HBE: Hand Branch Ensemble Network for Real-time 3D Hand Pose Estimation

Yidan Zhou[1], Jian Lu[2], Kuo Du[1], Xiangbo Lin[1*], Yi Sun[1], and Xiaohong Ma[1]

[1] Dalian University of Technology, Dalian, China
{shine0910,dumyy2728}@mail.dlut.edu.cn
{linxbo,lslwf,maxh}@dlut.edu.cn
[2] Dalian University, Dalian, China lujian@dlu.edu.cn

Abstract. The goal of this paper is to estimate the 3D coordinates of the hand joints from a single depth image. To give consideration to both the accuracy and the real time performance, we design a novel three-branch Convolutional Neural Networks named Hand Branch Ensemble network (HBE), where the three branches correspond to the three parts of a hand: the thumb, the index finger and the other fingers. The structural design inspiration of the HBE network comes from the understanding of the differences in the functional importance of different fingers. In addition, a feature ensemble layer along with a low-dimensional embedding layer ensures the overall hand shape constraints. The experimental results on three public datasets demonstrate that our approach achieves comparable or better performance to state-of-the-art methods with less training data, shorter training time and faster frame rate.

Keywords: hand pose estimation, depth image, Convolutional Neural Networks

1 Introduction

The research of 3D hand pose estimation is a hotspot in the field of computer vision, virtual reality and robotics [5, 18]. With the advent of depth cameras, studies based on depth image have made significant progress [28]. Nevertheless, there is still a challenge for the recovery of 3D hand poses due to the poor quality of depth images, high joint flexibility, local self-similarity and severe self-occlusions.

In general, depth based hand pose estimation can be categorized into two main approaches as either generative model-based or discriminative learning-based methods. Model based approaches assume a pre-defined hand model and then fit it to the input depth image by minimizing specific objective functions [21, 22, 24, 13, 26, 32, 31]. However, the accuracy of these methods is highly dependent on the objective function and sensitive to initialization. Additionally, such tracking-based model approaches are awkward to deal with large changes between two adjacent frames, which are common as the hand tends to move

fast. Alternatively, learning based approaches train a model with a large amount of data, and the hand pose parameters can be regressed directly. In this way, detecting hand pose frame by frame is easy to handle with fast hand movements.

Recently, learning based approaches have achieved remarkable performance in hand pose estimation from a single depth image. Although traditional machine learning methods have made significant progress, their performances are too dependent on the hand-craft features [28, 12, 35, 30, 27]. In recent years, Deep Learning methods have been paid more attention due to their abilities of learning effective features automatically. Early studies regressed joint locations from a depth image with a simple 2D Convolutional Neural Network [20, 33, 39, 16], which had high frame rate but low precision. To improve the accuracy, different strategies were proposed. One way was to improve the data quality. [19] used data augmentation to reduce the prediction error. [8, 17] converted the 2.5D depth image to 3D voxel representation to make use of the 3D spatial structure. [23] learned the feature mapping from a synthetic image with high quality to a real image. The other way was to design more complex network to extract more features. [19, 17, 10, 9] added residual module in their network. [17, 34] used encoder and decoder to learn features in the latent space. [8, 17] applied a 3D CNN instead of 2D CNN to estimate per-voxel likelihood of 3D locations for each hand joint. By combining the effective strategies mentioned above, [17] achieved the best results in the Hands In the Million (HIM2017) Challenge Competition so far [36]. However, their methods were too complex both in data preprocessing procedure and in network structure to get the efficient training and testing.

In order to improve the efficiency while ensuring accuracy, in this paper, we design a highly efficient and relatively simple Convolutional Neural Network structure named Hand Branch Ensemble network(HBE). The proposed network can achieve comparable accuracy with state-of-the-art studies even better than them using fewer training data and shorter training time but faster frame rates. Fig. 1 gives an overview of our proposed network structure. The core idea is to take advantage of the prior knowledge of the motion and the functional importance of different fingers [29, 2, 15, 4]. Since the thumb and the index finger play an even more important role in the grasping, manipulation and communication, while the middle finger, ring finger and little finger play an auxiliary role in most cases, we simplify the five-finger structure into three parts: thumb, index fingers and the other fingers. Correspondingly, the proposed HBE network learns the features of each part by each branch respectively. It makes full use of the shallow low-level image features that are more sensitive to the size, orientation and location information, which can greatly reduce the computational complexity and the training time. Moreover, we propose a branch ensemble strategy by concatenating features from the last fully connected layers of each branch and then the integrated features are used to infer the joint coordinates with the extra regression layers. Different from REN [10] training individual fully-connected layers on multiple feature regions and combining them as ensembles, our ensemble strategy directly exploits the features of different hand parts, which is more intuitive for the hand pose estimation. Motivated by Deep Prior [20], we add a

bottleneck layer as a low dimensional embedding to learn the hand pose physical prior before the output layer.

