3D Pose Estimation using Transformers

Ayush Chamoli[1] and Ahmed Tawfik Aboukhadra[2]

[1] chamoli@rhrk.uni-kl.de
[2] ahmed_tawfik.aboukhadra@dfki.de

Abstract. Human Pose Estimation is one of the important applications of Computer Vision. It involves determining the spatial configuration of different joints of human body. This is important for applications ranging from healthcare, improving human computer interaction, performance analysis in sports, among other fields. Up until recently, Convolutional Architectures have played an important role in Pose Estimation. However with the recent development in the Transformers Architecture, especially in the field of Neural Language Processing tasks, an emerging trend of the use of Transformer Architecture is seen in Pose Estimation. Here we go in depth into some of the architectures for Body and Hand Pose Estimation.

Keywords: Transformer, Pose Estimation

1 Introduction

3D Human Pose estimation aims to determine the joints of a human body. There has been a significant interest in 3D Human Pose estimation in the recent years in the computer vision community because of the wide range of application, which includes action recognition[7][8], virtual reality, sport motion analysis, neurodegenerative condition diagnoses and many more. However it is a complex problem because of issues like articulated motion, occlusions and depth ambiguity while determining the position of the 3D joints.

Convolutional network architectures have been the standard method to determine human poses [1][15]. However, they suffer from a number of issues which arise from the fact that CNNs rely on dilation techniques as they have limited temporal connectivity. To improve upon that, temporal convolutional neural architecture [3][12] and recurrent architecture [5] are also used in order to capture and understand global dependencies across multiple frames. However, the simple sequential correlation is a sub-problem for such recurrent networks.

Another technique to determine 3D human poses is 2D-to-3D human pose estimation. It uses the coordinates of the 2D joint as an input and determines the 3D pose based on it. However, convolutional network architectures have a hard time to work with such sort of data. Graph convolutional networks, which are great with structural information, are better at learning such representations of human pose. These GCN architecture [2][17] perform well but is often limited because of the small receptive fields.

Detecting the hand pose from an image is also an important application of Computer Vision. The idea behind it is pretty similar to that of human pose estimation, with the only difference being of detecting hand joints instead of body joints. Similar techniques can be employed for this application as well. However, we suffer from the issue of occlusion which is not easily solved with the techniques listed above.

An increasing trend has been observed in the use of Transformers, and it turns out it can not only be used in language processing tasks but also, recent works suggest they perform really well in the application of human and hand pose estimation. Transformers with their self-attention mechanism, which model the dependencies in inputs and outputs, provide an important framework for learning and estimating body poses from the images and videos.

In this seminar report, we will discuss some of the recent techniques for estimating body and hand pose and compare their results.

2 Background

2.1 Transformers

The transformer architecture which was introduced in the paper "Attention is All You Need"[16] has revolutionized natural language processing and computer vision tasks. At it's core, the transformer architecture uses self attention mechanism which lets it learn and capture long-range dependencies. Transformer can be parallelized with the use of multi-head self attention which helps in training and can therefore be scaled up. Recent works[19][18] shows that self-attention achieves state of the art results in computer vision tasks that which in the past involved the use of convolutional neural networks.

2.2 3D Pose Estimation

There are multiple ways of approaching the problem of 3D Pose Estimation. The two common ones are the one-stage approach and the two-stage approach. In one-stage approach, the input images are directly used in order to calculate the 3D pose. It does not require the use of an intermediate 2D pose. The two-stage approach on the other hand first evaluates the 2D pose from the input and then it uses the 2D pose as an input to calculate the 3D pose. There are approaches that exploit the information in the spatial domain, the temporal domain or uses convolutional-based, graph-based architecture and many more.

3 Methodologies

3.1 PoseFormer

PoseFormer[19] is a spatio-temporal transformer, which performs 3D human pose estimation by lifting from 2D to 3D. It comprises 3 modules: spatial transformer, temporal transformer and regression head.

