CrossInfoNet: Multi-Task Information Sharing Based Hand Pose Estimation

Kuo Du[1] Xiangbo Lin[2*] Yi Sun[2] Xiaohong Ma[2]
Dalian University of Technology, China
[1]dumyy2728@mail.dlut.edu.cn, [2]{linxbo,lslwf,maxh}@dlut.edu.cn

Abstract

This paper focuses on the topic of vision based hand pose estimation from single depth map using convolutional neural network (CNN). Our main contributions lie in designing a new pose regression network architecture named CrossInfoNet. The proposed CrossInfoNet decomposes hand pose estimation task into palm pose estimation sub-task and finger pose estimation sub-task, and adopts two-branch cross-connection structure to share the beneficial complementary information between the sub-tasks. Our work is inspired by multi-task information sharing mechanism, which has been few discussed in hand pose estimation using depth data in previous publications. In addition, we propose a heat-map guided feature extraction structure to get better feature maps, and train the complete network end-to-end. The effectiveness of the proposed CrossInfoNet is evaluated with extensively self-comparative experiments and in comparison with state-of-the-art methods on four public hand pose datasets. The code is available in[1].

1. Introduction

The research of vision based 3D hand pose estimation is a hotspot in the field of computer vision, virtual reality and robotics. It has been studied for decades and has made significant progress in recent years [3, 6, 19]. Nevertheless, it is still far from a solved problem due to the challenges of high joint flexibility, local self-similarity and severe occlusions. Different efforts have been made in vision based hand pose estimation. The input data changed from single RGB [2, 7], stereo RGB [24, 27], to depth maps which have made many achievements [26, 30, 39]. Recently, there seems to be a renewed interest to RGB images [24, 48, 18, 25]. The published hand pose estimation methods can be categorized into two main categories as either generative model-based [29, 35] or discriminative learning-based methods [11, 32, 36, 38]. Benefit from the increase of data amounts and computational ability, deep CNN has showed strong abilities and has become the leading method at present.

In 2017, Hands in the Million Challenge (HIM2017) [44] on depth maps based hand pose estimation attracted the attentions of many research teams. The issues discussed in the competition summary paper [43] are also our concerns.

Firstly, treating depth maps as 2D images and regressing 3D joint coordinates directly is a commonly used hand pose estimation pipeline. Although converting the 2.5D depth maps into 3D voxelized forms will reserve more information [12, 17], it suffers from heavy parameter loads and still exists information defect. In our work, we tend to be in line with the argument of [39] to leverage the advances of 2D CNNs, and try to excavate more information from 2D inputs.

Secondly, designing effective networks receives the most attentions. In machine learning, by sharing information, multi-task learning has the advantages of reserving more intrinsic information than single task learning. Learning multiple tasks simultaneously will be helpful to enforce a model with better generalizing ability [28]. However, multi-task learning has not been paid enough attention in CNN based hand pose estimation yet. As [39] claimed, they did the first attempt to fuse the hand pose estimation results of the holistic regression and the heat-map detection in a multi-task setup. Inspired by their achievements, we design a new CNN structure for hand pose estimation in a multi-task setup. Hierarchical model is one of hand pose estimation networks and has shown excellent performance in competition. It usually divides the pose estimation problem into sub-tasks by separately dealing with different fingers or different type of joints [4, 16, 47]. Intuitively, it would be easily understood that palm joints have closer tie-ups than those more flexible finger joints. The global hand pose will be mostly determined by the status of the palm joints, while the local hand pose will be reflected by the actions of the finger joints. Based on these knowledge, we design a new hierarchical model in a multi-task setup. The proposed architecture has two branches corresponding to the palm joint regression sub-task and the finger joint regression sub-task, respectively. By cross-connections between

[1]https://github.com/dumyy/handpose

the two branches, the noise in one branch becomes supplemental enhancement information in the other branch. This will help each branch to focus on its specific sub-task as is done in multi-task information sharing.

Thirdly, the output representations can be classified into the probability density map (heat-maps) or the 3D coordinates for each joint. Since the mapping between the 2D depth maps and the 3D joint coordinates is highly nonlinear, it will hamper the learning procedure and prevent the network from accurately estimating the joint coordinates. In contrast, the output representation with the heat-maps can provide more joint related information than a single joint location, which will help the network to get better feature maps. The analysis in [43] has concluded that the heat-map based method outperforms direct coordinate regression method. However, in heat-map based method, the final joint coordinates have usually to be inferred by maximum operation on the heat-maps. Maximum operation is non-differentiable, and it has to be tailored as a post-processing step, but not an end-to-end training. Taking into account of the advantages of the two representations, we propose a heat-map guided feature extraction network structure. In fact, our idea skillfully applies the multi-task parameter sharing.

In summary, for deep CNN based hand pose estimation from single depth map, our work has the following contributions:

• A new hand pose regression network in multi-task setup is proposed. It takes advantage of information sharing mechanism in multi-task learning. We use hierarchical model to decompose the final task into palm joint regression sub-task and finger joint regression sub-task. By branch cross-connection, the generated 'attention mask' guides one branch to focus on palm joint regression, and the other branch to focus on finger joint regression. Since the 'attention mask' enhances the sub-task features, the estimation accuracy is improved effectively.

• A heat-map guided feature extraction structure is proposed. It transfers more effective features from the heat-map detection task to the joint regressing task, without losing the end-to-end training advantage.

• We implement several baselines to investigate information sharing in a multi-task setup, which will provide valuable insights to this problem. We also carry out substantial experiments on commonly used datasets, and compare the performance with the state-of-the-art methods.

