Monocular 3D Human Pose Estimation from Static Images

Lorenzo Calda 3194670 **Alberto Paolo Lolli** 3224481 **Mariano Masiello** 3200991 **Luca Ricci** 3218444 **Ali Emre Senel** 3221337

Abstract

Accurate 3D human pose estimation is fundamental for numerous applications, ranging from augmented reality and robotics to entertainment, environmental awareness, or human-computer interaction. However, most existing methods heavily rely on video sequences, rather than individual static images, limiting their applicability in scenarios where only single frames are available. We propose a novel methodology that allows to generate 3D human pose renderings from single 2D inputs. Our approach leverages a two-stage machine learning framework, that employs state-of-the-art pretrained models to estimate 2D poses and pixel depth, followed by a specialized network designed to transform the 2D information into a coherent 3D pose representation. We train both Transformer and Convolutional models on the Human3.6M dataset and evaluate them on position-dependent and position-independent pose estimation tasks. Our results indicate a remarkable capability for the models of recovering accurate 3D poses from monocular images, demonstrating the feasibility of robust 3D pose estimation from single images and highlighting the benefits of combining advanced deep learning architectures for this task.

1 Introduction

The ability to reliably deduce 3D human pose from visual data is a fundamental building block for a wide variety of advanced applications. From enhancing augmented and virtual reality (AR/VR) to enabling natural human-robot interaction and decoding complex behaviors for environmental understanding, accurate 3D pose estimation is key. It allows computers to see and understand human movement in three dimensions, escaping simple 2D abstractions and capturing a richer understanding of actions and interactions. Such capabilities are essential for building more natural interfaces, more trusted autonomous devices, and more powerful analysis tools in many fields: from entertainment to healthcare and security.

There has been outstanding progress in 3D human pose estimation over the past few years, dominated by advances in deep learning. Most of the state-of-the-art methods [1], however, rely heavily on video sequences and multiple camera directions as they leverage the temporal consistency and the multiple angles to improve performance. These methods employ motion clues and sequential data to enhance accuracy as well as to break ambiguities in 2D projection-based inference of 3D depth. While highly effective with video-cluttered environments, reliance on them is a huge bottleneck: their effectiveness decreases drastically in scenarios involving only single static images. Real-world settings often present single frames in isolation—i.e., single images, isolated frames from video, or snapshots from security cameras—lacking temporal structure, precluding video-based methods from being applicable.

To address this lacuna, this project proposes a novel methodology specific to robust 3D human pose estimation from isolated 2D inputs. Our method outgrows the confine of sequential information and seeks to extract full 3D pose information from a single image.
We do this in a two-stage machine learning pipeline. In the first stage, our system employs two different pretrained models:

1. YOLO11X-Pose to determine 2D joints locations [2].

2. Depth Pro to estimate pixel depth [3].

In the second stage, we train a deep learning model to integrate the 2D representations and depth estimation for consistent and anatomically valid 3D pose reconstruction. Our main focus for the rest of the report will be on the second stage as the first stage solely makes use of pretrained models and is thus relatively straightforward.

2 Data

We utilized the Human3.6M dataset [4, 5] to develop and evaluate our 3D human pose estimation method. Introduced by Ionescu et al. in 2014, Human3.6M was designed to address the pressing need for large-scale, accurate, and diverse data to train and evaluate realistic human sensing systems. Earlier datasets suffered from either small scale or low motion variability, precluding the development of strong models that can operate in natural settings.

Human3.6M took the state-of-the-art forward by providing around 3.6 million accurate 3D human poses. This corpus was acquired by recording the performance of 11 professional actors (5 female and 6 male) performing 15 different activities of daily living. These actions range from simple activities like walking, eating, and talking on the phone, to subtle actions like posing, greeting, and various sitting poses, as can be seen in Figure 1. The diversity of these movements and poses gives a rich coverage of common daily human activities, making the dataset highly suitable for learning generalizable human pose models.

