IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 21, 2020, accepted June 1, 2020, date of publication June 11, 2020, date of current version June 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3001637

Fast and Accurate 3D Hand Pose Estimation via Recurrent Neural Network for Capturing Hand Articulations

CHEOL-HWAN YOO, SEOWON JI, YONG-GOO SHIN, SEUNG-WOOK KIM, (Member, IEEE), AND SUNG-JEA KO, (Fellow, IEEE)
School of Electrical Engineering, Korea University, Seoul 136-713, South Korea

Corresponding author: Sung-Jea Ko (sjko@korea.ac.kr)

This work was supported by the Institute for Information & Communications Technology Promotion (IITP) funded by the Korea Government through the Ministry of Science and Information and Communication Technology (MSIT) (Intelligent Defense Boundary Surveillance Technology Using Collaborative Reinforced Learning of Embedded Edge Camera and Image Analysis) under Grant 2017-0-00250.

ABSTRACT 3D hand pose estimation from a single depth image plays an important role in computer vision and human-computer interaction. Although recent hand pose estimation methods using convolution neural network (CNN) have shown notable improvements in accuracy, most of them have a limitation that they rely on a complex network structure without fully exploiting the articulated structure of the hand. A hand, which is an articulated object, is composed of six local parts: the palm and five independent fingers. Each finger consists of sequential-joints that provide constrained motion, referred to as a kinematic chain. In this paper, we propose a hierarchically-structured convolutional recurrent neural network (HCRNN) with six branches that estimate the 3D position of the palm and five fingers independently. The palm position is predicted via fully-connected layers. Each sequential-joint, i.e. finger position, is obtained using a recurrent neural network (RNN) to capture the spatial dependencies between adjacent joints. Then the output features of the palm and finger branches are concatenated to estimate the global hand position. HCRNN directly takes the depth map as an input without a time-consuming data conversion, such as 3D voxels and point clouds. Experimental results on public datasets demonstrate that the proposed HCRNN not only outperforms most 2D CNN-based methods using the depth image as their inputs but also achieves competitive results with state-of-the-art 3D CNN-based methods with a highly efficient running speed of 285 fps on a single GPU.

INDEX TERMS 3D hand pose estimation, recurrent neural network, hand articulations.

I. INTRODUCTION

Accurate 3D hand pose estimation has received considerable attention regarding a wide range of applications, such as virtual/augmented reality and human-computer interaction [1]. As commercial depth cameras have been released and become more common, depth-based hand pose estimation methods have attracted significant research interest in recent decades [2]. Nevertheless, it is still a challenging problem to accurately estimate 3D hand pose, because of the low quality of depth images, large variations in hand orientations, high joint flexibility, and severe self-occlusion.

Recently, most 3D hand pose estimation methods have been based on convolutional neural networks (CNNs) with a single depth image. In these conventional CNN-based methods, there are two major approaches to improve estimation accuracy. The first involves 3D data representation of 2D depth images. To utilize 3D spatial information, Ge *et al.* and Moon *et al.* converted a depth image into a volumetric representation, such as 3D voxels, and then applied a 3D CNN for 3D hand pose estimation [3]–[5]. In addition, 3D representation of inputs based on a 3D point cloud has been proposed [6]–[8]. Although these methods are effective for capturing the geometric properties of depth images [3], [5], they suffer from heavy parameters and complex data-conversion processes, resulting in high time complexity. For efficient training and testing, 2D CNN-based methods that attempt to extract more information from 2D inputs are still being widely researched. In this study, we adopt a 2D depth image itself as input without a

The associate editor coordinating the review of this manuscript and approving it for publication was Genoveffa Tortora.

further data representation process to utilize the efficiency of 2D CNN.

The second approach involves effective network architecture designs that utilize the structural properties of hands [9]–[12]. For a network model, hierarchical networks that divide the global hand pose estimation problem into sub-tasks have been proposed, where each one focuses on a specific finger or hand region. Sun *et al.* [11] divided global hand pose estimation into local estimations of palm and finger pose and then updated the finger location according to the palm pose in a cascade manner. Madadi *et al.* [10] designed a hierarchically tree-like, structured CNN using five branches to model the five fingers and an additional branch to model the palm orientation. Zhou *et al.* [12] designed a three-branch network, where the three branches correspond to the thumb, index finger, and the three other fingers, according to the differences in the functional importance of different fingers. More recently, Du *et al.* [9] proposed a two-branch cross-connection network that hierarchically regresses palm and finger poses through information-sharing in a multi-task setup. These studies have demonstrated that handling different parts of the hand via a multi-branch CNN can improve the accuracy of 3D hand pose estimation. However, these methods estimate all joints of the finger directly without explicitly considering finger kinematics. For a finger, the movements of different joints in the finger are dependent on each other and can be represented as a kinematic chain. To better capture spatial dependencies between adjacent joints, in our work, we adopt a recurrent neural network (RNN), which is mainly used to handle the sequential features of the joints in a finger.

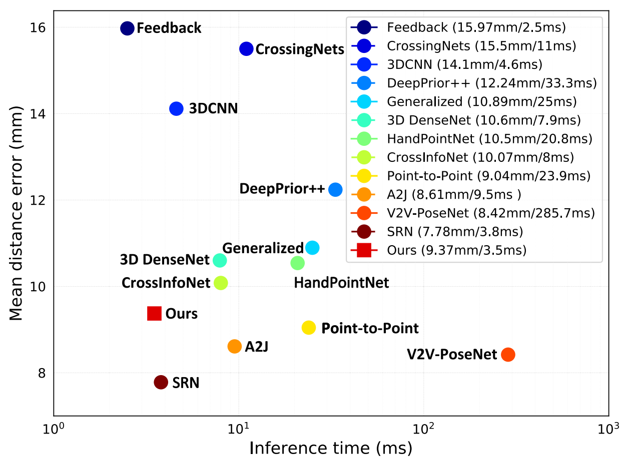


FIGURE 1. Performance/inference speed trade-off on NYU dataset. The proposed HCRNN has advantages both in estimation accuracy and fast inference speed.

Fig. 1 depicts the inference time (ms) versus mean distance error (mm) graph of some state-of-the-art 3D hand pose estimation methods on the NYU [13] dataset. Note that, in terms of estimation accuracy, the proposed method is ranked in the 5th place behind SRN [14], V2V-PoseNet [5], A2J [15], and Point-to-Point [8] which have higher computational complexity than the proposed method. The proposed HCRNN operates approximately 80, seven, and three times faster than V2V-PoseNet [5], Point-to-Point [8], and A2J [15], respectively. On the other hand, in terms of inference speed, the proposed HCRNN is in the 2nd place behind Feedback [16], while HCRNN improves the estimation accuracy of Feedback by about 40%. In summary, the HCRNN achieves both goals, effectiveness and efficiency, in 3D hand pose estimation.