2. Related works

The achievements in vision based hand pose estimation are very rich. Since our work focuses on deep CNN based hand pose estimation from single depth map, we will limit the discussions to those works related closely with our work. Please refer to [8, 33, 43] for more comprehensive reviews.

Pose parameterization: The object of hand pose estimation is to find the joint coordinates. Directly regressing these coordinates is the natural choice in the models for output pose representation [4, 10, 12, 22, 23, 46]. However, since only one 3D coordinate for each joint has to be regressed from the input, the highly non-linear mapping between the input and the 3D coordinates output hampers the learning procedure. To cope with this problem, Tompson *et al*. [38] firstly utilized 2D heat-maps for each hand joint as the pose parameters and then translated them into 3D coordinates by post-processing. They found that the intermediate heat-maps representation not only reduced required learning capacity but also improved generalization performance. Ge *et al*. [11] extended this method by exploiting multi-view CNN to estimate 2D heat-maps for each view. Moon *et al*. [17] adopted 3D heat-maps as the hand pose parameters. Wan *et al*. [39] decomposed the pose parameters into 2D heat-maps, 3D heat-maps, and unit 3D directional vector fields. Then these different outputs were translated into 3D joint coordinates by a vote casting scheme with a variant of mean shift post-processing. Different from their schemes, our work uses 3D coordinate regression under heat-map constraints. Such strategy can help the model to learn a better feature map, and get accurate joint coordinates without the need of post-processing.

Model design: Designing a network according to human hand kinematics or morphology has received competitive results in recent years [44]. Structured methods embed physical hand motion constraints into the model or in the loss function [16, 31, 46]. Hierarchical models divide the pose estimation problem into sub-tasks according to the hand structure. Chen *et al*. [40] applied constraints per finger and joint-type (across fingers) in their multiple regions extraction step, each region containing a subset of joints. The extracted feature regions were then integrated hierarchically and the hand pose were regressed by utilizing an iterative cascaded method. Madadi *et al*. [16] designed a hierarchically structured CNN, using five branches to model each finger and an additional branch to model the palm orientation. The final layers of all branches were concatenated into one layer to predict all joints. Zhou *et al*. [47] designed a three-branch network according to different finger functions in daily manipulation, where one branch correlated with the thumb finger, one branch modeled the index finger, and the last branch represented the other three fingers. These hierarchical models have their distinctive characteristics. Here we explore a new two-branch model with one branch for palm joint regression and the other branch for finger joint regression. It is a common sense that the finger joints are more flexible than the palm joints. If we use two different parameter sets to represent relatively stable palm pose and flexible finger pose separately, the regression task

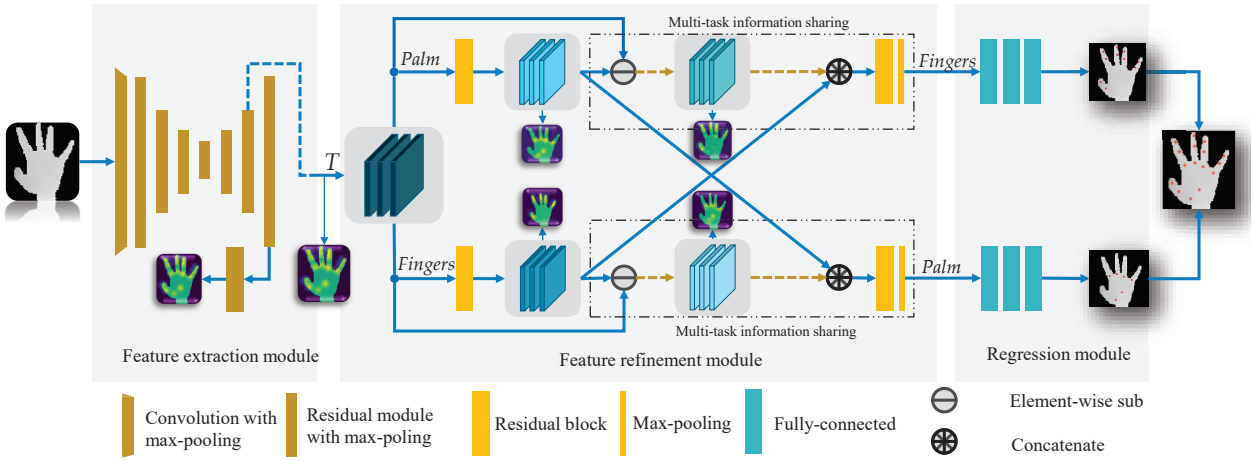


Figure 1. Overall network architecture with multi-task information sharing setup. We use the heat maps to highlight the regions with salient features in the first two modules. In the output part, the red dot represents the estimated joint positions.

will be implemented easier. In addition, we design cross-connections between the two branches, which helps each branch focus on its own task.

