

Real-time 3D Human Pose Estimation Model for Physiotherapy Exercises on Mobile Devices

Hamdi Yalın Yalıcı, Alaettin Uçan, Ali Yaşar Yiğit, Adem Ali Yılmaz
Department of Research and Development
Tiga Information Technologies
Ankara, Türkiye
{yalin.yalic, alaettin.ucan, ali, adem}@tigahealth.com

Abstract—We present a lightweight, real-time 3D human-pose estimation model specific to mobile devices that can be used for physiotherapy exercises. Our two-stage training strategy first learns 2D joint coordinates via an EfficientNetB2-based backbone, then freezes those weights and adds a compact regressor for the third (depth) dimension. Deployed in ONNX format, the model delivers full-body 3D poses with 30 joints, which is more comprehensive than common 17-point joints, while achieving accuracy on par with state-of-the-art methods. Extensive experiments on the public Human3.6M dataset and the TheraPose physiotherapy dataset demonstrate real-time inference (less than 15 ms on ARM CPUs) and mean per-joint position errors of less than 40 mm, making it ideal for mobile health and exercise-monitoring applications.

Keywords—3D human pose estimation, physiotherapy exercises, lightweight model optimization.

I. INTRODUCTION

3D human pose estimation has become more popular due to the rise of mobile health and remote exercise apps. To track full-body motion, study biomechanics, and give real-time feedback during exercise routines, you need accurate 3D pose information. This is especially important in physiotherapy, where even small changes in joint angles or posture can have big effects on how well the treatment works. 2D pose estimation works well for basic posture tracking, but it doesn't give us the depth information we need to measure range of motion, limb orientation, and spinal curvature, which are all important parts of therapeutic assessments and rehabilitation exercises.

Using 3D pose estimation models on mobile devices could make physiotherapy more accessible to everyone by allowing low-cost, real-time analysis without the need for special hardware. But getting to this vision is not without its own set of problems. Mobile and embedded platforms are limited in their capabilities because they do not have enough processing power, memory, or battery life. Most of the time, state-of-the-art (SOTA) 3D pose estimation models are big, require a lot of processing power, and are designed for desktop GPUs, which means they cannot be used directly on mobile devices.

To address these limitations, we present a real-time 3D pose estimation model explicitly optimized for mobile devices. Our method uses a two-stage training strategy: we first train a 2D joint coordinate regressor using an EfficientNetB2 backbone,

then freeze its weights and train a compact depth regressor to estimate the Z-coordinates. This modular approach makes optimization easier, reduces the cost of computation, and improves generalization in different poses. The model gives 30 joint coordinates, which are visualized in Figure 1, in 3D, providing more detailed information than most 18-joint COCO-based systems. This long skeleton has four spine-related joints that are very important to check and fix posture, balance, and alignment during physiotherapy exercises. This level of detail makes it easier to monitor areas such as lumbar extension, thoracic flexion, and cervical adjustments, which are areas that traditional skeleton layouts often overlook.

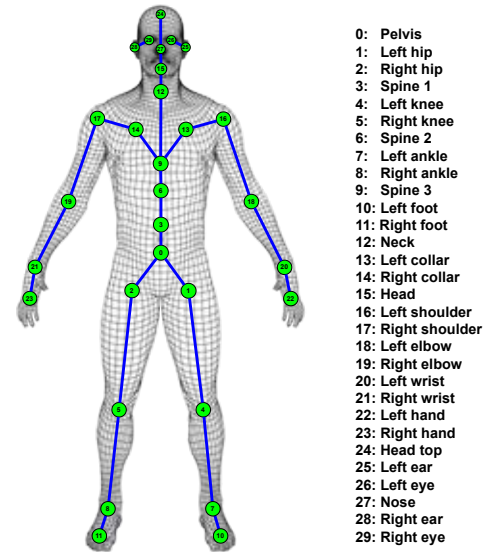


Fig. 1: Human pose model joint points

We use a number of optimization methods, such as post-training quantization and exporting to the ONNX format [1], to ensure that edge devices work in real time. Our last model is less than 10 MB in size and can process frames in less than 15 milliseconds on mid-range ARM CPUs. It also has comparable accuracy on 3D pose benchmarks.

