DEEP LEARNING FOR INTEGRATED HAND DETECTION AND POSE ESTIMATION

Min-Yu Wu (吳忞諭), Tzu-Yang Chen (陳子揚), Li-Chen Fu (傅立成), Chiou-Shann Fuh (傅楸善)

Dept. of Computer Science and Information Engineering,
National Taiwan University, Taiwan

E-mail: kyrtse@hotmail.com fuh@csie.ntu.edu.tw

ABSTRACT

In this paper, we will propose a novel framework which integrates human hand detection and pose estimation into one single pipeline. Unlike most of previous works which only focus on the pose estimation part subject to some strong assumptions or rely on a weak detector to detect human hands, we employ a deep learning architecture to complete both tasks we mentioned above. By having three different neural networks share the same convolutional layers, this deeply learning architecture can efficiently and accurately detect human hands and then compute their hand pose configuration. Moreover, we propose a new energy function to optimize the predicted result of convolutional neural network. To validate the proposed framework, experiments have been conducted and the results show that our approach is highly reliable and suitable for real-world applications.

Keywords: *Hand detection, Pose estimation, Convolutional neural network, Deep learning.*

1. INTRODUCTION

Accurate hand detection and pose estimation have become one popular research topic in recent years due to its wide applications, such as Human-Computer Interaction (HCI) and Virtual reality (VR). It provides a natural way for communication that achieves great user experience between human and cyberspace. Moreover, the emergence of depth sensor [1] in recent years, which is able to extract 3D information from environment, brings the possibility of accomplishing such a challenging task.

However, hand detection and pose estimation are still difficult missions due to some technical challenges. For the detection part, it is necessary for us to find out the position of a user's hands from complex environment where we will encounter several problems. First, the human hand may have various shapes and scales due to different viewpoints, distances and subjects which result in high intra-class variation. Secondly, the result of detection (i.e., the bounding box) are supposed to be pretty reliable in order to avoid the failure of the following stage of pose estimation. On the other hand, pose estimation is also another challenging task. First, hand poses usually have serious self-occlusion problems between different fingers, which may make the information we retrieve incomplete. Moreover, the degree of freedom of human hands is quite high. Last, the input data from depth sensor may contain a great deal of noise that misleads the estimator and deteriorates. Fig. 1 shows some ideal samples and some examples which are seriously broken due to self-occlusion and sensor noise.

Despite these difficulties, the current state-of-the-art work has shown that the great performance is still likely to be achieved on this task. Nevertheless, to make this problem more generalized, we can't deny that there is still room for improvement. Most of the previous works only focus on the pose estimation part subject to some assumptions or relying on a weak detector to detect human hands. In [8][9], the authors assume that human hands will only appear as the nearest object from the camera so that it implies this kind of systems may easily fail on several situations containing multiple

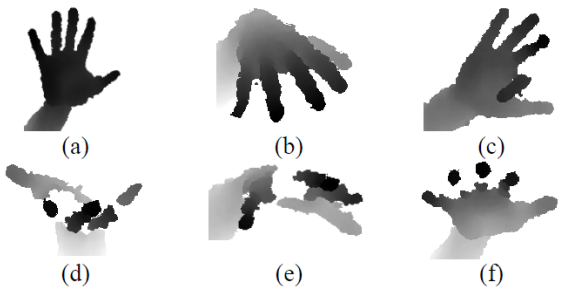

Fig. 1: Some samples of human hand. The first row are the ideal cases that human hands have complete contour and shape. On the contrary, the second row shows the broken samples which are commonly caused by self-occlusion or sensor noise.

