REGION ENSEMBLE NETWORK: IMPROVING CONVOLUTIONAL NETWORK FOR HAND POSE ESTIMATION

Hengkai Guo, Guijin Wang, Xinghao Chen, Cairong Zhang, Fei Qiao, Huazhong Yang

Department of Electronic Engineering
Tsinghua University, Beijing

ABSTRACT

Hand pose estimation from monocular depth images is an important and challenging problem for human-computer interaction. Recently deep convolutional networks (ConvNet) with sophisticated design have been employed to address it, but the improvement over traditional methods is not so apparent. To promote the performance of directly 3D coordinate regression, we propose a tree-structured Region Ensemble Network (REN), which partitions the convolution outputs into regions and integrates the results from multiple regressors on each regions. Compared with multi-model ensemble, our model is completely end-to-end training. The experimental results demonstrate that our approach achieves the best performance among state-of-the-arts on two public datasets.

Index Terms— Convolutional Network, Hand Pose Estimation, Ensemble Learning

1. INTRODUCTION

Hand pose estimation from single depth image plays an important role in applications of human-computer interface (HCI) and augmented reality (AR). Though has been studied for several years [1], it is still challenging due to large view variance, high joint flexibility, poor depth quality, severe self occlusion and similar part confusion.

Recently, convolutional networks (ConvNets) have witnessed great growth in several computer vision tasks such as object classification [2] and human pose estimation [3] because of great modeling capacity and end-to-end feature learning. ConvNets have also been introduced to solve the problem of hand pose estimation, often with complicated structure design such as multi-branch inputs [4][5] and multi-model regression [5] [6] [7] [8]. However, ConvNets remain unable to obtain significant advantage over traditional random forest based methods [9] [10].

Inspired by model ensemble and multi-view voting [2], we present a single ConvNet architecture named *Region Ensemble Net (REN)* (Fig.1) to directly regress the 3D joint coordinates in monocular depth images 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. As shown in our experiments, REN significantly promotes the performance of our ConvNet, which outperforms all state-of-the-art methods on two challenging hand pose benchmarks [4] [11].

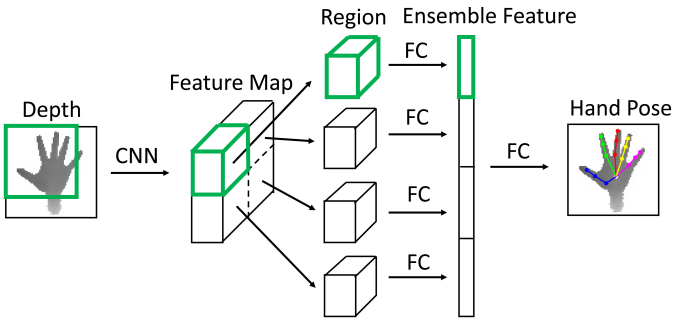

Fig. 1. Overview of region ensemble network (REN).

2. RELATED WORK

Hand pose estimation with ConvNets Recently deep ConvNets have been applied on hand pose estimation for depth imaging [12][13]. Tompson et al. [4] first use ConvNets to produce 2D heat maps and infer the 3D hand pose with inverse kinematics. Oberweger et al. [5] directly regress the 3D positions with multi-stage ConvNets using a linear layer as pose prior. In [6], a feedback loop is employed to iteratively correct the mistake, in which three ConvNets are used for initialization, synthesis and pose updating. Ge et al. [7] employ three ConvNets to separately regress 2D heat maps for each view with depth projections and fuse them to produce 3D hand pose. In [14], physical joint constraints are incorporated into a forward kinematics based layer in ConvNet. Similarly, Zhang et al. [8] embeds skeletal manifold into ConvNets and trains the model end-to-end to render sequential prediction.

Multi-model ensemble methods for ConvNets Traditional ensemble learning means that training multiple individual models and combining their outputs via averaging or weighted fusions, which is widely adopted in recognition competitions [2]. In addition to bagging [2] [15], boosting

is also introduced for people counting [16]. However, using multiple ConvNets requires large memory and time, which is not practical for applications.

Multi-branch ensemble methods for ConvNets We view single ConvNet with the fusion of multiple branches as a generalized type of ensemble. One popular strategy is to fuse different scaling inputs [4] [5] or different image cues [17] [18] [19] with multi-input branches. Another approach is to employ multi-output branches with shared convolutional feature extractor, either training with different samples [20] or learning to predict different categories [21]. Compared with multi-input, 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.

