Lightweight Bottom-up Approach For Multi-Human Pose Estimation Using Teacher Student Learning

Chinh La u7098799
Sneha Bahl u7006861
Amogh Dhaliwal u7053642
Safee Azam u7047994

Abstract

The current State of The Art (SOTA) approaches for 2D multi-human pose estimation have a very large number of parameters which smaller systems such as mobile devices or embedded systems cannot run in reasonable time. In this paper, we present SmallerHRNet: A lightweight bottom-up approach for multi-human pose estimation. We use teacher student learning on a current SOTA model to produce this model. The number of parameters and use of computational resources is significantly reduced. We demonstrate the advantage of our model by evaluating on the COCO Dataset. Our model outperforms the current SOTA lightweight models whilst having fewer parameters.

1 Introduction

2D human pose estimation is the task of identifying keypoints or joints (elbow, knee, etc.) for a given image of a person. This has been an increasingly popular area of research in the recent years. There are two broad approaches to human pose estimation; top-down approaches [4, 20, 26, 27, 29] use a person detector to encapsulate each person instance in a bounding box and then apply the task of single person pose estimation. This approach is accurate but is heavily dependent on the performance of the detector. Since the top-down methods need to run the pose estimation network for each detected human they are computational expensive specifically in an image crowded with people and they are not end-to-end networks. In comparison, the bottom-up approaches [2, 6, 15, 21, 24] identify keypoints and then groups them into person instances. This approach is faster but not as accurate as top-down methods due to some challenges such as scale variation, and generating heatmaps for accurate joint locations for people appearing further away.

In recent years, new SOTA results are often achieved by increasing model complexity [6, 18, 21]. These approaches focus on improving accuracy resulting in the computational resources being increased. As a result, many models cannot actually run on small devices like a mobile phone or embedded system with limited computational resources. Due to the practical needs, model compression has become a popular area of research. [8] has shown that there are a few techniques being used recently for model compression such as Pruning, Quantization, Low-rank factorization and Knowledge Distillation (or the teacher student learning). In the human pose estimation domain, some recent research shows promising results. [30] had shown that using knowledge distillation could achieve on par performance with many SOTA methods in keypoints detection with less than 20% resources on MPII benchmark. [13] had shown that using Knowledge Distillation could compress a model to run real time on IPhone for 3D human pose estimation. EfficientHRNet[17] attempts to compress a larger model by a close analysis of its structure. Lightweight OpenPose[19] changes the backbone of a large network to create a smaller network.

In this project, our aim is to combine the bottom-up approach of HigherHRNet with teacher student learning for the problem of multi-person pose estimation. Our aim is to achieve comparable accuracy to the current SOTA while reducing model complexity significantly. To be more specific:

1. We attempt to create a smaller model called SmallerHRNet based on HigherHRNet [6] using teacher student learning.
2. We compared our model with EfficientHRNet [17] and Lightweight OpenPose[19] in COCO 2017 keypoint detection benchmark [16].

2 Related works

2.1 2D Human Pose Estimation

There has been great progress in human pose estimation in the last decade, due to the advancements in deep learning networks [12, 18, 20]. However, the prior works in this field mainly focus on higher accuracy which come at the expense of increasing computational resources. This means models can be more accurate but few can be practically applied to real-world applications where computational resources are limited. There have been some prior attempts to improve runtime and complexity such as [28] and [1] but they show poor performance or propose general practices without providing a novel method.

Top-down methods Top-down methods first identify every person in the image with an existing object detector such as [4, 5, 23, 25], and then detect the keypoints for that instance. These keypoints are identified for a single person with methods like [9, 27], and for multi-person with methods like [4, 12, 20]. The performance of these methods rely heavily on their ability to detect people. This however, means that the inference time increases significantly for each additional person identified.

Bottom-up methods Bottom-up methods such as [6, 14, 21], detect all the keypoints in the image, and then group them into individuals. Grouping can be done through various methods such as [18] which uses associative embedding assigning each keypoint with a tag (vector) and then comparing and grouping the keypoints based on the L_2 distance. Another such bottom-up method is PifPaf [15], which as the name suggests uses PIF (Part Intensity Field) to identify body parts and PAF (Part Association Field) to link body parts to build human poses. Bottom-up methods are much faster for inference but have a trade-off with accuracy. Another problem which is encountered during bottom-up methods is that of scale variation. To reduce the negative impact of scale variations, feature pyramids are used but with cost of increased computational time [6].