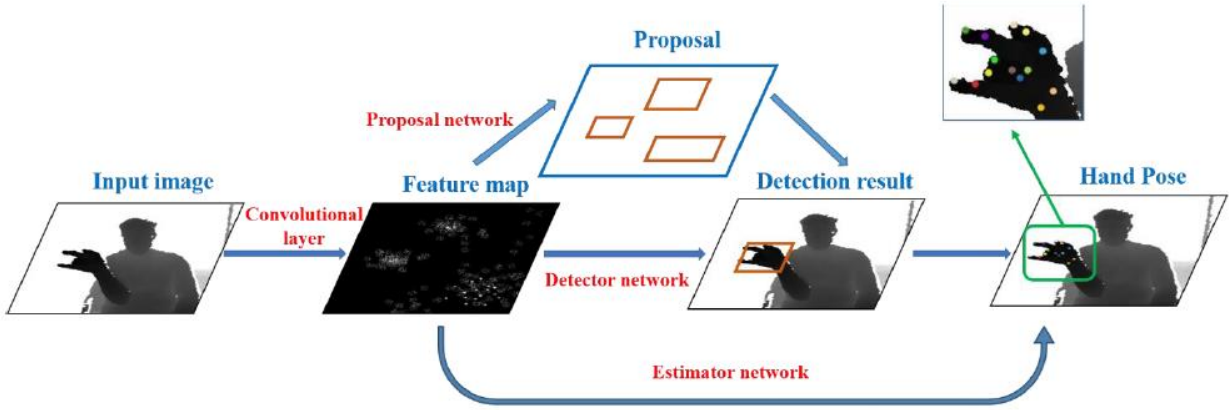

Fig. 2. Flowchart of our hand detection and pose estimation system. We have three networks following the shared convolutional layers, they are proposal network, detector network and estimator network, respectively. These three network use the feature map of the previous convolutional layers as input, and output results of proposal, detector and pose estimator one by one.

hands and users or other complex environments. To make this kind of system reliable in real-world applications, it is necessary to combine the pose estimation system with a robust and general hand detector.

In this paper, we propose a Convolutional neural network (CNN) based framework which combines both hand detection and pose estimation into one single pipeline. By letting the three neural networks share the convolutional layers, we can efficiently detect human hands and estimate their poses, and we only needs to compute the convolutional feature map once. Moreover, we propose an energy function to optimize the predicted result which can reduce the possible errors of CNN. The average precision of our detection system is almost 90% and the average error distance of pose estimation is 8.92 in pixels. Accordingly, it is confirmed that our proposed method is highly reliable and suitable for real-world applications.

The rest of content in this paper is organized as following: In section 2, we will discuss some works related to this research area. In section 3, we will explain the method of this work which includes the system overview, the network structure we employ and details of our implementation. In section 4 and 5, we will illustrate the result of our experiment and make a conclusion, respectively.

2. RELATED WORK

2.1. Deep Learning for Object Detection

It is a long history of developing a robust and reliable vision-based object detection system. Due to the advancement of computational power, the deep network has gained popularity among this research area. In [2], the authors propose a CNN framework with selective search proposal [3] which achieves powerful performance on [4]. Then, [5] designs a Spatial Pyramid Pooling (SPP) scheme which can generate fixed-length representation from CNN and greatly reduce the execution time by decreasing the number of computing feature map to 1. Later, fast R-CNN [6] develops a speed-up version of a training method which is much faster than the original one and allows us to train deeper as well as more complex neural networks. Recently, a boosted version of fast R-CNN is presented. The faster R-CNN [7] breaks the bottleneck of computation time with Region Proposal Network (RPN), which retrieve the region proposal in a much shorter time than it used to be, while it still retains the advantages of previous work. Since it shows a better performance and runs much more efficiently, our work is inspired by the framework, and we design a novel architecture that focuses on solving the hand pose estimation problem efficiently and effectively.

