Machine Perception Report CreditSuissers

Tim Ehrensperger Alihan Keresteciogl u Sören Lambrecht Rahul Steiger

ABSTRACT

This project endeavors to reconstruct the intricate surface of a human hand engaged in interaction, solely from a single RGB image. To capture the three-dimensional form of the hand, we employ the widely recognized MANO model, as introduced in [2] Our approach employs a comprehensive framework that initially extracts image features via a pre-trained backbone model. Subsequently, a Multilayer Perceptron (MLP) is employed to predict the camera translation, shape, and pose parameters of the MANO model. This methodology has proven instrumental in attaining remarkably precise 3D representations of right-hand images.

1 INTRODUCTION

The modeling of hand surfaces presents inherent challenges owing to several factors. Primarily, hands often suffer from substantial occlusion, exhibit rapid movements, and the acquisition of accurate ground truth data is prohibitively expensive. Consequently, to mitigate these challenges, we employ a meticulously pre-processed dataset sourced from the HO3D dataset, which encompasses precise input parameters for the MANO model. By utilizing these input parameters, the MANO model effectively generates the corresponding three-dimensional representation. [2] Consequently, our primary objective involves the prediction of these essential MANO input parameters, comprising hand pose (consisting of 6x16 parameters), shape (comprising 10 parameters), and camera translation (comprised of 3 parameters).

At a high level, our most successful outcomes are obtained through a modified version of the methodology employed in the work by Kanazawa et al. [1]. In the subsequent section, we provide a comprehensive explanation of the specific modifications and refinements implemented in our approach.

2 METHOD

The foundation of our implementation is an extension of the methodology presented in the work by Kanazawa et al. [1]. A visual overview of our approach is illustrated in Figure 1. To provide a more comprehensive understanding, we offer a detailed explanation as follows:

The hand images utilized in this project are sourced from the HO3D dataset. Specifically, we exclusively focus on right hand images for our analysis. These pixel values serve as input to our backbone model, which plays a crucial role in feature extraction and is commonly referred to as the "encoder." In contrast to the original skeleton code, which employs ResNet, we adopt the ResNeXt101 32X8D model with pretrained weights. Notably, this network diverges from conventional ResNet architectures by incorporating an additional dimension known as "cardinality" within its repeated building block system [3]. Extensive benchmark evaluations have demonstrated that this variant exhibits enhanced accuracy and performance.

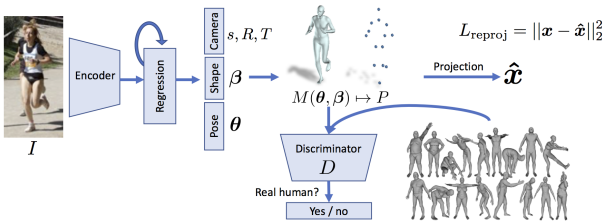

Figure 1: Framework overview from [1] where we adopted our model from. In comparison to [1] we do not use any Discriminator but only backpropagate with respect to MDJPE-loss.

After obtaining the 2048-dimensional feature vector from the last layer of the ResNeXt model through average pooling, we utilize it as the input for subsequent networks. To predict the required parameter groups, namely camera, shape, and pose, we employ separate Multilayer Perceptron (MLP) networks. The architecture of each MLP is sequential and consists of the following dimensions for their respective layers:

- Input dimension: 2048
- First hidden layer: 512
- Second hidden layer: 512
- Third hidden layer: 256
- Output layer: 3, 10, and 96 for Camera, Shape, and Pose, resp.

Similar to the provided skeleton code, we incorporate an HMR (Human Mesh Recovery) refinement layer, which refines our predictions based on previous values. The MLP architecture remains unchanged, with the only difference being the substitution of activation functions with Exponential Linear Unit (ELU) activations. Furthermore, we increase the number of refinement iterations from 3 to 100, allowing for more comprehensive optimization.

In terms of the loss function, we make modifications as follows: We assign a weight of 0.001 to all losses except for the Mean Per Joint Position Error (MPJPE) loss, denoted as kp3d in the code, which carries a weight of 1.0. Although our model is evaluated solely on MPJPE, we believe that the other losses serve as a regularization factor, preventing overfitting during training.

3 HYPERPARAMETERS DETAILS

Parameter	Value
learning rate	1e-4
batch_size	32
grad_clip	150
lr_decay	0.01

4 EVALUATION

In our evaluation process, we adopted a two-stage approach to assess our models' performance. Firstly, we sought to determine the epoch that achieved optimal performance on both the training set and the validation set. We utilized the model's loss function as a metric to guide our selection, ensuring it did not suffer from overfitting. Carefully avoiding hasty decisions based solely on the latest epoch, we identified the most promising epoch for each model. Subsequently, we compared the trained models based on their Mean Per Joint Position Error (MPJPE) loss on the public test set. This allowed us to identify the model with the lowest error and evaluate its performance. This section provides a detailed analysis of the influential factors and presents experimental results that highlight key findings.