As represented in Fig 2, the dataset includes:

1. High-frame-rate RGB video sequences from four different viewpoints captured simultaneously with high-frame-rate progressive scan cameras.
2. Accurate 3D human motion capture (MoCap) data, providing accurate ground-truth 3D joint positions.
3. Time-of-flight (TOF) depth data, with complementary depth measurements.
4. Accurate 3D body scans of all participating subjects, allowing for detailed body shape analysis and subject-specific model training.

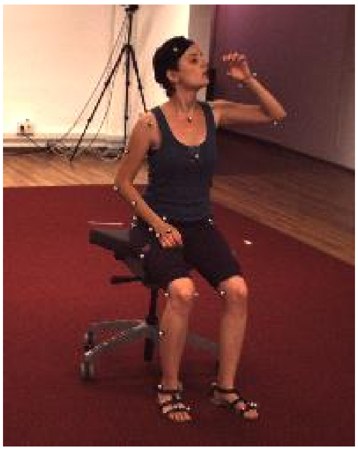


Figure 1: Example of a pose

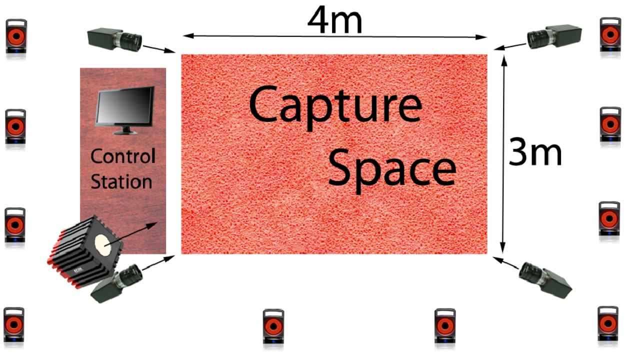


Figure 2: Cameras Arrangement

Furthermore, the dataset comprises rich annotations, including pixel-wise figure-ground segmentations and accurate bounding box annotations, received by projecting the 3D skeleton onto the image plane. It further contains controlled mixed-reality evaluation scenarios which insert realistic graphical characters into complex real environments with moving cameras and occlusions, essential to evaluate model robustness in challenging conditions. The subjects also acted in their daily clothes, which contributed to the realism of data recorded.

Owing to the goals of this project, we restrict ourselves to a monocular view, only considering data from one of the four available cameras. Moreover, due to computational constraints we were forced to work on a subset of the available dataset, working with approximately $1,000,000$ images comprising of scenes from all subjects we had access to. [1]

3 Preprocessing

The creation of the Human3.6M dataset was done meticulously, there are no corrupted images and no great imbalances between female and male subjects or between different poses. Additionally, there is no reason to doubt the quality of the ”ground-truth” 3D Pose since the Vicon T40 marker-based MoCap system it was recorded on has negligible spatial error, on the order of ~ 0.3 mm.

We now outline the preprocessing pipeline the data undergoes in preparation for training the 2D-to-3D model:

Each image is first reduced to a resolution of $500 \text{x} 500$ (from $1000 \text{x} 1000$), it is then passed through the first stage described in Section 1: YOLO11-X pose is used to find and estimate the 17-joint 2D pose of the subject in the image. In the edge case where no subjects are detected we discard the image, while if multiple subjects are detected we only keep the first one. The image is then passed through Depth Pro, producing a new image with each pixel representing the predicted distance from the camera. The original image, the 2D pose keypoints and the depth prediction will be used as the inputs to the model.

Following the convention of previous research onto Human3.6M pose estimation tasks, we split the data into training and testing dataset according to the subject. In particular, we use subjects 1, 5, 6 and 7 for training ($\sim 85\%$ of total data) and subjects 8, 9, 11 for testing ($\sim 15\%$ of total data). This allows us to better assess the true capabilities of the model as it tries to predict the pose of a new person never seen during training.