2.2 Knowledge Distillation

Knowledge Distillation was proposed in [11] in an attempt to compress a neural network by transferring knowledge from a large neural network (known as the teacher) to a smaller structure (known as the student). The comprehensive survey[10] talks about the 3 methods knowledge can be transferred from teacher to student. They are- response-based knowledge, feature-based knowledge and relation-based knowledge. Knowledge distillation can be used in situations with limited training data or limited computational and power resources. It helps to enhance accuracy with no major increase in complexity. Initial work using knowledge distillation was performed on classification problems [3, 22], but more recently progress has been made towards human pose estimation with methods such as [13] which applies knowledge distillation to 3D pose estimation, achieving accurate and fast inference time running on mobile devices. Another method [31] uses radio waves to estimate human pose by following a teacher-student design.

3 SmallerHRNet: Lightweight Bottom-up Network For Multi-Human Pose Estimation Using Teacher Student Learning

3.1 Teacher and Student

We use HigherHRNet [6] as our teacher to train our student. HigherHRNet uses HRNet[27] as its backbone. HRNet starts with a high resolution branch, then at each following stage, new branches are created with the resolution being half the lowest resolution of the previous branches. In Figure 2, they show 3 of the 4 stages of the network's architecture. Each branch of each stage consists of multiple resolution modules stacking on each other. HigherHRNet has 4 stages with 1, 4 and 3 resolutions at stage 2, 3 and 4. HigherHRNet uses a deconvolution module at the end to generate a

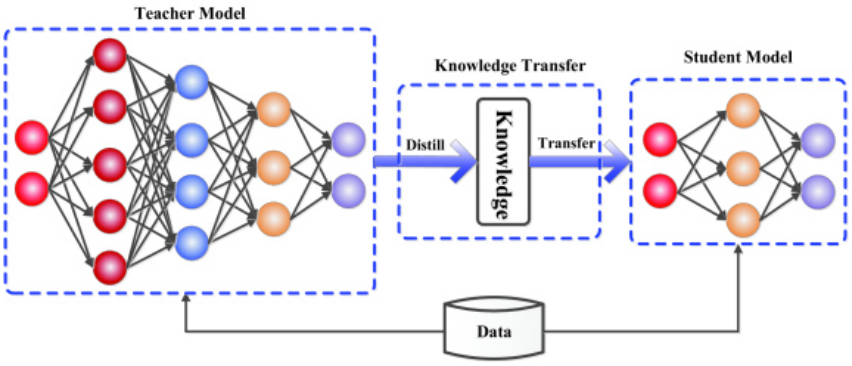

Figure 1: Basic Teacher-Student framework (image from [10])

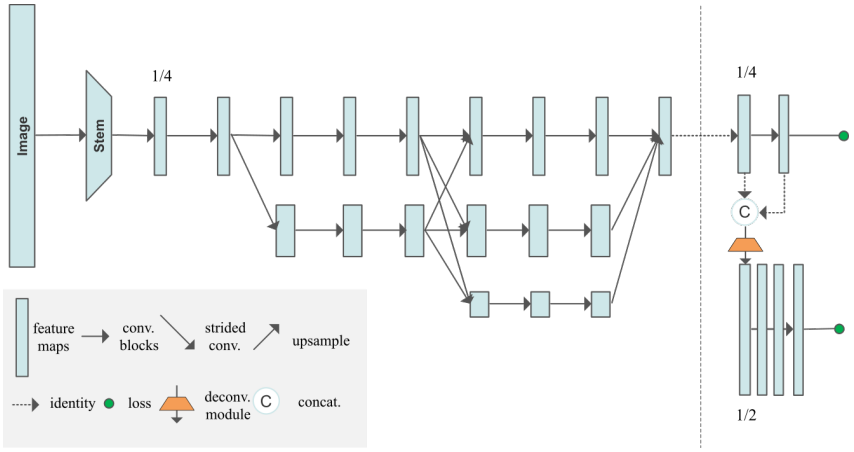

Figure 2: Architecture of HigherHRNet (Our teacher model) (image from [6])

high quality feature map with twice the resolution than the input feature maps. Predicted heatmaps are concatenated with the feature maps to form the input for the deconvolution module to predict heatmaps in a new scale. If we add more deconvolution modules, the output heatmaps will be multi-scale. Our pre-trained teacher model uses only 1 deconvolution module, as the result, we will have 2 predicted heatmaps with one being 2 times higher in resolution than the other. These multi-scale heatmaps are then scaled to the size of the input images using bilinear interpolation. Finally, the scaled heatmaps are aggregated by taking their average. This gives us the final prediction.

