Region Ensemble Network: Towards Good Practices for Deep 3D Hand Pose Estimation

Guijin Wang[a,*], Xinghao Chen[a,**], Hengkai Guo[a,**], Cairong Zhang[a]

[a] *Department of Electronic Engineering, Tsinghua University, Beijing 100084, China*

Abstract

3D hand pose estimation is an important and challenging problem for human-computer interaction. Recently convolutional networks (ConvNet) with sophisticated design have been employed to address it, but the improvement is not so significant. To exploit good practice and promote the performance for hand pose estimation, we propose a Region Ensemble Network (REN) for directly 3D coordinate regression. It first partitions the last convolutional outputs of ConvNet into several grid regions. Results from separate fully-connected (FC) regressors on each regions are integrated by another FC layer to perform estimation. By exploitation of several training strategies including data augmentation and smooth L_1 loss, REN significantly improves the performance of ConvNet for hand pose estimation. Experiments demonstrate that our approach achieves strong performance on par or better than state-of-the-art algorithms on three public hand pose datasets. We also experiment our methods on fingertip detection and human pose datasets and obtain state-of-the-art accuracy.

Keywords: Convolutional Network, Hand Pose Estimation, Human Pose Estimation, Fingertip Detection, Ensemble Learning, Depth Imaging

1. Introduction

3D hand pose estimation from depth images has drawn lots of attention from researchers [1] [2] [3] due to its important role in applications of augmented reality (AR) and human-computer interface (HCI) [4]. It aims to predict the accurate 3D positions for hand joints [5] from a single depth image[6] [7], which is critical for gesture recognition [8, 9, 10]. Though has been studied for several years [5], it is still challenging owing to high joint flexibility, large view variance, poor depth quality, severe self occlusion, and similar part confusion.

Recently, deep convolutional networks (ConvNets) have exhibited state-of-the-art performance across several computer vision tasks such as object classification [11], object detection [12], and image segmentation [13]. ConvNets have also been employed to solve the problem of hand pose estimation, often with complicated structure design such as multi-branch inputs [14][15] and multi-model regression [15] [16] [17] [18]. Thanks to the great modeling capacity and end-to-end feature learning, deep ConvNets have achieved competitive accuracy for methods [19] [3], which may result from the relatively shallow ConvNet structure (often 3 - 5 convolution layers [14] [16] [18]) and high risk of overfitting with relative small datasets compared to image classification.

In this paper, we explore multiple good practices with hand pose estimation in single depth images. Most importantly, inspired by model ensemble and multi-view voting [11, 26], we present a single deep ConvNet architecture named *Region Ensemble Net (REN)*[1] (Fig.1) to directly regress the 3D hand joint coordinates with end-to-end optimization and inference. We implement it by training individual fully-connected (FC) layers on multiple feature regions and combining them as ensembles. In addition, we adopt several approaches to enhance the performance including residual connection [20], data augmentation and smooth L_1 loss [21]. As shown in our experiments, REN significantly promotes the performance of our ConvNet, which outperforms state-of-the-art methods on three challenging hand pose benchmarks [14] [22] [19]. Evaluated on fingertip [14] and human pose benchmarks [23], our REN also achieves the best accuracy.

This paper builds upon our preliminary publication [24]. In this paper, we make extensions in following aspects: 1) Compared with [24], this paper describes more technical details and discusses several important factors for good practice, leading to slightly better results than [24] with different region settings. 2) We provide extensive ablation studies of region settings, fully-connected layers and model sizes. 3) We add results for one extra hand

[*] Corresponding author.

Email addresses: wangguijin@tsinghua.edu.cn (G. Wang), chenxh13@mails.tsinghua.edu.cn (X. Chen), guohengkaighk@gmail.com (H. Guo), zcr17@mails.tsinghua.edu.cn (C. Zhang)

[**] X. Chen and H. Guo are equally contributed to this work.

[1] Codes and models are available at `https://github.com/guohengkai/region-ensemble-network`

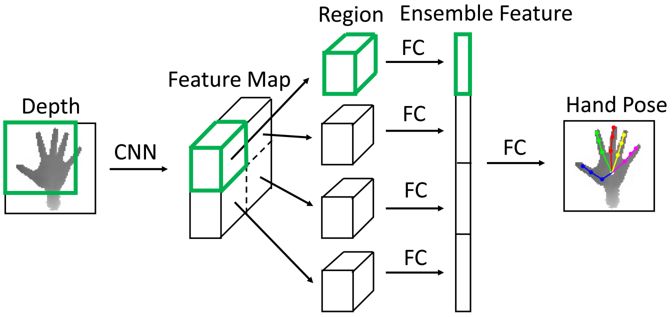

Figure 1: Region ensemble network (REN) with four regions: First deep ConvNet is used to extract features of depth image. The feature maps from ConvNet are then divided into regions. Each region is finally fed into fully-connected (FC) layers and then fused to predict the hand pose. The green rectangles represent the receptive field of the top-left region on the feature maps.

pose dataset [19], and further evaluate our REN for fingertip detection task on [14]. 4) We further apply our REN for depth-based human pose estimation with state-of-the-art performance, indicating the superiority of the proposed REN in articulated pose estimation tasks.

2. Related work

We briefly review relevant depth based hand pose estimation methods with ConvNets, and examine methodologies related to the proposed algorithm, including ensemble methods and multi-view testing for ConvNets. Finally we also introduce works using ConvNets for RGB-D fingertip detection and human pose estimation, which will be compared in our experiments.

2.1. Hand pose estimation with ConvNets

Recently deep ConvNets have been applied on hand pose estimation from depth images. Tompson et al. [14] first use ConvNets to produce 2D heat maps with multi-scale inputs and infer the 3D hand pose with inverse kinematics. Oberweger et al. [15] directly regress the 3D joint locations with multi-scale and multi-stage ConvNets using a linear layer as pose prior. In [16], a feedback loop is employed to iteratively correct the mistakes of inference, in which three ConvNets are used for pose initialization, image synthesis and pose updating. Ge et al. [17] employ three ConvNets from orthogonal views to separately regress 2D heat maps for each views with depth projections and fuse them to produce 3D hand pose. In [25], physical joint constraints are incorporated into a forward kinematics based layer in ConvNet. Similarly, Zhang et al. [18] embed skeletal manifold into ConvNets and train the model end-to-end to render sequential prediction.

2.2. Multi-model ensemble methods for ConvNets

Traditional ensemble learning means training multiple individual models and combining their outputs via averaging or weighted fusions, which is widely adopted in recognition competitions [26]. In addition to bagging [11] [27], boosting is also introduced for people counting [28]. However, using multiple ConvNets for both training and testing requires huge cost of memory and time, which is not practical for applications.

2.3. Multi-branch ensemble methods for ConvNets

Single ConvNet with the fusion of multiple branches can also be regarded as a generalized type of ensemble. One popular strategy is to fuse different scaling inputs [14] [15] or different image cues [29] [30] [31] with multi-input branches. Another approach is to employ multi-output branches with shared convolutional feature extractor, either training with different samples [32] or learning to predict different categories [33]. Compared with multi-input ensemble, multi-output methods cost less time because inference of FC layers is much faster than that of convolutional layers. Our method also falls into such category, but we apply ensemble on feature regions instead of inputs.

2.4. Multi-view testing for ConvNets

Multi-view testing is widely adopted to improve accuracy for object classification [11] [34] [35]. In [11], predictions from 10-crop (four corners and one center with horizontal flip) are averaged on single ConvNet. In [34] [35], fully-convolutional networks are employed in testing with multi-scale and multi-view inputs. Then spatially average pooling is applied on the class score map to obtain the final scores. To best of our knowledge, such strategy has not been applied on 3D pose regression yet[2].

2.5. RGB-D based fingertip detection and human pose estimation with ConvNets

Fingertips play an important role in human-computer interaction among the hand joints. Wetzler et al. [36] employ ConvNet for in-plane derotation of hand depth image and then use random forests or ConvNets for fingertip coordinate regression. Guo et al. [29] introduce a two-stream ConvNet to detect the 3D fingertips, which makes use of both depth information and edge information with slow fusion strategy.