A further step that we take (which was empirically witnessed to greatly increase performance) is to randomly shuffle the dataset instead of feeding the images in the chronological order they appeared as. This greatly increases the robustness of the training and prevents the model from overfitting to a single subject during training.

The final size of the raw dataset utilized for training was $\sim$ 70 GB, after preprocessing it becomes $\sim$ 285 GB. The size of the dataset posed considerable problems as the Virtual Machine provided for training did not contain enough storage to permanently store the dataset. Solving this issue took considerable time, but eventually it was solved through the usage of a cloud storage.

In addition to this, another issue encountered was that, regardless of the size of the file, no more than 900 files could be uploaded to the cloud storage in a period of 5 minutes (meaning it would take ~ 11.5 days to fully load the dataset). This problem was solved by creating self-contained ”chunks” of data, each one containing 1000 instances of preprocessed images, which could then be quickly loaded onto the machine.

4 Architecture

In this project we experimented with two main architectures: a CNN-based architecture and a Transformer-based architecture. Given that ours is a computer vision task, we believe these two to be the most useful architectures. CNNs have been one of the most used architecture in CV as they are able to extract features independently of the subjects position in the image. Visual Transformers (ViT) meanwhile have been a rising star in the field due to their ability to long-range dependencies and capture global context more effectively than traditional deep learning methods.

We consider two separate problems, one simpler and one more complex. The first problem is to predict the 3D pose independent of the actual position of the subject in the image. The second one also takes into consideration the position.

[1] We would like to note that subjects 2, 3, 4 and 10 were not made available to the public by the original authors of the Human3.6M dataset.

4.1 Convolutional Network

Following ideas from previous research [6], the first step in our convolutional network involves the creation of a gaussian heatmap from the 2D joint predictions outputted by YOLO. This procedure generates 17 images of resolution $500\text{x}500$ (same as the other inputs), one for each joint, containing a 2D gaussian distribution with fixed st.dev $\sigma = 10$ estimating the position of the joint. It is advantageous to work with this representation because:

- It offers a measure of uncertainty in the YOLO predictions
- It allows us to work solely with images. Indeed our input is now a 500x500 image with 21 channels - 3 from the original image, 1 from the depth estimation and 17 joint gaussian heatmaps.

After this the convolutional layers begin, with reference to fig. 3 the input is first fed to ”Sequential[conv1]”, which contains 2 standard convolution block with Batch norm. It is then passed through ”Sequential[0]” and ”Sequential[1]” which feature respectively 3 and 4 Inverted Residual Blocks with SE (IRB-SE). In ”Sequential[2]” 3 Dual Path Blocks [7] and 2 more IRB-SE are placed in alternate order. The output then passes through ”WASPModule[wasp]”, as the name implies it implements a WASP (Waterfall Atrous Spatial Pyramid Pooling) module, which is composed of multiple atrous (dilated) convolutions with different dilation rates implemented through a sequential “waterfall” structure where outputs from one branch feed into the next. This specific block has been successfully used in other 3D pose prediction models [6, 8].
The results are then passed through the final convolutional block before being flattened and fed to ”PoseRegressionHead”, which contains a 3 layer MLP with dropout layers and outputs the predicted (x, y, z) coordinates for each joint.

4.2 Transformer Network

Transformers require ample amounts of data to be effectively trained, during our experiments we attempted to train a ViT from scratch, but achieved little success due to our dataset not being big enough as well as computational and time constraints. We thus decided to follow a transfer learning approach and incorporate a pretrained transformer which we modify to fit our usecase. The model used is ”vit-base-patch16-384” [9], which was originally introduced by Dosovitskiy et al. in 2020 [10].