Multi-task information sharing: By sharing information between related tasks, multi-task learning will enable the model to generalize better on the tasks [28]. Multi-task learning in deep neural networks has led to success in many applications, such as in human pose estimation. Xia *et al*. [41] proposed to jointly solve the two tasks of human parsing and pose estimation. They trained two fully convolutional neural networks (FCNs), in which the estimated pose provided object-level shape prior to regularize part segments while the part-level segments constrained the variation of pose locations. Finally the two tasks were fused with a fully-connected conditional random field (FCRF). Nie *et al*. [20] proposed mutual learning to adapt model for joint human parsing and pose estimation. It effectively exploited mutual benefits by incorporating information from their counterparts, providing more powerful representations. Though effectively used in many applications, multi-task learning has not been paid enough attention in CNN based hand pose estimation yet. To our knowledge, [39] was the first one to give clear claim that they did hand pose estimation in a multi-task setup. In their work, they decomposed the hand pose parameters into 2D heat maps, 3D heat maps and unit 3D directional vector fields. The three representations were treated as three tasks, and were estimated via multi-task network cascades. Finally they fused these estimations by mean shift algorithm based post-processing. Our work is also built in multi-task learning framework, but it is quite different from [39]. We divide hand joints into two subsets, one set consisting of the palm joints, the other set consisting of the finger joints. The joint regression task is decomposed into palm joint regression sub-task and finger joint regression sub-task. By a cross-connection between the two sub-task regression branches, the information is shared.

3. Method

A hand is an articulated object and has high degree of freedom, and it is not easy to estimate hand pose accurately. In order to deal with the highly non-linear mapping from input depth data to output hand joint coordinates, the hand pose estimation problem can be simplified into sub-tasks, each of which is responsible of a sub-part or subset joint estimation. This is why designing hierarchical models to implement the task. Here we propose a new hierarchical model with an information sharing architecture named CrossInfoNet, as illustrated in Fig.1. The first part is the initial feature extraction module, where we integrate heat-maps as constraints to learn better feature maps and get all initial joint features. The second part is the feature refinement module, where the task is decomposed into two sub-tasks, one sub-task estimating the palm joints, the other sub-task estimating the finger joints. The information sharing strategy in this module guides the network to exploit useful clues from counterpart towards effectively improving the performance of hand pose estimation. The final part is the joint coordinate regression module.

We describe the details in the following sections. Section 3.1 describes the heat-map guided initial feature extraction module. Section 3.2 presents the baseline network with two independent sub-tasks without information sharing. Section 3.3 details how to provide complementary information by cross information sharing between the two sub-tasks. Loss function is introduced in Section 3.4 and implementation details are given in Section 3.5.

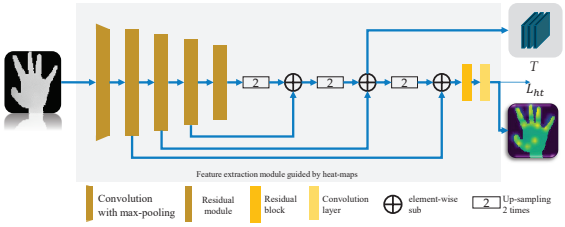


Figure 2. The initial feature extraction module. This network takes 2D depth map as input with the size of 96×96 and outputs the feature maps T with the size of 12×12. We use 2D heat maps with the size of 24×24 as supervision to guide the feature extractions.

3.1. Heat-map guided feature extraction

When a shallow CNN is used for feature extraction, the estimated results are usually not satisfactory. Given the problem, we design a novel feature extraction network with two stages, named as initial feature extraction module and feature refinement module. As for the initial feature extraction module, we choose the ResNet-50 [15] backbone network with four residual modules because it is highly efficient, as shown in Fig.2. In order to obtain more information, we apply the feature pyramid structure to merge different feature layers. We denote the feature maps for regressing initial joint locations as T. Different from previous heat-map based detection method, here the heat maps are only used as the constraints to guide the feature extraction and will not be passed to the subsequent module. The obtained feature map T with 256 channels will be input to the feature refinement module. The kernel size of the residual blocks is 3×3, and that of max-pooling layers is 2×2 with stride 2. We use a convolution layer with 3×3 filters to obtain the heat-map outputs for all joints.

3.2. Baseline feature refinement architecture

Some existing methods for hand pose estimation design tree-like branches, each of which is responsible for one independent sub-task, or extracts hand features from the output of one task to assist the other task at post-processing. They can neither extract powerful features nor strengthen the models. To fully utilize the extracted information, we proposed a novel feature refinement module based on multi-task information sharing. Before introducing our new multi-task feature refinement module, we first give the baseline multi-task architecture, which is illustrated in Fig.3.

Among all the joints, the palm joints have a smaller activity space compared to the finger joints, so the regressing complexity of the two parts is also different. If we use two different parameter sets to represent palm pose and finger pose, the hand pose would be regressed easier. Therefore, we separate the palm joint regression and finger joint regression into two independent branches. The feature maps

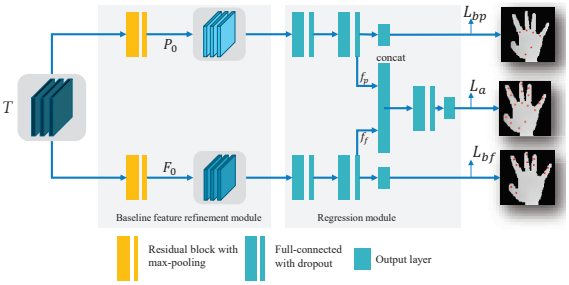


Figure 3. The baseline feature refinement module connected with the joint coordinate regression module. The kernel size of residual block is set to 3×3 and the dimension of full-connected layer is set to 2048.

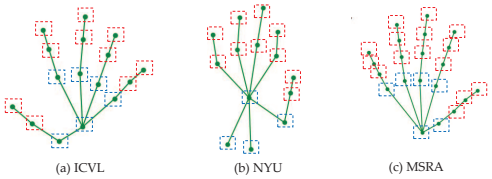
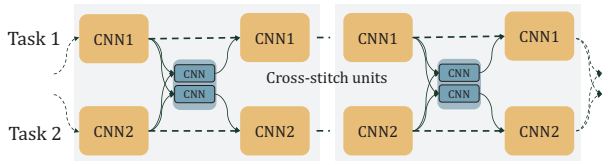


Figure 4. The palm joints subset (blue boxes) and the finger joints subset (red boxes) on different datasets.