Human pose estimation is also important for HCI applications such as action recognition [37] [38]. Though ConvNets are widely used in human pose estimation for RGB images [39] [40], there are limited number of works using ConvNets from depth images due to relatively small size of training datasets. Haque et al. [23] introduce a viewpoint invariant model using ConvNets and recurrent networks (RNNs) for human pose estimation. Local regions from depth images are transformed into a learned

[2] Please note the difference between the term “Multi-view testing”[35] and “Multi-view” methods (e.g. [17]). Multi-view testing is a widely used strategy at test time that extracts several cropped patches (e.g. 224 × 224) from the original image (e.g. 256 × 256) and averages the predictions from each patch. On contrast, the term “Multi-view” in [17] represents different images from several cameras on different viewpoints.

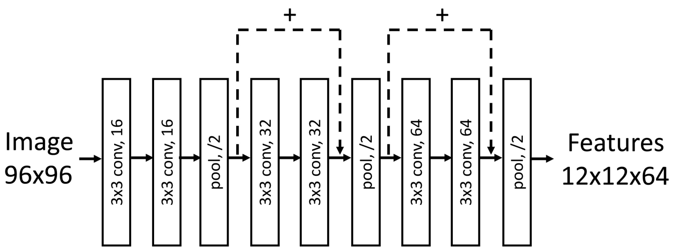

Figure 2: Structure of basic ConvNet for feature extraction. The ConvNet consists of six convolutional layers and three pooling layers. The dotted arrows represent residual connections with dimension increase [20]. The non-linear activation layers following each convolutional layers are not showed in the figure.

feature space via ConvNets and then RNNs are leveraged to predict the offsets of pose sequentially with multi-task setting.

3. Region Ensemble Network

As in Fig.1, Region Ensemble Network (REN) [3] starts with a ConvNet for feature extraction. Then the features are divided into multiple grid regions. Each region is fed into FC layers and learnt to fuse for hand pose estimation. In this section we introduce the basic network architecture, region ensemble structure and implementation details.

3.1. Network architecture with residual connection

The architecture of our ConvNet for feature extraction consists of six convolutional layers with 3×3 kernels (Fig.2) and three pooling layers with 2×2 kernels. Each convolutional layer is followed by a Rectified Linear Unit (ReLU) activation. The ConvNet accepts a 96×96 depth image as input and outputs the feature maps with dimension of $12 \times 12 \times 64$. To improve the learning ability, two residual connections are adopted between pooling layers with 1×1 convolution filters for dimension increase as in [20]. So there are totally eight convolutional layers in our model, which is deeper than ConvNets in [18] with five layers.

For regression, we use two 2048 dimension FC layers with dropout rate [41] of 0.5 for each regressor to avoid overfitting. The output of regressor is a $3 \times J$ vector representing the 3D world coordinates for hand joints, where J is the number of joints.

3.2. Region ensemble structure

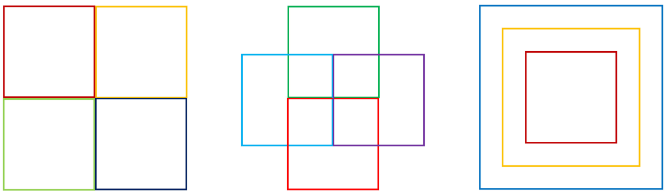

Figure 3: Different region setting for feature maps: four conners [24] (left), four centers in each edges (middle), and multi-scale regions with the same center (right). Proposed REN adopts nine regions with size of 6×6 including the center of feature maps and all the eight regions in left and middle figures.

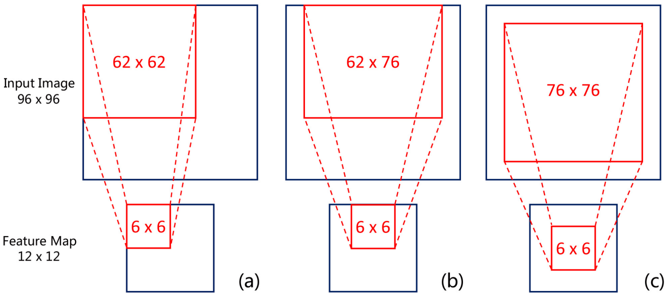

Figure 4: Receptive fields for different region positions: (a) 62×62 for conners, (b) 62×76 or 76×62 for centers in each edges, and (c) 76×76 for the center of feature maps.

Multi-view testing averages predictions from different crops of original input image, which reduces the variance for image classification [11]. Because image classification is invariant to translation and cropping, multi-view testing is easy to apply by directly cropping on the input image. When it comes to pose regression, each cropped parts will correspond to different hand pose configurations. So we should adapt the 3D coordinates of hand pose to the cropped view. Meanwhile, using multiple inputs to feed the ConvNet one-by-one is time-consuming.

Because each activation in the convolutional feature maps is contributed by a receptive field in the input image domain, we can project the multi-view inputs onto the regions of the feature maps. By using separate regions as features, we can train separate regressors instead of single regressor. So multi-view voting could be extended to regression task by utilizing each regions to separately predict the whole hand pose and then combining the results.

Based on this inspiration, we define a tree-structured network consisting of a single ConvNet trunk and several regression branches as shown in Fig.1. We first divide the feature maps of ConvNet into several regions. For each region, we feed it into the FC layers respectively as branches. There are several ways to combine different branches. A simple strategy is bagging, which averages all outputs of branches using average pooling. In order to boost the predictions from all the regions, we employ region ensemble strategy instead of bagging: features from the last FC layers of all regions are concatenated and used to infer the coordinates with an extra regression layer. The whole network can be trained end-to-end by minimizing the regression loss.

[3] We are aware that the term “ensemble” is commonly used to refer to methods that aggregate the results over several models. Though our proposed method has single model and strictly speaking it’s not an “ensemble” method, it’s inspired by model ensemble and that’s why we name it as “Region Ensemble Network”. In this paper we still keep the name of “Region Ensemble Network” to be consistent with our preliminary publication [24].

For region setting, we use nine regions with size of 6×6 located at four conners (left part in Fig. 3, which is also the whole setting in [24]), four centers near the edges (middle part in Fig. 3) and the center of the feature maps. The receptive fields of different regions within the 96×96 image bounding are shown in Fig. 4, which is similar to the corner and center crop in [11]. We will discuss the effect of different region settings on accuracy in Section 4.3.2.

There are three main differences between proposed methods and multi-view voting: 1) To our knowledge, all multi-view testing methods before are designed for image classification while our region ensemble can be applied on both classification and regression. By applying fusion FC layer in REN, different views of inputs are trained to simultaneously predict the same pose. 2) We adopt end-to-end training for region ensemble instead of testing only, making the ConvNet adjust the contributions from each views. 3) We replace the average pooling with one FC layer on concatenated features to learn the fusion parameters, which increases the learning ability of the network. We will perform the comparison in Section 4.3.3.

3.3. Implementation details

We implement our REN with Caffe [42] written in C++. We use stochastic gradient descent (SGD) with a mini-batch size of 128. The learning rate starts from 0.005 and is divided by 10 every 20 epochs, and the model is trained for total 80 epochs. In the meanwhile, we use a weight decay of 0.0005 and a momentum of 0.9. Our model is trained from scratch with random initialization [43]. Moreover, there are three important strategies for training: patch cropping, data augmentation, and smooth L_1 loss. The details are described below. And we will show the incremental contributions of these strategies in later section.

Patch cropping For ConvNet inputs, we extract a cube with fixed size of 150mm from the depth image centered in the hand region. Then the cube is resized into a 96×96 patch of depth values normalized to $[-1,1]$ as input for ConvNet. The 3D coordinates are also normalized to $[-1,1]$ according to the cube. To compute the center, we first segment the foreground with fixed thresholds and calculate the centroid of foreground.