Similarly to the CNN, we once again create a gaussian heatmap of the joints from the YOLO 2D frame input, this time of dimension 64x64 and $\sigma = 2$. Through Patch Embedding, we convert our inputs into vectors. The patch embeddings for the heatmap and the rest of the inputs are done separately as we first process the other inputs through the pretrained transformer.
The pretrained model was originally designed to take an image as input, but in addition to this we want to include our depth estimations as well. We incorporate them by extending the dimension of the embeddings of the pretrained transformer, so as to allow also for the encoding of depth in the embeddings. The rest of the transformer is kept the same, though the weights are not fixed to allow it adapt to the new changes through learning. The heatmap embeddings and the embeddings processed by the pretrained model are then passed through 2 consecutive Cross-Modal Fusion blocks before finally being concatenated. Cross-Modal Fusion Blocks are specifically designed to combine information from different data modalities, they work by setting up two symmetric Multi-Headed Cross-Attention blocks, one with input the first information source (ie. the heatmap embeddings) and with context the second information source (ie. the image and depth embeddings processed through the pretrained model) and the other one with the input and the context reversed.
Finally, the concatenated result is passed through 4 further standard Transformer Encoder blocks, each one containing a single multi-headed attention block ($h = 16$) and a 2 layer MLP (along with other regularization layers such as layer norm and dropout). The network ends with a 3 layer MLP connected to a classifier head which outputs the result.

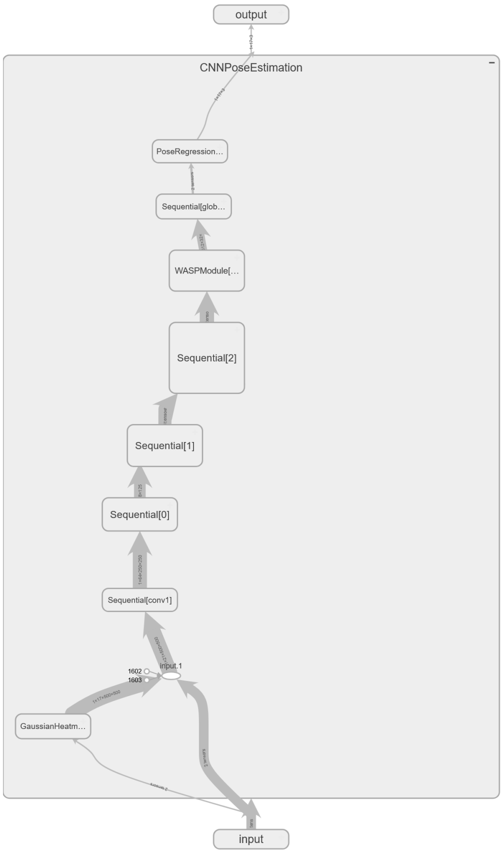
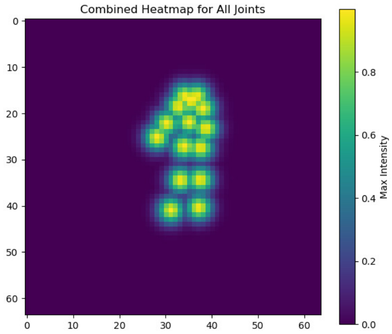


