Machine Perception Final Report
Project 2 – 3D Pose Estimation

Georges Pantalos
pgeorges@student.ethz.ch

Nikita Rudin
rudinn@student.ethz.ch

ABSTRACT

This work is produced in the context of the class « Machine Perception » of the spring semester 2019. The aim is to provide an improved pipeline for human 3D pose estimation from 2D single color images. Our approach is based on integral pose regression.

1 INTRODUCTION

Pose Estimation has been studied extensively over the past decade [4–6] with techniques ranging from RGB-D images (with one additional channel for depth) to simple feature extraction methods like edge direction histograms [7] or SIFT detectors [1]. State-of-the-art methods however are based on heat map representation through deep learning models.

Our approach is derived from [8] and consists in creating a 3D heat map with the grid containing values of a standard normal distribution centered at the position of each joint. The loss function is split into two functions weighted in a particular way for each stage of the training. This approach is particularly interesting because it allows an end-to-end training which was not possible with the traditional heat map methods. This was due to the non differentiability of the argmax operator when inferring the joint location from the heat map. We show in section 2 how this problem can be overcome by taking the expectation instead of the argmax.

Lastly, when creating a heat map, for efficiency reason, the 3D resolution is considerably reduced in comparison to the original image. This often leads to quantization problems in the precision of the prediction. Again, we will see how this problem can be reduced by using Gaussian valued heat maps in the regions of interest.

2 METHOD

Our design closely follows a method in [8] but with some non minor adjustments.

2.1 Integral Pose Regression

Heat map approaches generally work in the following manner. First a heat map $\mathbf{H}_k$ is created for each joint k for each image. Then the final joint's location $\mathbf{J}_k$ is obtained by taking the position $\mathbf{p}$ that maximizes the value of $\mathbf{H}_k$, *i.e.*

$$\mathbf{J}_k = \arg\max_{\mathbf{p}} \mathbf{H}_k(\mathbf{p}) \tag{1}$$

However, to avoid *quantization errors* (due to the resolution downscaling done by the heat map compared to the original image) and *non-differentiability* (which makes the decision step non learnable), an interesting approach is to take the expectation over the heat map, that is

$$\mathbf{J}_k = \int_{\mathbf{p}\in\Omega} \mathbf{p} \cdot \tilde{\mathbf{H}}_k(\mathbf{p}) \tag{2}$$

$$= \sum_{p_z=1}^{D} \sum_{p_y=1}^{H} \sum_{p_x=1}^{W} \mathbf{p} \cdot \tilde{\mathbf{H}}_k(\mathbf{p}) \tag{3}$$

where $\tilde{\mathbf{H}}_k$ represents the normalized (by softmax activation) heat map, *i.e.*

$$\tilde{\mathbf{H}}_k(\mathbf{p}) = \frac{e^{\mathbf{H}_k(\mathbf{p})}}{\int_{\mathbf{q}\in\Omega} e^{\mathbf{H}_k(\mathbf{q})}} \tag{4}$$

and D, H, W represent the depth, height and width of the heat map respectively. Note that expression (3) comes from the fact that the domain Ω is discrete.

The authors of [8] suggest that this approach (called *integral pose regression*) combines both the merits of heat map based and regression approaches mainly because it allows end-to-end training and is non-parametric, while solving the problems of non-differentiability and quantization errors.

2.2 Loss functions

We use two loss functions: the integral loss and the heatmap loss which we discuss in more detail hereafter.

2.2.1 Integral Loss. The integral loss is simply the mean absolute error between the predicted joint position $\mathbf{J}_k$ and the ground truth joint position $\mathbf{p}_k$. This is the main loss function which implements the regression of the joint positions.

2.2.2 Heatmap Loss. The heatmap loss is obtained as follows:

(1) First, we generate the heat map by assigning a value of 1 to the position of the respective joint and values decreasing in a gaussian manner with the Euclidean distance up until 5 units in all 3 cardinal directions (thus forming a cube of size 11 × 11 × 11). Note that all the values outside the cuboid are assigned value 0. That is,

$$(\mathbf{H}_k)_{ij\ell} = \frac{\exp\left(-d_{ij\ell}^2/2\right)}{\sqrt{2\pi}} \tag{5}$$

for all i, j, ℓ such that

$$-5 \leq i - p_{x,k} \leq 5 \tag{6}$$
$$-5 \leq j - p_{y,k} \leq 5 \tag{7}$$
$$-5 \leq \ell - p_{z,k} \leq 5 \tag{8}$$

and $(\mathbf{H}_k)_{ij\ell} = 0$ otherwise. The distance defined as

$$d_{ij\ell} := \sqrt{\left(i - p_{x,k}\right)^2 + \left(j - p_{y,k}\right)^2 + \left(\ell - p_{z,k}\right)^2} \tag{9}$$

is the Eucledian distance between the current cell (i, j, ℓ) and the joint k's cell $(p_{x,k}, p_{y,k}, p_{z,k})$. Note that $(i, j, \ell) \in \{0, W\} \times \{0, H\} \times \{0, D\}$.