Fig. 2 illustrates the proposed hierarchical convolutional RNN (HCRNN), which takes the 3D geometry of a hand into account for 3D hand pose estimation. The six separate branches are based on the observation that the hand is composed of six local parts (i.e. the palm and five fingers) with different amounts of variations due to the articulated structure of the hand. Inspired by the recent study in which sequential features of a sequence-like object are obtained in a single image [17], we first extract the sequential features of joints by applying the combination of convolutional and fully-connected (FC) layers to the feature from the encoder network. Then, we propose to make use of an RNN taking these joint features as inputs to capture the sequential information of a finger and to extract the interdependent information of the finger joints along the kinematic chain.

Our contributions can be summarized as follows:

1) We propose the HCRNN architecture that decomposes global hand estimation into sub-tasks of estimating the local parts of the hand. Based on the understanding that the palm and finger exhibit different flexibilities and degrees of freedom, the separate branches estimate the 3D positions of the palm and five fingers. We apply the RNN to utilize spatial information between the sequential-joints of the finger.
2) We design a relatively efficient network that not only achieves promising performance with the mean errors of 6.5, 9.4, and 7.7 on the ICVL [18], NYU [13], and MSRA [11] datasets, respectively, but also runs fast, at over 285 fps, on a single GPU. The speed versus accuracy trade-off graph can be seen in Fig. 1.

II. RECURRENT NEURAL NETWORKS AND ITS VARIANTS

RNNs learn a hidden representation for each time step of sequential data by considering both the current and previous information. Thanks to their ability to memorize and abstract the sequential information over time, RNN has achieved great success in sequential data modelings such as natural language processing (NLP) and speech recognition. Formally, the hidden state and the output feature at the current time step, t, can be respectively obtained by

$$h_t = \tanh(W_h x_t + U_h h_{t-1} + b_h), \quad (1)$$
$$y_t = W_y h_t + b_y, \quad (2)$$

where W_h, U_h, and b_h are the parameters for the hidden state and W_y and b_y are the parameters for the current output. This recurrent structure allows the RNN to convey the information in the past time steps to the current prediction process.

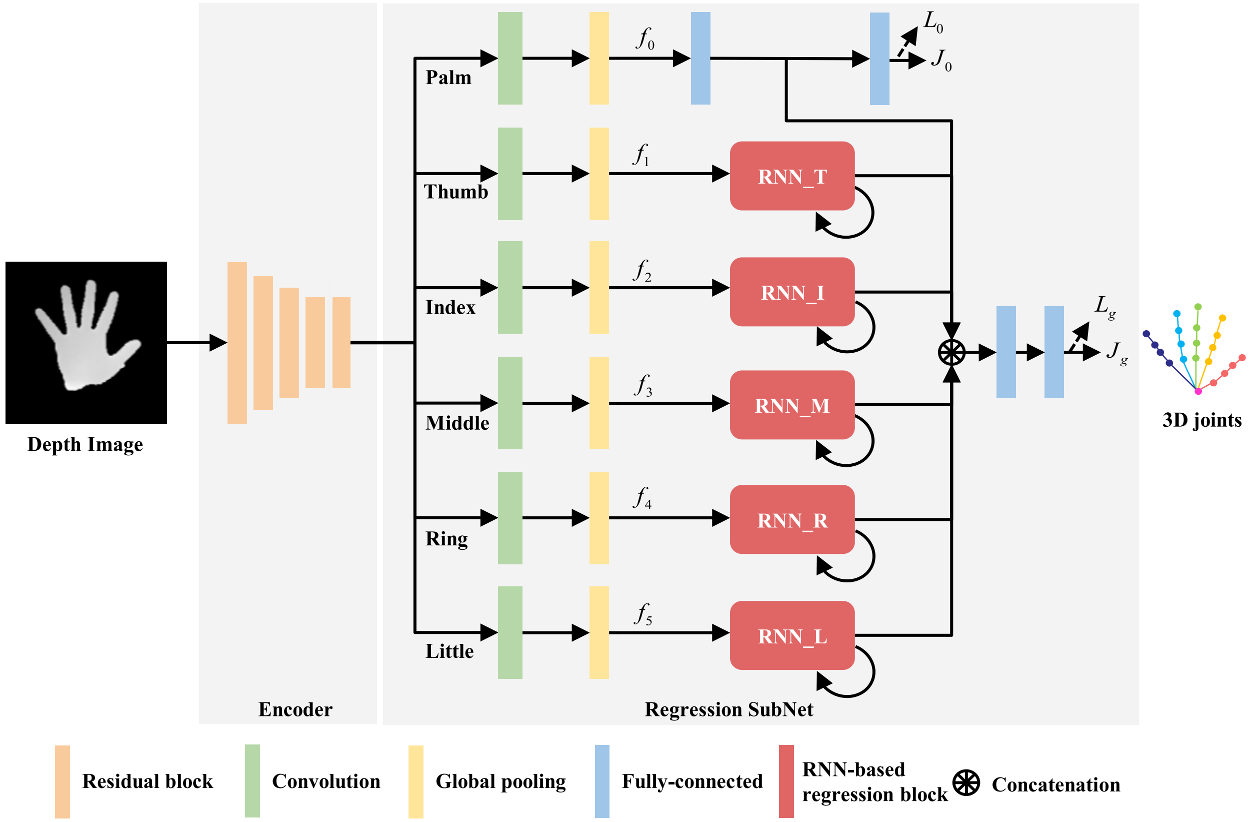


FIGURE 2. The overall architecture of the proposed method for 3D hand pose estimation. It is designed with a hierarchical structure, where each separate branch estimates the 3D positions of the palm and five fingers. In the finger branches, the RNN-based regression block is added to capture the spatial dependencies between the sequential-joints of the finger. f_k represents the feature of the kth finger, J_g and J_0 denote the global hand joint and the palm joint, respectively, L_g and L_0 are the global joint regression loss and the palm joint regression loss, respectively.

However, as the time gap between information grows, the basic RNN cannot preserve temporal memories and faces the problem of long-term dependencies due to the vanishing gradient [19].

To tackle the aforementioned problem, long short-term memory (LSTM) [20] has recently been proposed, which replaces the nonlinear units of the basic RNN. Among the numerous variants of LSTM, the gated recurrent unit (GRU) [21] is one of the most popular modules owing to its ability to reduce the complexity of LSTM. There are two key components in GRU, which are referred to as the update gate and reset gate. The update gate, z_t, controls the balance between the previous and current feature information, while the reset gate, r_t, is used to modulate the states of the previous hidden feature. At the time step t, the current gate output y_t is computed as follows:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r), \tag{3}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z), \tag{4}$$

$$\widetilde{h}_t = \tanh(W_h x_t + r_t \odot U_h h_{t-1} + b_h), \tag{5}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \widetilde{h}_t, \tag{6}$$

$$y_t = W_y h_t + b_y, \tag{7}$$

where W, U, and b terms with a specific subscript represent the parameters for each layer, respectively, $\sigma(\cdot)$ is a sigmoid function, $\widetilde{h}_t$ is the current memory content, h_t is a current hidden state, and $\odot$ represents element-wise multiplication. In this work, we adopt a GRU as a basic RNN module.