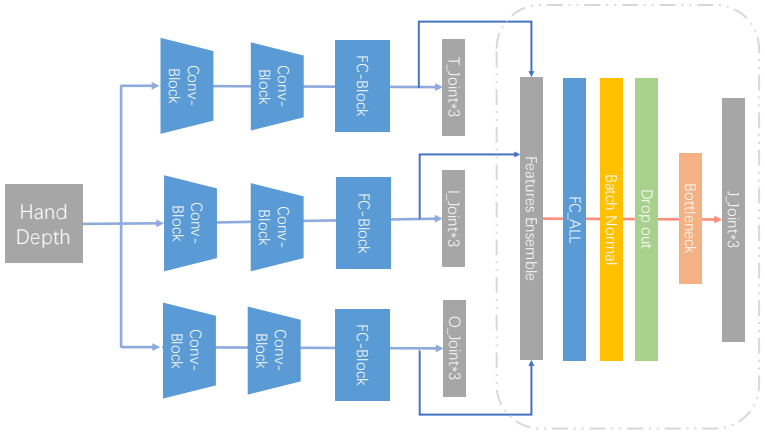

Fig. 1. The Hand Branch Ensemble(HBE) network: based on the activity space and functional importance of five fingers. The top branch handles the thumb, the median branch handles the index finger and the bottom branch handles the other fingers. The features are ensemble along with an additional fully connected layer and a bottleneck layer

The proposed HBE network is evaluated on three challenging benchmarks: the HIM2017 Challenge Dataset [37], the ICVL hand pose dataset [30] and the MSRA dataset [27]. The experiments show that our method achieves results comparable or better than state-of-the-art methods.

In summary, our contributions are:

1. We propose a new three-branch Convolutional Neural Network estimating full 3D hand joint locations from a single depth image. The structural design inspiration comes from the understanding of the differences in the functional importance of different fingers. In addition, a branch feature ensemble strategy is introduced to merge features of each branch along with a fully connected layer and a low-dimensional embedding layer, which emphasizes the correlation of different hand parts and ensures the overall hand shape constraints.

2. We design a relatively lightweight architecture and achieve comparable or better performance to state-of-the-art methods on publicly available datasets with less training data, shorter training time and faster frame rate.

The paper is organized as follows. After reviewing the related work in Section 2, we describe our proposed method in Section 3. Experimental results and discussions are reported in Section 4, and the conclusions are drawn in Section 5.

2 Related Work

In this section, we briefly discuss the Deep Learning based works on hand pose estimation, especially those closely related to our method. These approaches have achieved good performance due to the success of Deep Learning as well as the public large hand pose datasets [30, 27, 33, 38, 6]. However, most studies estimated the hand pose with all joints directly through a single-branch network. Deep Prior [20] proposed a bottleneck layer into the network for the first time to learn a pose prior, and Deep Model [39] adopted a forward kinematics based layer to ensure the geometric validity of the estimated poses. In spite of introducing hand physical constraints, the performances of these networks are not good enough.

To improve the accuracy, the single-branch network was designed more complicated to extract complex features. [19] improved Deep Prior greatly in accuracy by using residual network architecture, data augmentation, and better hand segmentation. [17] also used residual blocks and converted the depth image into a 3D representation form. They implemented an intricate 3D CNN in a voxel-to-voxel mapping manner for prediction. Although the accuracy is significantly improved, data conversion and network structure are too complex so that training and testing process are time-consuming. REN [10] also applied residual blocks in their feature extraction module and divided the feature maps of the last convolutional layer into several regions which were integrated in the subsequent fully connected layers. However, REN used uniform grid to extract region features without considering the spatial information of the hand feature maps.

Hierarchical branch structure can better model the hand topology. Based on REN, Pose-REN [3] boosted the accuracy by iterative refinement. Similar to our approach, they fused features of different joints and different fingers according to the topology of the hand. But they used posterior branch strategy focusing on the iterative refinement. In contrast, we use anterior branch to extract features of different hand parts. The network designed in this way can estimate simpler local poses, and let the training process converge faster. By a posterior branch structure [16] uses 6 branches to represent the wrist and each finger based on the hand geometric structure. Different from their work, we consider it both from the hand functional and kinematic features according to the biological viewpoint, designing an anterior branch structure with learning specific features of each functional part first and then merging them to learn the global features by a bottleneck layer. In addition, we group the last three fingers in one branch rather than one branch for each finger, which guarantees the muscle-association among them and speeds up the network convergence.

3 Methodology

In this section, we will elaborate on our proposed method, network structure and implementation details. Our goal is to estimate the 3D coordinates of J joints: $C = \{c_i\}_{i=1}^{J}$ with $c_i = [x_i; y_i; z_i]$ in the hand from a single depth image.

We design a novel three-branch Convolutional Neural Network based on the functional importance and activity space of different fingers, and then ensemble features to regress all 3D joint locations. The overview of our proposed HBE network is shown in Fig. 1.

