Unsupervised Human Pose Estimation through Transforming Shape Templates: Supplemental Material

Luca Schmidtke[1], Athanasios Vlontzos[1], Simon Ellershaw[1], Anna Lukens[3], Tomoki Arichi[2], and Bernhard Kainz[1]

[1]Imperial College London, [2]King's College London, [3]Evelina London Children's Hospital

1. Network Architectures

In this section, we present the details regarding network architecture. Each block represents a layer. Abbreviations are as following: k=kernel size, s=stride, d = dilation, ch= number of channels, n=number of (fully connected) neurons.

1.1. Our Approach

Conv (k=3, s=1, d=1, ch=32)	256x256
Conv (k=3, s=1, d=4, ch=32)	
Conv (k=3, s=2, d=1, ch=64)	128x128
Conv (k=3, s=1, d=8, ch=64)	
Conv (k=3, s=2, d=1, ch=128)	64x64
Conv (k=3, s=1, d=1, ch=128)	
Conv (k=3, s=2, d=1, ch=256)	32x32
Conv (k=3, s=1, d=1, ch=256)	
Conv (k=3, s=2, d=1, ch=32)	16x16
Conv (k=3, s=1, d=1, ch=256)	
Conv (k=3, s=2, d=1, ch=256)	8x8
Conv (k=3, s=1, d=1, ch=256)	
Conv (k=3, s=2, d=1, ch=256)	4x4
Conv (k=3, s=1, d=1, ch=256)	
Linear (n=512)	6k
Linear (n=6k)	

Figure 1: Architecture of **pose extractor** φ**.** All layers are followed by batch normalisation and LeakyReLU except for the last.

Conv (k=3, s=1, d=1, ch=32)	256x256
Conv (k=3, s=1, d=4, ch=32)	
Conv (k=3, s=2, d=1, ch=64)	128x128
Conv (k=3, s=1, d=8, ch=64)	
Conv (k=3, s=2, d=1, ch=128)	64x64
Conv (k=3, s=1, d=1, ch=128)	
Conv (k=3, s=2, d=1, ch=256)	32x32
Conv (k=3, s=1, d=1, ch=256)	
Bilinear Upsampling	64x64
Conv (k=3, s=1, d=1, ch=128	
Bilinear Upsampling	128x128
Conv (k=3, s=1, d=1, ch=32	
Bilinear Upsampling	256x256
Conv (k=3, s=1, d=1, ch=32	
Conv (k=3, s=1, d=1, ch=3	

Figure 2: Architecture of **image decoder** ϕ**.** All layers are followed by batch normalisation and LeakyReLU except for the last.

1.2. Jakab et al. [1]

For more details regarding the training procedure, refer to [1] and the accompanying supplemental material.

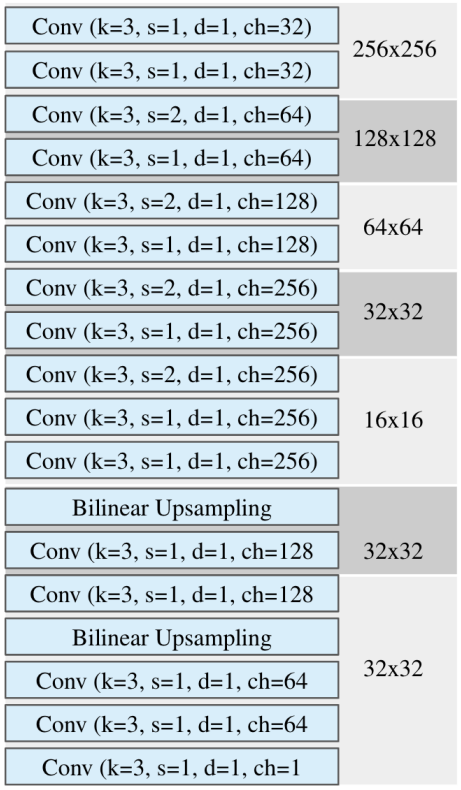

Figure 3: Architecture of **skeleton encoder** All layers are followed by batch normalisation and LeakyReLU except for the last.

