A 3D Human Pose Estimation Pipeline

Zimeng Jiang
zjiang@student.ethz.ch

Le Chen
lechen@student.ethz.ch

ABSTRACT

In this project, we study the task of 3D human pose estimation from a single RGB image with deep networks. We separate the task into two stages. We first estimate the 2D pose from the image by learning reliable high-resolution representations. Subsequently, fully-connected layers are applied to learn the 3D pose. Additionally, the training data is augmented by randomly placing occluders from the Pascal VOC dataset. Our method receives a PA-MPJPE of 58.58, ranked top 1 among all the participant groups.

1 INTRODUCTION

For a large number of applications, including virtual and augmented reality, robotics, human-computer interaction, and autonomous driving, detecting 3D human poses is of great importance. For example, in the case of autonomous driving, obtaining an accurate estimation of the pedestrian's pose is essential to driving safety. What's more, in the field of robotics, reliable human pose estimation is important for human-robot interaction. However, most of the existing depictions of humans are two dimensions, such as images. For that reason, estimating 3D human pose from monocular images attracts many researchers. Formally, the aim of this task is to estimate the 3-dimension coordinates of the 17 body joints of people given a 2D RGB monocular image.

In order to estimate a 3D pose from an image, an algorithm has to be invariant to many factors, including lighting, image imperfections, background scenes, and so on. Early methods use robust features such as SIFT descriptors to achieve invariance. In recent years, the research community has received significant progress on this problem, using deep Convolutional Neural Networks (CNNs). The methods of this task fall into two categories. One is to train a CNN architecture end-to-end to estimate the 3D human pose from a given monocular RGB image directly. The other one is the two-stage approach. Those methods proceed in two sequential steps. The first step is to estimate 2D joint locations given the image data, and the second step recovers the 3D pose from these 2D joint location estimations.

In this paper, we present a 3D human pose estimation pipeline following the two-staged protocol with occlusion augmentation technique. We train and evaluate our method on images extracted from a preprocessed version Human3.6M dataset, which is collected indoor by marker-based motion capture systems. Each image in the dataset is cropped around the human body and scaled to 256x256.

2 RELATED WORKS

2.1 Dataset

For 2D pose estimation, COCO [11] and MPII [2] datasets are mainly used for benchmarks. The COCO Keypoints dataset [11] contains more than 200,000 images and 250,000 person instances labeled with keypoints. MPII Human Pose dataset [2] is a state of the art benchmark for evaluation of articulated human pose estimation, which includes around 25,000 images containing over 40,000 people with annotated body joints, covering 410 human activities. The images in MPII dataset are systematically collected using an established taxonomy of every day human activities. For 3D pose estimation, most works report evaluation results on the Human3.6 dataset [5, 8], which is the main dataset for single-person 3D pose performance comparison. It consists of multi-view videos of a single person in a room, whose pose is captured with the OptiTrack motion capture system. The dataset contains 17 different scenarios including talking, smoking, and so on, as well as the ground-truth 2D and 3D pose annotations.

2.2 From image to 2D

For 2D human pose estimation, there are many models with good performance. Those approaches can be divided into two major categories: bottum-up [17] and top-down. The bottom-up methods first detect the parts or joints for humans in the image and then group them to get a person's pose. OpenPose [3, 4] is one of the most popular bottom-up methods for multi-human pose estimation. The authors use a nonparametric representation to learn the association between body joints and people in the image and apply a greedy bottom-up parsing step to maintain accuracy in real-time performance. Fang et al. [7] propose a popular top-down method, which uses the Symmetric Spatial Transformer Network to extract a high-quality single person region from an inaccurate bounding box. Stacked Hourglass Networks [14] is a landmark paper that introduces a novel architecture which beats all previous methods. The proposed network consists of steps of pooling and upsampling layers which look like an hourglass and perform repeated bottom-up, top-down processing with intermediate supervision. The HRNet (High-Resolution Network) model[20] maintains a high-resolution representation throughout the whole process and has outperformed all existing methods on pose estimation tasks in the COCO dataset.