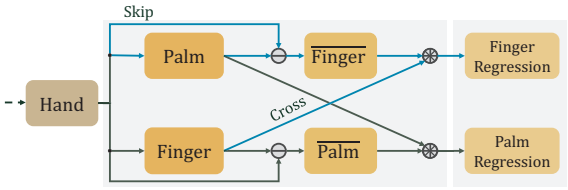
T from the initial feature extraction module are input to the residual block to extract more intrinsic local features of palm or fingers in different branch. Then the output of the full-connected layer f_p in the palm branch and f_f in the finger branch are concatenated to estimate all joint coordinates. We denote this architecture as the baseline network. Since ICVL, NYU and MSRA datasets have different label protocol, the joints subsets have some differences, as shown in Fig.4. The partition of HAND 2017 frame-based challenge dataset is the same as that of MSRA.

3.3. New feature refinement architecture

The baseline network only considers regressing palm and finger poses independently from each branch, which has no essential difference with the universal branch based network. There is little shared information between them, except the input features T. However, in the palm regression branch, there are residual finger features. These finger features may be noise for palm pose regression, but they are beneficial for finger pose regression. The same is in the finger branch. To make full use of the useful 'noise' information between the two branches, we try to design the network in a multi-task information sharing setup. Two-task Cross-stitch Network [28] is a universal multi-task network, as shown in Fig.5(a). It uses multiple cross-stitch units to leverage the knowledge of the other task by lazy fusion. Nevertheless, lazy cross-stitch may cause interference between sub-tasks, and lazy cross-stitch has no clear understanding of the sub-tasks – their similarity and rela-



(a) Cross-stitch network



(b) Our cross-connection model

Figure 5. Network comparisons with Cross-stitch Network

tionships.

We hope to actively guide how the sub-tasks should interact with each other. By guided information sharing, the features related to the same targets should be merged and enhanced. Fig.5(b) illustrates the proposed multi-task information sharing mechanism. It uses "skip line" to separate palm and fingers ($\overline{\text{Finger}}$) by subtracting the palm features from the global hand features, then uses cross line to concatenate the finger features from the two branches. It reduces the interference from the palm and enhances the finger features once more, and vice versa.

The detailed network structure is shown in Fig.6. Initial features T have palm related features and finger related features. By subtraction operation between T and palm pose dominated features P_0 via skip-connection, we get residual finger features F_-, regarded as finger 'attention mask'. This mask may be 'noise' for palm pose regression, but it will be beneficial for finger pose regression, which helps guide the branch to extract finer features. In the same way, we get the palm 'attention mask' P_-. By cross-connection, P_0 are concatenated with P_- and form the enhanced palm features P_1. The enhanced finger features F_1 are also obtained using the similar process. In this way, our new network architecture establishes associations between the different sub-tasks. The output features F_2 and P_2 are got from the followed residual block. In the end, the 3D hand joint coordinates are estimated through the final regression module. The network parameters are presented in Fig.6, and the main pose regression procedure is described in Algorithm 1.

3.4. Loss functions

We adopt the mean square error between the ground-truth and the estimated joint coordinates as the loss function. In the initial feature extraction module, we use a heat map as the constraint to guide the network for a better global

Algorithm 1 joint regression with multi-task information sharing.

Input:

Symbols:

$*$: spatial convolution operator

$\otimes$: feature concatenation operator

p_0, p_1 : convolutional layers for palm feature extraction in different stages

f_0, f_1 : convolutional layers for fingers feature extraction in different stages

f_c : Full-connected layers for regressing joint locations.

$T \in T^{w \times h \times c}$: regression feature

1: $P_0 = T * p_0$; $F_0 = T * f_0$ Preliminary features

2: $F_- = T - P_0$; $P_- = T - F_0$ Residual features

3: $P_1 = P_0 \otimes P_-$; $F_1 = F_0 \otimes F_-$ Enhanced features

4: $P_2 = P_1 * p_1$; $F_2 = F_1 * f_1$ The final features

5: $J_p = f_c(P_2)$; $J_f = f_c(F_2)$ The joint coordinates

6: $J = J_p \otimes J_f$ The final joint coordinates

Output: J

feature extraction, so the detection loss of heat-map is defined as:

$$L_{ht} = \sum_{n=1}^{A} \sum_{u,v} \left\| H_n^{a*}(u,v) - H_n^{a}(u,v) \right\|^2 \quad (1)$$

where A denotes the joint number of the whole hand. H_n^{a*} and H_n^{a} represent the ground-truth heat-map and estimated heat-map of joint n, respectively.

In the feature refinement module, we introduce two constraints, L_{bp} and L_{bf}, to extract the preliminary palm features P_0 and finger features F_0. They are defined as:

$$L_{bp} = \sum_{n=1}^{P} \sum_{u,v} \left\| H_n^{p*}(u,v) - H_n^{p}(u,v) \right\|^2 \quad (2)$$

$$L_{bf} = \sum_{n=1}^{F} \sum_{u,v} \left\| H_n^{f*}(u,v) - H_n^{f}(u,v) \right\|^2 \quad (3)$$

where H_n^{p*} and H_n^{f*} represent the ground-truth heat map of the nth palm joint and finger joint, respectively. H_n^{p} and H_n^{f} are the corresponding network outputs.