Multi-view testing for ConvNets Multi-view testing is widely used to improve accuracy for object classification [2] [22] [23]. In [2], predictions from 10-crop (4 corner and 1 center with horizontal flip) are averaged on single ConvNet. In [22] [23], fully-convolutional networks are employed in testing with inputs of multi-scale and multi-view and spatially average pooling is applied on the class score map to obtain the final scores. Such strategy has not been applied on hand pose estimation yet.

3. REGION ENSEMBLE NETWORK (REN)

As in Fig.1, REN 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 pose prediction. In this section we introduce the basic network architecture, region ensemble structure and implementation details.

Network architecture The architecture of our ConvNet for feature extraction consists of six 3×3 convolution layers (Fig.2) that accepts a 96×96 depth image as inputs. Each convolution layer is followed by a Rectified Linear Unit (ReLU) activation. Two residual connections [24] are adopted between pooling layers with 1×1 convolution filters for dimension increase. The dimension of output feature maps is $12 \times 12 \times 64$. For regression, we use two 2048 dimension FC layers with dropout rate of 0.5 for each regressor. 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.

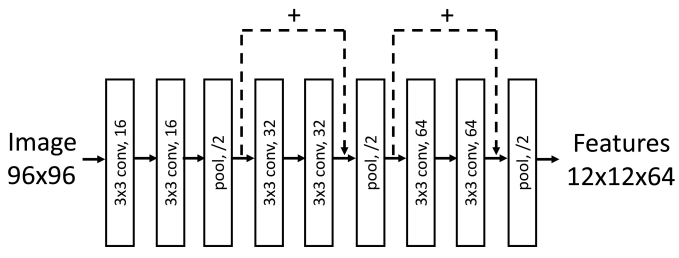

Fig. 2. Structure of ConvNet for feature extraction. The dotted arrows represent residual connections [24].

Region Ensemble Multi-view testing averages predictions from different crops of original image, which reduces the varaince for image classification [2]. Directly using multiple inputs is time-consuming. Because each activation in the convolutional feature maps is contributed by a receptive field in the image domain, we can project the multi-view inputs onto the regions of the feature maps. So multi-view voting is equal to utilizing each regions to separately predict the whole hand pose and combining the results. Based on it, we define a tree-structured network consisting of a single ConvNet trunk and several regression branches (Fig.1). We uniformly divide the feature maps of ConvNet into an $n \times n$ grid. For each grid region, we feed it into the FC layers respectively as branches. A simple strategy for combination of different branches is bagging, which averages all outputs of branches. To better 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 hand pose with an extra regression layer. The whole network can be trained end-to-end by minimizing the regression loss. We set $n = 2$ to balance the trade-off between performance and efficiency, so the receptive field of single region within the 96×96 image bounding is 62×62, which can be seen as the corner crop in [2]. Including the center crop does not provide any further increase in accuracy.

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 region ensemble can be applied on both classification and regression. 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 FC on concatenated features to learn to fuse, which increases the learning ability of the network.

Implementation We follow the practice in [4] [5] using Caffe [25]. We first segment the foreground and extract a cube of size 150mm from the depth image centered in the centroid of hand region. Then the cube is resized into 96×96 patch of depth values normalized to $[-1, 1]$ as input for ConvNet, with data augmentation of random translation, scaling and rotation. 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 after every 50000 iterations, and the model is trained for up to 200000 iterations. In the meanwhile, we use a weight decay of 0.0005 and a momentum of 0.9.

4. EXPERIMENTS

We apply our method on two publicly datasets: ICVL [11] and NYU [4]. The former dataset has 300K images for training and 1.6K for testing with 16 joints. The latter dataset has 72K images for training and 8K for testing with 14 joints. The performance is evaluated by two metrics: per-joint average Euclidean distance (in millimeters) and percentage of frames