HigherHRNet uses Associative Embedding [18] for keypoint grouping. We only use Associative Embedding in the the lowest resolution output.

Similarly to [30], we observe that these resolution modules are repeated. We attempt to minimize the number of resolution blocks as well as the number of channels at each branch to create our student.

3.2 Loss function

The original loss function of HigherHRNet consists of two losses: The heatmap loss or sum of mean squared errors for all resolution and the grouping loss from [18] to predict the tags for grouping the joints.

$$\mathcal{L}_{HigherHRNet} = \mathcal{L}_{mse} + \mathcal{L}_{group} \tag{1}$$

Heatmap distillation loss: We use pose distillation loss from [30] to "transfer" heatmap knowledge from our teacher to our student.

$$\mathcal{L}_{pd} = \frac{1}{K}\sum_{i=1}^{K} \|m_i^t - m_i^s\|_2^2 \tag{2}$$

where m_i^t and m_i^s are the heatmap for k^{th}-joint predicted by pre-trained model (or our teacher) and our student model.

Grouping distillation loss: We observed the grouping output from the teacher model and found that it was similar to a feature map. We also chose a feature-based distillation loss to transfer grouping knowledge from the teacher to the student. The most common feature-based loss is $\mathcal{L}_2$ [10]. The distillation loss for grouping is:

$$\mathcal{L}_{gd} = \sum_{i}^{WHC} (T_i - S_i)^2 \tag{3}$$

where T_i and S_i are pixel-wise grouping value output from the teacher and student respectively.

The final training loss to train our student is:

$$\mathcal{L} = \alpha \mathcal{L}_{HigherHRNet} + (1 - \alpha)(\mathcal{L}_{pd} + \mathcal{L}_{gd}) \tag{4}$$

where α is a hyperparameter.

4 Experiments

4.1 Implementation Details

Training process: Using pre-trained HigherHRNet [6] as the teacher, we tried multiple versions of students to compare results. The teacher was previously trained for 300 epochs with rate decay at epoch 200 and 260. We trained our student with 100 epochs and rate decay at epoch 60 and 90 using Google Colab Pro with a Tesla V100 16GB. We used a batch size of 16. Each epoch took about 45 to 50 minutes. The total training pipeline took 3 to 4 days to finish as Colab Pro only allows a maximum of 24 hours per session. Following [7], we chose $\alpha = 0.9$

Data preparation: We used COCO 2017[16] dataset with 64115 train images and validate on validation set. We use the same data augmentation as in [6].

Students: We tried four different ways to construct our student:

- **SmallerHRNet1.5M**: decrease all number of stacked modules by 2, decrease number of channels by more than half, decrease deconvolution module parameters by 4.
- **SmallerHRNet2.3M**: keep same number of stacked modules like our teacher, decrease number of channels to be the same as SmallerHRNet1.5M, keep same deconvolution parameters as teacher.
- **SmallerHRNet2.84M**: decrease all number of stacked modules by 2, decrease number of channels by 2, decrease deconvolution module parameters by 2.
- **SmallerHRNet 2.96M**: decrease all number of stacked modules by 2, decrease number of channels by 2, keep same deconvolution parameters as teacher.

Details about each student model can be found in our configuration files

Testing: We compare our students' performance on COCO2017 val set with EfficientHRNet [17] and Lightweight OpenPose [19].

4.2 Results

In Table 1, we present the results of our students compared with other models. Our SmallerHRNet2.3M has only 2.3M parameters and performs on par with EfficientHRNet$_{-3}$ which has 3 times as many parameters. We also outperform EfficientHRNet$_{-4}$ and Lightweight OpenPose which have almost double the number of parameters.

We can clearly see that choosing the same number of modules in each stage as in the teacher and keeping the deconvolution module unchanged gives us the best result.

However, from Table 1, we could see that in exchange for a better performance, the computational complexity increases significantly. Based on the variations of the models tested, we conclude that the deconvolution module requires high computation cost.

Looking at the output results in Figure 3, we can see that some output images have been estimated very well (1,2,3,4,8). However, we notice that pictures with large people often results in some errors (our model confuses joints of one person with another). Smaller people, however, are estimated better, as can be seen in image 5 and image 6. We notice that partially visible joints can cause grouping errors, even for smaller people, see image 6 and 9.