III. PROPOSED METHOD

A. OVERALL NETWORK ARCHITECTURE

Fig. 2 illustrates the overall architecture of the proposed 3D hand pose estimation methods. The proposed network mainly consists of two parts: an encoder that transforms an input depth image onto the abstracted feature space; a joint regression sub-networks (SubNets) that are composed of six branches corresponding to five fingers and a palm. The input depth image is firstly fed into an encoder for low-level feature extraction. Then, the regression SubNets take the obtained feature map from the encoder as an input and predict the 3D pose of a palm and fingers.

B. ENCODER NETWORK

The encoder of the proposed 3D hand pose estimation method is based on ResNet with full pre-activation [22] as described in Table 1. The encoder has five residual blocks, each of which consists of two 3×3 convolutional layers. For the skip connection in the residual block, we use a 1×1 convolutional shortcut. Average-pooling layers for down-sampling are appended after each residual block except for the last

TABLE 1. **Detailed architecture of the encoder network for initial feature extraction.**

Layers	Kernel size	# channels	Output size
Residual block	3×3	64	96
Average pooling	2×2	64	48
Residual block	3×3	64	48
Average pooling	2×2	64	24
Residual block	3×3	128	24
Average pooling	2×2	128	12
Residual block	3×3	256	12
Residual block	3×3	256	12

two blocks. We design the encoder to be shallow because an input depth map has a plainer texture as compared with the inputs used for classification or segmentation tasks. Unless otherwise noted, the encoder takes the input with a spatial size of 96×96, and thus the output feature map has a spatial size of 12×12 with 256 channels.

C. REGRESSION SUBNET

As the different parts of the hands have different amounts of variation and degrees of freedom (DoF) due to the articulated structure of the hand [11], it is inefficient directly regressing all parts together from the encoded feature. Among hand joints, the palm is much more stable than fingers and mainly affects global hand positions. The five fingers are largely independent and have more flexibility than palm [11]. By decomposing the global hand pose estimation problem into sub-tasks of palm and finger pose estimation, the optimization of the network parameters can be simplified by limiting the search space [23], [24]. Based on the aforementioned properties of hands, we design our joint regression SubNet as hierarchically structured networks with six separate branches for the estimation of the palm and five fingers: thumb, index, middle, ring, and little finger.

1) SEQUENTIAL MODELING OF FINGER JOINTS

Although recent state-of-the-art 3D hand pose estimation methods mostly adopt discriminative learning-based methods as deep learning technology advances, model-based methods still have their own advantages [25]–[27]. As considering the joint connection over the finger, the movements of different joints of a finger are highly related to each other. With a finger root, i.e. metacarpophalangeal (MCP), as the base position, each finger is composed of sequential-joints, which can be represented as a kinematic chain. Using the kinematic structure of a hand, the model-based methods can constrain the solution space of the hand joint positions. To take the advantages of both the model- and discriminative-learning-based methods, we estimate the nth joint of the kth finger, $J_k^{(n)}$, by using the feature sequence for the previous links as follows:

$$J_k^{(n)} = \mathcal{F}_k^{(n)}\left(f_k^{(n)}, f_k^{(n-1)}, \cdots, f_k^{(1)}\right), \tag{8}$$

where $\mathcal{F}_k^{(n)}(\cdot)$ is a mapping function from the feature sequence onto the 3D coordinate and $f_k^{(i)}$, $i = 1, \cdots, n$, is the abstracted feature of the kth finger's joint. Note that $f_k^{(1)}$ represents the feature of the MCP joint. Reflecting the kinematic structure of the finger, we design a regression subnetwork (SubNet) of (8) by using a recurrent model in which the hidden layer containing the information of previous links controls the current estimation as follows:

$$h_k^{(n)} = \mathcal{G}\left(h_k^{(n-1)}, f_k^{(n)}; \Theta_k\right), \tag{9}$$

$$J_k^{(n)} = W_k h_k^{(n)} + b_k, \tag{10}$$

where $h_k^{(n)}$ and $h_k^{(n-1)}$ are the current and previous hidden states, respectively, $\mathcal{G}(\cdot)$ is a GRU, Θ_k represents the parameter for the FC layers in the GRU, and W_k and b_k are the parameter for the linear regression.

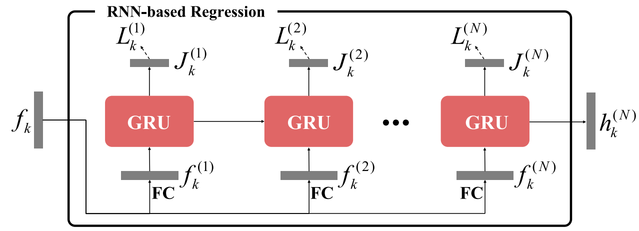


FIGURE 3. **Architecture of the RNN-based regression block, which takes a feature sequence of a finger as input. It conveys the feature information of the spatially adjacent joints, which is utilized for estimating the position of sequential finger joints. $f_k^{(i)}$, $i = 1, \cdots, N$, represents the abstracted feature of the kth finger's joint. $J_k^{(n)}$ and $L_k^{(n)}$ are the nth joint of the kth finger and its corresponding finger joint regression loss, respectively, and $h_k^{(N)}$ is the output hidden state.**

2) FINGER BRANCH

Fig. 3 depicts the RNN-based regression block in the finger branch. For a finger, each 3D joint position is recurrently estimated by using its input encoded feature and the previous joint information in the latent space. Inspired by the recent study where sequential features of a sequence-like object are obtained in a single image [17], we first extract the sequential features of joints from the output of the encoder network. To this end, we first apply a 3×3 convolution with 256 channels followed by a global average pooling to the feature map from the encoder for each branch in order to extract the hand-part-specific information. For example, let f_k be such a feature vector of kth finger branch. Taking f_k as an input, N different FC layers are employed to obtain the feature vectors of each joint, $f_k^{(1)}, \cdots, f_k^{(N)}$, where N is the number of joints in the finger, as shown in Fig. 3. To estimate the first joint of the finger $J_k^{(1)}$, i.e. the MCP joint, $f_k^{(1)}$ and a zero vector as an initial hidden state are fed into the GRU as in (9) and (10). Then, the hidden state in the GRU conveys the sequential information to the end joint along the kinematic chain. The 3D coordinates of adjacent finger joints $J_k^{(n)}$ for $n = 1, \cdots, N$ are sequentially estimated by utilizing the previous hidden state.