4.1 Activation Functions

We explored different activation functions , including ELU and GELU. ELU is a smooth activation function that effectively addresses the "dead ReLU" issue associated with negative values. It has demonstrated superior performance compared to ReLU, contributing to improved results in our experiments. Similarly, GELU also outperformed ReLU, but ELU yielded the best results in our specific context. ELU proved to be the optimal activation function for achieving superior performance in our experiments.

4.2 Initialization of MANO Parameters

We discovered that initializing MANO parameters significantly influenced the overall performance. Initially, the skeleton code initialized shape and pose parameters as zero vectors, while camera translation parameters were initialized using an MLP. However, our experiments revealed that initializing all parameters as MLPs yielded superior results. Instead of only initializing camera translation parameters with an MLP, we initialized all MANO parameters using MLPs. This approach effectively captured complex relationships and intricacies in the data, resulting in improved performance compared to zero vector initialization.

4.3 Learning Rate

During training, we initially used a learning rate of 1e-5 and later transitioned to a learning rate of 1e-4. Our decision to adjust the learning rate was based on the observation that the models trained with a learning rate of 1e-4 yielded better results.

4.4 Choice of Backbone Model

In our experimentation, we compared the performance of two different convolutional neural network (CNN) architectures: ResNet18 and ResNext101. Our findings consistently indicated that ResNext101 outperformed ResNet18 across various settings and conditions.

4.5 Loss Function

We trained our models using different loss functions: the skeleton code's weighted average of mean squared errors for MANO parameters and outputs, MPJPE alone, and a weighted average where MPJPE had a weight of 1 and the other losses had a weight of 0.001. The last one was the best performing among all.

Activation	Backbone	LR	Param. Init.	Loss	**Error**
RELU	ResNet18	1e-5	Zeros	Skeleton	46.526
RELU	ResNet18	1e-5	NN	Skeleton	37.711
GELU	ResNet18	1e-5	NN	Skeleton	34.131
ELU	ResNet18	1e-4	NN	Skeleton	32.122
ELU	ResNext101	1e-4	NN	Skeleton	31.327
ELU	ResNext101	1e-4	NN	MPJPE	28.086
ELU	ResNext101	1e-4	NN	Our	26.215

Table 1: Test Performance in Different Settings

5 DISCUSSION

Our model achieved accurate 3D model predictions, surpassing the provided ground-truth in Figure 2. We suspect the ground-truth's accuracy may vary within the training dataset. To further improve our model's accuracy, fine-tuning on a more precise dataset is recommended. This could involve manually selecting samples with the most accurate ground-truth labels from the training dataset.

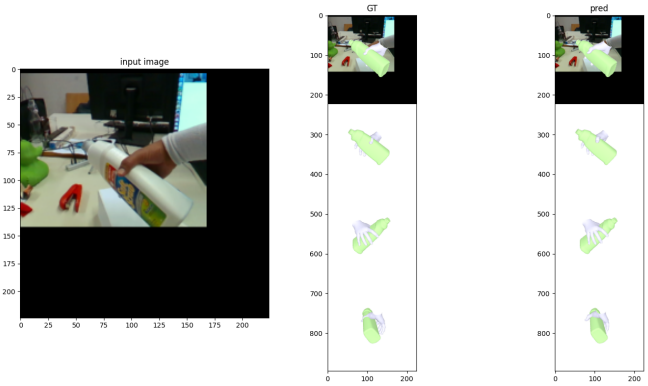

Figure 2: Example of prediction

6 CONCLUSION

In conclusion, our modifications to the skeleton code yielded substantial improvements in the model's performance. Leveraging the MANO model, we successfully achieved accurate 3D representations of right hands. However, it's important to note that our model is not flawless in predicting 3D hand shape and pose, particularly when significant occlusions occur in the 2D image. Moreover, the limitations of the provided dataset's accuracy suggest the potential for future work to focus on fine-tuning the model using a subset of more precise training data.

REFERENCES

[1] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. 2018. End-to-end Recovery of Human Shape and Pose. In *Computer Vision and Pattern Recognition (CVPR)*.

[2] Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)* 36, 6 (Nov. 2017).

[3] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2016. Aggregated Residual Transformations for Deep Neural Networks. *CoRR* abs/1611.05431 (2016). arXiv:1611.05431 http://arxiv.org/abs/1611.05431