3.1 Network Architecture

Hands are frequently used to deal with different tasks, and each finger has different importance and occupies different activity space [2, 15]. The thumb has a unique structure as the opposable characteristic, which plays an important role in in communication or dexterous manipulation. Therefore, the thumb is the most important due to the highest DOF and the largest activity space, so we use a separate branch to learn its features. Although each of the other four fingers have the same DOF, the index finger is closest to thumb and the two fingers alone can generate some gestures, thus the index finger is the second most important and is assigned to a separate branch. Considering the muscle-associated movement among the last three fingers and high correlation in activity, we group them in a single branch.

We design the hand pose estimation network based on above mentioned fingers functional importance. The five-finger structure of the hand is simplified into three parts, corresponding to the three branches of the network respectively. As shown in Fig. 1, three convolutional branches in this network are used to extract the features of each hand part. Since the function of the middle, the ring and the little finger is less important and similar in movements, we merge them into one part and abstractly understand the 5-finger structure of the hand as a 3-part structure. Each part is of equal importance. Therefore, the feature extraction network structure of each branch is the same.

The features from each branch are fused to predict the hand pose. Here we introduce the branch ensemble strategy: features from the last fully connected layers in all branches are concatenated and used to infer 3D joint coordinates with an extra regression layer. It should be pointed out that before the output layer, inspired by the idea of Deep Prior [20], we add a linear bottleneck layer. The bottleneck embedding forces the network to learn a low dimensional representation of the hand pose as a global physical constraint of the hand shape in the network. The label dimensions ($J \times 3$) of the training data are reduced by Principal Component Analysis (PCA) and used as the ground truth of the bottleneck embedding layer. The principal component and the mean value of the low-dimensional data are used as the weights and the biases of the output layer respectively. Finally, the output layer recovers the low-dimensional predictions of the bottleneck layer to the original $J \times 3$-dimensional joint positions.

3.2 Branch Details

When designing the feature extraction layers, we believe that the regression problem of predicting joint positions is rather different from the classification

problem of object recognition, because semantic features are crucial to the latter one. Since shallow network learns low-level spatial features that are more susceptible to the size, direction, and position of an object. Common convolutional layers and max pooling layers for the feature extraction module in each branch are shown in Fig. 2. The estimation of the complex global pose is reduced to the estimations of simpler local poses, enabling the network to be more lightweight and easier to train. A larger convolution kernel that can obtain more

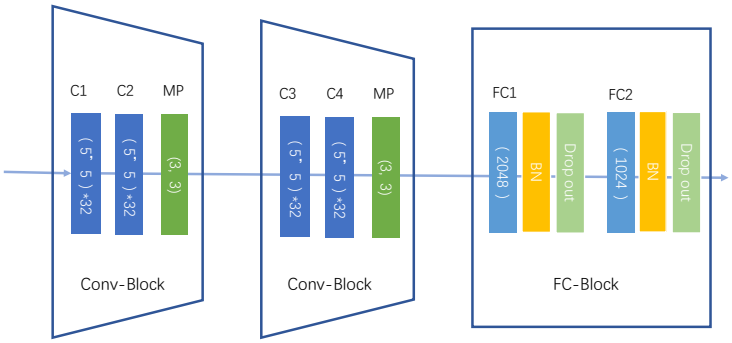

Fig. 2. The structure details of the feature extraction branch. *Ci* represents the convolutional layer, *MP* represents the maxpooling layer, and *FC* represents the fully connected layer

spatial information and lager receptive fields, is very useful for location regression and effective to infer the occluded joints. At each branch, we use a stack of two 5×5 convolutional layers instead of a single larger one, which gains the same size of effective receptive field to a single 9×9 convolutional layer as well as decreases the number of parameters, as calculated in [25]. In the feature mapping module, we add Batch Normalization(BN) layer after each fully connected layer. The distribution change of the training data is accumulated after processed by the middle hidden layers, which will affect the network training. The BN layer has the ability to solve this data distribution change problem, which makes the gradient transfer more fluent and improves the robustness and generalization ability of the training model [11]. All layers use Rectified Linear Unit (ReLU) activation functions.

3.3 Loss Function

The loss function of our network is defined as:

$$Loss = L + \lambda R(w) \tag{1}$$

where $\lambda R(w)$ is the L2-norm regularization term and the regularization coefficient λ is set to 0.001 in our experiments. L is the mean square error between the predicted value and the ground truth. Specifically, we define the loss term L in the form:

$$L = \alpha \times L_{thumb} + \beta \times L_{index} + \gamma \times L_{others} + \sigma \times L_d \tag{2}$$

where L_{thumb} is the loss of the thumb branch, L_{index} is the loss of the index finger branch, L_{others} is the loss of the other fingers branch, L_d is the loss of low-dimensional embedding layer, and $\{\alpha, \beta, \gamma, \sigma\}$ are factors to balance these losses. In our experiment we set them to be 1 for simplification.