Table 1: Comparisons with other approaches on the COCO2017 valset. (* See remarks)

Network	**Accuracy(%)**	**Correct(%)**	**Incorrect(%)**
OpenPose [2]	25.9M	160B	65.3
85.2	71.3	62.2	70.7
-			
PersonLab [21]	68.7M	405.5B	66.5
86.2	71.9	62.3	73.2
-			
HRNet [27]	28.5M	38.9B	64.1
86.3	70.4	57.4	73.9
HigherHRNet [6]	28.6M	47.9B	68.4
88.2	75.1	64.4	74.2
74.9			
Lightweight OpenPose	4.1M	9.0B	42.8
-	-	-	-
-			
EfficientHRNet$_0$	23.3M	25.6B	64.8
85.3	70.7	-	-
-			
EfficientHRNet$_{-1}$	16M	14.2B	59.2
82.6	64.0	-	-
-			
EfficientHRNet$_{-2}$	10.3M	7.7B	52.9
80.5	59.1	-	-
-			
EfficientHRNet$_{-3}$	6.9M	4.2B	44.8
76.7	48.2	-	-
-			
EfficientHRNet$_{-4}$	3.7M	2.1B	35.7
69.6	33.7	-	-
-			
SmallerHRNet1.5M	1.5M	8.66B	27.6
61.4	20.9	26.7	28.2
36.2			
SmallerHRNet2.3M	2.3M	15.77B	44.3
72.7	45.5	39.8	50.9
51.4			
SmallerHRNet2.8M*	2.84M	9.87	33.4
-	-	-	-
-			
SmallerHRNet2.9M	2.96M	14.92B	39.7
69.0	39.6	36.1	44.4
47.5			

Figure 3: Some output images from SmallerHRNet2.3M

4.3 Ablation Studies

Effect of associative embedding distillation: At first, the distillation loss only contained the heatmap loss between the teacher and the student. The results of our first student model, which had only 1.5M parameters can be seen in Table 2. SmallerHRNet1.5 is the model which was trained with only distillation heatmap loss, SmallerHRNet1.5+ is the model which was trained with both distillation losses. As we can see, adding the tagmap distillation loss gives us a better result.

Table 2: Comparison in performance for with and without Tagmap distillation loss on the COCO2017 valset, + means that model was trained with Tagmap distillation

Model	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
SmallerHRNet1.5M	24.4	57.8	16.7	23.9	24.1	33.3
SmallerHRNet1.5M+	**27.6**	**61.4**	**20.9**	**26.7**	**28.2**	**36.2**

Effect of different distillation weight: As [7] mentioned, a popular choice for α is $\alpha = 0.9$. However, upon examination of [30], we found that they used $\alpha = 0.5$. We tested model performance with $\alpha = 0.9$ and $\alpha = 0.5$ on SmallerHRNet2.84M. In Table 3 we can clearly see that $\alpha = 0.9$ gave us a much better result.

Table 3: Comparison in performance for different distilling weights on the COCO2017 valset with SmallerHRNet2.8M

α	AP
$\alpha = 0.9$	**33.4**
$\alpha = 0.5$	22.8

Effect of training time: Our teacher model was trained for 300 epochs, so we tested the effect of training our students for 300 epochs as well. We trained with a rate decay at epoch 200 and epoch 260. This results in Table 4, clearly show that longer training has positive effects on our models' performance. This shows that our model has not been overfit to the dataset, and improves with further training.

Table 4: Comparison in performance for more training epochs on the COCO2017 valset. (* See remarks)

Model	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
SmallerHRNet2.3M 100 epochs	44.3	72.7	45.5	39.8	50.9	51.4
SmallerHRNet2.3M 300 epochs	**44.6**	**73.5**	**45.8**	**40.0**	**51.4**	**51.6**
SmallerHRNet2.8M 100 epochs*	33.4	-	-	-	-	-
SmallerHRNet2.8M 300 epochs	**36.2**	**70.6**	**34.6**	**34.6**	**38.0**	**44.9**

Remarks: The original result of SmallerHRNet2.8M (with 100 epoch) was lost, we tried to to retrain it again and we realized that performance was different. It was much worse (with $AP = 6.4$) than the first time we trained it. We then tried to retrain it with $\alpha = 0.5$ and again got worse results (with $AP = 6.3$). We assume that training for this model might not be easy or they might have overfitted at some earlier epoch. Due to time constraints we decided not to train this model further.