3) PALM BRANCH

Like the finger branches, the hand-part-specific feature f_0 for the palm branch is first extracted. As mentioned at the beginning of this section, the palm is inflexible and more stable than fingers. Thus, the palm joint J_0 is directly estimated by applying a series of FC layers to the palm feature f_0.

4) ENSEMBLE STRATEGY

To incorporate the inter-finger correlation, instead of directly concatenating the joint predictions of each branch, we adopt the feature ensemble strategy as in [9], [10], [12]. The features from the last FC layers of the palm branch and the hidden states $h_k^{(N)}$ for $k = 1, \cdots, 5$ containing the previous joints information from the finger branches are concatenated and followed by 1,024 dimension FC layers to estimate the global 3D hand position J_g.

D. LOSS FUNCTIONS

To train the proposed network, we adopt the smooth ℓ_1 loss [28] defined as

$$\text{smooth}_{\ell_1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 0.01, \\ 0.01(|x - 0.005|), & \text{otherwise,} \end{cases} \tag{11}$$

The smooth ℓ_1 loss is more robust to outliers than ℓ_2 loss and it is experimentally proven that smooth ℓ_1 loss is also effective in 3D hand pose estimation tasks [6], [14], [15]. In the joint regression SubNet, we define the global joint regression loss L_g along with the local joint regression loss L_l as the constraint to enforce each finger branch to be specialized in predefined subsets of the hand joints as follows:

$$L_g = \sum_{i=1}^{T} \left| \widetilde{J}_g^{(i)} - J_g^{(i)} \right|_{smooth}, \tag{12}$$

$$L_l = L_0 + \sum_{k=1}^{5} \sum_{n=1}^{N} \left| \widetilde{J}_k^{(n)} - J_k^{(n)} \right|_{smooth}, \tag{13}$$

with

$$|y|_{smooth} = \sum_{j} \text{smooth}_{\ell_1}(y_j), \tag{14}$$

where T and N are the number of whole joints and the number of joints in the finger, respectively, y_j is the jth component of y, and L_0 is a palm joint regression loss. $\widetilde{J}_g^{(i)}$ and $J_g^{(i)}$ represent the ground truth and estimated 3D coordinates of the ith hand joint, $\widetilde{J}_k^{(n)}$ and $J_k^{(n)}$ denote the ground truth and estimated 3D coordinates of the nth joint from the kth finger branch. The total loss function is defined as:

$$L = L_g + \lambda L_l, \tag{15}$$

where λ is a weight factor to balance the two losses. In our experiment, λ is set to 1.

E. IMPLEMENTATION DETAILS

As mentioned in the previous subsection, the input feature vectors of each branch f_k are obtained by 3×3 convolutional layers with 256 channels followed by global average pooling layers. The number of output dimensions of the FC layers in the palm branch and the RNN-based regression block is set to 256 except those of the last FC layer for joint regression and the ensemble layer with 1,024 dimensions. We add a batch normalization (BN) layer after each convolution layer to simplify the learning procedure and improve the generalization ability. The rectified linear unit (ReLU) is employed as an activation function after the convolutional and FC layers except for the last FC layer, which performs final joint regression. To simplify the hand pose estimation task, we first extract a fixed-size cube around the center points from the depth image. Because the center point which is obtained by the center of mass after simple depth thresholding can be inaccurate, some methods [5], [29] trained a simple 2D CNN which takes cropped depth image using depth thresholding and generates the 3D offset from the current center point to the center of ground-truth joint locations. Following the method in [5], we used refined center points which can be obtained by adding the output offset value of the network to the center of the mass. A cropped hand region from the bounding area is then resized to a fixed size of 96×96. The depth values inside the cropped cube region are normalized to $[-1, 1]$. The points outside the cube are assigned a depth of 1. During training, we apply the following online data augmentation tricks: Random rotation in the range $[-180°, 180°]$; random translation of $[-10, 10]$ pixels; random scaling of $[0.9, 1.1]$. We use the Adam optimizer [30] with an initial learning rate of 1e-3, a batch size of 32, and a weight decay of 1e-5. The learning rate is multiplied by a factor of 0.96 at every 2k iterations. We trained our networks from scratch with parameters randomly initialized using the Xavier initializer [31]. The entire network is trained for 120 epochs in an end-to-end manner. Our model is implemented by Tensorflow [32], and a single NVIDIA Titan X GPU (Pascal architecture) is used for training and testing. To design the branch details for different hand pose datasets, we define the joint subsets as shown in Fig. 4.

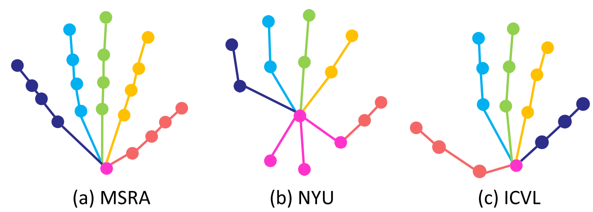


FIGURE 4. Subset of joints on the MSRA, NYU, and ICVL datasets. Violet color indicates the palm joints subset, and other colors indicate the finger joints subset.

IV. EXPERIMENTAL RESULTS

A. DATASETS AND EVALUATION METRICS

We evaluated our network on the following three public hand pose datasets: ICVL [18], NYU [13], and MSRA [11]. The NYU and ICVL datasets provide initially divided training and test images. The MSRA dataset consists of sequences from 9 different subjects and we use a leave-one-subject-out cross-validation strategy for evaluation. More details of these datasets are as follows:

1) MSRA DATASET

The MSRA dataset [11] contains 76k frames from nine different subjects with 17 different gestures. This dataset was captured with Intel's Creative Interactive Gesture Camera [33] and has 21 annotated joints, including 1 palm joint and four joints for each finger as shown in Fig. 4(a). Following the most commonly used protocol [11], we used a leave-one-subject-out cross-validation strategy for evaluation on this dataset.

2) NYU DATASET

The NYU dataset [13] was captured with three Microsoft Kinects. It contains 72k training and 8k testing images from three different views. The training set was collected from one subject, while the testing set was collected from two subjects. According to the evaluation protocol that most previous works follow, we used only a frontal view and a subset of 14 annotated joints which is depicted in Fig. 4(b) for both training and testing.

3) ICVL DATASET

The ICVL dataset [18] was captured with an Intel Realsense Camera. In this dataset, there are 22k frames from 10 different subjects for training and 1.5k images for testing. The training set includes an additional 300k augmented frames with in-plane rotations, but we did not use them because we applied online data augmentation during training, as described in Section III-E. This dataset has 16 annotated joints, including 1 palm joint and 3 joints for each finger, as shown in Fig. 4(c).

4) EVALUATION METRICS

To evaluate the performance of the different 3D hand pose estimation methods, we used two metrics. The first metric is the average 3D distance error between the ground truth and predicted 3D position for each joint. The second one is the percentage of succeeded frames whose errors for all joints are within a threshold.