Data augmentation We apply online data augmentation during training, including translation within $[-10,10]$ pixels, scaling within $[0.9,1.1]$ and rotation within $[-180,180]$ degrees. Random augmentation effectively increases the size of training dataset, so it can improve the generalization performance.

Smooth L_1 loss To deal with noisy annotations, we adopt similar smooth L_1 loss in [21]:

$$\mathrm{smooth}_{L_1}(x,\tilde{x}) = \begin{cases} 50(x-\tilde{x})^2 & \text{if}|x-\tilde{x}| < 0.01 \\ (|x-\tilde{x}|-0.005) & \text{otherwise} \end{cases} \quad (1)$$

where x is the predicted label and $\tilde{x}$ is the groundtruth. Because it is less sensitive to outliers than the L_2 loss, it can benefit the training of ConvNet.

4. Experiments

In this section, we first introduce the evaluation datasets and metrics for our experiments. Then we compare our REN with several state-of-the-art methods on public hand pose datasets. Next we explore several good practices of training ConvNets for hand pose estimation, discuss different region settings and also compare with traditional ensembles and multi-view testing. Finally we apply our REN on fingertip detection and human pose estimation for public benchmarks.

4.1. Experiment setup

4.1.1. Datasets

We first conduct our experiments on three public depth-based hand pose datasets: ICVL hand pose dataset [22], NYU hand pose dataset [14], MSRA hand pose dataset [19]. For self-comparison, ICVL dataset is mainly used. More details for these datasets are as follows:

ICVL dataset The training set of ICVL dataset contains 300K images with different rotations, and the testing set contains 1.6K images. All the depth images are captured by Intel RealSense. Totally 16 hand joints are initialized by the output of camera and manually refined.

NYU dataset The NYU dataset has 72K images for training and 8K for testing with 36 3D annotated joints, collected from Microsoft Kinect camera. Following [14], 14 hand joints with front-view image are used in experiments. And this dataset is also used to evaluate fingertip detection on the 5 fingertip joints in [36] [29].

MSRA dataset The MSRA dataset contains 9 subjects with 17 gestures for each subject. 76K depth images with 21 annotated joints are collected with Intel's Creative Interactive Camera. For evaluation, each subject is alternatively used as testing data when other 8 subjects are used for training. This is repeated 9 times and the average metrics are reported.

To further demonstrate the power of REN, we evaluate the fingertip detection task on NYU dataset [14]. What's more, without bells and whistles, we apply REN for depth-based human pose estimation task on ITOP human pose dataset [23], see Section 4.4. The details of ITOP dataset are as follow:

ITOP dataset The ITOP dataset consists 18K training images and 5K testing images for front view and top view acquired by two Kinect cameras. Each depth image is labelled with fifteen 3D joint locations of human body.

4.1.2. Evaluation metrics

We employ different metrics for hand pose estimation and human pose estimation following the literatures [22] [14] [23]. For hand pose, the performance is evaluated by two metrics: 1) **average 3D distance error** is computed as the average Euclidean distance for each joint (in millimeters). 2) **percentage of success frames** is defined as the rate of frames in which all Euclidean errors of joints are below a variant threshold [15]. In addition, mean precision (mP) with a threshold of 15mm as defined in [36] is calculated for fingertip detection.

For human pose, we compute the **mean average precision (mAP)** [23], which is defined as the average detected rate for all human body joints. A joint is counted as detected when the Euclidean distance between predicted position and ground truth is below 10cm.

4.2. Comparison with the state of the art

We compare our methods against several state-of-the-art approaches on ICVL dataset [22] [15] [19] [25] [3] [44], NYU dataset [14] [15] [16] [1] [17] [25] [18] [44] [45], and MSRA dataset [19] [46] [17] [3] [44] [45]. Overall, Fig.5 - Fig.7 show that proposed REN obtains the best accuracy among all the algorithms for hand pose estimation.

In details, on ICVL dataset our method surpasses other methods with a large margin. And the mean error $7.31mm$ obtains a $0.80mm$ decrease compared with LSN [3], which is a 9.87% relative improvement. Similarly on NYU dataset, our results are more accurate ($12.69mm$) than other approaches, and reduce the error of [18] by 10.3%. For MSRA dataset, our algorithm achieves similar performance with 3DCNN [45] and significantly outperforms all other state-of-the-art methods for nearly all thresholds, with an average error of $9.79mm$. Surprisingly, it reduces the mean error of [17] by 25.7%. Note that either LSN [3] or multi-view ConvNets [17] employ multiple models with complicated design and 3DCNN [45] uses 3D volumetric representation, while our REN only uses single model and 2D CNN without multi-stage regression, which indicates the power for the proposed region ensemble strategy. Fig. 9 shows some good cases and bad cases for all datasets. We can find that the failure cases are often caused by severe occlusion and bad depth images.

For MSRA dataset we also report the average joint errors distributed over all yaw and pitch viewpoint angles as in [19] and [17], shown in Fig. 8. On all angles our method outperforms [19, 17] and achieves comparable accuracy with 3DCNN [45]. Our method still get considerably good results for large viewpoints, which indicates the robustness for viewpoint variance.

4.3. Self-comparison

We perform self comparison experiments for different strategies and setting of region ensemble network on ICVL dataset [22].

4.3.1. Exploration study

In this section, we focus on the investigation of good practices. Specifically, we incrementally introduce five strategies on a basic shallow network in Fig. 10: 1) adding one convolution layer after each convolution layer to increase the depth of ConvNet. 2) adding residual connection edges across pooling layers as described in Section 3.1. 3) using smooth L_1 loss [21] instead of Euclidean L_2 loss for regression optimization. 4) augmenting the input patches as described in Section 3.3. 5) proposed region ensemble.

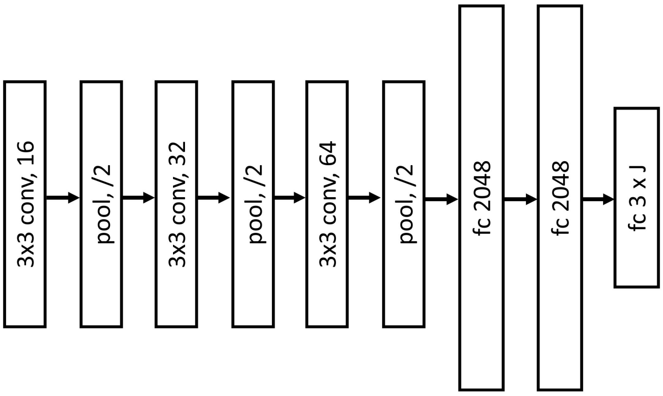

Figure 10: Structure of basic shallow ConvNet with three convolution layers and three pooling layers. The non-linear activation layers following each convolution layers are not showed in the figure.

Table 1: Average 3D distance error (mm) of incremental strategies on ICVL dataset [22]. Lower is better.

Strategy	Error(mm)
Shallow	10.48
+Deeper	10.02
+Residual Edge	9.73
+Smooth L_1 Loss	8.59
+Augmentation	8.36
+Region Ensemble	**7.31**

The experimental results are summarized in Table 1. Combining all the strategies reduces the errors by 3.17mm (relative 30.2%), which is a significant improvement of accuracy. Among them, L_1 loss and region ensemble are two most important factors for performance boosting, because L_1 loss is more suitable for labels with relative large noise and region ensemble can help improve the generalization for model.

Qualitative comparison on ICVL dataset are shown in Fig.11 for region ensemble (second row, corresponding to the sixth row in Table 1) and basic network (third row, corresponding to the fifth row in Table 1). The estimations are more accurate for region ensemble especially for fingers.