Figure 3: CNN Architecture Workflow



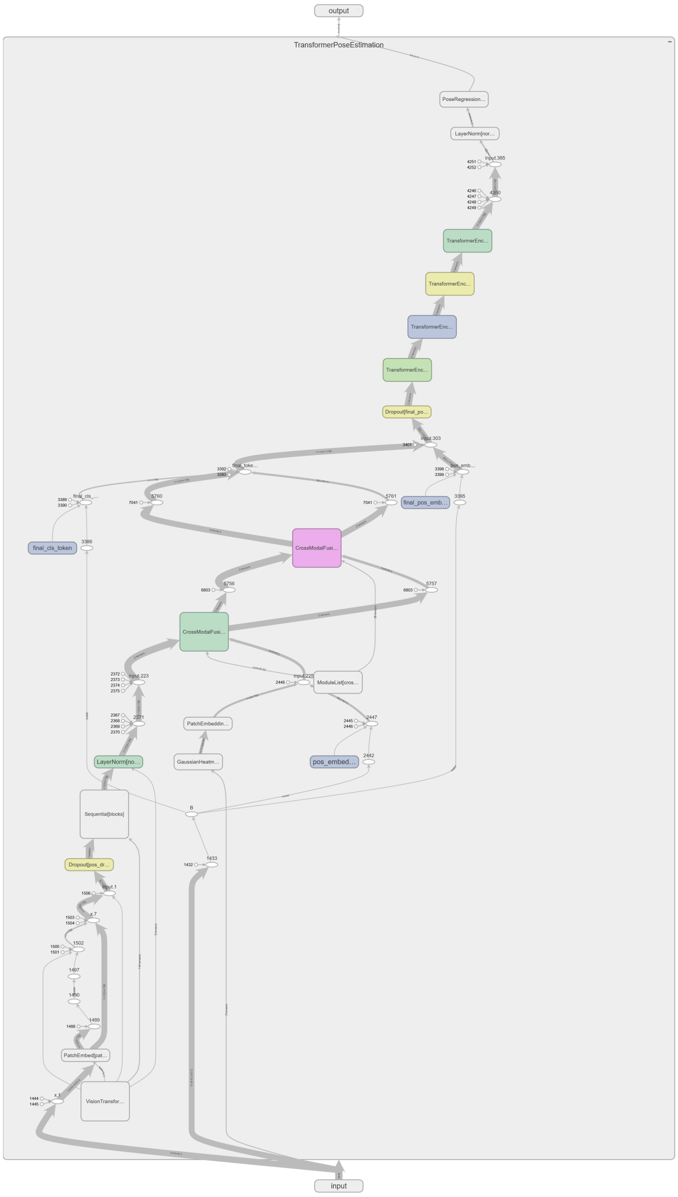


Figure 4: Combined Gaussian Heatmap of Joints

Figure 5: Transformer Architecture Workflow

5 Experimental Set-up and Results

5.1 Implementation Details

As mentioned above we trained models for two different tasks: the first one only predicting the pose of the subject without regard for its position within the image, and the second one also taking the position into consideration. Given the performance of the ViT we decided to only train a CNN for the second more difficult task. Moreover the architecture for the CNN in both tasks was kept the same, what was changed was the loss to be optimized.

Due to the great time length required to fully train a single model, performing rigorous and complete hyperparameter tuning through cross-validation was not feasible. We based our final hyperparameter choices both by relying on previous research and making quick exploratory runs lasting a short number of iterations ($\sim$ 2000) and extrapolating based on their performance[2]. Each model is trained using the AdamW optimizer and the training time for a model with optimized hyperparameters ranges from 1 to 3 days.

[2] We prefer reasoning in terms of iterations instead of epochs once again due to slow training time. As a general rule 1 epoch $\sim$ 8700 iterations.

5.1.1 Loss function

In the position independent (IND-P) estimation problem we automatically translate the pose so as to normalize the root joint (found on the pelvis) to be at $(0, 0, 0)$ in both prediction and the label. Given a parameter θ and a single labeled data-point $(\overline{x}^{\mu}, y^{\mu})$, the IND-P loss function has 2 components:

$$\mathscr{L}_{\mu}^{\text{IND-P}}(\theta) = \lambda_1 \mathscr{L}_{\mu}^{\text{MSE}}(\theta) + \lambda_2 \mathscr{L}_{\mu}^{\text{Inter-joint}}(\theta), \qquad \text{with } \lambda_1 = 1, \lambda_2 = 1$$

Where:

- $\mathscr{L}_{\mu}^{\text{MSE}}$ is the MSE loss and it computes the Mean Squared Error between each predicted and actual joint.
- $\mathscr{L}_{\mu}^{\text{Inter-joint}}$ is the Inter-Joint loss. It computes the distance between each unique pair of joints in the predicted pose, compares it to that of the actual pose and takes the average.
  Thus $\mathscr{L}_{\mu}^{\text{Inter-joint}} = \langle \, | \, \| \, \hat{y}_i^{\mu} - \hat{y}_j^{\mu} \, \|_1 - \| \, y_i^{\mu} - y_j^{\mu} \, \|_1 \, | \, \rangle_{i \neq j \, \in \, J}$ with J being the set of joints. We introduce this novel loss term because by considering not only how close a joint is to its actual value but also how the predicted joints interact with each other to form a coherent and realistic pose we can the models predict more realistic poses and limb sizes.

The position dependent (DEP-P) estimation problem the root joint is not normalized at the root, this allows us to take into consideration also the position of the subject and not only the pose. The DEP-P loss function features two additional components to the loss:

$$\begin{aligned} &\mathscr{L}_{\mu}^{\text{DEP-P}}(\theta) = \lambda_1 \mathscr{L}_{\mu}^{\text{MSE}}(\theta) + \lambda_2 \mathscr{L}_{\mu}^{\text{Inter-joint}}(\theta) + \lambda_3 \mathscr{L}_{\mu}^{\text{ABS-ROOT}}(\theta) + \lambda_4 \mathscr{L}_{\mu}^{\text{L1}}(\theta) \\ &\text{with } \lambda_1 = 1, \, \lambda_2 = 100, \, \lambda_3 = 1, \, \lambda_4 = 1 \end{aligned}$$

Where the first two components are the same as the ones in INDP-P while:

- $\mathscr{L}_{\mu}^{\text{ABS-ROOT}}$ is the absolute root distance loss and it measures the absolute distance between the predicted and actual root joint.
- $\mathscr{L}_{\mu}^{\text{L1}}$ is the L1 loss and it computes the mean absolute error between the predicted and actual joints.

5.2 Results

A summary of our results is presented in table 1, we report the values through the conventional evaluation metrics utilized in the literature.
Mean Per Joint Positional Error (MPJPE) measures the average distance between the predicted joints and the ground truth joints, while Procrustes Aligned MPJPE (PA-MPJPE) applies a rigid alignment (eg. rotation, translation, and scaling) to the predicted poses before error computation. Following the notation established previously, we denote IND-P the position independent estimation problem, while we denote with DEP-P the position dependent problem.
We can see that the ViT model performs significantly worse than the Convolutional model, we attribute this difference in performance to two main reasons: lack of data for the ViT to learn properly, the transformer was trained on significantly less iterations compared to the CNN due to its per-iteration time requirements being much higher. The transformer-based model seems to properly learn to predict a human pose but the output pose doesn't seem to vary based of the input image.
For a visual comparison between the models we include figures 6, 7, 8 that contain randomly sampled predictions from the training dataset along with their actual label and the original image.

Metric	**IND-P**		**DEP-P**
	CNN	ViT	CNN
MPJPE	84.83	209.50	120.70
PA-MPJPE	106.10	211.20	95.53

Table 1: MPJPE and PA-MPJPE for independent-pose (IND-P) and dependent-pose (DEP-P) models.