B. SELF-COMPARISONS

To analyze the contributions of each component of the proposed method, we conducted self-comparative experiments on the NYU [13] dataset. First, we evaluated the effect of the number of branches. For this experiment, as shown in Fig. 5(a), we designed a two-branch network consisting of one branch for palm joint regression and the other one for unified finger joint regression, which is a similar approach in [9]. In other words, in the finger branch of the two-branch network, the RNN-based regression block estimates recurrently the joints of the five fingers simultaneously. For a fair comparison with the proposed architecture, we adjusted the number of convolution channels and dimensions of FC layers in the finger branch of the two-branch network so that each network has a similar number of parameters. As shown in Table 2 and Fig. 6, the proposed network architecture with six branches performs better and reduces the mean 3D distance error (mm) by 0.51 (from 9.88 to 9.37) as compared with the baseline network architecture. This result demonstrates that regressing all fingers jointly is inefficient because

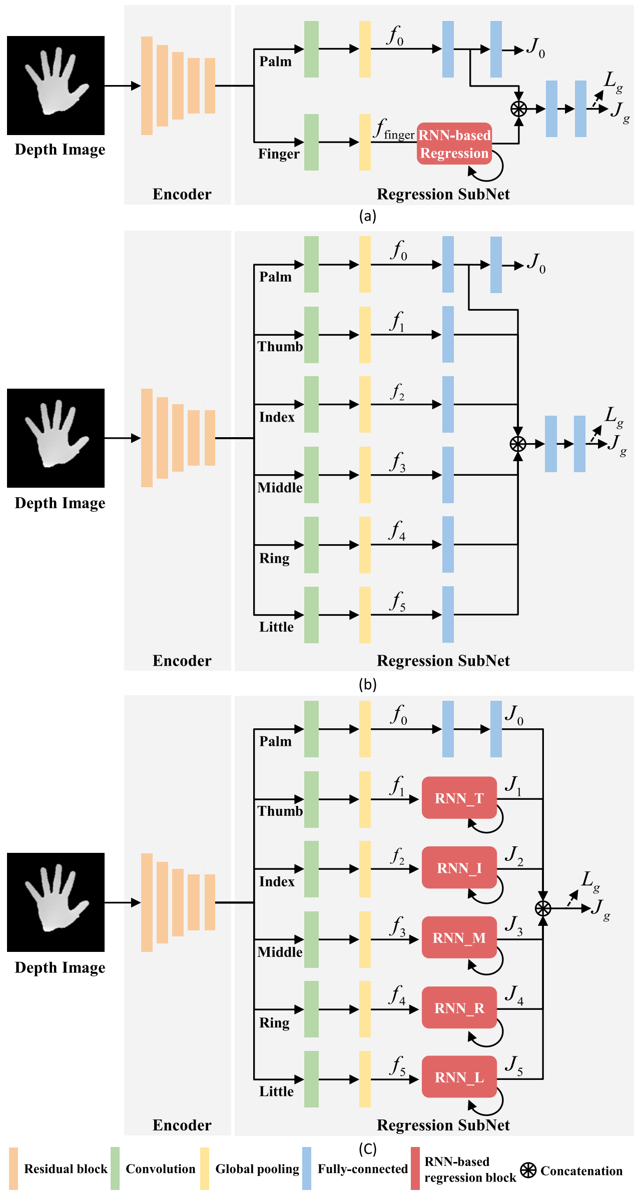


FIGURE 5. Baseline architectures for self-comparisons. a) Two-branch baseline network with one branch for palm and the other branch for finger joint regression. b) Baseline network where the RNN-based regression block is replaced with fully connected layers. c) Baseline network where the joint predictions of each branch are directly concatenated instead of adopting ensemble strategy.

the five fingers are largely independent [11]. By building a fine-grained branch for each finger, the network can learn richer features for finger pose estimation.

We also evaluated the effect of the RNN-based regression on finger joint estimation. We designed another network architecture, where the RNN-based regression block is replaced with FC layers, as shown in Fig. 5(b). In each finger branch of the FC-layer-based network, the output features are directly extracted from the input finger features with a simple FC layer instead of considering the structural properties of finger joints. As shown in Table 2 and Fig. 6, the proposed RNN-based network architecture achieves a better result than the FC-layer-based network with direct regression

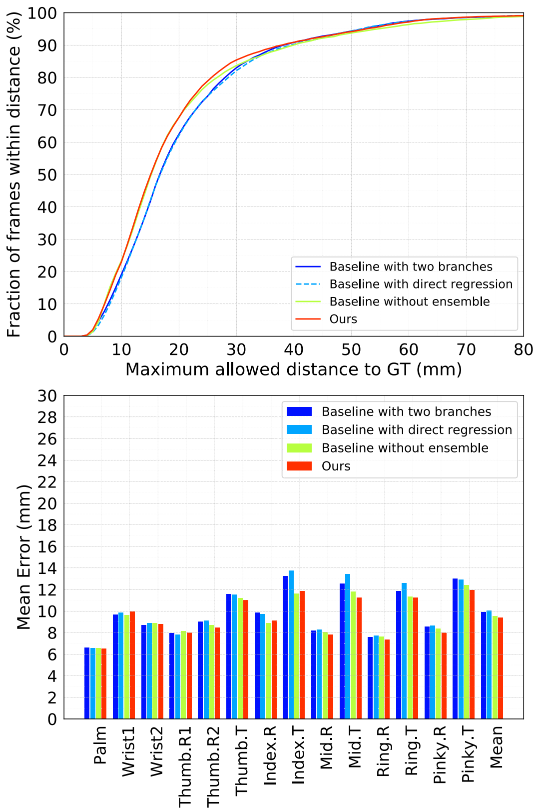


FIGURE 6. Self-comparison results on 3D distance errors (mm) per hand joint.

TABLE 2. Self-comparison results on NYU dataset.

Method	Complexity (MB)	Inference time (ms)	Mean error (mm)
Baseline (two branches)	39.4	2.4	9.88
Baseline (direct regression)	39.2	2.7	10.05
Baseline (without ensemble)	39.5	3.5	9.52
HCRNN	39.5	3.5	9.37

and reduces the mean 3D distance error (mm) by 0.68 (from 10.05 to 9.37). This experiment confirms that the proposed RNN-based regression can drive the network to utilize spatial dependencies between the sequential-joints of the finger for accurate 3d hand pose estimation.

Finally, we evaluated the effect of the ensemble strategy. We designed a baseline architecture, where the joint predictions of each branch are directly concatenated instead of fusing features from each branch as in Fig. 5(c). As shown in Table 2 and Fig. 6, the mean 3D distance error (mm) of the baseline network achieves 9.52mm, while the mean error of the proposed HCRNN achieves 9.37, which shows that adopting the ensemble strategy in the regression layer improves the prediction accuracy.