Let c_i be the outputs of the branch predicting joint positions in 3D form and C_i be the ground-truth, both c_i and C_i have the form of $[x_i; y_i; z_i]$. We define the loss of each branch as:

$$L_b = \sum_{i=1}^{J_b} \|c_i - C_i\|_2^2, \quad b \in \{thumb, index, others\} \tag{3}$$

where J_b is the number of joints in each branch.

As for the bottleneck embedding, let D be the number of reduced dimension which is much less than $J \times 3$, p_i be the output of the bottleneck layer, P_i be the dimension reduced training label as the ground truth. We define the loss of the low-dimensional embedding as:

$$L_d = \sum_{i=1}^{D} \|p_i - P_i\|_2^2 \tag{4}$$

3.4 Implementation Details

The input of our network is a hand-only depth image, which is generated after a series of preprocessing steps on the dataset. First of all, we cut out the hand area according to the ground truth labels provided by the dataset, then fill the cropped image up into a square, at last resize it to 128×128, and in the meanwhile, we normalize the hand depth value in [-1,1]. Pixel values that are larger than the maximum hand depth or unavailable because of noise are set to 1. This depth normalization step is important for the network to adapt to different distances from the hand to the camera.

Our model is trained and tested on a computer with Intel Core i7 CPU, 32GB of RAM and an NVIDIA GTX1080 GPU. Our network is implemented in Python using the Tensorflow [1] framework. Except for the output layer, all weights are initialized from the zero-mean Normal distribution with 0.01 standard deviation. The network is trained with back propagation using Adam [14] optimizer with a batch size of 128 for 100 epochs. We use a dynamic learning rate with an initial value of 0.001 and reduce it by a factor 0.95 for every epoch. And the dropout rate is set to be 0.85 (keep probability).

4 Experiments

In this section we evaluate our Hand Branch Ensemble(HBE) network on several challenging public hand pose datasets. First of all, we introduce these datasets and the parameters of our methods. Then we describe the evaluation metrics, and finally we present and discuss our quantitative as well as qualitative results.

4.1 Datasets

We evaluate our network on three recent public hand pose datasets: the latest high-quality HIM2017 Challenge dataset [37], the traditional widely used ICVL dataset [30] and MSRA dataset [27].

ICVL Dataset [30] includes a training set of 330K hand pose depth frames with additional in-plane rotations augmented frames and 1.5K testing depth images. In our experiments, we only use 110K training data by random sampling. The dataset provides 16 annotated 3D joints.

MSRA Dataset [27] contains 76K depth frames from 9 subjects with 21 annotated joints. Following [27], we use the leave-one-subject-out cross-validation strategy and average the results over the 9 subjects.

Hands In the Million (HIM2017) Challenge Dataset contains the frame based hand pose estimation dataset and the continuous action tracking dataset [36]. We focus on the frame based estimation dataset, which samples poses from BigHand2.2M dataset [38] and FHAD datasets [6] consisting 957K training and 295K testing depth images. The training data is randomly shuffled instead of continuous action sequence. Including both the first-person view and the third-person view hand pose depth images, this dataset is more challenging for its abundant perspectives and hand poses. Moreover, this dataset provides accurate 21-joint 3D location annotations.

In our experiment, we randomly sample 72K training data from the original HIM2017 Challenge dataset as our training set. Since the original test set provided by the Challenge does not contain the ground truth, we have difficulties to measure the accuracy of our method by ourselves. In order to evaluate more fairly, considering that the original test set contains a total of 295,510 frames of SEEN and UNSEEN subjects, we randomly sample 295,510 frames from the original training set to form a new test set (not included in our training set). Since our test set only contains the SEEN subject, we only compare the results of SEEN in the Challenge leaderboard.

4.2 Evaluation metrics

We follow the common evaluation metrics on hand pose estimation:

1. Mean joint error: The mean 3D distance error for all joints for each frame and average across all testing frames.

2. Correct frame proportion: The proportion of frames that have all joints within a certain distance to ground truth annotation.

4.3 Self-comparisons

Firstly, we compare the effect of the number of branches on the results, as shown in the left figure of Fig. 3. *Single-branch* means that we do not decompose the hand by part but predict all the joints of the hand directly through a single branch CNN. With regard to the *Two-branch*, we train a two-branch network with one branch handles the thumb and the other branch manages the other fingers. Obviously, the *Three-branch* stands for the original three-branch network. As for the *Four-branch*, the last branch handles the ring and the little finger together, the other branches handle the other fingers one by one. The *Five-branch* means that each branch corresponds to one finger. By adjusting the number of convolution channels, the parameters of each network remain roughly constant. These networks are trained and tested on the HIM2017 Challenge dataset.

As shown in the left figure of Fig. 3, the original three-branch structure achieves the best accuracy. The horizontal ordinate of Fig. 3 represents each joint. C means the wrist, and $Ti(i = \{1, 2, 3, 4\})$, Ii, Mi, Ri and Li represent the joint in the thumb, index, middle, ring, and little finger, respectively. And Avg means the mean joint error. For each finger, take the thumb for example, *T1*, *T2*, *T3* and *T4* represent the MCP joint, PIP joint, DIP joint and the fingertip respectively. The following graphs are represented in the same way.