4.3.2. Region setting

According to the analysis in Section 3.2, different region partitions are equal to different patterns of multi-view inputs. Here we explore the effect of different settings of regions, including: **1) Multi-scale**: multi-scale regions with three regions of size 12×12, 8×8 and 4×4, which is similar to multi-scale inputs as in [14] [15]. **2) $\mathbf{4 \times 6 \times 6}$**: four regions of size 6×6 (left parts in Fig. 3), which is the setting in [24]. **3) $\mathbf{9 \times 6 \times 6}$**: nine regions of size 6×6 (four as left parts, four as middle parts in Fig. 3 and one in the center), which is the setting in this paper. **4) $\mathbf{9 \times 4 \times 4}$**:

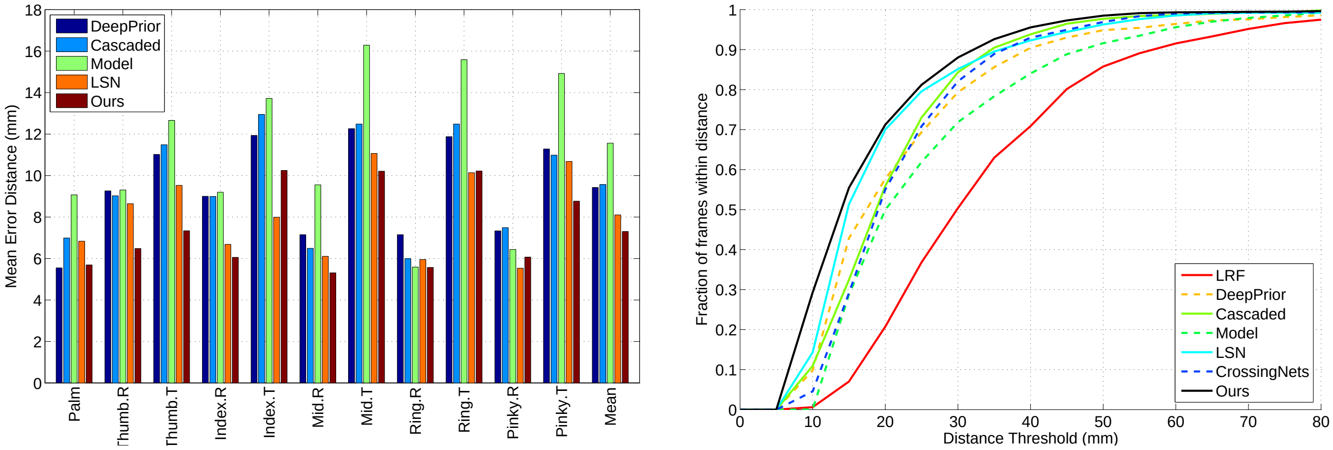

Figure 5: Comparison with state-of-the-arts (LRF [22], DeepPrior [15], Cascaded [19], Model [25], LSN [3] , CrossingNets [44]) on **ICVL** [22] dataset: distance error (left) and percentage of success frames (right).

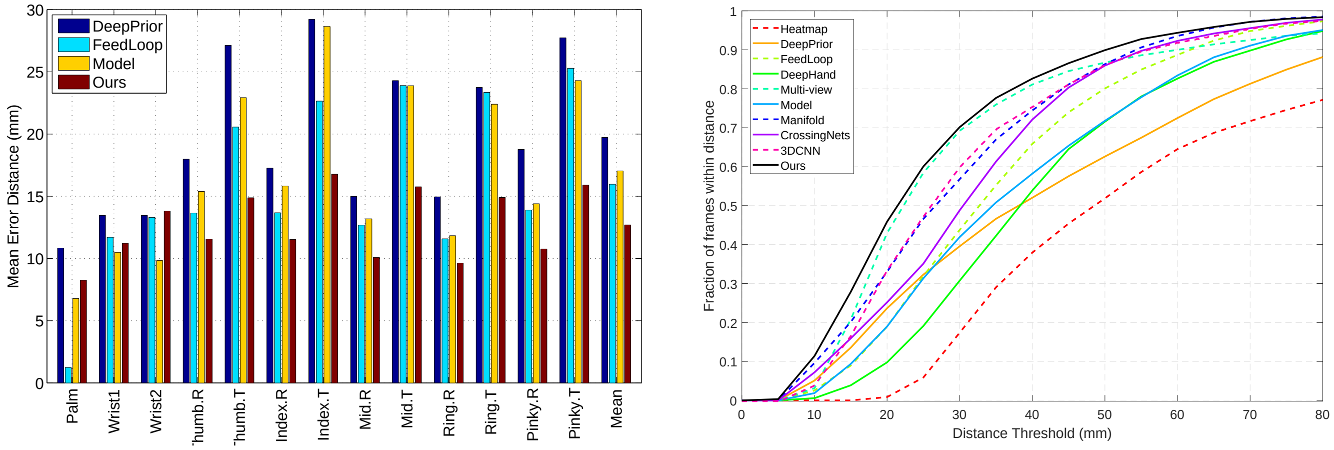

Figure 6: Comparison with state-of-the-arts (HeatMap [14], DeepPrior [15], Feedloop [16], DeepHand [1], Multi-view [17], Model [25], Manifold [18], CrossingNets [44], 3DCNN [45]) on **NYU** [14] datasets: distance error (left) and percentage of success frames (right).

nine regions of size 4×4 with similar positions as (3). **5) $9 \times 8 \times 8$**: nine regions of size 8×8 with similar positions as (3). **6) $9 \times 6 \times 6$+multi-scale**: Combination of nine regions of size 6×6 as (3) and multi-scale regions as (1), resulting in totally 12 regions.

Results for different settings can be seen in Fig. 12. Regions with same size are significantly more accurate than multi-scale regions due to the balance parameter number for FC layers of different regions. And more regions with moderate size (i.e. $9 \times 6 \times 6$) obtain slightly better performance. Too large or too small receptive field (i.e. $9 \times 4 \times 4$ and $9 \times 8 \times 8$) hurts the accuracy of hand pose estimation. Combination of $9 \times 6 \times 6$ and multi-scale regions does not further improve the performance. We empirically conclude that the regions setting with $9 \times 6 \times 6$ is a better choice for REN, with a good balance of receptive field and region number.

4.3.3. Comparison with ensembles and multi-view testing

We compare with traditional ensembles and multi-view testing in this section. In details, we implement three baselines: 1) *Basic* network has the same convolution structure in Fig.2 and single regressor on the full feature map with two FC layers of 2048 dimensions. 2) *Basic Bagging* network has nine basic networks as (1) that trained independently on the same data with different random order and augmentation. The average predictions of all the networks form the final prediction. 3) *Multi-view Testing* trains single basic network as (1) but averages the predicted 3D hand poses with nine multi-view inputs. The inputs are cropped as in Section 3.3, but on different centers with bias of $-d/0/d$mm on their x and y coordinates relative to the centroid. We use $d = 26.5625$ to approximately match the nine region positions in REN.

Results in Fig.13 shows that ensemble based methods (both basic bagging and region ensemble) are significantly more effective that baseline network. And the performance of our region ensemble is much better than traditional bagging. Because REN only employs multiple FC layers instead of multiple complete ConvNets, it also costs less time and memory than traditional bagging. Meanwhile, the improvement from multi-view testing is limited for hand pose