Although HCRNN with the RNN-based regression block estimates the joint of the fingers recurrently through the shared GRU and therefore takes longer inference time than baseline networks, HCRNN improves the accuracy of baseline networks by about 5%, 7%, and 2%, respectively. These results show that designing a CNN network exploiting the articulated structure of the hand achieves an improvement in the accuracy of 3D hand pose estimation.

TABLE 3. Comparison of the proposed method with state-of-the-art methods on three 3D hand pose datasets. Mean error indicates the average 3D distance error. And "∗" means that ResNet-50 pre-trained on ImageNet is used as the backbone network.

Method	Mean error (mm)			Input
	ICVL	NYU	MSRA	
3D CNN [3]	–	14.1	9.58	3D
SHPR-Net [6]	7.22	10.78	7.96	3D
3D DenseNet [4]	6.7	10.6	7.9	3D
Hand PointNet [7]	6.94	10.5	8.5	3D
Point-to-Point [8]	6.33	9.04	7.71	3D
V2V-PoseNet [5]	6.28	8.42	7.49	3D
Multi-view CNNs [36]	–	–	13.1	2D
DISCO [37]	–	20.7	–	2D
DeepPrior [38]	10.4	19.73	–	2D
Feedback [16]	–	15.97	–	2D
Global2Local [10]	–	15.60	12.8	2D
CrossingNets [39]	10.2	15.5	12.2	2D
HBE [12]	8.62	–	–	2D
REN (4x6x6) [40]	7.63	13.39	–	2D
REN (9x6x6) [41]	7.31	12.69	9.79	2D
DeepPrior++ [31]	8.1	12.24	9.5	2D
Pose-REN [42]	6.79	11.81	8.65	2D
Generalized [43]	–	10.89	–	2D
CrossInfoNet [9]	6.73	10.07	7.86	2D
A2J* [15]	6.46	8.61	–	2D
SRN [14]	6.26	7.78	7.16	2D
HCRNN (Ours)	6.54	9.37	7.70	2D

C. COMPARISION WITH STATE-OF-THE-ART METHODS

We compared the proposed network on three public 3D hand pose datasets with the most recently proposed methods using 2D depth maps as an input, including DISCO [35], DeepPrior [36], its improved version DeepPrior++ [29], Feedback [16], Multi-view CNNs [34], REN-4 × 6 × 6 [38], REN-9 × 6 × 6 [39], Pose-REN [40], Generalized [41], Global2Local [10], CrossingNets [37], HBE [12], CrossInfoNet [9], A2J [15], and SRN [14] as well as methods using 3D inputs, including 3D CNN [3], SHPR-Net [6], 3D DenseNet [4], HandPointNet [7], Point-to-Point [8], and V2V-PoseNet [5]. The average 3D distance error per joint and percentage of success frames over different error thresholds are respectively shown in Table 3 and Fig. 7. It is seen that our results outperform most 2D CNN-based methods on three datasets. Our method shows similar results to A2J [15] on ICVL dataset but slightly worse performance on NYU dataset. Nevertheless, due to the adoption of a complex ResNet-50-based backbone network, A2J requires a higher computational complexity than the proposed method as described in Table 4. As compared to SRN [14] which shows the best performance in both 2D and 3D input data-based methods, our method achieves comparable results on ICVL and MSRA datasets. For NYU dataset, our method

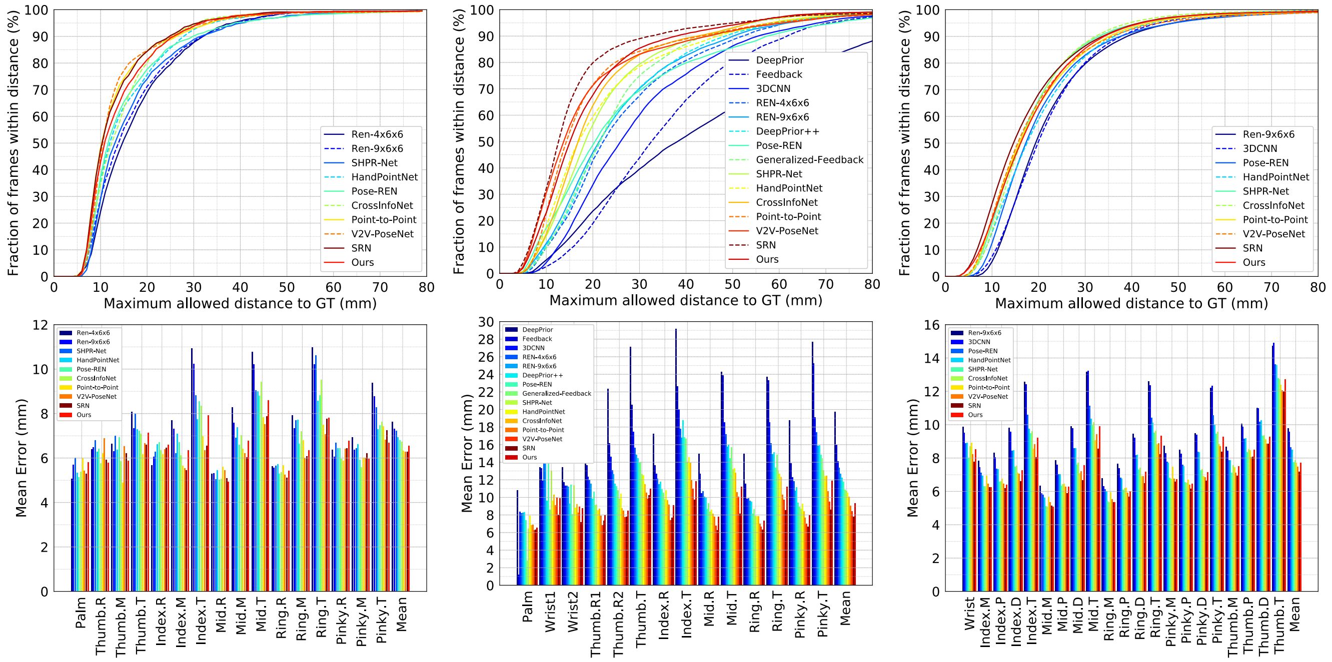


FIGURE 7. Comparison with state-of-the-art methods. Top row: percentage of success frames over different error thresholds. Bottom row: 3D distance errors per hand joint. Left: ICVL dataset, Center: NYU dataset, Right: MSRA dataset.