In the regression module, three losses are used to supervise the final outputs of each subtask and the total hand joints. They are palm joint regression loss L_{ep}, finger joint regression loss L_{ef}, and total hand joint regression loss L_a.

$$L_{ep} = \sum_{n=1}^{P} \left\| J_n^{p^*} - J_n^{p} \right\|_2^2 \quad (4)$$

$$L_{ef} = \sum_{n=1}^{F} \left\| J_n^{f^*} - J_n^{f} \right\|_2^2 \quad (5)$$

$$L_{a} = \sum_{n=1}^{A} \left\| J_n^{a^*} - J_n^{a} \right\|_2^2 \quad (6)$$

where J_n^{p*} and J_n^{p} denote the ground-truth and estimated 3D coordinates of the nth palm joint, J_n^{f*} and J_n^{f} are the

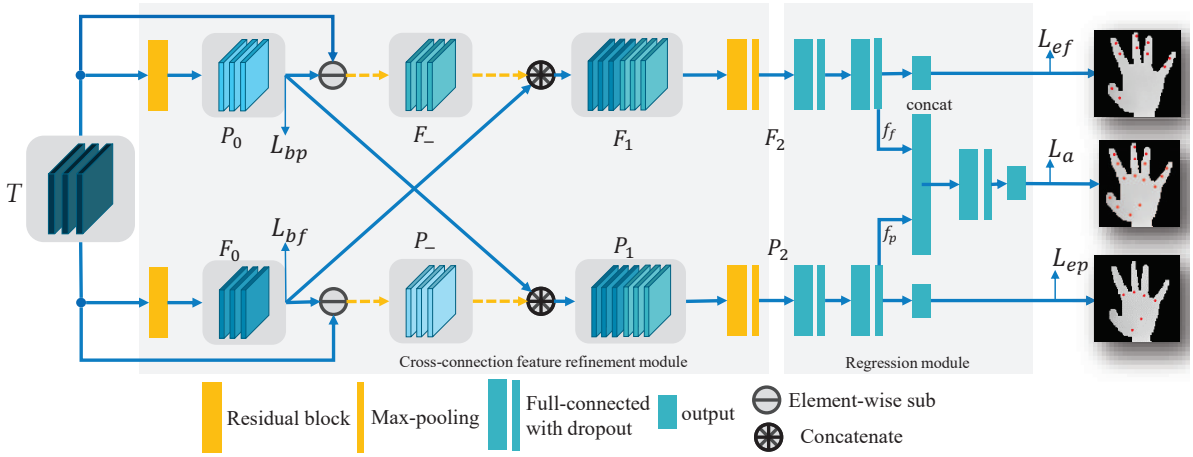


Figure 6. The new feature refinement module connected with the joint regression module. It is designed based on multi-task information sharing mechanism. The kernel size of the residual block is set to 3×3 and the dimension of full-connected layer is set to 2048.

ground-truth and estimated 3D coordinates of the nth finger joint, J_n^{a*} and J_n^a denote the ground-truth and estimated 3D coordinates of the nth hand joint.

The total loss function is:

$$L = \alpha \times (L_{ht} + L_{bp} + L_{bf}) + \beta \times (L_{ep} + L_{ef} + L_a) \quad (7)$$

where α, β are the factors to balance detection loss and regression loss. In our experiments, α and β are set to be 0.01 and 1, respectively.

3.5. Implementation details

A hand area is firstly cropped from the original image and resized to a fixed size of 96×96. The depth values within the cropped region are normalized to [-1, 1] and the labels are also normalized to keep the correspondence with the cropped depth map. We apply online data augmentation during training, including random rotation ([-180, 180] degree), translation ([-10, 10] pixel) and scaling ([0.9, 1.1]).

The proposed network is trained in an end-to-end manner. All weights are initialized from the zero-mean normal distribution with $\sigma = 0.01$. We choose Adam algorithm to train the model with an initial learning rate 1e-3, batch size 128 and weight decay 1e-5. The learning rate is reduced by a factor of 0.96 every epoch, and the dropout rate is set to be 0.6 to prevent over-fitting.

Our network is implemented by Tensorflow [1] and the RTX 2080 TI GPU is used for training and testing. We trained the model for 110 epochs. The training time of our model is 15 hours for ICVL dataset, 6.5 hours for NYU and MSRA datasets, and 3 days for HANDS 2017 challenge dataset, respectively. While testing, our model runs at 124.5 fps on a single GPU.

4. Experiments and results

4.1. Datasets and evaluation metrics

ICVL Dataset. The ICVL dataset [34] has 330K frames for training and 1.5k for testing. The training set consists of the genuine 22k frames and an additional 300K augmented frames with in-plane rotations. This dataset has 16 annotated joints. We use complete frames for training, while in the self-comparisons we only use the genuine 22k.

NYU Dataset. The NYU dataset [38] contains 72k training frames and 8k testing frames from three different views. The training set is collected from subject A, while the testing set is collected from subject A and B. Most previous works only used view 1 and 14 annotated joints for training and testing, we also use the same setup for comparison purposes.

MSRA Dataset. The MSRA dataset [32] consists of 76.5k depth images with 21 annotated joints. It has 9 subjects and 17 different gestures for each one. Following the common evaluation protocol [32], we also use the leave-one-subject-out method to evaluate the result.

HANDS 2017 Challenge Frame-based Dataset. This dataset [44] contains 957k training and 295k testing depth frames, which are sampled from BigHand2.2M [45] and FHAD [9] datasets. The training set has 5 subjects, while the testing set has 10 subjects, including 5 unseen subjects. This dataset has 21 annotated joints.

