Machine Perception Report [Machine Polpette]

Alessandro Burzio Elisabetta Fedele Lorenzo Liso Ravi Srinivasan

ABSTRACT

The goal of this work is to estimate the 3D body pose from monocular 2D images, without any additional information. This task is becoming relevant in a growing number of fields and has indeed gained increased interest from the computer vision community.

Based on previous work done in this field by Kocabas et al. [4] we aim at implementing a neural network which is able to perform robustly for this task, even if trained on a smaller dataset.

1 INTRODUCTION

Regressing 3D human pose has several applications (e.g. human motion analysis, human computer interaction, robots). This area of research has recently become relevant in the computer vision community, and a large number of approaches have been proposed over the past decade, largely improving the performance on existing benchmarks. In particular, the community is now focused on obtaining models which are robust to small perturbations in the observations of the body and its part (e.g. object- and self-occlusion [4]). The task is to take a monocular RGB 2D image and estimate the 3D human pose and shape (HPS). In our case, our model regresses the parameters of SMPL [5].

In this project we have decided to integrate PARE [4] in the given skeleton code. The main differences with the original PARE implementation are the end-to-end loss (we do not compute an intermediate loss for the part segmentation through the whole training process) and the fact that we don't use occlusions during the training phase. Some other minor modifications in the architecture were also implemented.
Our model was trained on 3DPW [7] and MPII [1], while evaluation and testing was done only on 3DPW.

2 METHOD

You may refer to Figure 1 for the model architecture.

Given the input image, it is first passed to the backbone module. We implemented the backbone network using HRNet-W32 [6] trained on MPII. This architecture extracts high resolution feature maps that are then fed into a convolutional layer to extract the part-features.

The part-features are then processed by two separate branches: the 3D Body Branch and the 2D Part Branch. The 2D Part Branch is denoted as $P \in \mathbb{R}^{H \times W \times (J+1)}$ where J is the number of joints and an additional channel has been added to model the background mask. The 3D body branch is denoted as $F \in \mathbb{R}^{H \times W \times C}$ and is used for estimating the camera and shape parameters. Both branches use convolutional layers to extract F and P.

P and F are then passed to two different part attentions modules.

The first one (F' in Figure 1) uses P as a soft attention mask to aggregate the features in F. F' rows are then passed to MLPs to obtain the pose parameters θ_i.

The second one (F'' in Figure 1), instead, is the same as the first module but uses a downsampled version of P instead, to reduce the computational cost. The obtained matrix F'' is then used to extract the camera and shape features through the use of MLPs.

As mentioned before, we used an end-to-end loss to train our models. Overall, our total loss is:

$$\mathcal{L} = \lambda_{\text{keypoints}}(\mathcal{L}_{3D} + \mathcal{L}_{2D}) + \lambda_{\text{pose}}\mathcal{L}_{\text{pose}} + \lambda_{\beta}\mathcal{L}_{\beta}$$

Let us now explain the meaning of each loss term.

$\mathcal{L}_{3D} = ||\mathcal{J}_{3D} - \hat{\mathcal{J}}_{3D}||$ is the loss related to the positions of the 3D keypoints. In particular, $\hat{\mathcal{J}}_{3D}$ are the 3D joint locations regressed using a pretrained linear regressor W as $\hat{\mathcal{J}}_{3D} = W\mathcal{M}(\theta, \beta)$.

$\mathcal{L}_{2D} = ||\mathcal{J}_{2D} - \hat{\mathcal{J}}_{2D}||$ is the loss related to the positions of the 2D keypoints. In particular, $\hat{\mathcal{J}}_{2D}$ is the 2D projection of the 3D joints, obtained using the regressed camera parameters.

The last two losses $\mathcal{L}_{\text{pose}}$ and $\mathcal{L}_{\beta}$ are the losses computed directly on the SMPL parameters.

The λ factors are the weights assigned to each loss. In particular, as suggested in the skeleton code, we set the following weights: $\lambda_{\text{keypoints}} = 5$, $\lambda_{\text{pose}} = 1$, $\lambda_{\beta} = 0,001$.

3 EVALUATION

In order to define our final model we have tried different approaches. In particular, we conducted experiments to choose the backbone network, the learning rate and the batch size. In addition, as suggested in the PARE approach, we tried to perform image augmentation on our training set (i.e. synthetic occlusions).

In our first attempts, we used ResNet50 [3] trained on MPII as a backbone. We then performed some experiments using HRNet-W32 trained on the same dataset, which led to improved results in the validation phase. As a consequence, we decided to keep the latter in our final architecture.

With the HRNet-W32 backbone we then tested the performance of the model trained with synthetic occluded input images, as suggested in [4]. We observed an increase in the loss and we decided not to train the model with occlusions.