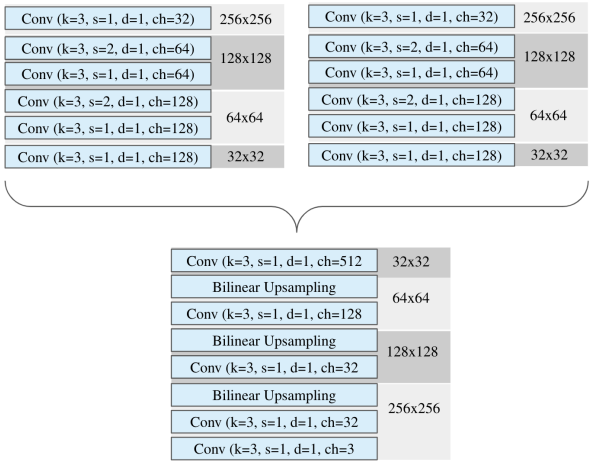

Figure 4: Architecture of **image decoder.** All layers are followed by batch normalisation and LeakyReLU except for the last.

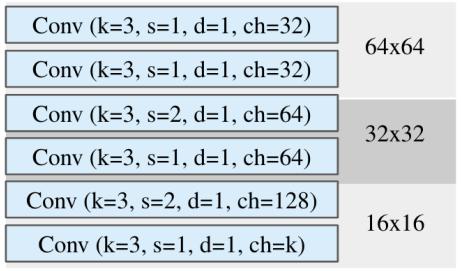

Figure 5: Architecture of **skeleton regressor.** All layers are followed by batch normalisation and LeakyReLU except for the last.

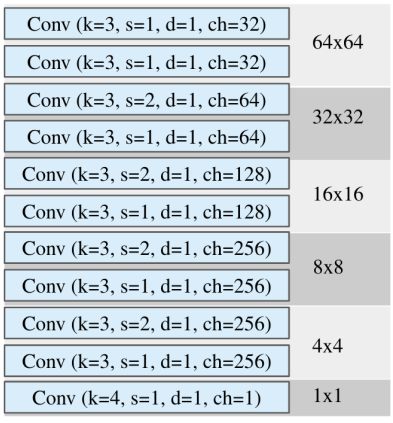

Figure 6: Architecture of **the skeleton discriminator** All layers are followed by instance normalisation and LeakyReLU except for the last.