In summary, our key contributions are:

- A two-stage 2D→3D training pipeline that decouples depth learning, improving efficiency and robustness.
- A lightweight EfficientNetB2-based architecture with ONNX deployment support, under 10 MB in size.
- A 30-joint pose representation specifically designed for physiotherapy, with enhanced spinal joint coverage.
- Real-time inference on commodity mobile hardware with competitive accuracy compared to existing 3D pose estimation models.

II. RELATED WORK

There have been a number of attempts to estimate poses on platforms with limited resources, with a focus on real-time 2D pose estimation. BlazePose by Bazarevsky et al. [2] was the first landmark model made for mobile phones. It ran at over 30 FPS on a Pixel 2 device. It predicts 33 full-body landmarks using a regression head without a heatmap, which makes it very useful for real-time uses like fitness tracking. Zhang et al.'s MobiPose [3] also aims to improve multi-person 2D pose estimation on smartphones by combining motion tracking, lightweight convolutional architectures, and parallel scheduling for mobile CPUs. These works show excellent real-time 2D performance, but they do not estimate 3D joint positions, which are very important for understanding depth and posture in physiotherapy situations. Another work has explored two-stage architectures that decouple 2D estimation and 3D lifting. Zhao et al. [4] proposed a transformer-based 3D regressor that leverages frozen 2D features and contextual image cues, improving depth accuracy while maintaining modularity.

Researchers have explored methods to compress and deploy models to bring 3D pose estimation to mobile devices. Hwang et al.'s MoVNect [5] uses knowledge distillation to train a MobileNetV2-based model to figure out the 3D pose of one person. MoVNect runs at more than 20 frames per second on mobile CPUs when used with joint-angle filters for smoothing over time. In the same way, Choi et al.'s MobileHumanPose [6] uses a U-Net-like structure with skip connections and parametric activations to get 3D estimates in less than 13 ms per frame. In TransSMPL [7], a compressed transformer-based SMPL model, other methods like pruning and quantization have been looked into. TransSMPL uses early-exit logic and dynamic quantization to significantly reduce memory use while still accurately estimating the shape and pose of a 3D body.

Recent works have begun customizing pose estimation systems for clinical and rehabilitation use. Mercadal-Baudart et al. [8] demonstrated that a lightweight transformer model trained on exercise data could achieve high accuracy (less than 10 degrees error) in estimating joint angles critical to physiotherapy, such as pelvic tilt and knee flexion. They use 3D keypoints estimated from monocular RGB to measure full-body movement during rehabilitation exercises. Lee et al. [9] did another study that was based on physiotherapy. They used a 2D-to-3D lifting pipeline to find forward head posture, a common musculoskeletal problem, and got promising classification results using only upper-body joint data. These studies show how useful high-resolution joint tracking is for clinical

feedback, but most of them use pre-trained general-purpose skeletons (like 17 joints), which might miss important spinal or therapy-specific areas.

III. METHODOLOGY

A. Model Architecture

Our model features a two-stage architecture designed to accelerate and enhance the accuracy of estimating 3D human poses. We use an EfficientNet-B2 backbone as the basis for our method. This backbone is a lightweight but powerful feature extractor. Two task-specific heads come after this backbone: a 2D pose estimation head and a depth regression head. Model architecture is illustrated in Figure 2.