5 Conclusion

We successfully created SmallerHRNet a lightweight bottom up network based on HigherHRNet which was trained using teacher student learning. After experimentation, we found the best student model to be the one with the same deconvolution module and stacked modules as HigherHRNet, but with fewer than half the number of channels. This student is composed of less than 10% of the number of parameters and uses less than 15% of the FLOPs compared to HigherHRNet. Validation of our

model on the COCO2017 dataset gives an AP of 44.3, outperforming other lightweight models with fewer parameters than them as well. Our model does well on certain difficult poses and single person pose estimation. While our model is small in terms of size, our model's computation complexity is higher compared to other lightweight models.This is due to it retaining the full deconvolution module from HigherHRNet, which is computationally complex, but clearly important for accuracy.

We successfully demonstrated the applicability of teacher student learning in 2D multi-person pose estimation. This is a promising method for compressing large models. For future works we would extend this research into 3D pose estimation and real time pose estimation in videos.

6 Acknowledgement

We would like to thank Dylan Campbell, our course convener for facilitating this course and enabling our research as well as our tutors Sameera Ramasinghe and Lin Li for providing feedback and support. We would also like to thank Nguyen Huu Duc, a masters student from RWTH Aachen University for helping with experiment setups.

References

[1] A. Bulat and G. Tzimiropoulos. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. doi: 10.1109/iccv.2017.400. URL `http://dx.doi.org/10.1109/ICCV.2017.400`.

[2] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields, 2019.

[3] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker. Learning efficient object detection models with knowledge distillation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 742–751. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/6676-learning-efficient-object-detection-models-with-knowledge-distillation.pdf`.

[4] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun. Cascaded pyramid network for multi-person pose estimation, 2017.

[5] B. Cheng, Y. Wei, H. Shi, R. Feris, J. Xiong, and T. Huang. Decoupled classification refinement: Hard false positive suppression for object detection, 2018.

[6] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *CVPR*, 2020.

[7] J. H. Cho and B. Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[8] T. Choudhary, V. Mishra, A. Goswami, and J. Sarangapani. A comprehensive survey on model compression and acceleration. *Artificial Intelligence Review*, 53(7):5113–5155, 2020. doi: 10.1007/s10462-020-09816-7.

[9] M. Dantone, J. Gall, C. Leistner, and L. van Gool. Human pose estimation using body parts dependent joint regressors. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3041–3048, Portland, OR, USA, June 2013. IEEE.

[10] J. Gou, B. Yu, S. J. Maybank, and D. Tao. Knowledge distillation: A survey, 2020.

[11] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL `http://arxiv.org/abs/1503.02531`.

[12] S. Huang, M. Gong, and D. Tao. A coarse-fine network for keypoint localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3028–3037, 2017.

[13] D.-H. Hwang, S. Kim, N. Monet, H. Koike, and S. Bae. Lightweight 3d human pose estimation network training using teacher-student learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[14] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele. Arttrack: Articulated multi-person tracking in the wild, 2016.

[15] S. Kreiss, L. Bertoni, and A. Alahi. Pifpaf: Composite fields for human pose estimation, 2019.

[16] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft coco: Common objects in context, 2015.

[17] C. Neff, A. Sheth, S. Furgurson, and H. Tabkhi. Efficienthrnet: Efficient scaling for lightweight high-resolution multi-person pose estimation, 2020.

[18] A. Newell, Z. Huang, and J. Deng. Associative embedding: End-to-end learning for joint detection and grouping, 2016.

[19] D. Osokin. Real-time 2d multi-person pose estimation on cpu: Lightweight openpose. In *arXiv preprint arXiv:1811.12004*, 2018.

[20] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[21] G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model, 2018.

[22] M. Phuong and C. Lampert. Towards understanding knowledge distillation. volume 97 of *Proceedings of Machine Learning Research*, pages 5142–5151, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL `http://proceedings.mlr.press/v97/phuong19a.html`.

[23] 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, 2015.

[24] Y. Raaj, H. Idrees, G. Hidalgo, and Y. Sheikh. Efficient online multi-person 2d pose tracking with recurrent spatio-temporal affinity fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[25] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2015.

[26] K. Su, D. Yu, Z. Xu, X. Geng, and C. Wang. Multi-person pose estimation with enhanced channel-wise and spatial information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[27] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[28] J. G. Umer Rafi, Bastian Leibe and I. Kostrikov. An efficient convolutional network for human pose estimation. In E. R. H. Richard C. Wilson and W. A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 109.1–109.11. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.109. URL `https://dx.doi.org/10.5244/C.30.109`.

[29] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao. Deep high-resolution representation learning for visual recognition, 2020.

[30] F. Zhang, X. Zhu, and M. Ye. Fast human pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[31] M. Zhao, T. Li, M. Abu Alsheikh, Y. Tian, H. Zhao, A. Torralba, and D. Katabi. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.