2.3 From 2D to 3D

Classic ways to solve the problem of inferring 3D joints from their 2D projections can be traced back to the work of Lee and Chen [10], who make use of a binary decision tree where each split correspond to two possible states of a joint with respect to its parent. Most of the methods in recent years learn the mapping from between 2D keypoints and 3D pose with deep neural networks. Pavlakos et al. [16] applied a stacked hourglass architecture deep convolutional neural network [14] that maps to probability distributions in 3d space instead of regressing 2d joint probability heatmaps. Martinez et al.[13] use a fully-connected network with residual connections which takes a 2D pose from a 2D detector as input and predicts a 3D pose with a regression loss from a single image. This simple model has a pretty good performance on Human3.6 dataset. Zhou et al. [24] regress both 2D and 3D poses simultaneously and reach similar performance to [13]. Sun et al. apply integral regression for joint 2D and 3D pose estimation, which is compatible with any

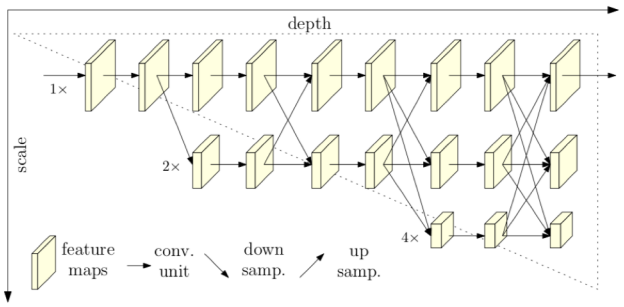

Figure 1: Architecture of HRNet

heat map-based methods. Zhao et al. [23] propose Semantic Graph Convolutional Networks (SemGCN) which operates on 3D pose regression tasks with graph-structured data.

3 METHOD

We utilize a two-stage protocol and the solution is mainly based on existed methods in the literature. The first step is to generate 2D keypoint heatmap via learning high-resolution representation (HRNet) [20], after which the 2D pose keypoints are extracted via taking the expectation of the heatmap. In the second step, the 2D pose locations are fed into another deep model to estimate 3D joint positions. We have tried two methods in the second stage: 1) learning a novel semantic graph convolutional network for 3D pose regression(SemGCN) [23], 2) training a deep feed-forward network proposed by [13].

3.1 From image to 2D keypoints: HRNet

HRNet learns reliable high-resolution and is currently the state-of-the-art for detecting locations of human joints in 2D images. Starting from a high-resolution subnetwork, it gradually adds high-to-low resolution subnetworks to build more stages, and connect the multi-resolution subnetworks in parallel. The novelty of this work is that it connects high-to-low resolution subnetworks in parallel instead of sequentially, maintaining the high resolution rather than recovering the resolution through a low-to-high process as done in many existed works. Therefore, the predicted heatmap potentially offers spatially higher precision. Another contribution of HRNet is that it performs repeated multi-scale fusions by exchanging the information among parallel multi-resolution subnetworks through the whole process, accordingly generates high-resolution representations that are rich for pose estimation. As a result, the predicted heatmap is expected to be more accurate and spatially more precise.

The network structure that we use shown in Figure 1 is the same as [20]. It consists of four stages with four parallel subnetworks. The first stage contains 4 ResNet-50 residual units, while the 2nd, 3rd, 4th stages contain 1, 4, 3 exchange blocks respectively. Each exchange block includes 4 residual units where each unit contains two 3×3 convolutions in each resolution and an exchange unit across resolutions. In our setting, the output is a 17×64×64 heatmap, where 17 is the number of joints in Human3.6 dataset.

3.2 2D Integral Loss

Instead of taking argmax, we take expectations over the heatmap in different layers to extract keypoint of each joint, inspired by[21]. The resulting keypoint location is a weighted sum of its coordinate:

$$\hat{\mathbf{J}}_k = \sum \mathbf{p} * \mathbf{H}_k(\mathbf{p}) \tag{1}$$

, where $\mathbf{p} = (u, v)$ is the location of the keypoint, and $H_k(\mathbf{P})$ is its probability computed by performing softmax operation on the heatmap:

$$\mathbf{H}_k(\mathbf{p}) = \frac{e^{\mathbf{H}_k(\mathbf{p})}}{\sum_{\mathbf{q}} e^{\mathbf{H}_k(\mathbf{q})}} \tag{2}$$

The loss function is simply mean-square-errors of 2D keypoint locations:

$$L_{2D} = \frac{1}{N} \sum_k (J_{k2D} - \hat{J}_{k2D})^2 \tag{3}$$

3.3 From 2D keypoints to 3D pose: SemGCN & Feed-forward

SemGCN [23] is state-of-the-art for 2D to 3D human pose regression in monocular settings. Given a 2D human pose (and the optional relevant image) as the input, it predicts the locations of its corresponding 3D joint. The key idea of the proposed Semantic Graph Convolution (SemGConv) is to learn channel-wise weights for edges as priors implied in the graph, and then combine them with kernel matrices. The architecture of SemGConv is illustrated in Figure 2. Integrating SemGConv and non-local layers[22] yields SemGCN, which captures both local and global relationships among graph nodes. SemGCN can also incorporate image content, such as deep image features, to boost the 3D pose regression performance. Since we've found training SemGCN is very time-consuming, we only explore the setting when 2D joint locations are the only input.