There is a linkage between the middle finger and the ring finger, which is forcibly destroyed by the structure of the Four-branch and the Five-branch. Further more, in most cases, the last three fingers are in the same activity range, and the Three-branch networks can extract their associated features and reduce the redundancy in the feature combining and mapping. Therefore, the performance of the Three-branch outperforms the others.

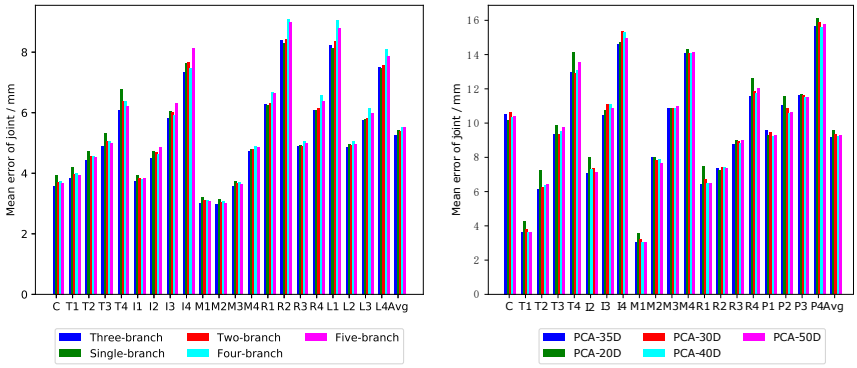

Fig. 3. Self-comparisons. *Left:* Distribution of joint errors in different branch-structures. *Right:* Distribution of joint errors in different bottleneck dimensions

The effect of the bottleneck layer with low-dimensional embedding has been proved in the paper of Deep Prior [20]. In our experiments, we also use this

method to introduce the physical prior of the overall hand pose shape. As for the ICVL dataset, we follow [20] using a 30-dimensional embedding bottleneck layer. And on the MSRA and HIM2017 dataset, we use a 35-dimensional embedding layer according to our experimental results as shown in the right at Fig. 3, which is evaluated on the MSRA P0 test set. The distribution of joint errors shows that the 35 dimensions out of the original 63-dimensional pose spaces performs best. The evaluation shows that enforcing a pose prior is beneficial compared to direct regression in the full pose space, which is in line with the conclusion of [20], but it is not significant in the improvement of accuracy according to our experiments.

Then we evaluate the importance of our ensemble strategy on the HIM2017 dataset. When we directly concatenating the joint predictions of three branches instead of fusing features of each branch as our ensemble strategy, the mean joint error reaches 5.71mm, while the mean joint error of our original network with feature ensemble achieves 5.26mm, and the distribution of joint errors and the correct frame proportion are shown in Fig. 4, which shows that the ensemble strategy in fully connected layer achieves the best performance and further confirms the effectiveness of the ensemble method used in our network.

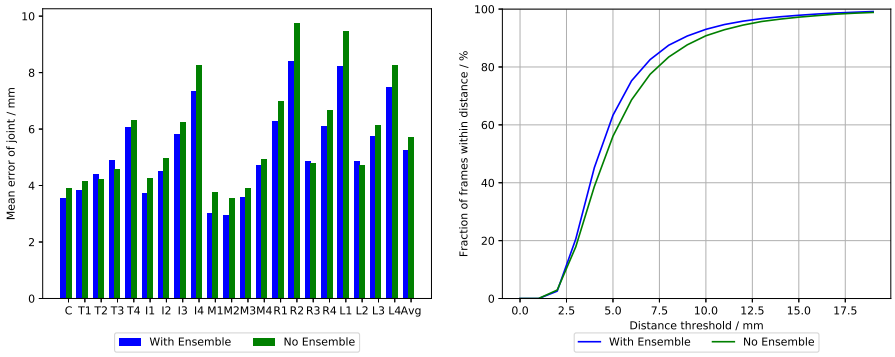

Fig. 4. Self-comparisons of the ensemble strategy. *Left:* Distribution of joint errors. *Right:* Correct frame proportion

Qualitative Results: We present qualitative results on the ICVL, MSRA and HIM2017 dataset in Fig. 5. As we can see, most of the hand poses can be predicted correctly on the three datasets.

