Human Pose Estimation via Deep Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[30] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional Pose Machines. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[31] W. Yang, W. Ouyang, H. Li, and X. Wang. End-to-End Learning of Deformable Mixture of Parts and Deep Convolutional Neural Networks for Human Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[32] Y Yang and D Ramanan. Articulated Pose Estimation using Flexible Mixtures of Parts. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.