2.2. Hand Pose Estimation

Human hand pose estimation is a competitive area since it is an important component for a wide range of applications. Therefore, there are so many different methods to pursue this goal recently. In [10], they propose an innovative method which is able to successfully estimate human body pose from a single depth image at 200 frames per second. They trained a pixel-wise classifier from a highly varied dataset that the classifier is invariant to poses, shapes and human's clothing. Besides, the method can also be applied to hand pose estimation as well. Kondori *et al*. [12] present a 3D gesture-based interaction system which directly estimates 3D hand motion from an optical flow constraint equation. Qian *et al*. [11] model a hand using a number of spheres and compare the input depth image with hand model based on a defined cost function. Tang *et al*. [20] present a Semi-supervised Transductive Regression (STR) forest algorithm for real-time articulated hand pose estimation. Later, they continue to

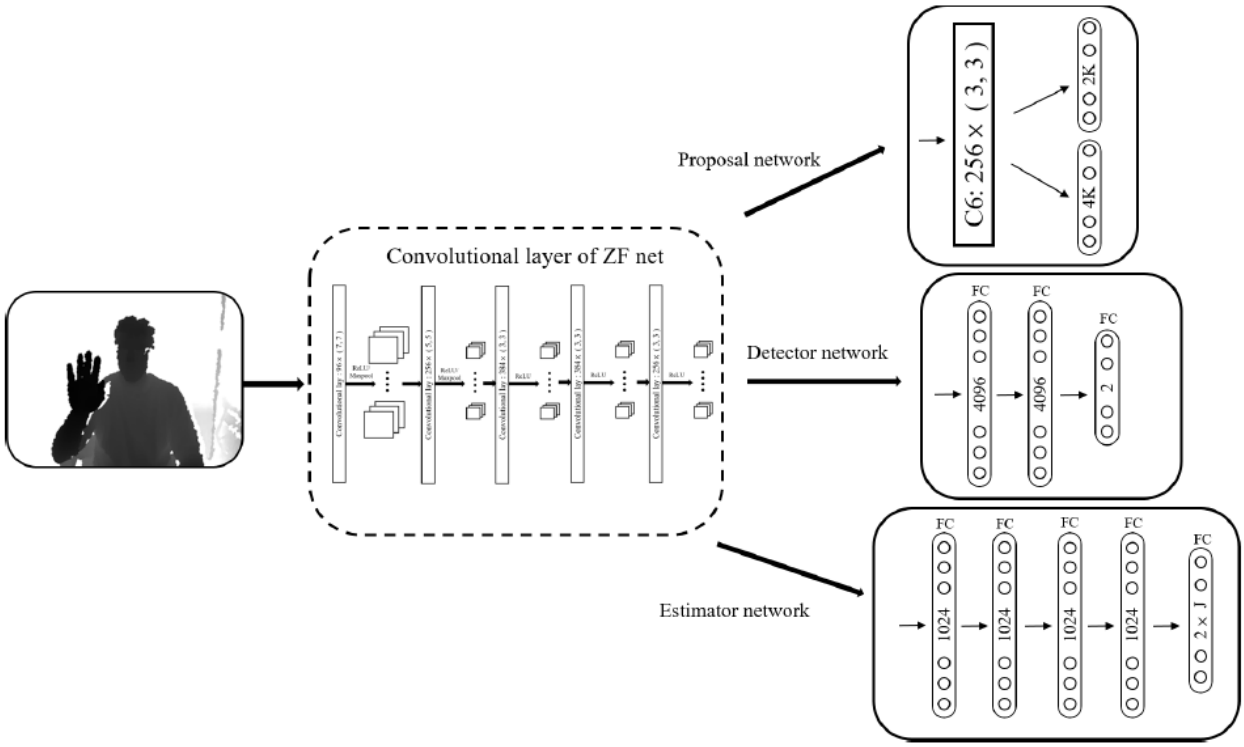

Fig. 3: The architecture of our deep networks. We use the convolutional layers of ZF net and connect it with different designed networks.

propose a Latent Regression Forest method [19] which can search for all the joint locations from a point cloud based on the hierarchical topology of hand. Tompson *et al*. [13] propose a pipeline for human hand detection and then pose estimation. They detect human hand by a randomized decision forest classifier and extract dense feature with a convolutional neural network from cropped hand images. Finally, they recover human hand pose by inverse kinematics. In [8], the authors compare different deep learning architectures for hand pose estimation and enforce a prior on 3D pose which results in the better performance for deep network. Afterwards, they improve the result of deep network by training a feedback loop [9]. In [14], the authors conduct an extensive survey and analysis of the state-of-the-art. They compare the result of different methods and implement a baseline by their own.