4.4 Comparison with state-of-the-art methods

We compare the performance of the Hand Branch Ensemble(HBE) network on three public challenging 3D hand pose datasets (HIM2017, ICVL and MSRA) with some of the state-of-the-art methods, including Deep Prior [20], Deep Model [39], latent random forest (LRF) [30], Crossing Nets [34], V2V-PoseNet [17],

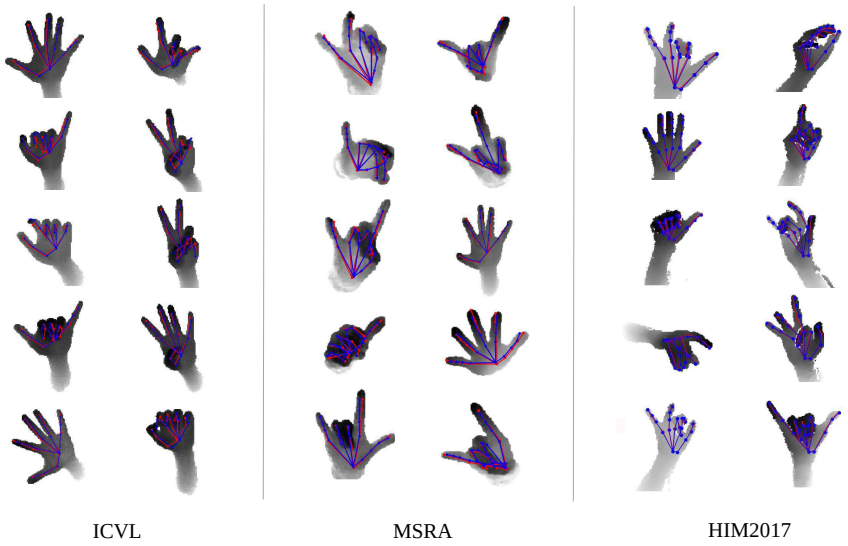

Fig. 5. The qualitative results on the MSRA, ICVL and HIM2017 Challenge dataset. The ground truth is marked in blue lines and the prediction is marked in red lines.

Cascade [27], MultiView [7], Pose-REN [3] and Global2Local [16]. Some reported results of previous works [30, 20, 39, 17] are calculated by their prediction available online. Other results [27, 16, 34, 3, 7] are calculated from the figures and tables of their papers.

Table 1. The mean Joint Error on the ICVL Dataset

Methods	Mean Joint Error (mm)
LRF	12.58
Deep Prior	11.56
Deep Model	10.4
Crossing Net	10.2
Cascade	9.9
Ours	**8.62**
V2V-PoseNet	6.28

Our network is evaluated on the ICVL dataset and compared with the state-of-the-art methods. As shown in Table 1, we get better results than Cascade but inferior to V2V-PoseNet. However, we use less data than them to train our method and the parameters complexity is much less than them. Fig. 6 shows the correct frame proportion on the ICVL dataset compared with Deep Prior [20], Deep Model [39], latent random forest (LRF) [30], Crossing Nets [34] and Cascade [27], where the horizontal axis represents the maximum allowed

distance to ground truth. In general, we achieve comparable performance with state-of-the-art methods on the ICVL dataset in standard evaluation metrics.

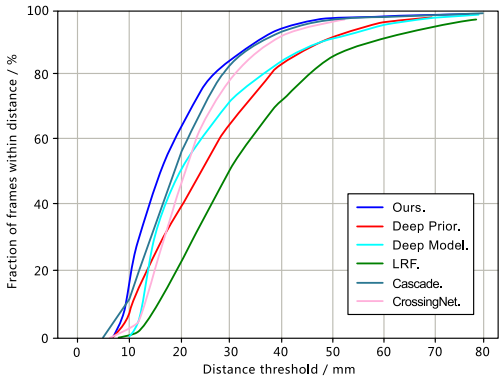

Fig. 6. Correct frame proportion on the ICVL dataset

On the MSRA dataset, we compared with Cascade [27], MultiView [7], Crossing Nets [34] and Global2Local [16] as shown in the left figure in Fig. 7. Global2Local [16] also uses a branch-like structure, but our method is quite different from them as described in Section 2. The result also proves that our three-branch anterior branch structure achieves a better performance.

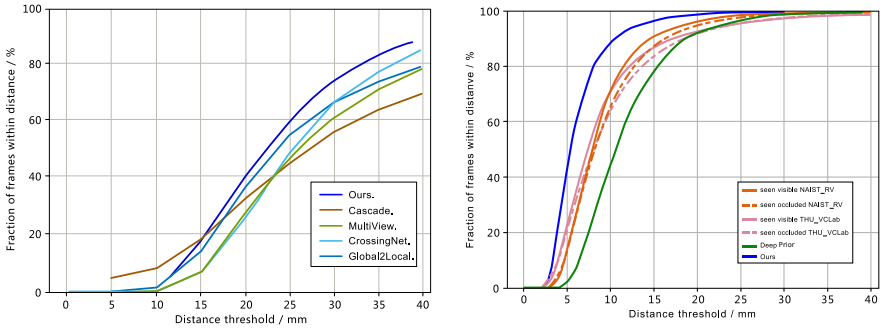

Fig. 7. Comparison with state-of-the-art methods. *Left:* Correct frame proportion on the MSRA dataset. *Right:* Correct frame proportion of SEEN subjects on the HIM2017 dataset. The curves of THU VCLab and NAIST RVLab are from [36]

We also implement our HBE network and Deep Prior network on HIM2017 Challenge dataset, and get the prediction results of all the joints on our test set.