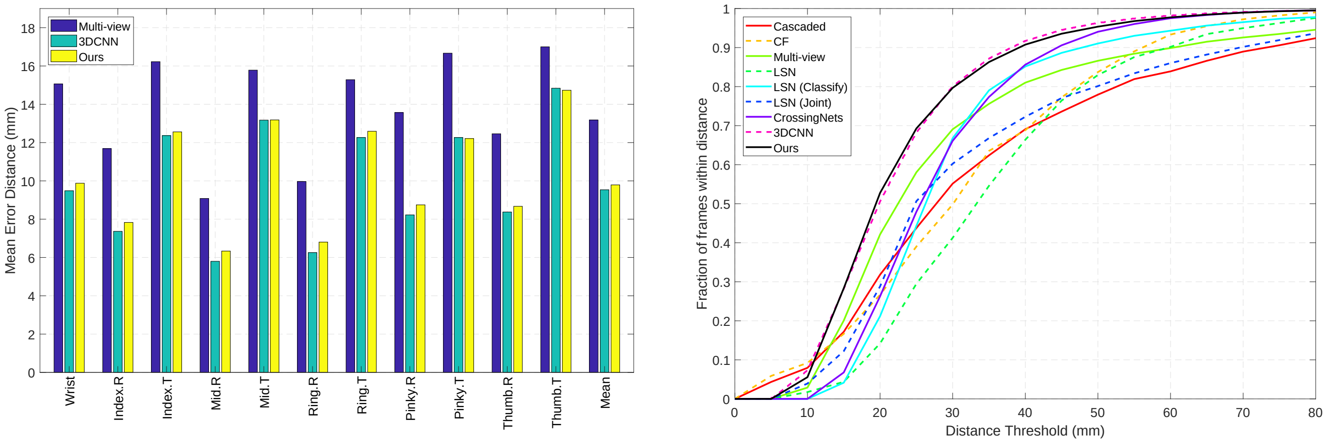

Figure 7: Comparison with state-of-the-arts (Cascaded [19], CF [46], Multi-view [17], LSN, LSN(Classify), LSN(joint) [3], CrossingNets [44], 3DCNN [45]) on **MSRA** [19] datasets: distance error (left) and percentage of success frames (right).

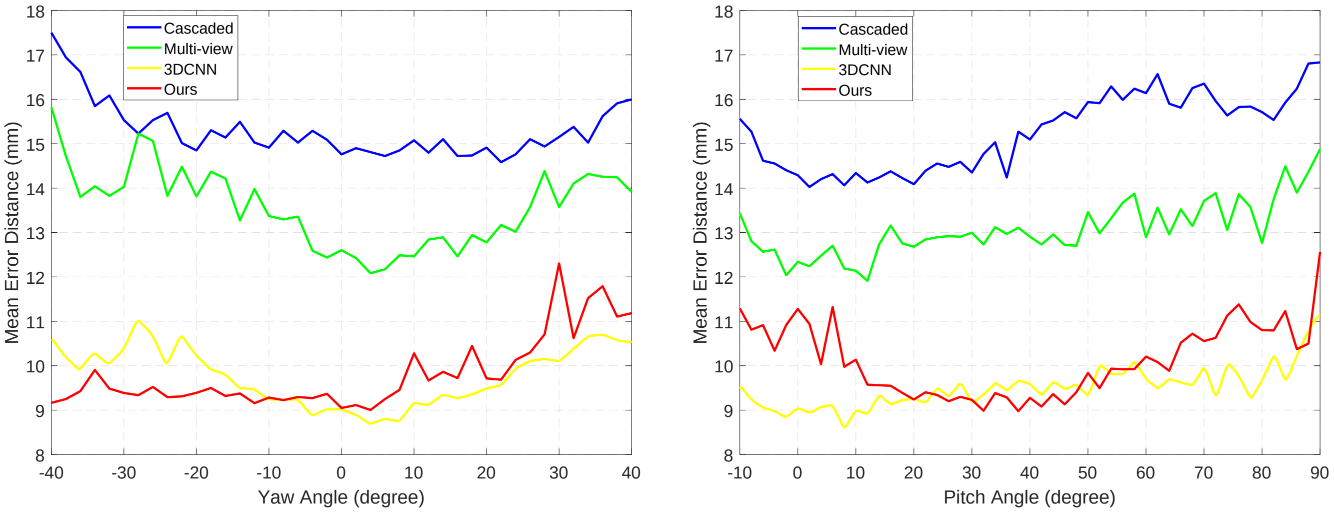

Figure 8: The average joint errors distributed over all yaw/pitch viewpoint angles on MSRA [19] dataset. Cascaded [19], Multi-view [17], 3DCNN [45].

estimation, because the model is more sensitive to translation in regression tasks than that in classification.

4.3.4. Model size and run time analysis

In this section we discuss how the number of parameters (model size) affect the accuracy and run time.

Firstly we will demonstrate that the performance improvement of proposed REN is not purely due to larger model size but actually due to our propose region ensemble strategy. We compare REN with a network that have more dimensions in FC layers so that it has the same model size with proposed REN. More specifically, when comparing to REN with $4\times6\times6$ region setting (REN-$4\times6\times6$), we use 8192 dimensions in the second FC layer in Basic Large network (denote this network as Basic Large-1). Similarly, when comparing to REN-$9\times6\times6$, we increase the dimensions of two FC layers as 4608 and 8192 (denote as Basic Large-1) respectively to ensure similar number of parameters to REN. The mean joint error on ICVL dataset of different methods and the corresponding model size are shown in Table 2. Larger model size for basic network can lead to slightly better performance. However, the proposed REN outperforms the basic large network that has same number of parameters to REN. REN-$4\times6\times6$ still perform much better than Basic-Large-2 even though REN has much smaller model size. These observations convince the contribution of the proposed region ensemble structure.

Furthermore, we also compare the run time performance of our method and the basic network. We run the experiments on a Nvidia Titan X GPU and report the forward time of different networks on GPU, as shown in Table 2. Our propose REN takes slightly more time but still sufficiently fast for real-time applications.

4.3.5. Influence of fully connected (FC) layer

Fully connected (FC) layers are critical for hand pose regression from convolutional features. In this section we

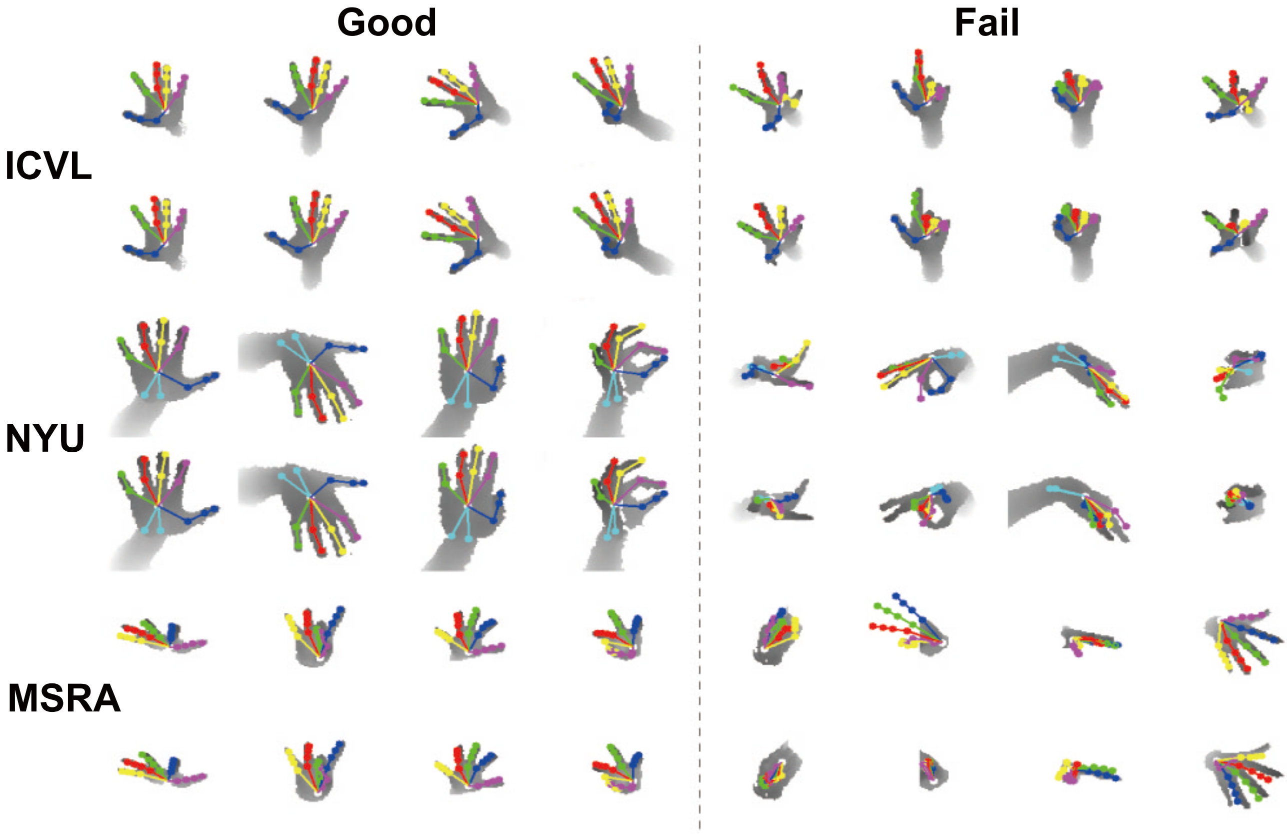

Figure 9: Example results on ICVL [22], NYU [14] and MSRA [19] datasets: ground truth (first row) and region ensemble network (second row) for each datasets.