The SemGCN in our implementation is depicted in Figure 3. It is similar to the original paper, except that our inputs are 2D keypoint locations in shape 17×2. The model has 4 repeated building blocks. Each building block is one residual block composed of 2 SemGConv layers with 128 channels and 3×3 convolution kernel, followed by 1 non-local layer. All SemGConv layers are followed by batch normalization and ReLU activation, except the last layer.

Besides SemGCN, we also tried a deep feed-forward network proposed by [13]. The residual block of the model, depicted in Figure 4 contains linear layers of 1024 neurons, followed by batch normalization, dropout, a ReLU activation, and skip connection. The network simply contains two repeated building blocks and two extra linear layers: One is applied to the input to increase the dimensionality to 1024, and one is employed before the output to reduce the dimensionality to $3N$, where N is the number of joints.

The loss function that we use is the mean-squared-errors of 3D joint locations:

$$L_{3D} = \frac{1}{N} \sum_k (J_{k3D} - \hat{J}_{k3D})^2 \tag{4}$$

3.4 Occlusion Augmentation

For the 3D human pose estimation task, data augmentation is of great significance since most of the methods suffer from overfitting. Recent research on the occlusion robustness of 3D pose

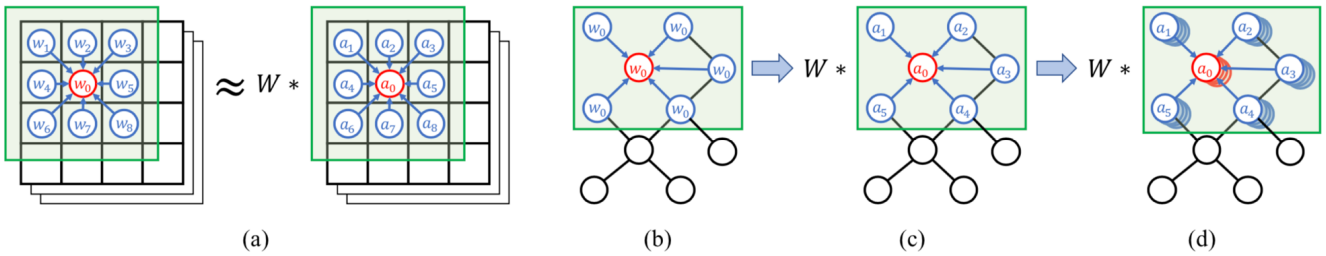

Figure 2: Architecture of SemGConv block

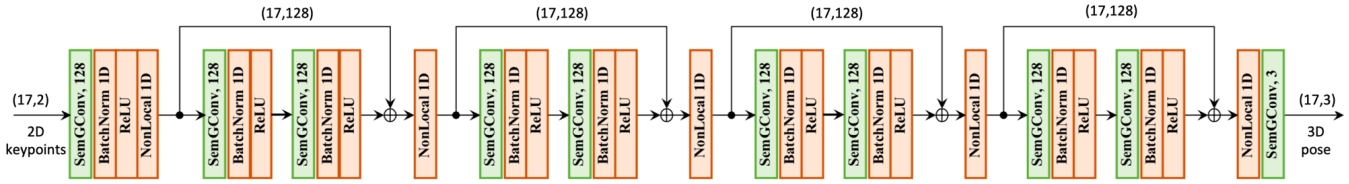

Figure 3: Architecture of SemGCN

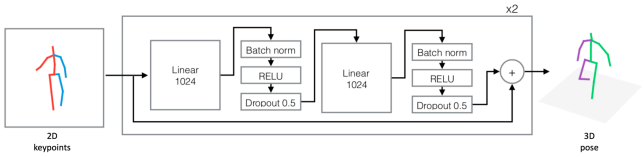

Figure 4: Architecture of a Feed-forward Network

Figure 5: Occlusion Augmentation

estimation [18] claims that augmenting image dataset with synthetic occlusion acts as an effective regularizer. Inspired by [19], we integrate occlusion augmentation as a preprocessing step in our pipeline. In this step, we utilize the Pascal VOC dataset [6]. The persons, segments labeled as difficult or truncated, and segments with the area below 500 px in Pascal VOC dataset are filtered out. Subsequently, as shown in Figure 5, with probability p_{occ}, a random number (from 1 to 8) of those objects are pasted at random locations in each frame of the Human3.6 training set. In addition to the occlusion augmentation, we also randomly change the brightness, contrast, and saturation of the training images.