2. Results

2.1. Human3.6m

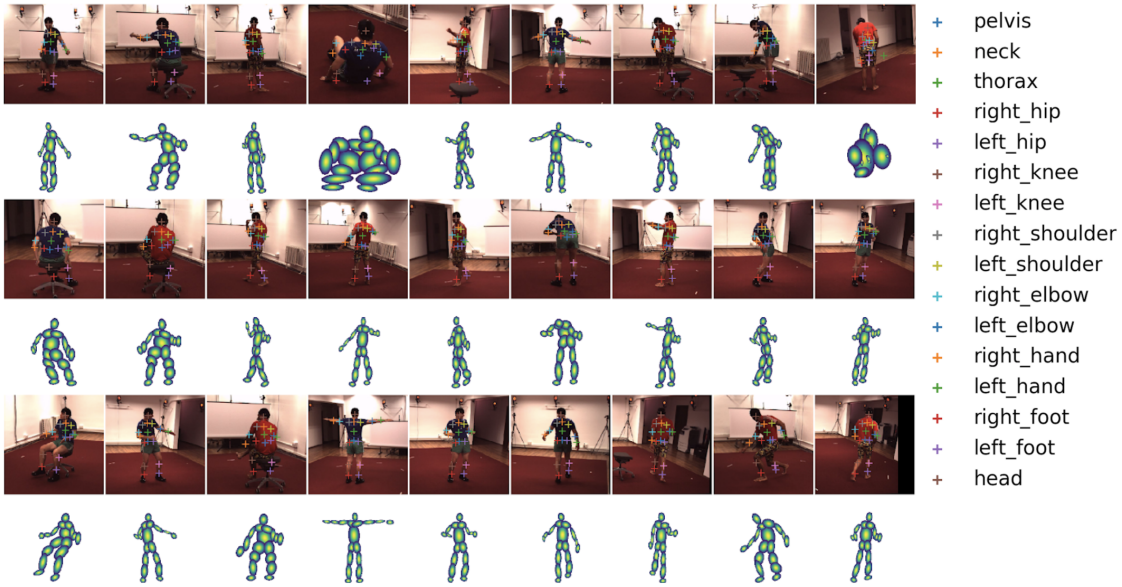

Figure 7: Results for Human 3.6m. Markers denote detected landmarks.

2.2. Infants

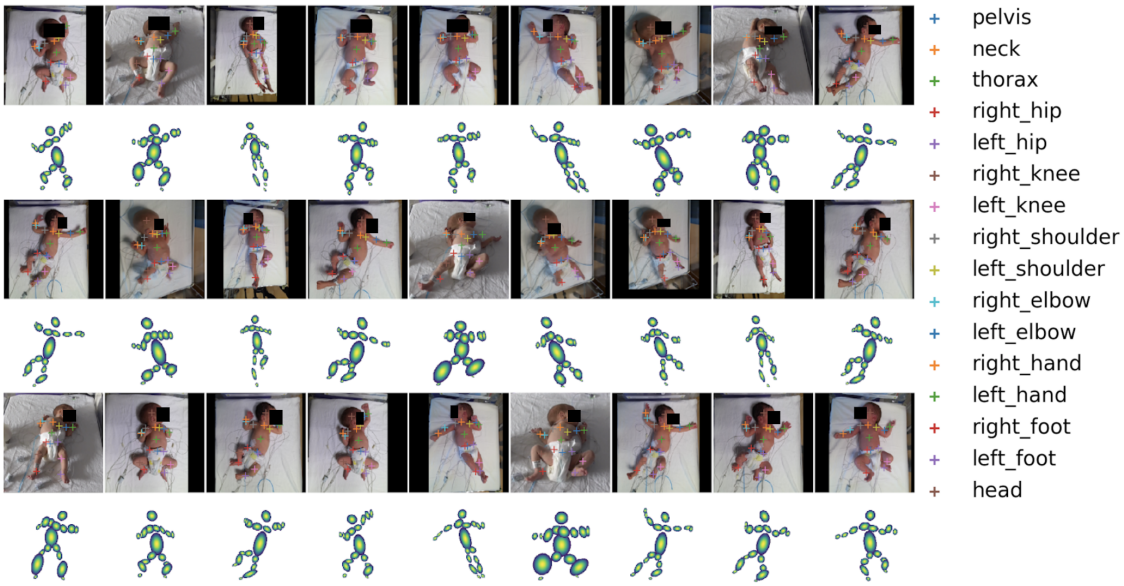

Figure 8: Results for infants. Markers denote detected landmarks.

3. Training and evaluation details

3.1. Human3.6m

The model is trained with the Adam optimizer [2], a learning rate of 1e-4 and a batch size of 48. For evaluation, we choose 16 keypoints corresponding to following indices: 0, 13, 12, 1, 6, 2, 7, 25, 17, 26, 18, 27, 19, 3, 8, 14. While most of these keypoints conincide with our anchor points, we place two additional points on the thorax and the head in our default template.

3.2. Infants

For the infants, both models are trained the Aadam optimizer [2], a learning rate of 1e-4 and a batch size of 48. In order to train [1] with labels from the synthetic infants dataset, we project 3d coordinates into the image plane via a perspective camera transformation. Since the camera position in our clinical infant dataset is varying in terms of tilting angles, we augment the data via random rotations of the camera before projection in an effort to mimic these conditions.

References

[1] Tomas Jakab, A. Gupta, Hakan Bilen, and A. Vedaldi. Self-supervised learning of interpretable keypoints from unlabelled videos. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8784–8794, 2020.

[2] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.