(2) This procedure is done once for the ground truth of each image and each joint. It is repeated at each iteration for the predicted poses.

(3) The loss function is the binary crossentropy computed between the target heat map and the current heat map.

Moreover, when implementing the heat map loss, we have experimented with replacing the gaussian by a one-hot encoding of the joint to a single position. Supposedly this improve performance according to [8]. However it was not the case in our experiments.

2.3 Data Augmentation

The original images are augmented using the python library `imgaug`. The specific parameter details and functions used are listed below.

- `Affine(translate_percent=0.05, scale={"x": (0.75, 1.25), "y": (0.75, 1.25)})`: random translation and scaling
- `AddToHueAndSaturation((-60, 60))`: random color changes
- `ContrastNormalization((0.7, 1.3))`: random contrast change
- `Multiply((0.5, 1.5), per_channel=True)`: random brightness changes, independently for each channel

This step is performed on the raw images (*i.e.* before the normalization). Some augmented pictures are displayed in Figure 1.

2.4 Other Details

Below are listed some important complementary details.

Architecture. The model design used as a backbone was ResNet with depth 50. The expectation over the heat map step is done there which allows for end-to-end training.

Dataset. The training dataset consisted of a pre-processed version of the Human3.6m [3] dataset containing a total of 312188 training images. The model's performance is then evaluated on 10987 images of the same original data set.

3 INSTRUCTIONS

Below are displayed precise instruction to reproduce our best performing model.

(1) Unizip `code.zip`
(2) Install imgaug: `pip install imgaug`
(3) Run the training for about 40 hours (≈ 25 epochs × 1:30 hours per epoch): `./run_train.sh`
(4) After successful training run `./run_test.sh` to generate labels for the test set

4 RESULTS

We tried 4 different approaches:

(1) Use the provided code and implement normal regression using a ResNet50 with some data augmentation
(2) Integral pose regression, only with integral loss (no heat map loss)
(3) Final pipeline as described above
(4) Integral pose regression but on a ResNet50 pretrained on ImageNet, which was not allowed in the context of this project.

The test MPJPE for each of the approaches used during this project is displayed in Figure 2.

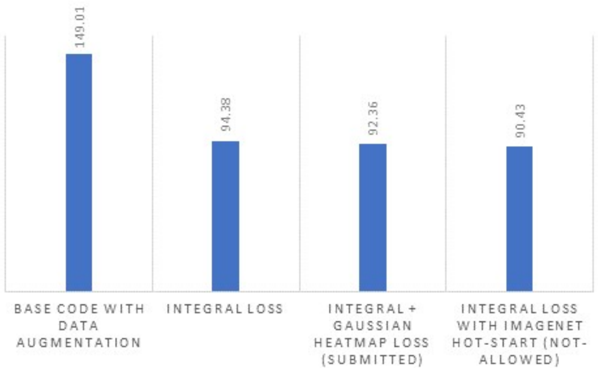

Figure 2: Performance of each training method

5 DISCUSSION

We will now discuss some of the observations made throughout our experiments.

First, we trained the model without the heat map loss (*i.e.* only the integral loss) with a backbone ResNet50 pretrained on ImageNet [2] and saw that the performance was only slightly improved than training it from scratch. This is probably due to the large size of our dataset.