4 EXPERIMENT

We use PyTorch [15] as our deep learning framework. Since the preprocessed version Human3.6M dataset is given in TFRecord format which is designed for Tensorflow [1], we first extract and save images as well as ground truth information from TFRecord files. For the extracted training images, we randomly place occluders on them with probability $p_{occ} = 0.5$ and obtain the augmented dataset. We train two stages separately. For image->2D stage, we use the Adam [9] optimizer with a learning rate of 1e-5. For 2D->3D stage, we use the AdamW [12] optimizer with a learning rate of 5e-4. For all training experiments, we set weight decay to 0.05. The evaluation results are shown in Table 1.

Method	Public PA-MPJPE
HRNet+SemGCN	62.2902558307
HRNet+Feed-forward	58.5883454863

Table 1: Public PA-MPJPE of Our Pipeline

5 DISCUSSION

We have shown our solution to 3D human pose prediction from monocular 2D RGB images. The task is splitted into two stages: 1) detecting 2D pose keypoints and 2) predictinng 3D pose from 2D keypoint locations. We take advantages from existed work of HRNet, SemGCN and a feed-forward network. Plus, for pre-processing, we occlude images using Pascal VOC dataset as augmentation.

We train these two stages separately and achieve our best result, PA-MPJPE of 58.58, using HRNet + feed-forward network with occlusion augmentation. We've found that HRNet is extremely prone to over-fitting, as the validation error grows after only two epochs of training. This is alleviated by occlusion augmentation because we've seen a boost of performance both in SemGCN and feed-forward network, which means that the upstream HRNet provides keypoint locations with higher precision. Since training SemGCN is very time-consuming, we were unable to spend too much time tuning parameters. But we believe it is a very promising method to solve this task.

There are multiple directions of our future work: 1) Explore various loss functions, considering bone consistency, bone symmetry, and bone length. 2) Incorporate image content in SemGCN rather than using solely 2D keypoint locations as input. 3) Fine-tune the model on MPII dataset to avoid overfitting. 4) Train the whole pipeline end-to-end, instead of training two stages separately.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*. 265–283.

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. [n. d.].

[3] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2018. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *arXiv preprint arXiv:1812.08008* (2018).

[4] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7291–7299.

[5] Cristian Sminchisescu Catalin Ionescu, Fuxin Li. 2011. Latent Structured Models for Human Pose Estimation. In *International Conference on Computer Vision*.

[6] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2015. The pascal visual object classes challenge: A retrospective. *International journal of computer vision* 111, 1 (2015), 98–136.

[7] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. 2017. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*. 2334–2343.

[8] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2014. 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 (jul 2014), 1325–1339.

[9] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[10] Hsi-Jian Lee and Zen Chen. 1985. Determination of 3D human body postures from a single view. *Computer Vision, Graphics, and Image Processing* 30, 2 (1985), 148–168.

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

[12] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[13] J. Martinez, R. Hossain, J. Romero, and J. J. Little. 2017. A Simple Yet Effective Baseline for 3d Human Pose Estimation. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 2659–2668.

[14] Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*. Springer, 483–499.

[15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*. 8026–8037.

[16] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. 2017. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7025–7034.

[17] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. 2016. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4929–4937.

[18] István Sárándi, Timm Linder, Kai O Arras, and Bastian Leibe. 2018. How robust is 3d human pose estimation to occlusion? *arXiv preprint arXiv:1808.09316* (2018).

[19] István Sárándi, Timm Linder, Kai O Arras, and Bastian Leibe. 2018. Synthetic occlusion augmentation with volumetric heatmaps for the 2018 eccv posetrack challenge on 3d human pose estimation. *arXiv preprint arXiv:1809.04987* (2018).

[20] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. 2019. Deep High-Resolution Representation Learning for Human Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[21] Xiao Sun, Bin Xiao, S. Liang, and Y. Wei. 2018. Integral Human Pose Regression. In *ECCV*.

[22] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-Local Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[23] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas. 2019. Semantic Graph Convolutional Networks for 3D Human Pose Regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 3420–3430.

[24] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. 2017. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *Proceedings of the IEEE International Conference on Computer Vision*. 398–407.