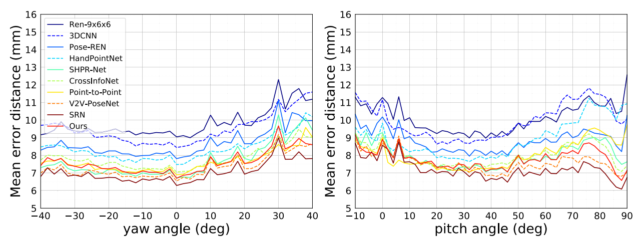


FIGURE 8. Comparison of mean error distance over different yaw (left) and pitch (right) viewpoint angles on MSRA dataset.

is inferior to SRN on accuracy but shows better running efficiency. As compared with the methods using 3D inputs, our method performs better than 3DCNN [3], SHPR-Net [6], HandPointNet [7], and 3D DenseNe [4] and achieves comparable performance with Point-to-Point [8] on the ICVL and MSRA datasets. On the NYU dataset, the results of the proposed method are worse than those of V2V-PoseNet [5] but are better in terms of percentage of success frame rates when the error threshold is larger than 30mm. On the MSRA dataset, following the evaluation protocol of prior works [3], [11], [40], we also measured the mean joint error over various viewpoint angles. As shown in Fig. 8, our method can obtain promising results from large yaw and pitch angles, which demonstrates the robustness of our proposed method to viewpoint changes. The qualitative results of our method on three datasets are shown in Fig. 9. It can be seen that our method can accurately estimate 3D hand joint locations on the three datasets.

D. RUNTIME

Because 3D hand pose estimation tasks play an important role as a sub-system in the overall human-computer interaction (HCI) system such as in-vehicle infotainment (IVI), virtual reality (VR), and augmented reality (AR), the inference speed is an important factor for the practical application. Table 4 compares the inference speed of conventional and the proposed methods on a single GPU. While top-ranked methods using 3D inputs have a higher inference time owing to the time-consuming 3D convolution operation or data conversion procedure, our method has a faster inference speed owing to its efficiency of 2D CNN-based architecture. The proposed HCRNN is ranked in 2nd place among the compared methods behind Feedback. Based on the aforementioned results, it can be seen that our proposed HCRNN not only achieves competitive performance compared with state-of-the-art methods but also is very efficient, having a high frame rate, which shows the applicability to real-time applications.

TABLE 4. Comparison of inference speed with state-of-the-art methods. The inference speed is measured with a single GPU.

Method	Test speed (fps)	Input
V2V-PoseNet [5]	3.5	3D
Point-to-Point [8]	41.8	3D
HandPointNet [7]	48	3D
3D DenseNet [4]	126	3D
3D CNN [3]	215	3D
DeepPrior++ [31]	30	2D
Generalized [43]	40	2D
CrossingNets [39]	90.9	2D
A2J [15]	105.1	2D
CrossInfoNet [9]	124.5	2D
SRN [14]	263.1	2D
Feedback [16]	**400**	2D
HCRNN (ours)	<u>285</u>	2D

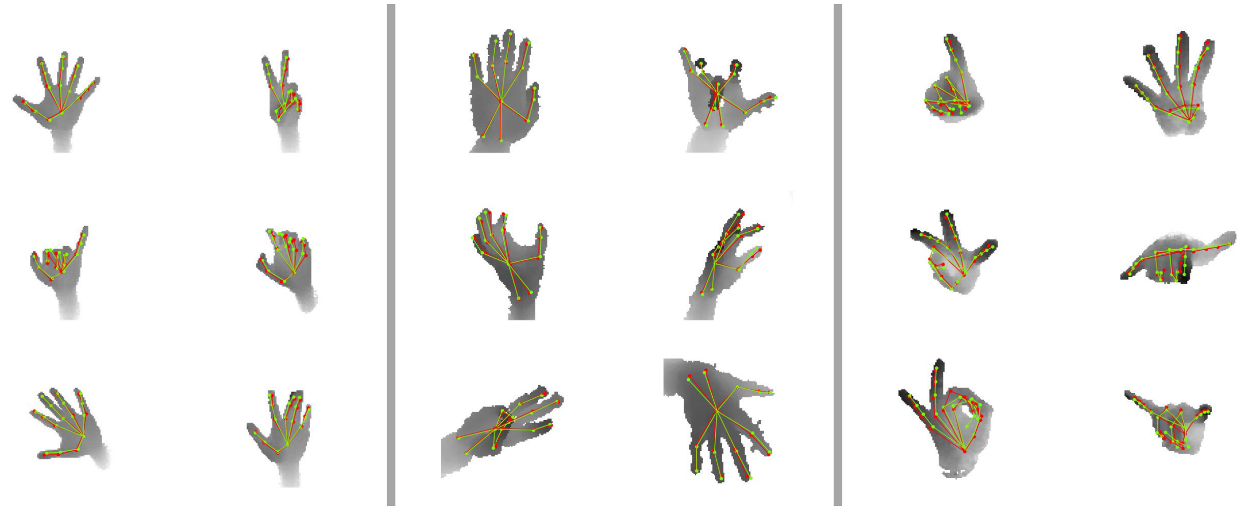


FIGURE 9. Qualitative results on the three public datasets. Left: ICVL dataset. Center: NYU dataset. Right: MSRA dataset. The ground truth is shown as red lines, and the prediction is shown as green lines.

V. CONCLUSION

To design a practical architecture for 3D hand pose estimation, we considered the articulated structure of the hand and proposed an efficient regression network, namely termed HCRNN. The proposed HCRNN has a hierarchical architecture, where six separate branches are trained to estimate the position of each local part of the hand: the palm and five fingers. In each finger branch, we adopted an RNN to model spatial dependencies between the sequential-joints of the finger. In addition, HCRNN is built on a 2D CNN that directly takes 2D depth images as inputs, making it more efficient than 3D CNN-based methods. The experimental results showed that the proposed HCRNN not only achieves competitive performance compared with state-of-the-art methods but also has a highly efficient running speed of 285 fps on a single GPU.

REFERENCES

[1] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *Comput. Vis. Image Understand.*, vol. 108, nos. 1–2, pp. 52–73, Oct. 2007.

[2] S. Yuan *et al.*, "Depth-based 3D hand pose estimation: From current achievements to future goals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2636–2645.

[3] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "3D convolutional neural networks for efficient and robust hand pose estimation from single depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1991–2000.

[4] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Real-time 3D hand pose estimation with 3D convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 956–970, Apr. 2019.

[5] J. Y. Chang, G. Moon, and K. M. Lee, "V2V-PoseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5079–5088.

[6] X. Chen, G. Wang, C. Zhang, T.-K. Kim, and X. Ji, "SHPR-NET: Deep semantic hand pose regression from point clouds," *IEEE Access*, vol. 6, pp. 43425–43439, 2018.

[7] L. Ge, Y. Cai, J. Weng, and J. Yuan, "Hand PointNet: 3D hand pose estimation using point sets," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8417–8426.

[8] L. Ge, Z. Ren, and J. Yuan, "Point-to-point regression pointnet for 3D hand pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 475–491.

[9] K. Du, X. Lin, Y. Sun, and X. Ma, "CrossInfoNet: Multi-task information sharing based hand pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9896–9905.

[10] M. Madadi, S. Escalera, X. Baró, and J. Gonzalez, "End-to-end global to local CNN learning for hand pose recovery in depth data," 2017, *arXiv:1705.09606*. [Online]. Available: http://arxiv.org/abs/1705.09606

[11] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun, "Cascaded hand pose regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 824–832.