The spatial transformer extracts high dimensional feature embedding from each frame. Considering a 2D pose with J joints, the coordinates of each joint $j \in J$ is projected to a higher dimension with a trainable linear projection. Learnable spatial positional embeddings $E_{SPos} \in \mathbb{R}^{J \times c}$ are added to linear projection. Hence the input for i-th frame becomes $x_0^i \in \mathbb{R}^{J \times c}$ where c is the spatial embedding dimension. These joint sequences of features are passed to the spatial transformer which applies self-attention over all J joints. The output for i-th frame with L layers is $x_{L-spatial}^i \in \mathbb{R}^{J \times c}$

The temporal transformer extract the dependencies across the frames. For i-th frame, the output of spatial transformer $x_L^i \in \mathbb{R}^{J \times c}$ is flattened to a vector $\boldsymbol{x}^i \in \mathbb{R}^{1 \times (J.c)}$ and then they are concatenated for f input frames as $X_0 \in \mathbb{R}^{f \times (J.c)}$. The learnable temporal positional embeddings $E_{TPos} \in \mathbb{R}^{f \times (J.c)}$ is also added at this point. Then this is passed through the temporal transformer which follows the same architecture as temporal transformer consisting of multi-head self attention and multi-layer perceptron blocks. The output of this module is $X_{temporal} \in \mathbb{R}^{f \times (J.c)}$.

In order to estimate the 3D Pose of the frame, the output of the temporal transformer is passed through a regression head that is a simple MLP with Layer norm and a linear layer and a weighted mean operation with learnable parameters is used. The output of this layer $\boldsymbol{y} \in \mathbb{R}^{1 \times (J.3)}$ gives the 3D pose estimate of the center frame.

This model is trained by MPJPE (Mean Per Joint Position Error) where the goal is to minimize the error between predicted and ground truth pose coordinates.

3.2 Graformer

Graformer[18] is a transformer architecture combined with graph convolutions to perform 3D pose estimation. It consists of two modules which are stacked one after another repeatedly. These modules are the GraAttention and ChebGConv block.

The input of the Graformer is the 2D joint coordinate $x_0 \in \mathbb{R}^{j\times 2}$. The input is firstly preprocessed by ChebGConv layer. It is used to handle graph-structed data. The Chebyshev graph convolution is evaluated for layer l as:

$$X_{l+1} = \sigma_{k=0}^{K-1} T_k(\tilde{L}) X_l \theta_k$$

where $T_k(x) = 2xT_{k-1}(x) - 2_{k-1}(x)$ denotes the Chebyshev polynomial of degree k, $T_0 = 1$, $T_1 = x$ and $\tilde{L} \in \mathbb{R}^{j\times j}$ denotes the re-scaled Laplacian. $\theta_k \in \mathbb{R}^{D_l \times D_{l+1}}$ denotes the trainable parameter of the graph convolutional layer. This block is able to capture information among the K top neighbors of the joint and therefore increases the receptive field.

After preprocessing, we stack the GraAttention and ChebGConv block for N times. This inherits the multi-head self attention mechanism from the Transformer architecture but removes the MLP layer. This is followed by the use of a dropout layer in order to regualize the self attention output. The global interaction is exploited here as each element of the block has 2D joint information. Normalization is done on the output followed by GCN layers and ReLU activation.

GraAttention and ChebGConv layers are used in repeated succession as GraAttention block gives a global receptive field while the ChebGConv gives the local receptive field. It is represented well in the following figure.

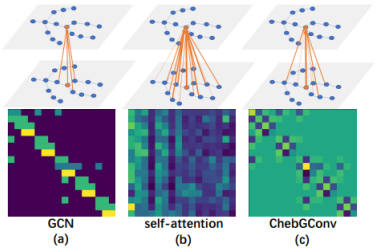

Fig. 1: Receptive field of blocks of GraFormer

The model is train by using MPJPE to minimize the error between the predicted and ground truth coordinates of the 3D Pose.