Table 2: Model size and run time analysis.

Methods	Mean Error (mm)	Model Size (MB)	GPU time (ms)
Basic	8.36	93	0.21
Basic Large-1	8.18	137	0.22
REN-4 × 6 × 6	7.47	137	0.31
Basic Large-2	7.89	309	0.24
REN-9 × 6 × 6	7.31	309	0.47

will discuss how different configurations of FC layers (e.g. different number of feature channels, shared weights between different branches) affect the final performance of REN.

First we explore the influence of different dimensions for FC layers. As described in Section 3.1, we use two FC layers with 2048 nodes for each regressor. We change the dimensions of the FC layers to 512, 1024, 3072, 4096 respectively and compare the mean errors as well as the model sizes, as shown in Fig.14. Using FC layers with 512 dimensions results in quite small model size. Nevertheless, REN still get considerable good results which are even slightly better than several existing methods. This observation indicates that REN produces quite discriminative features that can enable good performance even with small dimensions of FC layers, which demonstrates the superiority of REN. Increasing the FC dimensions to a rather big scale does not further observably improve the performance, but dramatically increases the model size. To balance the model complexity and accuracy, we choose the dimension of FC layers as 2048.

Furthermore, we explore whether sharing the weights of FC layers in different branches of REN helps to improve the performance. As shown in Fig.14, sharing the weights of FC layers hinders the accuracy of hand pose estimation. This is likely due to the fact that different FC layers of each region focus on different aspects of the hand pose and implicitly impose constraints on hand pose estimation. Therefore, separately regressing on each regions using FC layers and integrating them in the following layer help to improve the performance.

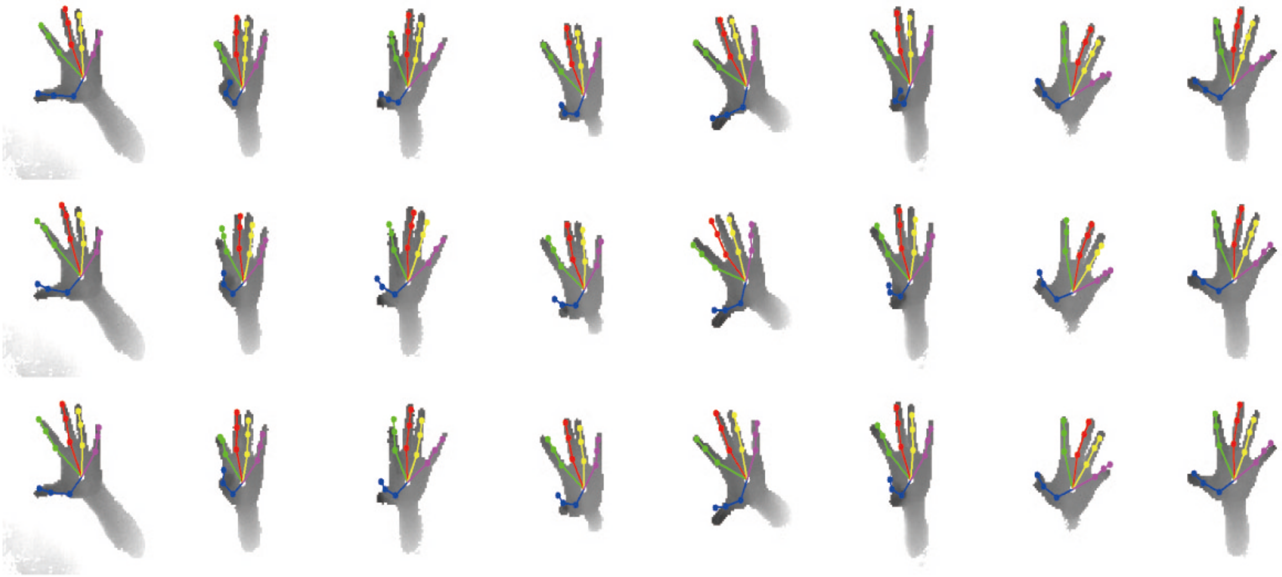

Figure 11: Example results on ICVL [22] dataset: ground truth (first row), basic network (second row, corresponding to the fifth row in Table 1), and region ensemble network (third row, corresponding to the seventh row in Table 1).

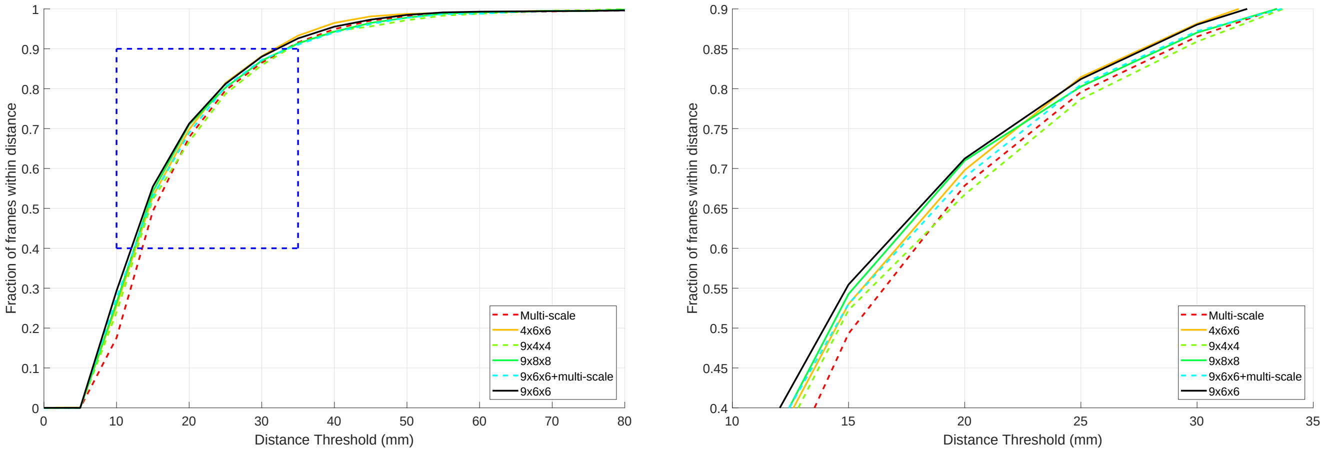

Figure 12: Comparison of different region settings (region number × region width × region height, multi-scale) for percentage of success frames on ICVL dataset [22]. The right figure is zoomed in from the blue region in the left figure.

4.4. *Evaluation on other tasks*

Here we also test our REN on challenging benchmarks for fingertip detection and human pose estimation and compare with state-of-the-art methods.

4.4.1. *Fingertip detection*

We compare the fingertip detection results to several state-of-the-art algorithms [36, 29] on NYU dataset without retraining our REN model. Table 3 illustrates that our REN achieves the best performance among all the methods, with an average error of 15.6mm.