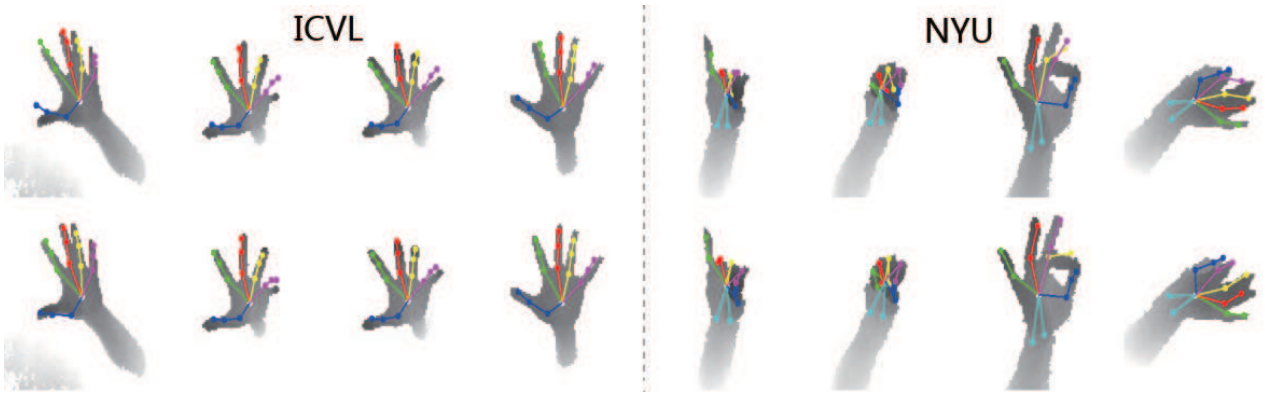

Fig. 3. Example results on ICVL [11] and NYU [4]: basic network (first row) and region ensemble network (second row).

in which all errors of joints are below a threshold [5]. First we compare our REN with baseline and different ensemble settings on ICVL dataset. Then we compare it with several state-of-the-art methods on both datasets.

Self-comparison For comparison, we implement four baseline: 1) *Basic* network has the same convolution structure in Fig.2 and single regressor on the full feature map. 2) *Basic Large* network is the same as basic network except for using 8192 dimensions in the second FC layer, which contains the similar number of parameters to REN. 3) *Basic Bagging* network has four basic networks trained independently on the same data with different random order and augmentation. The average predictions of all the networks form the final prediction. 4) *Region Bagging* network shares the same region division with REN but predicts independent hand pose for each region and averages them as prediction.

Results in Fig.4 shows that: 1) bagging or ensemble based methods are more effective since all the ensemble versions significantly outperform the single one. 2) region ensemble is much better than basic bagging and slightly better than region bagging. Qualitative results of region ensemble (second row) and basic network (first row) are shown in Fig.3.

Table.1 further compares the running time (Nvidia Titan X GPU) and the number of parameters for different approaches. Our REN obtains the most accurate results with nearly the same number of parameters as basic large network and region bagging, while the basic bagging takes significantly more parameters and time. And it runs up to over 3000fps on a single GPU, which is fast enough for practical use.

Comparison with state-of-the-arts We compare our methods against several state-of-the-art approaches on ICVL dataset [11] [5] [9] [14] [10] and NYU dataset [4] [5] [6] [26] [7] [14] [8]. Fig.5 and 6 show that proposed REN obtains the best accuracy among all the algorithms.

In details, on ICVL our method outperforms LSN [10] on the threshold of $(5mm, 15mm)$ and $(20mm, 60mm)$, and surpasses other methods with a large margin. And the mean errors obtains $0.63mm$ decrease compared with LSN, which is a 7.77% relative improvement. Similarly on NYU, our results are better than multi-view ConvNets [7] on the threshold of $(5mm, 15mm)$ and $(40mm, 80mm)$ and significantly more accurate than other approaches. Note that either LSN or multi-view ConvNets employ multiple models with complicated inputs, while our REN only uses single model without multi-stage regression, which indicates the power for proposed region ensemble strategy.

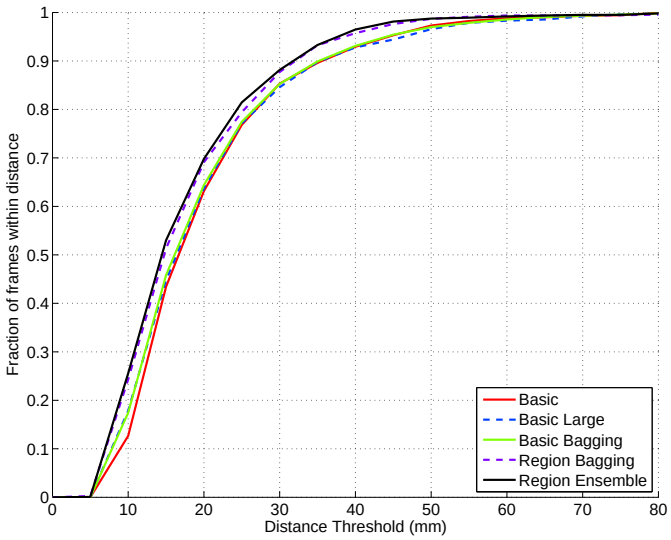

Fig. 4. Self-comparison for percentage of success frames on ICVL dataset [11].