3.3 METRO

Human pose mesh estimation is superior to joint estimation which is discussed in the methods above as it offers a more detailed 3D representation of human pose which is crucial for applications which require precise body shape. MEsh TRansfOrmer[10] uses transformer encoder to model vertex-vertex and vertex-join interaction in a 3D human pose. It consists of 2 modules: Convolutional Neural Network and a Multi-Layer Transformer Encoder.

The task of the convolutional neural network block is to extract features. In order to do this, the CNN is pre-trained on ImageNet classification task [14]. The feature vector $X \in \mathbb{R}^{2048\times 1}$ from the last hidden layer is taken as the input for the transformer layer. Here X has a dimension of 2048.

Now the transformer encoder with several modifications is used. The output of the convolutional neural network block is passed through several blocks to perform dimensionality reduction gradually.

This block is refered as the Multi-Layer Transformer Encoder. The image feature vector X with template 3D coordinates of the pose is concatenated to form the set of joint queries $Q_J = q_1^J, q_2^K, ...q_n^J$. Similarly for mesh vertices v, a set of vertex queries Q_V are formed.

In order to obtain the bi-directional attention in the transformer model, Masked Vertex Modeling (MVM) is used for the regression task. MVM forces transformer to regress 3D coordinates by taking other vertices and joints in consideration which helps in learning the local and global interaction of each joint.

This model applies MPJPE loss on the output of the transformer module and tries to minimize it. 2D reprojection is also used in order to refine the image-mesh alignment.

3.4 HandOccNet

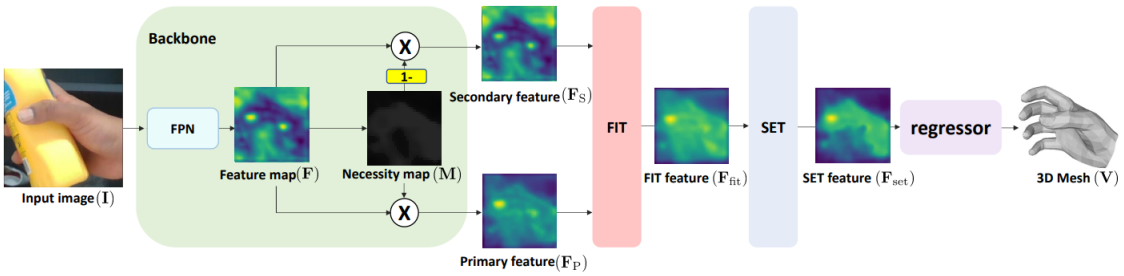

Fig. 2: Architecture of HandOccNet

HandOccNet[13] focuses on the domain of hand pose detection. Despite there being a lot of models for this application, HandOccNet targets the problem where the Hands are occluded by objects which is a challenging problem. It tackles this problem by its use of several modules: backbone, feature injecting transformer (FIT), self-enhancing transformer (SET) and regressor.

A hand image $I \in \mathbb{R}^{512\times512\times3}$ is passed through ResNet50-based FPN [11] and a feature map F is obtained. A necessity map M is also obtained from the feature map F. These maps are used to extract the primary features X_P and secondary features X_S using the element-wise multiplication operation.

$$X_P = F \bigotimes M$$

$$X_S = F \bigotimes (1 - M)$$

Primary feature X_P describe the information of the region of the hand and the secondary feature X_S describe the information of the occluded region. The query is extracted from X_S and the key is extracted from X_P.

Feature Injecting transformer injects the information of the primary features X_P into secondary features X_S. It includes a softmax-based and a sigmoid-based attention module. The task of the softmax based attention module is to find the relevant information of X_P from X_S. It generates a correlation map $C_{soft} \in \mathbb{R}^{1024\times1024}$:

$$C_{soft} = softmax(\frac{q_{soft}k_{soft}^T}{\sqrt{d_{k_{soft}}}})$$

where key k_{soft} is extracted from X_P, query q_{soft} from X_S with a two 1×1 convolutional layer. $d_{k_{soft}} = 256$ denotes the feature dimension of k_{soft}.