4.4.2. *Human pose estimation*

The results for human pose estimation are reported in Table 4, where we compare our method with RTW [47] and REF [23] using mAP metric on ITOP dataet. For frontal view, proposed REN with 84.9 mAP significantly outperforms RTW and REF. And the accuracy for lower body is much higher. For top-down view, our method is better than RTW and shows comparable performance with

Table 3: Mean precision (mP) and average 3D distance error (mm) for fingertips of different methods on NYU dataset [14]. Higher is better for mP and lower is better for error.

Methods	mP	Error(mm)
CNN-DeROT [36]	0.63	-
DeepPrior [15]	0.43	26.4
FeedLoop [16]	0.38	23.2
TwoStream [29]	0.50	19.3
Model [25]	0.40	24.4
REN (Ours)	**0.66**	**15.6**

Table 4: Mean average precision (mAP, unit: %) of different methods on frontal view and top view of ITOP dataset [23] using a 10cm threshold. Higher is better.

	mAP (front-view)			mAP (top-view)		
Body Part	RTW [47]	REF [23]	REN (Ours)	RTW [47]	REF [23]	REN (Ours)
Head	97.8	98.1	98.7	98.3	98.1	98.2
Neck	95.8	97.5	99.4	82.2	97.6	98.9
Shoulders	94.1	96.6	96.1	91.8	96.1	96.6
Elbows	77.9	73.3	74.7	80.1	86.2	74.4
Hands	70.5	68.6	55.2	76.9	85.5	50.7
Torso	93.8	85.6	98.7	68.1	72.9	98.1
Hips	80.3	72.0	91.8	55.7	61.1	85.5
Knees	68.8	69.0	89.0	53.9	51.6	70.0
Feet	68.4	60.8	81.1	28.6	51.5	41.6
Mean	80.5	77.2	**84.9**	68.5	**75.5**	**75.5**

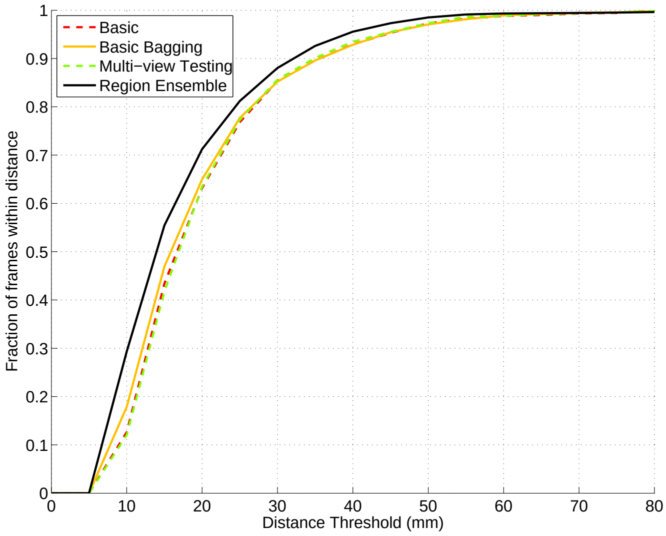

Figure 13: Comparison of ensembles and mutli-view testing for percentage of success frames on ICVL dataset [22].

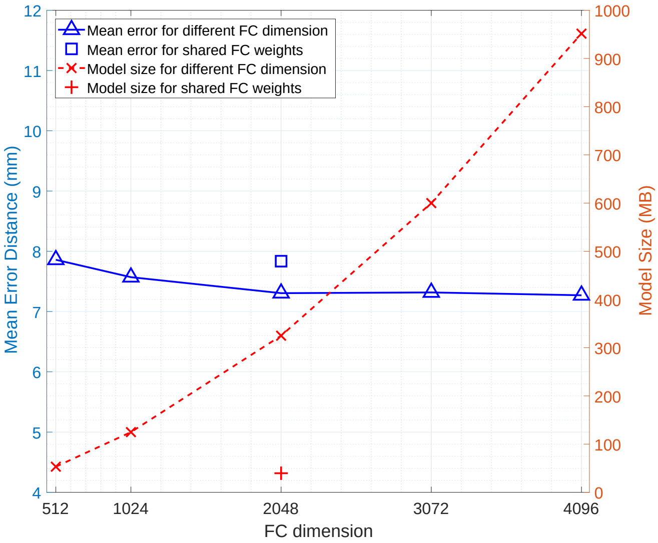

Figure 14: Comparison of different configurations of fully connected (FC) layer (dimensions, shared weights) on ICVL dataset [22].

REF, which contains deeper ConvNets with 16 convolution layers in their models. See Fig. 15 for some visualization results.

Implementation details For human pose, a small ConvNet is trained to predict the torso position as center. We extract a 3D cube with fixed size from the depth image and resize it into 96 × 96 image patch, similarly to the pre-processing procedure for hand pose. To better segment the body and remove useless objects in background, the size of cube is 800 × 1200 × 800mm^3 for front-view and 600 × 600 × 1000mm^3 for top-view. Note that we use ITOP-top-view and ITOP-side-view as separate datasets and conduct experiments on them respectively. For data augmentation, random flip of image with probability of 0.5 is also used.

5. Conclusion

To boost the performance of single ConvNet for 3D hand pose estimation, we exploit several good practices and present a simple but powerful region ensemble structure by dividing the feature maps into different regions and jointly training multiple regressors on all regions with fusion. Such strategies significantly improve the accuracy of ConvNet. The experimental results demonstrate that our method outperforms all the state-of-the-arts on three hand pose datasets and one human pose dataset. Since region ensemble is easy to be introduced into ConvNets, we believe that proposed structure could be applied on more computer vision tasks and achieve more promising results.

Acknowledgments

This work was partially supported by National High-Tech R&D Program (863 Program) of China (No. 2015AA016304).

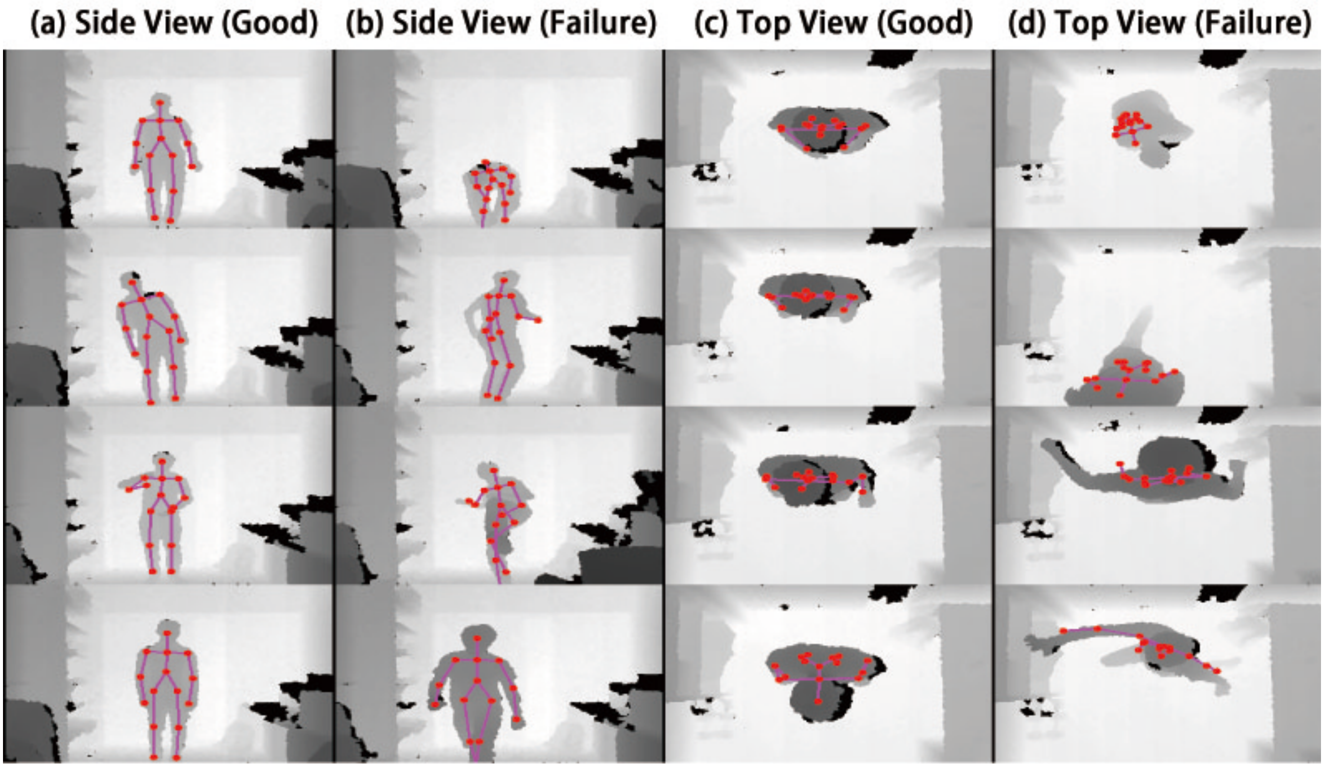

Figure 15: Example results on ITOP [23] dataset.