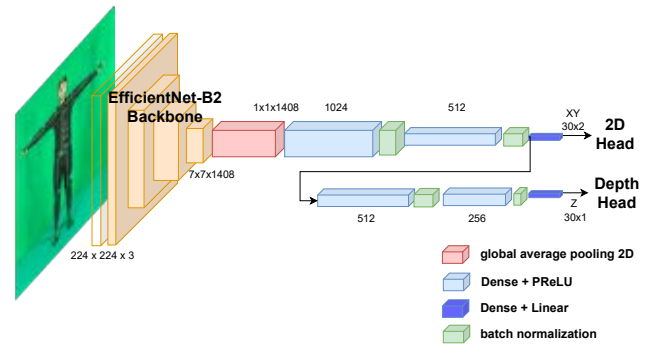


Fig. 2: 3D Pose Estimation Model Architecture

1) *2D Head*: After the convolutional layers of EfficientNet-B2, the extracted feature maps are globally averaged and passed through two fully connected layers, each followed by batch normalization and a Parametric ReLU (PReLU) activation function. The last layer of output is a dense linear layer with 60 units, which are the (x, y) coordinates of 30 joints. We chose PReLU as the activation function because it learns the negative slope of the activation on its own. This makes it easier to generalize across different poses and improves gradient flow in small models, which is especially useful for mobile devices that have limited computational resources.

2) *Depth Head*: After training and freezing the 2D backbone and prediction layers, a second head is added to the 2D output. This head gets the flattened 2D joint coordinates as input and sends them through two dense layers, each of which is followed by batch normalization and PReLU activation. The last linear layer gives the Z-coordinates of the 30 joints, which offers complete 3D spatial information. This modular design allows for efficient fine-tuning of the depth regressor while preserving 2D accuracy.

B. Loss Functions

1) *2D Pose Loss*: During the first stage, the model is trained to minimize the mean squared 2D distance between predicted and ground truth keypoints. Let $\hat{Y}_{2D} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{30}$ and

$Y_{2D} = \{(x_i, y_i)\}_{i=1}^{30}$ be the predicted and true 2D coordinates, respectively. The loss is defined as:

$$\mathcal{L}_{2D} = \frac{1}{30} \sum_{i=1}^{30} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]$$

This function encourages the network to minimize the Euclidean distance between joint predictions in image space.

2) *3D Fine-tuning Loss*: In the second stage, we train the Z-coordinate head while keeping the 2D model frozen. The loss function is a combination of MSE on Z and a small-weighted consistency term with the already learned 2D outputs. Let $\hat{Z} = \{\hat{z}_i\}_{i=1}^{30}$, $Z = \{z_i\}_{i=1}^{30}$, and reuse $\hat{Y}_{2D}, Y_{2D}$ as before. The loss becomes:

$$\mathcal{L}_{3D} = \underbrace{\frac{1}{30} \sum_{i=1}^{30} (z_i - \hat{z}_i)^2}_{\text{Z loss}} + \lambda \cdot \underbrace{\frac{1}{30} \sum_{i=1}^{30} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]}_{\text{Reprojection consistency}}$$

Here, λ is a small scalar (e.g., 1×10^{-6}) to preserve the previously learned 2D accuracy while focusing training on the Z dimension.

C. Data Preparation

Our model is trained on a curated subset of the TheraPose [10] dataset, which provides synchronized RGB frames and 3D joint annotations from an IMU-based motion capture system. The full dataset contains millions of frames covering over 120 physiotherapy exercises; we selected approximately 177,000 representative samples covering a wide range of body movements, including those critical to spinal alignment, balance, and posture. This subset was manually reviewed and corrected to ensure annotation quality, particularly for spine and limb joints.

We applied data augmentation using the Albumentations [11] library to improve model robustness. Changes in hue and saturation, random changes in brightness and contrast, and limited-angle rotations are all part of the transformations. These additions ensure spatial consistency between images and labels by considering keypoints. All images are resized to 224×224 , and the keypoints are also scaled to fit this size. Before being fed into the network, the RGB inputs are brought down to the range of $[0,1]$. This standardization makes sure that it will work the same way on all devices and input resolutions.

D. ONNX Export and Mobile Optimization

We used the `tf2onnx` library to convert our trained TensorFlow model into ONNX format for real-time inference on mobile devices. Post-training quantization improved the model even more by lowering memory and computing overhead without losing much accuracy. We used the XNNPACK delegate with ONNX Runtime for inference. This is the best option for ARM CPU architectures. This setup can process frames in less than 15 milliseconds on mid-range mobile devices, which shows that it is possible to run full 3D pose estimation pipelines in real time, even in edge devices where GPUs are not available.