Evaluation Metrics. We use two metrics to evaluate the performance of different 3D hand pose estimation methods. One is the average 3D distance error between the ground truth and the predicted 3D joint location for each joint, the other is the percentage of success frames below a threshold, which is the same as [37].

Strategy	Average 3D distance error (mm)	
	ICVL	NYU
Base	9.28	11.17
Base+HM	9.08	10.84
Cross	8.79	10.57
Cross+HM	8.48	10.08

Table 1. Self-comparison results on average 3D distance error (mm). Base: baseline network without the heat-map constraints; Base + HM: baseline network with the heat-map constraints; Cross: cross-connection network without the heat-map constraints; Cross + HM: cross-connection network with the heat-map constraints.

4.2. Self-comparisons

We conduct ablation experiments on both ICVL[34] and NYU[38] datasets. To evaluate the advantages of the heat-map constraints, we compared the results of baseline network with or without heat-map constraints. To demonstrate the performance of the multi-task information sharing network, we compared it with baseline network.

As shown in Tab.1, the baseline network with heat-map constraints reduces the mean 3D distance error by 0.2mm (from 9.28 to 9.08) on the ICVL dataset and by 0.33mm (from 11.17 to 10.84) on the NYU dataset, compared to the one without heat-map constraints. It proves that the heat-map constraints enforce the model to get better features and the estimated errors decrease. Then based on the initial feature extraction network with heat-map constraints, we compared the effect of two different feature refinement modules on the average 3D distance error. The proposed model with cross-connection significantly lowers the errors by 0.60mm (from 9.08 to 8.48) on the ICVL dataset and by 0.76mm (from 10.84 to 10.08) on the NYU dataset, compared to the one in the baseline model with two separated branches. Obviously, the result of this comparative experiment supports our viewpoint that multi-task information sharing can get more accurate hand pose estimation.

Based on the comprehensive self-comparisons, it can be concluded that the proposed model with multi-task information sharing via cross-connected two-branch architecture and heat-map guided initial feature extraction, has the best performance in hand pose estimation.

4.3. Comparisons with state-of-the-art methods

We compared the performance of the proposed CrossInfoNet on three public 3D hand pose datasets with most of state-of-the-art methods, including methods using depth maps (2D) as inputs: latent random forest (LRF)[34], model-based method (DeepModel)[46], feedback loop training (Feedback) [23], Lie-X [42], DeepPrior with refinement (DeepPrior) [22], improved DeepPrior (DeepPrior++) [21], region ensemble network (Ren-4x6x6 [14], Ren-9x6x6 [40]), Pose-guided REN (Pose-Ren) [4], dense regression network (DenseReg) [39], and methods using point cloud or voxel (3D) as input: 3DCNN [12], SHPR-Net [5], HandPointNet [10], Point-to-Point [13], V2V [17]. The results of some methods used for comparisons are obtained from the online available prediction labels, others are extracted from their papers.

Methods	Mean error (mm)			Input
	ICVL	NYU	MSRA	
Feedback [23]	-	-	15.97	2D
Lie-X [42]	-	-	14.51	2D
LRF [34]	12.58	-	-	2D
DeepModel [46]	11.56	17.04	-	2D
DeepPrior [22]	10.4	19.73	-	2D
Ren-4x6x6 [14]	7.63	13.39	-	2D
Ren-9x6x6 [40]	7.31	12.69	9.7	2D
DeepPrior++ [21]	8.1	12.24	9.5	2D
Pose-Ren [4]	6.79	11.81	8.65	2D
DenseReg [39]	7.3	10.2	**7.2**	2D
CrossInfoNet(Ours)	**6.73**	**10.08**	7.86	2D
3DCNN [12]	-	14.1	9.6	3D
SHPR-Net [5]	7.22	10.78	7.76	3D
HandPointNet [10]	6.94	10.54	8.5	3D
Point-to-Point [13]	6.3	9.1	7.7	3D
V2V [17]	**6.28**	**8.42**	**7.59**	3D

Table 2. Comparisons with state-of-the-art methods on three datasets. Mean error indicates the average 3D distance error.

Methods	Testing on single GPU (fps)
V2V [17]	3.5
DenseReg [39]	27.8
Point-to-Point [13]	41.8
CrossInfoNet (Ours)	124.5

Table 3. Comparison of inference time while testing.

As shown in Tab.2 and Fig.7, our results outperform the results of the state-of-the-art methods whose input is a depth map on ICVL and NYU datasets. Compared to those methods using 3D inputs, our results are worse than V2V [17] and Point-to-Point [13], but have larger improvement than 3DCNN [12] and SHPR-Net [5]. For the MSRA dataset, our method gets comparable results with the best 3D CNN method. DenseReg [39] is better than our method on this dataset. Nevertheless, when the thresthod is below 10mm, our method is better on percentage of success frames metric. The qualitative results of our method on three datasets are shown in Fig.8.