3. DETECTION AND ESTIMATION WITH DEEP LEARNING

In this section, we will explain the method we employ in this paper which we can integrate hand detection and pose estimation into one single pipeline. For each input depth image, we can accurately generate the position of a hand and its pose configuration.

3.1. System Overview

Our system is composed of three convolutional neural networks including proposal, detector and estimator network and one optimization stage. Fig. 2 shows the flowchart of our system. The proposal and detector networks work similarly as in [7]. At first, we will compute the convolutional feature map for the entire input image. Then, the proposal network will generate several candidates which have higher objectness scores. After that, the detector network will compute the probability of truly being a hand for each candidate by directly forwarding the sub feature map, which is generated from the feature map of the entire image, into the fully connected layers. Therefore, the candidates which have higher probability than our defined threshold will be considered as a hand. Afterwards, for each bounding box of hand, the estimator will also directly forward the sub-feature map into the fully connected layers and generate the hand pose result. Last, we will optimize the predicted locations of all joints to get the final result.

3.2. Network architecture and process

The network architecture is composed of one convolutional section and three fully connected components with respect to proposal, detector and estimator. Basically, we can regard it as an extension of ZF net [15], which has five layers of convolution, and is then followed by different fully connected layers for different purpose. The structure of our network can be seen in Fig. 3. After computing the feature map of the entire image, we will slide a window along the feature map and transfer each sliding window onto the proposal

network. The number of output values from the proposal network for each sliding window will be 2K plus 4K, where K denotes the number of "anchor" (We set K = 9 in our experiment). One can regard anchor as templates of bounding box with respect to different scales and aspect ratios. The 2K output values mean the probability of being foreground or background for K anchors individually and 4K means their spatial position (i.e. X, Y, width, height). More details about proposal network can be referred in [7]. Then, we will pick up 300 proposals with higher scores and forward their corresponding feature map into detector network by Region of Interest (ROI) pooling [6]. The ROI pooling is actually a max pooling layer which generates a smaller but fixed-size output from previous feature map according to the desired ROI, and the dimension can also be decreased in this layer so that the full processing time of several convolution layers can be reduced. The detector network contains three fully connected layers, of which the first and the second layer have 4096 neurons each. The last layer has only two neurons which determine whether the proposal is a hand or not. Finally, we will use the estimator network to generate the hand pose from ROI-pooled feature map of hands. This network contains four fully connected layers which have 1024 neurons in total. The last layer has $2 \times J$ neurons which means the spatial positions (i.e. (X,Y)) of J desired joints.

3.3. Implement detail

3.3.1. Data prepocessing

Before training, we need to preprocess the raw data from the dataset. Since the ground truth of hand pose in the dataset is the set of absolute positions in the image, we first transform them into the relative positions with respect to the center of hand in a way similar to that in [16]. The transformation is indicated below:

$$\begin{bmatrix} x_j' \\ y_j' \end{bmatrix} = \begin{bmatrix} \frac{1}{w} & 0 \\ 0 & \frac{1}{h} \end{bmatrix} \begin{bmatrix} x_j - x_c \\ y_j - y_c \end{bmatrix} \quad (1)$$

where x_j, y_j and x_j', y_j' are denoted as original position and transformed position of joint j. Furthermore, x_c, x_c is the absolute position of hand's center and w, h are the width and height of hand's bounding box. As a result, each element of position vector will be a value in the range of $[-0.5, 0.5]$.