IV. EXPERIMENTS

We trained the proposed model in two phases, using the Adam optimizer with a starting learning rate of 1×10^{-3} for 2D training and a lower rate of 1×10^{-4} for the Z-coordinate regressor. We set the batch size to 32 and the input resolution to 224×224 . Training was performed on an NVIDIA RTX A6000 ADA GPU with 48GB of VRAM, enabling the processing of large batches and high-resolution data efficiently. We used early stopping with a patience of 40 epochs during training to keep from overfitting and speed up convergence, in a total of 200 epochs of training. In addition, a learning rate scheduler was used to reduce the learning rate by half if the validation loss remained unchanged, which made the model more stable during fine-tuning. Both phases were monitored using validation loss, and the best-performing weights were saved for deployment. An example output of the model on a single RGB frame is visualized in Figure 3, which contains both the pose model for the RGB input frame drawn on top of it, and a slight side-angle representation of the pose within the 3D grid.

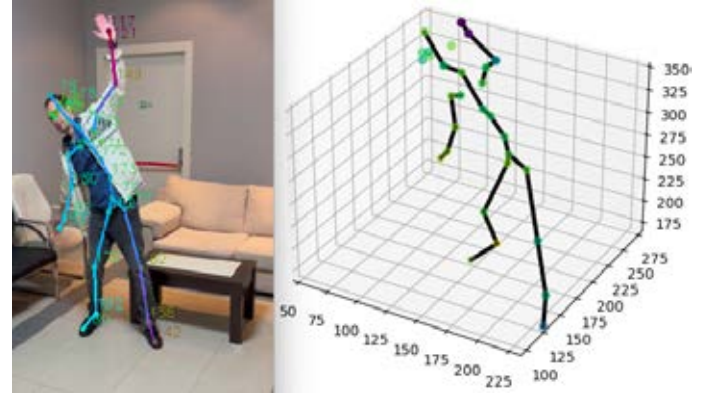


Fig. 3: Model output visualization overlaid on the RGB image and inside the 3D grid

We evaluated multiple backbone architectures to identify the optimal trade-off between accuracy and efficiency on our curated physiotherapy dataset. We used the MPJPE (Mean Per Joint Position Error) metric, which is defined as the average Euclidean distance between the predicted and ground truth 3D joint positions across all joints and frames. Among the tested variants, as shown in Table I, EfficientNet-B2 achieved the lowest 3D error (45.4 mm), outperforming both smaller (B0, B1) and larger (B3) EfficientNet versions. While EfficientNet-B3 has a higher parameter count, it surprisingly produced the worst accuracy, suggesting overfitting or reduced generalization in our scenario. MobileNet-v2 offered a good balance between compactness and accuracy, whereas MobileNet-v3Small yielded the smallest ONNX model size, but at the cost of higher error. As a result, EfficientNet-B2 was selected for the final model due to its favorable accuracy-to-size trade-off.

To enable a fair comparison with prior work, we retrained our model on the Human3.6M dataset using its standard 16-joint layout. The results are given in Table II. Our model

TABLE I: Training results using various backbones

Backbone	3D Error (MPJPE)	# Params (M)	ONNX Model size (MB)
EfficientNet-B0	50.3	5.8	6.2
EfficientNet-B1	48.7	8.3	8.8
EfficientNet-B2	45.4	9.7	9.8
EfficientNet-B3	83.4	12.5	13.0
MobileNet-v2	49.5	4.2	4.5
MobileNet-v3Small	61.2	3.1	3.3
MobileNet-v3Large	70.5	5.9	6.3

achieved a competitive 3D error of 39.8 mm MPJPE, on par with SOTA methods like PoseMamba [12], while maintaining real-time inference speed (<15 ms per frame) on mobile devices. Most lightweight models in the literature either omit runtime benchmarks on mobile platforms or do not report evaluation results on Human3.6M. For example, models like P-STMO [13] and Tiny-HourglassNet [14] do not share latency metrics, making real-time deployment viability unclear. In contrast, our model is one of the few that delivers both high accuracy and verified mobile efficiency, highlighting its practical value for edge-based physiotherapy applications.