The authors are grateful to the reviewers for their constructive comments.

References

[1] A. Sinha, C. Choi, K. Ramani, Deephand: robust hand pose estimation by completing a matrix imputed with deep features, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[2] Q. Ye, S. Yuan, T.-K. Kim, Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation, in: European Conference on Computer Vision, 2016.

[3] C. Wan, A. Yao, L. Van Gool, Hand pose estimation from local surface normals, in: European Conference on Computer Vision, 2016.

[4] Y. Zhou, G. Jiang, Y. Lin, A novel finger and hand pose estimation technique for real-time hand gesture recognition, Pattern Recognition 49 (2016) 102–114.

[5] J. S. Supancic III, G. Rogez, Y. Yang, J. Shotton, D. Ramanan, Depth-based hand pose estimation: methods, data, and challenges, in: IEEE International Conference on Computer Vision, 2015.

[6] C. Shi, G. Wang, X. Yin, X. Pei, B. He, X. Lin, High-accuracy stereo matching based on adaptive ground control points, IEEE Transactions on Image Processing 24 (4) (2015) 1412–1423.

[7] G. Wang, X. Yin, X. Pei, C. Shi, Depth estimation for speckle projection system using progressive reliable points growing matching, Applied optics 52 (3) (2013) 516–524.

[8] X. Chen, C. Shi, B. Liu, Static hand gesture recognition based on finger root-center-angle and length weighted mahalanobis distance, in: SPIE Photonics Europe, International Society for Optics and Photonics, 2016, pp. 98970U–98970U.

[9] X. Chen, H. Guo, G. Wang, L. Zhang, Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition, in: Image Processing (ICIP), 2017 24th IEEE International Conference on, IEEE, 2017, pp. 2881–2885.

[10] Q. De Smedt, H. Wannous, J.-P. Vandeborre, Skeleton-based dynamic hand gesture recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 1–9.

[11] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[12] R. Girshick, J. Donahue, T. Darrell, J. Malik, Region-based convolutional networks for accurate object detection and segmentation, IEEE transactions on pattern analysis and machine intelligence 38 (1) (2016) 142–158.

[13] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, arXiv preprint arXiv:1606.00915.

[14] J. Tompson, M. Stein, Y. Lecun, K. Perlin, Real-time continuous pose recovery of human hands using convolutional networks, ACM Transactions on Graphics 33 (5) (2014) 169.

[15] M. Oberweger, P. Wohlhart, V. Lepetit, Hands deep in deep learning for hand pose estimation, Computer Vision Winter Workshop.

[16] M. Oberweger, P. Wohlhart, V. Lepetit, Training a feedback loop for hand pose estimation, in: IEEE International Conference on Computer Vision, 2015.

[17] L. Ge, H. Liang, J. Yuan, D. Thalmann, Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[18] Y. Zhang, C. Xu, L. Cheng, Learning to search on manifolds for 3d pose estimation of articulated objects, arXiv preprint arXiv:1612.00596.

[19] X. Sun, Y. Wei, S. Liang, X. Tang, J. Sun, Cascaded hand pose regression, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 824–832.

[20] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, arXiv preprint arXiv:1512.03385.

[21] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.

[22] D. Tang, H. J. Chang, A. Tejani, T.-K. Kim, Latent regression forest: Structured estimation of 3d articulated hand posture, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2014, pp. 3786–3793.

[23] A. Haque, B. Peng, Z. Luo, A. Alahi, S. Yeung, L. Fei-Fei, Towards viewpoint invariant 3d human pose estimation, in: European Conference on Computer Vision, Springer, 2016, pp. 160–177.

[24] H. Guo, G. Wang, X. Chen, C. Zhang, F. Qiao, H. Yang, Region ensemble network: Improving convolutional network for hand pose estimation, in: IEEE International Conference on Image Processing, IEEE, 2017.

[25] X. Zhou, Q. Wan, W. Zhang, X. Xue, Y. Wei, Model-based deep hand pose estimation, in: IJCAI, 2016.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision (IJCV) 115 (3) (2015) 211–252. `doi:10.1007/s11263-015-0816-y`.

[27] Y. Sun, X. Wang, X. Tang, Deep convolutional network cascade for facial point detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3476–3483.

[28] E. Walach, L. Wolf, Learning to count with cnn boosting, in: European Conference on Computer Vision, Springer, 2016, pp. 660–676.

[29] H. Guo, G. Wang, X. Chen, Two-stream convolutional neural network for accurate rgb-d fingertip detection using depth and edge information, arXiv preprint arXiv:1612.07978.

[30] H. Li, Y. Li, F. Porikli, Deeptrack: Learning discriminative feature representations online for robust visual tracking, IEEE Transactions on Image Processing 25 (4) (2016) 1834–1848.

[31] X. Chen, G. Wang, H. Guo, Accurate fingertip detection from binocular mask images, in: Visual Communications and Image Processing (VCIP), 2016, IEEE, 2016, pp. 1–4.

[32] H. Li, Y. Li, F. Porikli, Convolutional neural net bagging for online visual tracking, Computer Vision and Image Understanding (2016) 120–129.

[33] K. Ahmed, M. H. Baig, L. Torresani, Network of experts for large-scale image categorization, arXiv preprint arXiv:1604.06119.

[34] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: Integrated recognition, localization and detection using convolutional networks, International Conference on Learning Representations (ICLR).

[35] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE transactions on pattern analysis and machine intelligence 37 (9) (2015) 1904–1916.

[36] A. Wetzler, R. Slossberg, R. Kimmel, Rule of thumb: Deep derotation for improved fingertip detection, arXiv preprint arXiv:1507.05726.

[37] J. Zhang, W. Li, P. O. Ogunbona, P. Wang, C. Tang, Rgb-d-based action recognition datasets: A survey, Pattern Recognition 60 (2016) 86–105.

[38] H. Chen, G. Wang, J.-H. Xue, L. He, A novel hierarchical framework for human action recognition, Pattern Recognition 55 (2016) 148–159.

[39] J. Carreira, P. Agrawal, K. Fragkiadaki, J. Malik, Human pose estimation with iterative error feedback, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4733–4742.

[40] A. Bulat, G. Tzimiropoulos, Human pose estimation via convolutional part heatmap regression, in: European Conference on Computer Vision, Springer, 2016, pp. 717–732.

[41] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting., Journal of Machine Learning Research 15 (1) (2014) 1929–1958.

[42] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, arXiv preprint arXiv:1408.5093.

[43] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.

[44] C. Wan, T. Probst, L. Van Gool, A. Yao, Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 680–689.

[45] L. Ge, H. Liang, J. Yuan, D. 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, 2017, pp. 1991–2000.

[46] C. Choi, A. Sinha, J. Hee Choi, S. Jang, K. Ramani, A collaborative filtering approach to real-time hand pose estimation, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2336–2344.

[47] H. Yub Jung, S. Lee, Y. Seok Heo, I. Dong Yun, Random tree walk toward instantaneous 3d human pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2467–2474.