Sigmoid based attention module is used for filtering the undesired high correlation. The key query pair are extracted in the same way as above and we form the correlation map $C_{sig} \in \mathbb{R}^{1024\times1}$:

$$C_{sig} = sigmoid(pool(\frac{q_{sig}k_{sig}^T}{\sqrt{d_{k_{sig}}}}))$$

The pooling here makes the correlation map robust to noisy correlations. The final correlation map $C \in \mathbb{R}^{1024\times1024}$:

$$C = C_{soft} \bigotimes C_{sig}$$

This correlation map C is used for injecting the primary features in the occluded region. The output X_{FIT} is passed through the self enhancing transformer whose task is to refine the output of the FIT module. It utilizes the self attention of X_{FIT} by using a three 1×1 convolutional layers to extract key k', query q' and value v'. It uses just the softmax based attention module to generate the correlation map.

A regressor is finally used to produce the 3D Hand pose and the hand mesh. It is trained by minimizing the loss function which is defined as the distance between predicted and ground truth 3D coordinates.

3.5 MHFormer

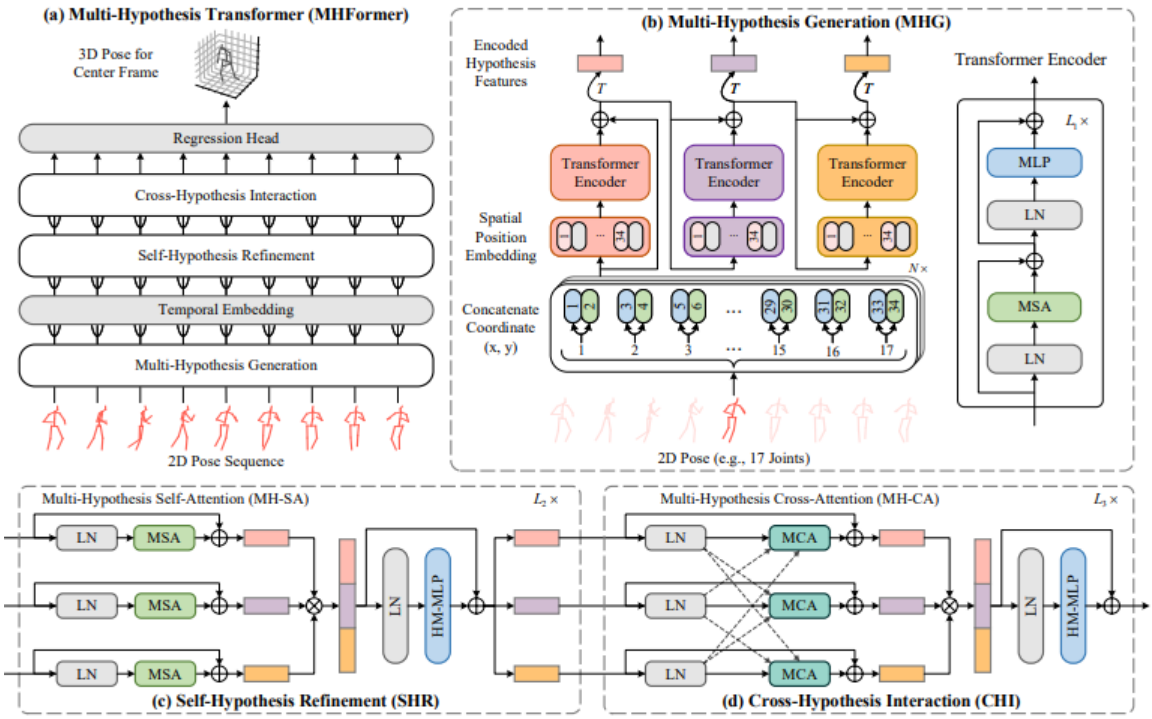

Fig. 3: Architecture of MHFormer

Multi-Hupothesis Transformer[9] exploits the idea that a given video for a human pose can have depth ambiguity and self-occlusion and therefore can have multiple different solutions (hypothesis). It trains over the spatio-temporal representation of these hypothesis. The task here is done by the three major modules: Multi-hypothesis generation (MHG), self-hypothesis refinement (SHR) and cross-hypothesis interaction (CHI) along with two minor modules: temporal embedding and regression head.