Since our test set has the same size of the original test set, but only contains the SEEN subjects, we only compare the results of SEEN in the Challenge leaderboard. Table 2 shows the Challenge leaderboard and our comparison of mean joint error in millimeter. The right figure in Fig. 7 shows the correct frame proportion of SEEN subjects whose mean joint error within a certain value. The result of Pose-REN from the THU VCLab and the result of NAIST RVLab are from [36]. What we need to emphasize is that the comparison of the above is only an approximate comparison. In spite of this, it can be seen from the results that our method has a superior performance.

Table 2. The approximate comparison on the HIM2017 Challenge Dataset

Teams	Methods	AVG	**SEEN**	UNSEEN
mks0601	V2V-PoseNet	9.95	**6.97**	12.43
NVResearch & UMontreal	RCN+	9.97	**7.55**	12.00
NTU	3D CNN	11.30	**8.86**	13.33
THU VCLab	Pose-REN	11.70	**9.15**	13.83
NAIST RVLab	5-branch 3D CNN	11.90	**9.34**	14.04
Deep Prior	Deep Prior	- -	**9.63**	- -
Ours	**HBE**	**- -**	**5.26**	**- -**

4.5 Computational complexity

We take the HIM2017 Challenge dataset as an example to compare the computational complexity of the proposed HBE network and V2V-PoseNet. We train our network on a single GPU for 100 epochs taking 26250.24s (7.2h). The input generation and data pretreatment take 435s, and loading the input data takes 7.04s. In the testing stage, it takes 1.5ms for processing a frame.

Table 3 compares the computational complexity of our HBE network with V2V-PoseNet. We only use part of the original training set for training, while V2V-PoseNet uses the entire training set spending 6 days training including time-consuming I/O operations. With regard to the testing stage, we can achieve 673 fps on a single GPU, while V2V-PoseNet reaches 3.5 fps on a single GPU and 35 fps in a multi-GPU environment. Unlike them, we don't need to do voxel data conversion and epoch models ensemble for testing, and our network has a fast forwarding due to its simplicity. Besides, the number of parameters in our proposed method is much less than V2V-PoseNet regressing 3D coordinates. In summary, we use a much less training set and simpler network structure but reach the same level as their result even better than them. Our method is faster, more efficient and suitable for real-time applications.

Table 3. The comparison of computational complexity on the HIM2017 Challenge dataset

Items	Parameter Quantity	Testing on single GPU	Testing on muti-GPU
V2V-PoseNet	457.5M	3.5fps	35fps
Ours	**67.27M**	**673 fps**	**- -**

5 Conclusions

We propose a novel three-branch network called the Hand Branch Ensemble(HBE) network for 3D hand pose estimation from a single depth image. According to fingers activity space and functional importance we decompose the hand to three parts: the thumb, the index and the other fingers. Each branch corresponds to one part. The features of three branches are ensemble to predict all 3D joint locations. Our network is trained with a small amount of training data and evaluated on three challenging datasets. Both the training and testing time are quite short, and the experimental results demonstrate that our method outperforms the state-of-the-art methods on the HIM2017 Challenge dataset and achieves comparable performance on the ICVL and MSRA dataset. Our method has less complexity and can adapt to a large range of view-points and varied hand poses. Our proposed method provides a technical approach for tracking and analyzing the complex interaction between humans and environment.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: a system for large-scale machine learning. In: OSDI. vol. 16, pp. 265–283 (2016)
2. Braido, P., Zhang, X.: Quantitative analysis of finger motion coordination in hand manipulative and gestic acts. Human movement science **22**(6), 661–678 (2004)
3. Chen, X., Wang, G., Guo, H., Zhang, C.: Pose guided structured region ensemble network for cascaded hand pose estimation. arXiv preprint arXiv:1708.03416 (2017)
4. Cotugno, G., Althoefer, K., Nanayakkara, T.: The role of the thumb: study of finger motion in grasping and reachability space in human and robotic hands. IEEE Transactions on Systems, Man, and Cybernetics: Systems **47**(7), 1061–1070 (2017)
5. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: A review. Computer Vision and Image Understanding **108**(1-2), 52–73 (2007)
6. Garcia-Hernando, G., Yuan, S., Baek, S., Kim, T.K.: First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In: CVPR. vol. 1 (2018)
7. Ge, L., Liang, H., Yuan, J., Thalmann, D.: Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3593–3601 (2016)