3.3.2. Training

Since the parameters within the network will deterministically affect the result, we train our networks through several steps. At first, we train a proposal network based on the pre-trained model from ImageNet [17]. Secondly, we train a detector network, which is also initialized by aforementioned pre-trained model, based on the output of the previous proposal network. Thirdly, we copy the weights from the convolutional layers of the detector network to first initialize the estimator network and then train it until convergence. Finally, we train the proposal and detector networks again using the convolutional weights of estimator network for initialization but hold them as fixed values so that we only fine-tune the weights of the fully connected layers. As a result, the three different networks can share the same convolutional layers. In addition, we use Euclidean distance as cost function to train our estimator network. Therefore, the objective function can be derived as below:

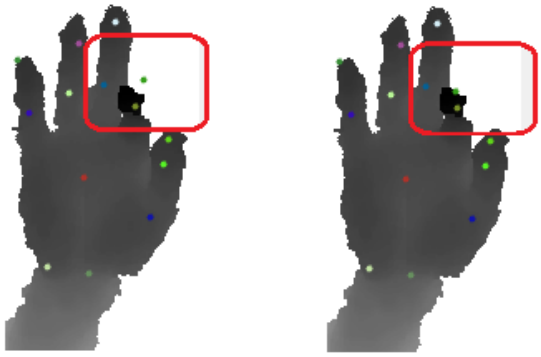

Fig. 4. The result of our energy-based optimization. We can correct the locations of joints which are located outside the hand.

$$\hat{\Phi} = argmin_{\Phi} \sum_t \|Estimator(I_t) - P_t\|_2 + \lambda \|\Phi\|_2^2 \quad (2)$$

where I_t and P_t are the image's and pose's ground truths of instance t in the training-set and $\hat{\Phi}$ is the optimized parameters of the neural network. We set the weight decay rate $\lambda = 0.001$ for regularization. Besides, our implementation is based on the framework of [7] and uses Caffe [18] as deep learning library.

3.4. Pose estimation

Although we can directly obtain a reasonably good result from the deeply learned network, we find that there are some minor errors which tends to be easily observed on the other hand. In fact, we can eliminate these errors by some post-processing methods. That is, the predicted joints may be located at background area whose depth values in this environment setting are far away from which of the hand. If we don't cope with the incorrect depth values, it may deteriorate the results quite easily. Therefore, we propose an energy function in order to optimize the location for each joint which is located at background:

$$\hat{\boldsymbol{p}}_{\boldsymbol{j}} = \boldsymbol{argmin}_{\boldsymbol{p_j'}} (\boldsymbol{\alpha} \cdot \mathbf{s}(\boldsymbol{p_j'}, \boldsymbol{p_j}) + \boldsymbol{\beta} \cdot \boldsymbol{d}(\boldsymbol{p_j'}, \boldsymbol{p}, \boldsymbol{I})) \quad (3)$$

where function s and d denote the shift loss and the depth loss, respectively, and α, β are their weights. Here, $p_j \in R2$ is the predicted result of joint j from the estimator network (*i.e.*, $p_j = (x_j, y_j)$) and $\hat{p}_j$ is the final optimized result.

Table 1: The Detection Result of NYU Hand Pose Dataset.

Recall	.1	.2	.3	.4	.5	.6	.7	.8	.9	1	AP	AUC
Precision	1	1	1	0.997	0.991	0.981	0.974	0.961	0.929	0.107	0.894	0.955

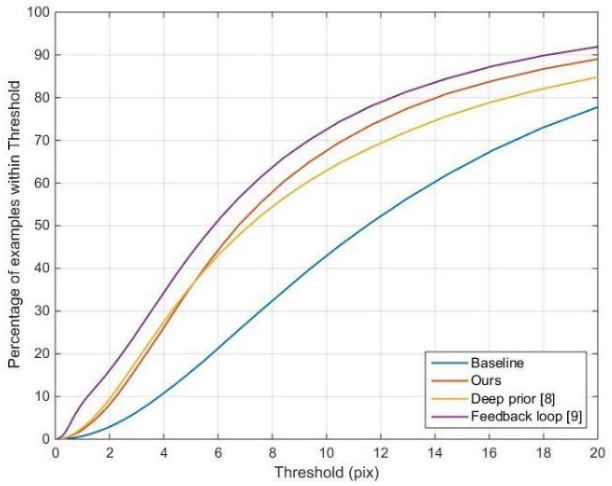

Fig. 5: The comparison between our system and other state-of-the-art methods. In the figure, the *X*-axis represents the threshold error distance and the *Y*-axis represents the fraction of samples that satisfies the threshold.