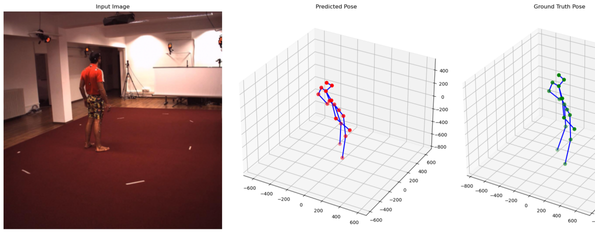


Figure 6: IND-P CNN Testing Sample

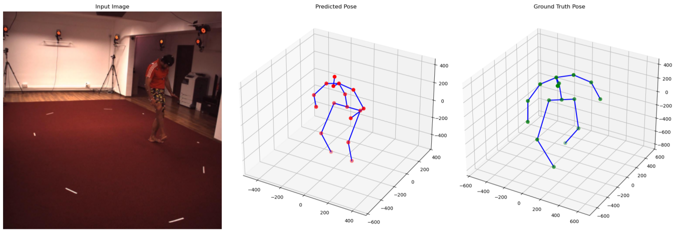


Figure 7: IND-P Transformer Testing Sample

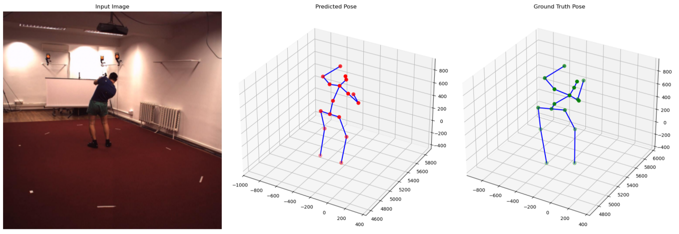


Figure 8: DEP-P Testing Sample

6 Conclusion

In this work we present several models and approaches to estimating human 3D pose starting from a single static image. By using a convolutional architecture we are able to obtain good results both in the position independent and the position dependent problems.

The main limitations to this study come from the limited time and from the computational constraints: for these reasons, it was not possible to fully preprocess the whole Human3.6M dataset, nor was it viable to experiment with deeper and more complex networks trained for more epochs, or with different architectures like diffusion models, which have been shown to perform 3D pose estimation in a remarkable way [11]. Moreover, it is important to notice that the model could reflect bias induced by the use of a single dataset: indeed, it may learn some features that are specific to the Human3.6M dataset setting (e.g., the room where the actors were recorded), rather than to the poses themselves.

Future development on this project could explore several key areas. First, the model could be expanded to work with more diverse datasets that feature new human poses and varied backgrounds. Although we had access to EMDB, another dataset of human poses and shapes in the wild [12], we chose not to use it due to time constraints and limited computational resources. Furthermore, a simplification of the pipeline, by removing heavy dependencies on YOLO and Depth Pro, could yield a lighter model with comparable performance that could possibly be deployed on small devices, such as a smartphone. Additionally, the model could be improved to handle multiple human pose predictions simultaneously. Lastly, it could be extended to generate complete scenes by incorporating predictions of the surrounding environment such as rooms, furniture, animals, and other contextual objects.

References

[1] Jiajie Liu, Mengyuan Liu, Hong Liu, and Wenhao Li. Tcpformer: Learning temporal correlation with implicit pose proxy for 3d human pose estimation, 2025.

[2] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. Ultralytics YOLO, January 2023.

[3] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R. Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *International Conference on Learning Representations*, 2025.

[4] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.

[5] Cristian Sminchisescu Catalin Ionescu, Fuxin Li. Latent structured models for human pose estimation. In *International Conference on Computer Vision*, 2011.

[6] Bruno Artacho and Andreas Savakis. Unipose: Unified human pose estimation in single images and videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[7] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[8] Bruno Artacho and Andreas Savakis. Unipose+: A unified framework for 2d and 3d human pose estimation in images and videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9641–9653, 2022.

[9] Ross Wightman. Pytorch image models.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.

[11] Jia Gong, Lin Geng Foo, Zhipeng Fan, Qiuhong Ke, Hossein Rahmani, and Jun Liu. Diffpose: Toward more reliable 3d pose estimation, 2023.

[12] Manuel Kaufmann, Jie Song, Chen Guo, Kaiyue Shen, Tianjian Jiang, Chengcheng Tang, Juan Zarate, and Otmar Hilliges. Emdb: The electromagnetic database of global 3d human pose and shape in the wild, 2023.