Moreover, we tested the influence of the learning rate on the training and validation phase. We trained our model first with $\text{lr} = 5 \cdot 10^{-4}$ and we saved the best 30 checkpoints. We then resumed the training from the checkpoint with the lowest validation loss, reducing the learning rate to $\text{lr} = 5 \cdot 10^{-5}$. Using this technique we observed significantly faster convergence rate and better results.

In general, we achieved the best results by using a batch size of 32 (assume a batch size of 32 unless stated otherwise).

In our last experiment we decided to change the sampling ratios from the two training datasets, giving higher weight to 3DPW [7] (0.6) than to MPII [1] (0.4) and it improved even more the perforfance of our architecture.

All the details of the previously discussed results are presented in *Table* 1.

4 DISCUSSION

In this section we will discuss the previously presented choices.

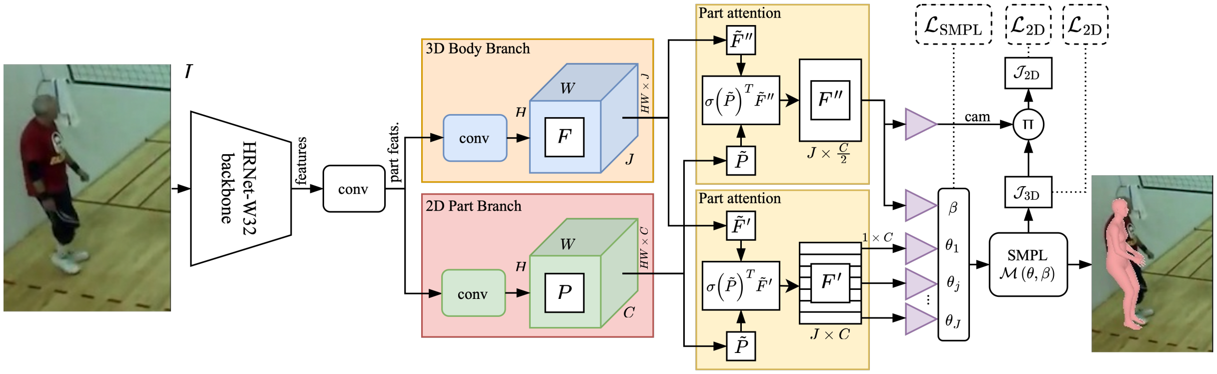

Figure 1: Model architecture: given an input image, the model extracts two feature maps, which are combined by two different part attention layers: the resulting F' and F'' are then used for camera and SMPL body regression.

Table 1: Validation scores during our experiments.

Method	PA-MPJPE (mm) ↓
ResNet50	82.15
HRNet-W32 with occlusions, batch size= 64	61.08
HRNet-W32, no occlusions, batch size = 64	60.01
HRNet-W32, lr = $5 \cdot 10^{-5}$	58.99
HRNet-W32, decreasing lr	57.5

The first important decision regarded the selection of the backbone network. We decided to use HRNet-W32 rather than ResNet50 to extract relevant features as it was performing significantly better in our experimental tests. This decision was also supported by further experiments conducted by Kocabas et al. [4].

Furthermore, we decided not to train our models on occlusions since the model had poorer performance. Even if usually data augmentations techniques are known to lead to better results, in our case including occlusion would have lead to worse results, but we think it may be due to the small dimensions of the training dataset.

We believe that using a learning rate of 10^{-4} rather than $5 \cdot 10^{-5}$ not only improves the overall performance of the model, but also improves the learning phase as allows the trainer not to get stuck in local minima. In addition, we reduced the learning rate in the last epochs since models have proven to often benefit from reducing the learning rate once learning stagnates.

The choice of a smaller batch size (32 instead of 64) is known to offer a regularizing effect [2]. To reduce the training time and the memory footprint when training with batch size = 32, we used the automatic mixed precision of 16-bit

We are strongly convinced that sampling more from 3DPW improved the performances as it contains more accurate predictions of the joints by construction.

5 CONCLUSION

In this project we presented an integration of the architecture presented in PARE, which aims to achieve higher generalization even if trained on smaller datasets. By trying different approaches and fine-tuning our model we achieved a precision of 57.50 mm on the joints' predictions, which we consider a satisfactory result, given the limitations previously discussed.

REFERENCES

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2014. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.03385 (2015). arXiv:1512.03385 http://arxiv.org/abs/1512.03385

[4] Muhammed Kocabas, Chun-Hao P. Huang, Otmar Hilliges, and Michael J. Black. 2021. PARE: Part Attention Regressor for 3D Human Body Estimation. *CoRR* abs/2104.08527 (2021). arXiv:2104.08527 https://arxiv.org/abs/2104.08527

[5] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16.

[6] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. 2019. Deep High-Resolution Representation Learning for Human Pose Estimation. arXiv:arXiv:1902.09212

[7] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. 2018. Recovering Accurate 3D Human Pose in The Wild Using IMUs and a Moving Camera. In *European Conference on Computer Vision (ECCV)*.