Although on ICVL and NYU datasets, V2V and Point-to-Point methods with 3D input are better, and on MSRA dataset, DenseReg method with 2D input is better, they have a higher inference time on test data than our method. The

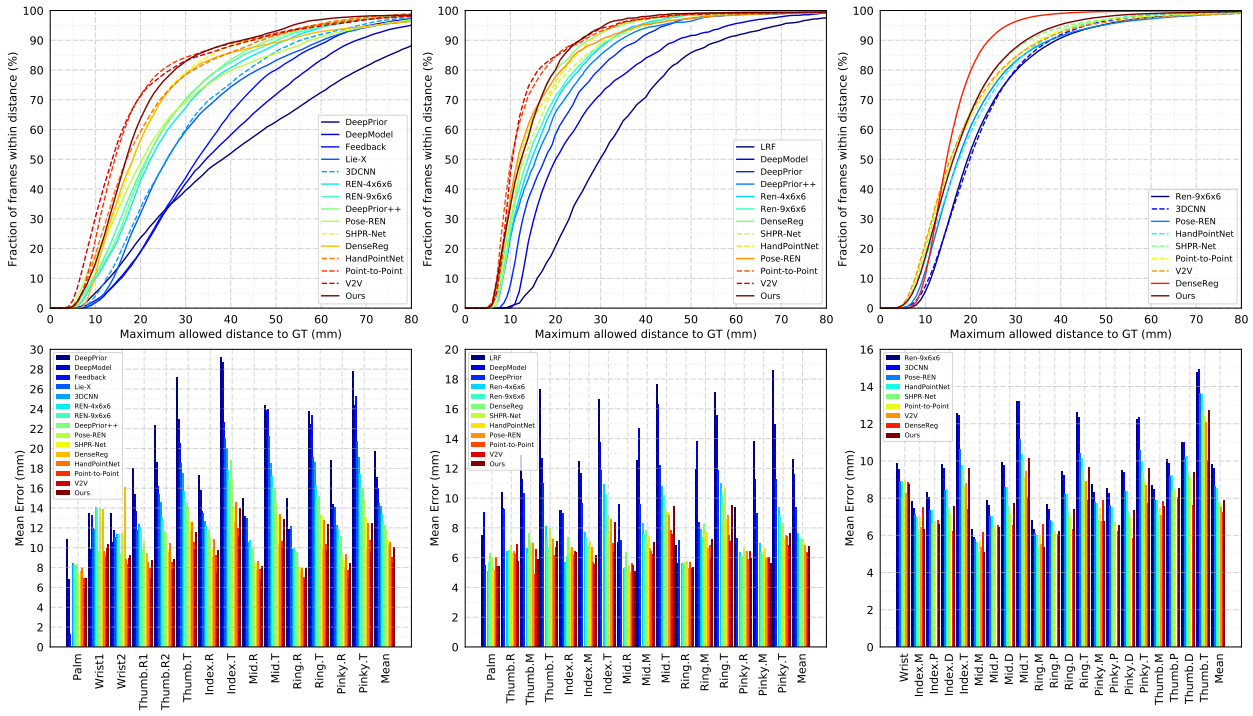


Figure 7. Comparisons with state-of-the-art methods. Top row: the percentage of good frames over different error thresholds. Bottom row: 3D distance errors per hand joints. Left: NYU [38] dataset. Middle: ICVL [34] dataset. Right: MSRA [32] dataset.

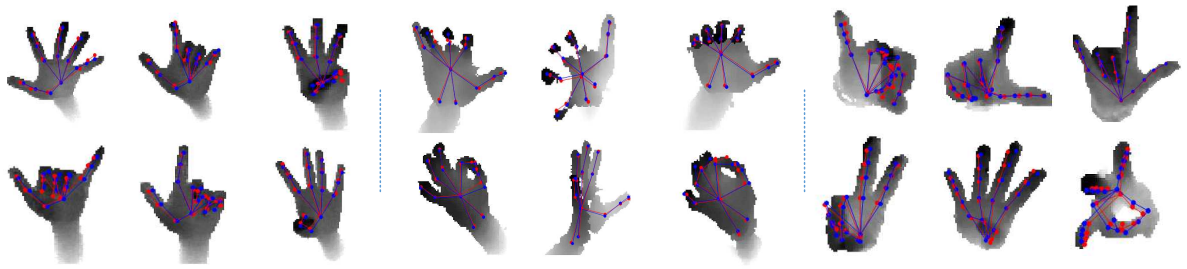


Figure 8. The qualitative results of our method on three datasets. Left: ICVL [34] dataset. Middle: NYU [38] dataset. Right: MSRA [32] dataset. Ground truth is shown in blue, and the estimated pose is shown in red.

comparisons about inference time are listed in Tab.3.

We also tested the performance of our method on the HANDS 2017 frame-based challenge dataset [44] on Feb.2, 2019. Our method won the first place, and had the best performance on the Unseen data.

5. Conclusion

Our work aims at exploring an effective CNN network to get the hand joint coordinates from depth data input. Our designed two-branch cross-connection network hierarchically regresses the palm pose and the finger pose by information sharing in a multi-task setup. It also uses heat-map guidance to get better feature maps. The experimental results prove that the proposed strategies are beneficial to get more accurate results, and the results of our method on three 3D hand pose datasets outperform most of previous works. Moreover, the proposed method also achieves the best result in the hand pose estimation challenge, compared to all previous participants. We hope this work can provide a new idea of network design for hand pose estimation.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[2] Vassilis Athitsos and Stan Sclaroff. Estimating 3d hand pose from a cluttered image. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–432. IEEE, 2003.

[3] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Augmented skeleton space transfer for depth-based hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8330–8339, 2018.

[4] Xinghao Chen, Guijin Wang, Hengkai Guo, and Cairong Zhang. Pose guided structured region ensemble network for cascaded hand pose estimation. *Neurocomputing*, 2018.

[5] Xinghao Chen, Guijin Wang, Cairong Zhang, Tae-Kyun Kim, and Xiangyang Ji. Shpr-net: Deep semantic hand pose regression from point clouds. *IEEE Access*, 6:43425–43439, 2018.