8. Ge, L., Liang, H., Yuan, J., Thalmann, D.: 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. vol. 1, p. 5 (2017)
9. Guo, H., Wang, G., Chen, X., Zhang, C.: Towards good practices for deep 3d hand pose estimation. arXiv preprint arXiv:1707.07248 (2017)
10. Guo, H., Wang, G., Chen, X., Zhang, C., Qiao, F., Yang, H.: Region ensemble network: Improving convolutional network for hand pose estimation. In: Image Processing (ICIP), 2017 IEEE International Conference on. pp. 4512–4516. IEEE (2017)
11. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. pp. 448–456 (2015)
12. Keskin, C., Kıraç, F., Kara, Y.E., Akarun, L.: Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In: European Conference on Computer Vision. pp. 852–863. Springer (2012)
13. Khamis, S., Taylor, J., Shotton, J., Keskin, C., Izadi, S., Fitzgibbon, A.: Learning an efficient model of hand shape variation from depth images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2540–2548 (2015)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
15. Lin, Y., Sun, Y.: Robot grasp planning based on demonstrated grasp strategies. The International Journal of Robotics Research **34**(1), 26–42 (2015)
16. Madadi, M., Escalera, S., Baró, X., Gonzalez, J.: End-to-end global to local cnn learning for hand pose recovery in depth data. arXiv preprint arXiv:1705.09606 (2017)
17. Moon, G., Chang, J.Y., Lee, K.M.: V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In: CVPR. vol. 2 (2018)
18. Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., Theobalt, C.: Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In: Proceedings of International Conference on Computer Vision (ICCV) (2017), https://handtracker.mpi-inf.mpg.de/projects/OccludedHands/
19. Oberweger, M., Lepetit, V.: Deepprior++: Improving fast and accurate 3d hand pose estimation. In: ICCV workshop. vol. 840, p. 2 (2017)
20. Oberweger, M., Wohlhart, P., Lepetit, V.: Hands deep in deep learning for hand pose estimation. In: Computer Vision Winter Workshop (2015)
21. Oikonomidis, I., Kyriazis, N., Argyros, A.: Efficient model-based 3d tracking of hand articulations using kinect. bmvc 2011. In: BMVC (2011)
22. Qian, C., Sun, X., Wei, Y., Tang, X., Sun, J.: Realtime and robust hand tracking from depth. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1106–1113 (2014)
23. Rad, M., Oberweger, M., Lepetit, V.: Feature mapping for learning fast and accurate 3d pose inference from synthetic images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4663–4672 (2018)
24. Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., et al.: Accurate, robust, and flexible real-time hand tracking. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. pp. 3633–3642. ACM (2015)

25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
26. Sridhar, S., Mueller, F., Oulasvirta, A., Theobalt, C.: Fast and robust hand tracking using detection-guided optimization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3213–3221 (2015)
27. Sun, X., Wei, Y., Liang, S., Tang, X., Sun, J.: Cascaded hand pose regression. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 824–832 (2015)
28. Supancic, J.S., Rogez, G., Yang, Y., Shotton, J., Ramanan, D.: Depth-based hand pose estimation: data, methods, and challenges. In: Proceedings of the IEEE international conference on computer vision. pp. 1868–1876 (2015)
29. Susman, R.L.: Hand function and tool behavior in early hominids . Journal of Human Evolution **35**(1), 23–46 (1998)
30. Tang, D., Jin Chang, H., Tejani, A., Kim, T.K.: Latent regression forest: Structured estimation of 3d articulated hand posture. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3786–3793 (2014)
31. Taylor, J., Tankovich, V., Tang, D., Keskin, C., Kim, D., Davidson, P., Kowdle, A., Izadi, S.: Articulated distance fields for ultra-fast tracking of hands interacting. Acm Transactions on Graphics **36**(6), 1–12 (2017)
32. Tkach, A., Pauly, M., Tagliasacchi, A.: Sphere-meshes for real-time hand modeling and tracking. ACM Transactions on Graphics (TOG) **35**(6), 222 (2016)
33. Tompson, J., Stein, M., Lecun, Y., Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. ACM Transactions on Graphics (ToG) **33**(5), 169 (2014)
34. Wan, C., Probst, T., Van Gool, L., Yao, A.: Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2017)
35. Xu, C., Cheng, L.: Efficient hand pose estimation from a single depth image. In: Proceedings of the IEEE international conference on computer vision. pp. 3456–3462 (2013)
36. Yuan, S., Garcia-Hernando, G., Stenger, B., Moon, G., Chang, J.Y., Lee, K.M., Molchanov, P., Kautz, J., Honari, S., Ge, L., et al.: Depth-based 3d hand pose estimation: From current achievements to future goals. In: IEEE CVPR (2018)
37. Yuan, S., Ye, Q., Garcia-Hernando, G., Kim, T.K.: The 2017 hands in the million challenge on 3d hand pose estimation. arXiv preprint arXiv:1707.02237 (2017)
38. Yuan, S., Ye, Q., Stenger, B., Jain, S., Kim, T.K.: Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. pp. 2605–2613. IEEE (2017)
39. Zhou, X., Wan, Q., Zhang, W., Xue, X., Wei, Y.: Model-based deep hand pose estimation. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. pp. 2421–2427. AAAI Press (2016)