(a) The directly cropped image (b) The finely processed image

Fig. 6: The comparison of directly cropped image (a) and finely processed image (b). The right one eliminates the noise of background and enhance the contrast of depth value within the hand. On the other hand, the left one still has background, and it may become the noise of the input image into hand pose estimator.

In the shift loss term, we want to measure the penalty based on the distance between the predicted and optimized locations. It is modeled as a Gaussian distribution:

$$\mathbf{s}(\boldsymbol{p}'_j, \boldsymbol{p}_j) = -\frac{\boldsymbol{h_g}(x'_j - x_j, y'_j - y_j)}{\sum_{x\in\tau}\sum_{y\in\tau}(x - x_j, y - y_j)} \quad (4)$$

$$\boldsymbol{h_g}(\boldsymbol{x}, \mathbf{y}) = \boldsymbol{e}^{\frac{-(x^2+y^2)}{2\sigma^2}} \quad (5)$$

where τ is the search space of potential candidates, and

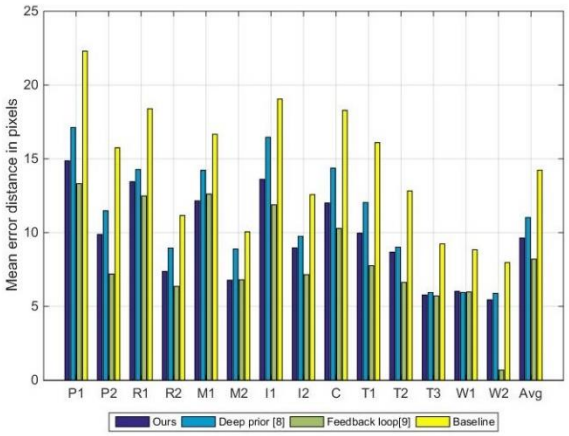

Fig. 7: The error distance of different joint types on a hand. The shorter the bar is, the better the performance is.

we use a bounding box whose size is 15 × 15 pixels around the predicted joint in our experiment.

In terms of depth loss, we assume most of the predicted results will exactly lie inside the hand. Therefore, we choose the median of depth value from all predicted joints as the standard. The depth loss function is derived below:

$$\mathbf{d}(\boldsymbol{p}'_j, \boldsymbol{p}_j, \boldsymbol{I}) = \left|\boldsymbol{I}(\boldsymbol{p}'_j) - \boldsymbol{median}(\boldsymbol{I}(\boldsymbol{p}))\right| \quad (6)$$

where I is the depth image, and p is the set of all predicted joints (*i.e.*, $p = \{p_1, p_2, \ldots p_J\}$). Fig. 4 illustrates one example of our optimization from which we can correct the locations of joints that are located outside the hand.

4. EXPERIMENT

In this section, we will give an overview of our experiment. We will introduce the dataset we used for evaluation and then show the performance of our system and make some comparisons with other existing methods.

4.1 NYU Hand Pose Dataset [13]:

This dataset is a public dataset, and it is commonly regarded as a pretty challenging dataset because it contains 8252 testing-set and 72757 training-set of RGB-D images captured by the Primesense Carmine 1.09. Each frame in this dataset is annotated with ground truth of each hand pose, and all the frames are highly various. Moreover, the training data includes the samples from a single user and the testing samples are from two different subjects. As done in [8] and [13], following their strategy, we use only the depth image, and the joint number *J* is 14 in our experiment.