5. CONCLUSION

To boost the performance of single ConvNet for hand pose estimation, we 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 strategy significantly improves the Con-

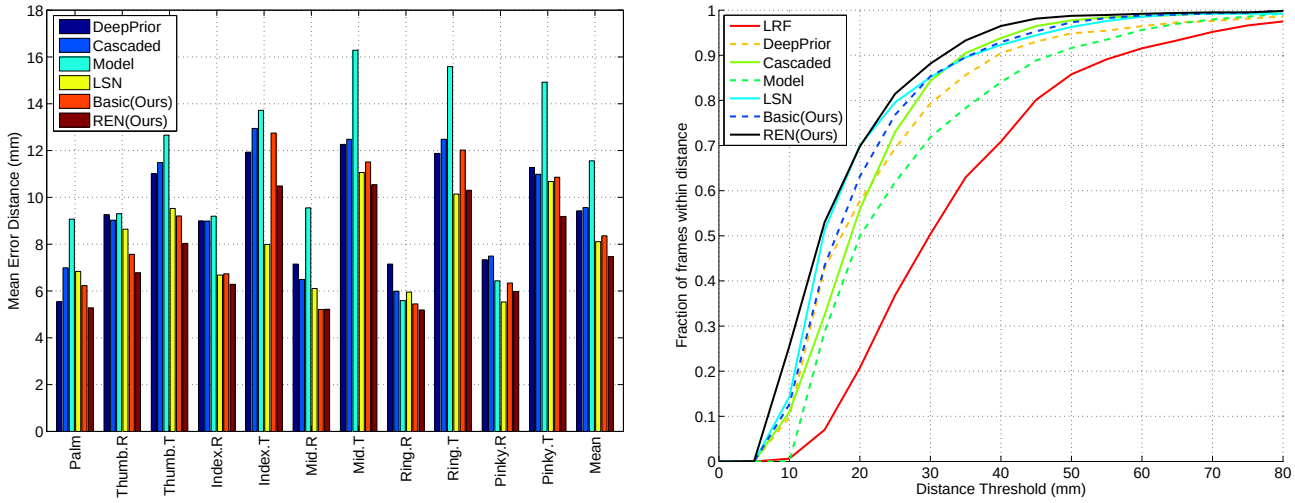

Fig. 5. Comparison with state-of-the-arts on ICVL [11] dataset: distance error (left) and percentage of success frames (right).

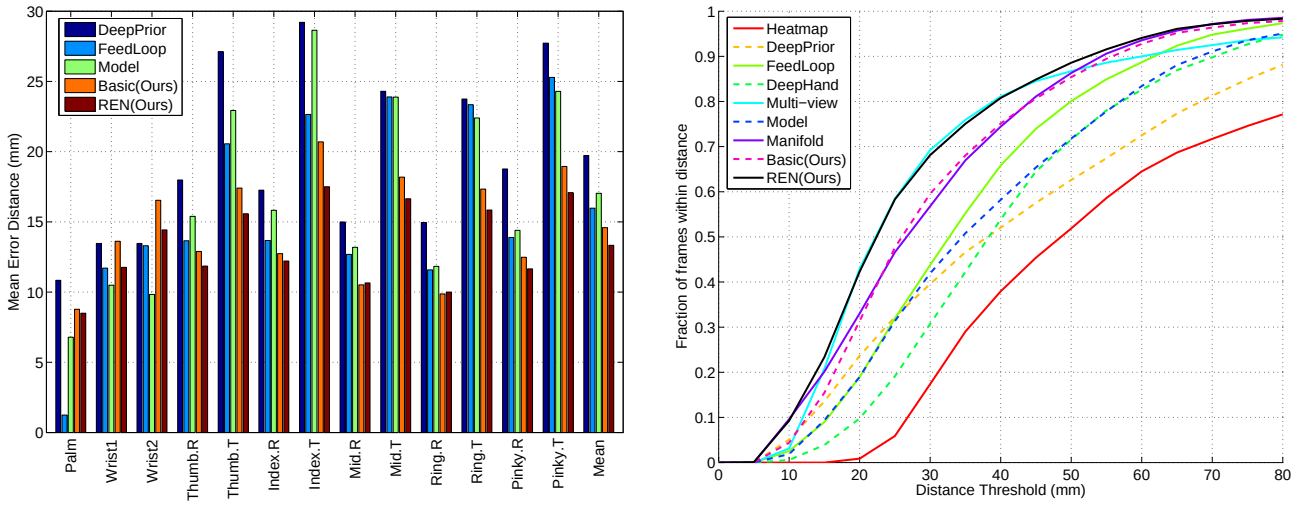

Fig. 6. Comparison with state-of-the-arts on NYU [4] datasets: distance error (left) and percentage of success frames (right).