TABLE II: Lightweight models comparison on Human3.6M dataset

Models	3D Error (MPJPE)	# Params (M)	Inference time (ms/frame) on mobile device
PoseMamba	38.1	6.7	-
P-STMO	42.8	6.2	-
MobileHumanPose	51.4	5.6	$\sim 30-40$
MoVNect	97.3	1.3	~ 22
Tiny-Hourglassnet	64.9	-	-
Ours	39.8	9.8	< 15

V. CONCLUSION

In this work, we introduced a lightweight and mobile-optimized 3D human pose estimation model tailored for physiotherapy applications. By adopting a two-stage training strategy and selecting an EfficientNet-B2 backbone, we achieved high accuracy with real-time inference performance on mobile devices. Our model offers full-body 30-joint coverage, including critical spinal landmarks, and demonstrates competitive results compared to SOTA lightweight methods. In future work, we aim to enhance robustness under occlusion by incorporating temporal priors and visibility-aware modules, and explore partial body models that can estimate joint confidence scores when certain limbs or body parts are not visible.

REFERENCES

- [1] ONNX — open neural network exchange, <https://onnx.ai/>, [Accessed 29-07-2025].
- [2] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, “Blazepose: On-device real-time body pose tracking,” *arXiv preprint arXiv:2006.10204*, 2020.
- [3] J. Zhang *et al.*, “Mobipose: Real-time multi-person pose estimation on mobile devices,” in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 136–149.
- [4] Q. Zhao, C. Zheng, M. Liu, and C. Chen, “A single 2d pose with context is worth hundreds for 3d human pose estimation,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 27 394–27 413, 2023.
- [5] 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*, 2020, pp. 479–488.
- [6] S. Choi, S. Choi, and C. Kim, “Mobilehumanpose: Toward real-time 3d human pose estimation in mobile devices,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2328–2338.
- [7] Y. Kim, H. Yoo, J.-H. Ryu, S. Lee, J. H. Lee, and J. Kim, “Transsmpl: Efficient human pose estimation with pruned and quantized transformer networks,” *Electronics*, vol. 13, no. 24, p. 4980, 2024.
- [8] C. Mercadal-Baudart *et al.*, “Exercise quantification from single camera view markerless 3d pose estimation,” *Heliyon*, vol. 10, no. 6, 2024.
- [9] H. Lee, B. Oh, S.-C. Kim, *et al.*, “Recognition of forward head posture through 3d human pose estimation with a graph convolutional network: Development and feasibility study,” *JMIR Formative Research*, vol. 8, no. 1, e55476, 2024.
- [10] H. Y. Yalıç, A. E. Usta, Ü. Atila, A. Uçan, A. A. Yılmaz, and A. Y. Yiğit, “Therapose: A large video dataset for physiotherapy exercises,” in *In Proceedings of 15th Turkish Congress of Medical Informatics*, 2024, pp. 88–97.
- [11] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, “Albumentations: Fast and flexible image augmentations,” *Information*, vol. 11, no. 2, p. 125, 2020.
- [12] Y. Huang, J. Liu, K. Xian, and R. C. Qiu, “Posemamba: Monocular 3d human pose estimation with bidirectional global-local spatio-temporal state space model,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, 2025, pp. 3842–3850.
- [13] W. Shan, Z. Liu, X. Zhang, S. Wang, S. Ma, and W. Gao, “P-stmo: Pre-trained spatial temporal many-to-one model for 3d human pose estimation,” in *European Conference on Computer Vision*, Springer, 2022, pp. 461–478.
- [14] B. Shi *et al.*, “Tiny-hourglassnet: An efficient design for 3d human pose estimation,” in *2020 IEEE international conference on image processing (ICIP)*, IEEE, 2020, pp. 1491–1495.