Starting with multi-hypothesis generation, assuming M different hypothesis, the module takes the sequence of 2D poses $X \in \mathbb{R}^{N \times J \times 2}$ where N is the number of frames in the video and J is the number of joints. It outputs multiple hypothesis $X^m_{MHG} \in \mathbb{R}^{(J.2) \times N}$. The spatial information of each joint is also retained with the use of spatial position embedding and these are used in the encoders for the MHG. The output here usually contains diverse information assuming different depths and occlusion and therefore are enhanced further.

In order to build a strong relationship across different hypothesis features, the temporal dependencies are exploited. Each hypothesis X^m_{MHG} generated by MHG are embedded to a high-dimensional feature using transposition operation and linear embedding. Then the learnable temporal positional embeddings are added to it.

The next module, self hypothesis refinement (SHR) works to refine each individual hypothesis in the temporal domain. In order to do that, it uses two blocks: multi-hypothesis self attention (MH-SA) and hypothesis mixing MLP (HM-MLP). THe MH-SA tries to learn from each hypothesis independently in order to model the long-range dependency. In order to exchange the information across hypothesis, hypothesis mixing MLP is used. Here the concatenated features of the multiple hypothesis are fed and it produces even non-overlapping chunks.

Finally the CHI module takes a look at the interaction among multi-hypothesis features. In order to do this, it also uses several blocks: multi-hypothesis cross attention (MH-CA) and hypothesis mixing MLP. Like before, multi-hypothesis self-attention does not model the connection across multiple hypothesis. In order to fix this issue, the MH-CA computes the correlation among cross hypothesis features. It significantly boosts the power of model. Hypothesis mixing MLP serves a similar process as the one in SHR where the output of MH-CA are passed through it and it generate a single representation $X_{CHI} \in \mathbb{R}^{N \times (C.M)}$.

Finally a regression head is used to produce the 3D pose sequence for the given center frame. The model is trained on MPJPE loss.

4 Comparison

In order to evaluate the reviewed model, a number of datasets are used. These include Human3.6M [6] for body pose estimation and HO-3D[4] for hand pose estimation.

The Human3.6M dataset includes 3.6 million images captured from 4 cameras. It includes 15 daily activities by 11 humans. There are generally 2 protocols for comparing the models. Protocol 1 takes into consideration the MPJPE metric for the analysis. As mentioned before, this measure is the mean Euclidean distance calculated between the predicted and the ground truth 3D pose in millimeters.

$$L = \frac{1}{J} \sum_{k=1}^{J} ||y_k - \hat{y}_k||_2$$

where y_k is the ground truth pose and $\hat{y}_k$ is the predicted 3D pose. Protocol 2 considers P-MPJPE as an evaluation metric for the models. P-MPJPE is MPJPE which is calculated after the rigid alignment of the 3D pose using pose processing. For our analysis, we take a look at Protocol 1 (MPJPE) in 2 cases: using images as input and using ground truth as input.

Method	Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
PoseFormer	41.5	44.8	**39.8**	42.5	46.5	51.6	42.1	42.0	**53.3**	60.7	45.5	43.3	46.1	31.8	32.2	44.3
GraFormer	49.2	53.9	54.1	55.0	63.0	69.8	51.1	53.3	69.4	90.0	58.0	55.2	60.3	47.4	50.6	58.7
METRO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	54.0
MHFormer	**39.2**	**43.1**	40.1	**40.9**	**44.9**	**51.2**	**40.6**	**41.3**	54.5	**60.3**	**43.7**	**41.1**	**43.8**	**29.8**	**30.6**	**43.0**

Table 1: Protocol 1: MPJPE metrics for Poseformer, GraFormer, METRO and MHFormer on Human3.6M Dataset

Table 1 shows that MHFormer performs the best in most of the categories and on average with MPJPE (in mm) of 43.0. However it can also be obeserved that there are several categories where PoseFormer works better than MHFormer, which are Eat and Sit. GraFormer performs the worst on average. Considering temporal information helps MHFormer and Poseformer perform significantly better than the other two models which only consider the spatial information. However no comment can be made in comparison of GraFormer and METRO on each category as their average MPJPE is close to each other with METRO being 9% better.