Table 1. Average 3D distance error (mm) and GPU forward time (ms) of different methods on ICVL dataset [11].

Method	Error(mm)	Time(ms)
Basic	8.36	0.21
Basic Large	8.18	0.22
Basic Bagging	7.94	0.88
Region Bagging	7.63	0.31
Region Ensemble	**7.47**	0.31

vNet without extra large computation overhead. The experimental results demonstrate that our method outperforms all the state-of-the-arts on two datasets. In the future we will investigate and analysis more ensemble methods for ConvNets. Since proposed region ensemble is general, we will also try to apply them on more tasks such as human pose estimation.

Acknowledgments This work is supported by NSFC (No. 61271390), and 863 Plan (No. 2015AA016304). Thanks Shulan Pan for paper edition.

6. REFERENCES

[1] James Steven Supancic III, Gregory Rogez, Yi Yang, Jamie Shotton, and Deva Ramanan, "Depth-based hand pose estimation: methods, data, and challenges," in *IEEE International Conference on Computer Vision*, 2015.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional

neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[3] Adrian Bulat and Georgios Tzimiropoulos, "Human pose estimation via convolutional part heatmap regression," in *European Conference on Computer Vision*. Springer, 2016, pp. 717–732.

[4] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Transactions on Graphics*, vol. 33, no. 5, pp. 169, 2014.

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

[6] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit, "Training a feedback loop for hand pose estimation," in *IEEE International Conference on Computer Vision*, 2015.

[7] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann, "Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[8] Yu Zhang, Chi Xu, and Li Cheng, "Learning to search on manifolds for 3d pose estimation of articulated objects," *arXiv preprint arXiv:1612.00596*, 2016.

[9] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun, "Cascaded hand pose regression," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 824–832.

[10] Chengde Wan, Angela Yao, and Luc Van Gool, "Hand pose estimation from local surface normals," in *European Conference on Computer Vision*, 2016.

[11] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun 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.

[12] Guijin Wang, Xuanwu Yin, Xiaokang Pei, and Chenbo Shi, "Depth estimation for speckle projection system using progressive reliable points growing matching," *Applied optics*, vol. 52, no. 3, pp. 516–524, 2013.

[13] Chenbo Shi, Guijin Wang, Xuanwu Yin, Xiaokang Pei, Bei He, and Xinggang Lin, "High-accuracy stereo matching based on adaptive ground control points," *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 1412–1423, 2015.

[14] Xingyi Zhou, Qingfu Wan, Wei Zhang, Xiangyang Xue, and Yichen Wei, "Model-based deep hand pose estimation," in *IJCAI*, 2016.

[15] Yi Sun, Xiaogang Wang, and Xiaoou Tang, "Deep convolutional network cascade for facial point detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3476–3483.

[16] Elad Walach and Lior Wolf, "Learning to count with cnn boosting," in *European Conference on Computer Vision*. Springer, 2016, pp. 660–676.

[17] Hengkai Guo, Guijin Wang, and Xinghao Chen, "Two-stream convolutional neural network for accurate rgb-d fingertip detection using depth and edge information," *arXiv preprint arXiv:1612.07978*, 2016.

[18] Hanxi Li, Yi Li, and Fatih Porikli, "Deeptrack: Learning discriminative feature representations online for robust visual tracking," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1834–1848, 2016.

[19] Xinghao Chen, Guijin Wang, and Hengkai Guo, "Accurate fingertip detection from binocular mask images," in *Visual Communications and Image Processing (VCIP), 2016*. IEEE, 2016, pp. 1–4.

[20] Hanxi Li, Yi Li, and Fatih Porikli, "Convolutional neural net bagging for online visual tracking," *Computer Vision and Image Understanding*, pp. 120–129, 2016.

[21] Karim Ahmed, Mohammad Haris Baig, and Lorenzo Torresani, "Network of experts for large-scale image categorization," *arXiv preprint arXiv:1604.06119*, 2016.

[22] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *European Conference on Computer Vision*. Springer, 2014, pp. 346–361.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[25] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[26] Ayan Sinha, Chiho Choi, and Karthik Ramani, "Deephand: robust hand pose estimation by completing a matrix imputed with deep features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.