[6] Chiho Choi, Sang Ho Yoon, Chin-Ning Chen, and Karthik Ramani. Robust hand pose estimation during the interaction with an unknown object. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3123–3132, 2017.

[7] Martin de La Gorce, David J Fleet, and Nikos Paragios. Model-based 3d hand pose estimation from monocular video. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1793–1805, 2011.

[8] Ali Erol, George Bebis, Mircea Nicolescu, Richard D Boyle, and Xander Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1-2):52–73, 2007.

[9] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[10] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. Hand pointnet: 3d hand pose estimation using point sets. In *CVPR*, June 2018.

[11] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 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*, pages 3593–3601, 2016.

[12] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 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*, volume 1, page 5, 2017.

[13] Liuhao Ge, Zhou Ren, and Junsong Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *ECCV*, September 2018.

[14] Hengkai Guo, Guijin Wang, Xinghao Chen, Cairong Zhang, Fei Qiao, and Huazhong Yang. Region ensemble network: Improving convolutional network for hand pose estimation. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 4512–4516. IEEE, 2017.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Meysam Madadi, Sergio Escalera, Xavier Baró, and Jordi Gonzalez. End-to-end global to local cnn learning for hand pose recovery in depth data. *arXiv preprint arXiv:1705.09606*, 2017.

[17] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *CVPR*, June 2018.

[18] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[19] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of International Conference on Computer Vision (ICCV)*, volume 10, 2017.

[20] Xuecheng Nie, Jiashi Feng, and Shuicheng Yan. Mutual learning to adapt for joint human parsing and pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 502–517, 2018.

[21] Markus Oberweger and Vincent Lepetit. Deepprior++: Improving fast and accurate 3d hand pose estimation. In *ICCV Workshops*, Oct 2017.

[22] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Hands deep in deep learning for hand pose estimation. In *Computer Vision Winter Workshop*, 2015.

[23] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3316–3324, 2015.

[24] Paschalis Panteleris and Antonis Argyros. Back to rgb: 3d tracking of hands and hand-object interactions based on short-baseline stereo. *Hand*, 2(63):39, 2017.

[25] Paschalis Panteleris, Iasonas Oikonomidis, and Antonis A. Argyros. Using a single rgb frame for real time 3d hand pose estimation in the wild. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 436–445, 2018.

[26] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1106–1113, 2014.

[27] Romer Rosales, Vassilis Athitsos, Leonid Sigal, and Stan Sclaroff. 3d hand pose reconstruction using specialized mappings. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 378–385. IEEE, 2001.

[28] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[29] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, et al. Accurate, robust,

and flexible real-time hand tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3633–3642. ACM, 2015.

[30] Jamie Shotton, Ross Girshick, Andrew Fitzgibbon, Toby Sharp, Mat Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, Alex Kipman, et al. Efficient human pose estimation from single depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2821–2840, 2013.

[31] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 2, page 7, 2017.

[32] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[33] James S Supancic, Gregory Rogez, Yi Yang, Jamie Shotton, and Deva Ramanan. Depth-based hand pose estimation: data, methods, and challenges. In *Proceedings of the IEEE international conference on computer vision*, pages 1868–1876, 2015.

[34] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3786–3793, 2014.

[35] Danhang Tang, Jonathan Taylor, Pushmeet Kohli, Cem Keskin, Tae-Kyun Kim, and Jamie Shotton. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *Proceedings of the IEEE international conference on computer vision*, pages 3325–3333, 2015.

[36] Danhang Tang, Tsz-Ho Yu, and Tae-Kyun Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *Proceedings of the IEEE international conference on computer vision*, pages 3224–3231, 2013.

[37] Jonathan Taylor, Jamie Shotton, Toby Sharp, and Andrew Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 103–110. IEEE, 2012.

[38] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33(5):169, 2014.

[39] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Dense 3d regression for hand pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[40] Guijin Wang, Xinghao Chen, Hengkai Guo, and Cairong Zhang. Region ensemble network: towards good practices for deep 3d hand pose estimation. *Journal of Visual Communication and Image Representation*, 2018.

[41] Fangting Xia, Peng Wang, Xianjie Chen, and Alan L Yuille. Joint multi-person pose estimation and semantic part segmentation. In *CVPR*, volume 2, page 7, 2017.

[42] Chi Xu, Lakshmi Narasimhan Govindarajan, Yu Zhang, and Li Cheng. Lie-x: Depth image based articulated object pose estimation, tracking, and action recognition on lie groups. *International Journal of Computer Vision*, 123:454–478, 2017.

[43] Shanxin Yuan, Guillermo Garcia-Hernando, Björn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Liuhao Ge, et al. Depth-based 3d hand pose estimation: From current achievements to future goals. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[44] Shanxin Yuan, Qi Ye, Guillermo Garcia-Hernando, and Tae-Kyun Kim. The 2017 hands in the million challenge on 3d hand pose estimation. *arXiv preprint arXiv:1707.02237*, 2017.

[45] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2.2m benchmark: Hand pose dataset and state of the art analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[46] Xingyi Zhou, Qingfu Wan, Wei Zhang, Xiangyang Xue, and Yichen Wei. Model-based deep hand pose estimation. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2421–2427. AAAI Press, 2016.

[47] Yidan Zhou, Jian Lu, Kuo Du, Xiangbo Lin, Yi Sun, and Xiaohong Ma. Hbe: Hand branch ensemble network for real-time 3d hand pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 501–516, 2018.

[48] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.