Method	Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
PoseFormer	32.5	34.8	32.6	34.6	35.2	39.3	**32.1**	32.0	42.8	48.5	34.8	32.4	35.3	24.5	26.0	34.6
GraFormer	32.0	38.0	30.4	34.4	**34.7**	43.3	35.2	31.4	38.0	46.2	34.2	35.7	36.1	37.4	30.6	35.2
MHFormer	**27.7**	**32.1**	**29.1**	**28.9**	39.9	**33.9**	33.0	**31.2**	**37.0**	**39.3**	**30.0**	**31.0**	**29.4**	**22.2**	**23.0**	**30.5**

Table 2: Protocol 2: P-MPJPE metrics for Poseformer, GraFormer and MHFormer on Human3.6M Dataset

Table 2 paints a similar picture as in table 1 where MHFormer performs the best among all the models on average. However, it can also be seen that PoseFormer performs better in categories of Phone and Pose than MHFormer. For the category Phone, even GraFormer performs better than MHFormer. Despite that, on average, MHFormer performs 12% better than PoseFormer and 15% better than GraFormer.

Method	Joint	Mesh	F@5	F@15
METRO	10.4	11.1	48.4	94.6
HandOccNet	**9.1**	**8.1**	56.4	96.3

Table 3: Comparison metrics for METRO and HandOccNet using PA-MPJPE using HO-3D Dataset

In order to evaluate the performance of Hand Pose Estimation model, we use HO-3D dataset. The common way of comparing different models is with the use of mean joint error and mean mesh error (in mm) and also the F-scores

As seen in the table 3, HandOccNet performs better than METRO at detecting hand pose. An improvement of 12% and 27% is observed in the joint and mesh error for HandOccNet than METRO which proves that HandOccNet is robust to severe occlusions.

5 Conclusion

In this report, we've summarized the recent development on transformer-based 3D human and hand pose estimation models. The architecture, results and metrics for each model is analyzed and

compared. In this analysis, it can be said that MHFormer and HandOccNet perform the best in class in their domain of pose analysis.

References

1. Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014.
2. Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.
3. Tianlang Chen, Chen Fang, Xiaohui Shen, Yiheng Zhu, Zhili Chen, and Jiebo Luo. Anatomy-aware 3d human pose estimation with bone-based pose decomposition, 2021.
4. Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses, 2020.
5. Mir Rayat Imtiaz Hossain and James J. Little. Exploiting temporal information for 3d human pose estimation. In *Computer Vision – ECCV 2018*, pages 69–86. Springer International Publishing, 2018.
6. 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, 2013.
7. Songyao Jiang, Bin Sun, Lichen Wang, Yue Bai, Kunpeng Li, and Yun Fu. Skeleton aware multi-modal sign language recognition, 2021.
8. Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition, 2019.
9. Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation, 2022.
10. Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers, 2021.
11. Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.
12. Ruixu Liu, Ju Shen, He Wang, Chen Chen, Sen-ching Cheung, and Vijayan Asari. Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5064–5073, 2020.
13. JoonKyu Park, Yeonguk Oh, Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Handoccnet: Occlusion-robust 3d hand mesh estimation network, 2022.
14. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015.
15. Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.
16. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017.
17. Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3425–3435, 2019.
18. Weixi Zhao, Yunjie Tian, Qixiang Ye, Jianbin Jiao, and Weiqiang Wang. Graformer: Graph convolution transformer for 3d pose estimation, 2021.
19. Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers, 2021.