We then trained it from scratch using only the integral loss once more. This model gave us a *mean per joint position error* (MPJPE) of 94.38 after 23 epochs. Finally, we added the heat map loss with

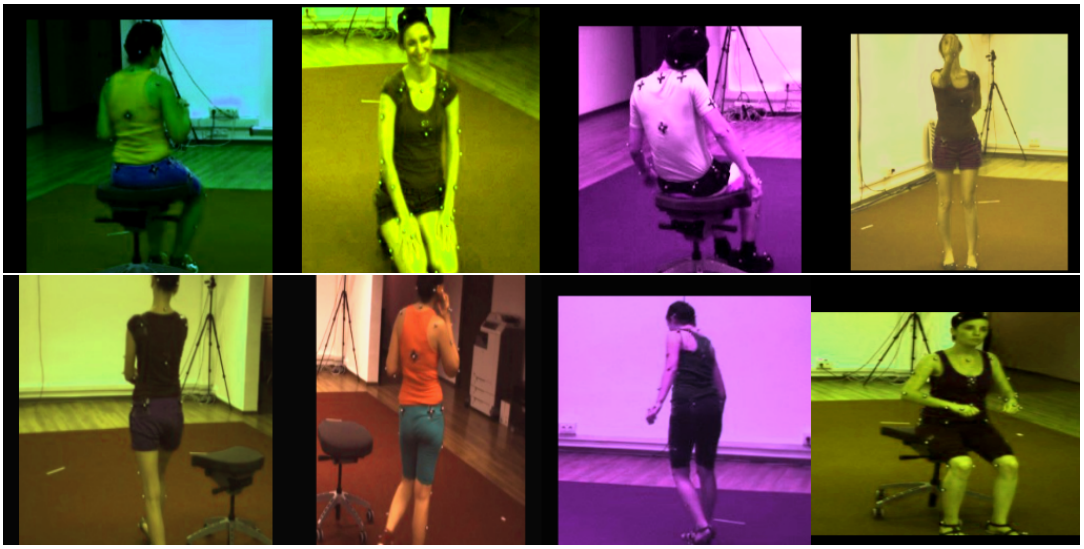

Figure 1: Some examples of augmented images from our dataset

a Gaussian centered around the joint positions and got a score of 92.36 after 2 more epochs.

We can therefore see the positive influence of the heat map loss. A severe drawback of this second loss is the fact that it is far more computationally expensive. One epoch took more than 7 hours. A better performance could probably be achieved by learning for more epochs with the gaussian loss, but because of time restrictions, we were not able to do so.

It is worth mentioning, that while the test score improved when re-training with the two losses, the training error actually increased. Taking the expectation of the position of the joint weighted by its heat map value allows to learn the position smoothly as the distribution of the joint position estimation may be multimodal. Adding the gaussian heat map target, forces the predicted heat map to adopt a more gaussian structure, which seems to improve the generalization.

Note that the heat map loss function serves more as a help to the integral loss function as we have tried to train the model using only heat map loss and it revealed itself to be very inefficient on its own. Whereas integral loss alone was still able to perform decently.

6 CONCLUSION

We have verified that the challenging problem of 3D human pose estimation from single color images can be tackled efficiently with integral regression approaches combined with heat map representations which constitute the state-of-the-art methods. We have based our implementation on [8]. Our main differences are the fact that we do not use a pre-trained backbone network and our gaussian heat map loss is also different from their description. We achieve better results than claimed by the paper for the simple integral regression loss. When combining it with the heat map loss our results improve, but we did not have the capacity to train it for enough time to achieve the best possible results.

REFERENCES

[1] L. Bo, C. Sminchisescu, A. Kanaujia, and D. Metaxas. 2008. Fast algorithms for large scale conditional 3D prediction. (June 2008), 1–8. https://doi.org/10.1109/CVPR.2008.4587578

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. (2009).

[3] 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 Trans. Pattern Anal. Mach. Intell.* 36, 7 (July 2014), 1325–1339. https://doi.org/10.1109/TPAMI.2013.248

[4] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. 2017. End-to-end Recovery of Human Shape and Pose. *CoRR* abs/1712.06584 (2017). arXiv:1712.06584 http://arxiv.org/abs/1712.06584

[5] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. 2017. A simple yet effective baseline for 3d human pose estimation. *CoRR* abs/1705.03098 (2017). arXiv:1705.03098 http://arxiv.org/abs/1705.03098

[6] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. 2016. Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose. *CoRR* abs/1611.07828 (2016). arXiv:1611.07828 http://arxiv.org/abs/1611.07828

[7] Gregory Shakhnarovich, Paul Viola, and Trevor Darrell. 2003. Fast Pose Estimation with Parameter-Sensitive Hashing. (2003), 750–. http://dl.acm.org/citation.cfm?id=946247.946721

[8] Xiao Sun, Bin Xiao, Shuang Liang, and Yichen Wei. 2017. Integral Human Pose Regression. *CoRR* abs/1711.08229 (2017). arXiv:1711.08229 http://arxiv.org/abs/1711.08229