[12] Y. Zhou, J. Lu, K. Du, X. Lin, Y. Sun, and X. Ma, "HBE: Hand branch ensemble network for real-time 3D hand pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 501–516.

[13] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Trans. Graph.*, vol. 33, no. 5, p. 169, Sep. 2014.

[14] P. Ren, H. Sun, Q. Qi, J. Wang, and W. Huang, "SRN: Stacked regression network for real-time 3D hand pose estimation," in *Proc. BMVC*, 2019, pp. 1–14.

[15] F. Xiong, B. Zhang, Y. Xiao, Z. Cao, T. Yu, J. T. Zhou, and J. Yuan, "A2J: Anchor-to-joint regression network for 3D articulated pose estimation from a single depth image," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 793–802.

[16] M. Oberweger, P. Wohlhart, and V. Lepetit, "Training a feedback loop for hand pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3316–3324.

[17] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017.

[18] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim, "Latent regression forest: Structured estimation of 3D articulated hand posture," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3786–3793.

[19] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies," in *A Field Guide to Dynamical Recurrent Networks*, S. C. Kremer and J. F. Kolen, Eds. Hoboken, NJ, USA: Wiley, 2001.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 630–645.

[23] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[24] A. Sinha, C. Choi, and K. Ramani, "DeepHand: Robust hand pose estimation by completing a matrix imputed with deep features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4150–4158.

[25] J. Wöhlke, S. Li, and D. Lee, "Model-based hand pose estimation for generalized hand shape with appearance normalization," 2018, *arXiv:1807.00898*. [Online]. Available: http://arxiv.org/abs/1807.00898

[26] Y. Wu, J. Y. Lin, and T. S. Huang, "Capturing natural hand articulation," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Jul. 2001, pp. 426–432.

[27] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei, "Model-based deep hand pose estimation," 2016, *arXiv:1606.06854*. [Online]. Available: http://arxiv.org/abs/1606.06854

[28] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[29] M. Oberweger and V. Lepetit, "DeepPrior++: Improving fast and accurate 3D hand pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 585–594.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[31] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[32] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.

[33] S. Melax, L. Keselman, and S. Orsten, "Dynamics based 3D skeletal hand tracking," in *Proc. Graph. Interface*. Mississauga, ON, Canada: Canadian Information Processing Society, 2013, pp. 63–70.

[34] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Robust 3D hand pose estimation in single depth images: From single-view CNN to multi-view CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3593–3601.

[35] D. Bouchacourt, P. K. Mudigonda, and S. Nowozin, "DISCO Nets: Dissimilarity coefficients networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 352–360.

[36] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands deep in deep learning for hand pose estimation," 2015, *arXiv:1502.06807*. [Online]. Available: http://arxiv.org/abs/1502.06807

[37] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Crossing nets: Combining GANs and VAEs with a shared latent space for hand pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 680–689.

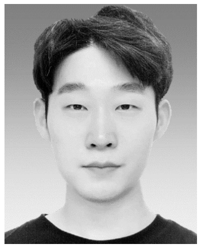
[38] H. Guo, G. Wang, X. Chen, C. Zhang, F. Qiao, and H. Yang, "Region ensemble network: Improving convolutional network for hand pose estimation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 4512–4516.

[39] G. Wang, X. Chen, H. Guo, and C. Zhang, "Region ensemble network: Towards good practices for deep 3D hand pose estimation," *J. Vis. Commun. Image Represent.*, vol. 55, pp. 404–414, Aug. 2018.

[40] X. Chen, G. Wang, H. Guo, and C. Zhang, "Pose guided structured region ensemble network for cascaded hand pose estimation," *Neurocomputing*, vol. 395, pp. 138–149, Jun. 2020.

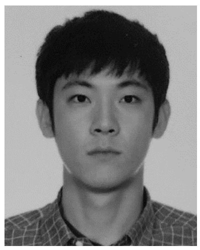
[41] M. Oberweger, P. Wohlhart, and V. Lepetit, "Generalized feedback loop for joint hand-object pose estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Mar. 27, 2019, doi: 10.1109/TPAMI.2019.2907951.

CHEOL-HWAN YOO received the B.S. degree in electrical engineering from Korea University, in 2014, where he is currently pursuing the Ph.D. degree. His research interests include deep learning, image processing, and computer vision.



SEOWON JI received the B.S. degree in electrical engineering from Korea University, Seoul, South Korea, in 2015, where he is currently pursuing the Ph.D. degree. His research interests include image processing, computer vision, and deep learning.

YONG-GOO SHIN received the B.S. and Ph.D. degrees in electronics engineering from Korea University, Seoul, South Korea, in 2014 and 2020, respectively. He is currently a Research Professor with the Department of Electrical Engineering, Korea University. His research interests include digital image processing, computer vision, and artificial intelligence.



SEUNG-WOOK KIM (Member, IEEE) received the B.S. and Ph.D. degrees in electronics engineering from Korea University, Seoul, South Korea, in 2012 and 2019, respectively. He is currently a Research Professor with the School of Electrical Engineering, Korea University. His research interests include deep learning-based applications, including image processing and computer vision.

SUNG-JEA KO (Fellow, IEEE) received the B.S. degree in electronic engineering from Korea University, in 1980, and the M.S. and Ph.D. degrees in electrical and computer engineering from the State University of New York at Buffalo, in 1986 and 1988, respectively.

From 1988 to 1992, he was an Assistant Professor with the Department of Electrical and Computer Engineering, University of Michigan–Dearborn. He joined the Department of Electronic Engineering, Korea University, in 1992, where he is currently a Professor. He has published over 210 international journal articles. He holds over 60 registered patents in fields such as video signal processing and computer vision.

Dr. Ko is a member of the National Academy of Engineering of Korea. He is also a member of the editorial board of the IEEE Transactions on Consumer Electronics. He was a recipient of the 1999 LG Research Award. He received the Hae-Dong Best Paper Award from the Institute of Electronics and Information Engineers (IEIE), in 1997, the Best Paper Award from the IEEE Asia Pacific Conference on Circuits and Systems, in 1996, a Research Excellence Award from Korea University, in 2004, a Technical Achievement Award from the IEEE Consumer Electronics (CE) Society, in 2012, a 15-year Service Award from the TPC of ICCE, in 2014, the Chester Sall Award (First Place Transaction Paper Award) from the IEEE CE Society, in 2017, and the Science and Technology Achievement Medal from the Korean Government, in 2020. He has served as the General Chairman of ITC-CSCC 2012 and IEICE 2013. He was the President of the IEIE, in 2013. From 2013 to 2016, he was the Vice-President of the IEEE CE Society. From 2015 to 2017, he was a Distinguished Lecturer of the IEEE.