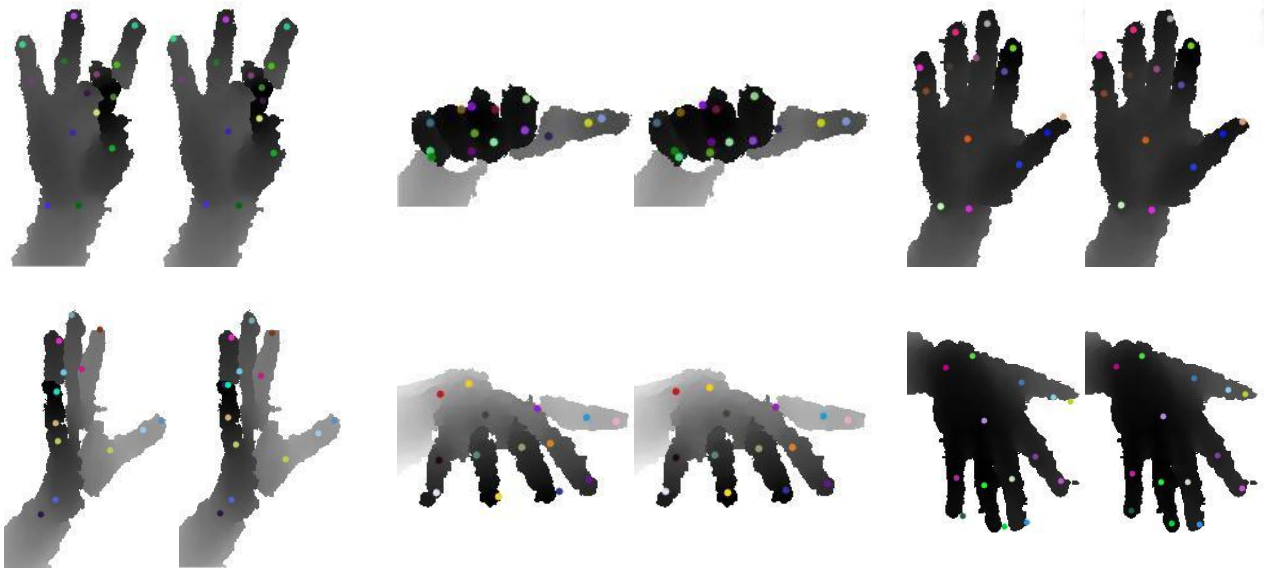

Fig. 8: Samples of our pose estimation. The left side within the pair is our predicted result and the right side is the corresponding ground truth. The experiments are operated on NYU hand pose dataset, which contains 14 joint labels of a hand pose.

4.2 Hand detection result:

We evaluate the performance of our detector on the testing-set of NYU hand pose dataset. Like most detection work, we show the precision under different recall and calculate the Average Precision (AP) as well as Area Under Curve (AUC). However, according to our survey, there is no other hand detection work reporting their results on this dataset and the detector in [13] is pixel based, so we can just directly show the content of our evaluation. Table 1 shows the final result. It indicates the robustness and reliability of our system that we achieve precision 0.929 under recall 0.9 and the average precision reaches almost 0.9.

4.3 Pose estimation result:

Adopting the same process as we do in detection part, we evaluate the performance of our estimator on the testing set of NYU hand pose dataset. We compare our estimator with different state-of-the-art works in deep learning, and we implement a simple CNN as the baseline. The baseline CNN has 4 convolutional layers with 32 kernels and 3 fully connected layers with 1024 neurons, but it is randomly initialized without cross fine-tuning as what we have done for our estimator network. We can see the result in Fig. 5. Similar to Feedback loop [9] and Deep prior [8], we use the evaluation criterion that measures the fraction of samples whose distance between all predicted joints and their ground truth is less than a certain threshold. This criterion is generally regarded as a great challenge since a single mistaken joint may deteriorate the judgement of the entire hand pose. Moreover, we also compare the error distance of different joints and calculate their average value in Fig. 7. The result indicates that we have a slight improvement over Deep prior [8]; however, it is not as good as Feedback loop [9] despite the fact that we even use a deeper network architecture. The possible reason is that the estimator's input images are very different in quality. For the efficiency and coherence of our system, we directly use the cropped feature map from the previous network (*i.e.* the detection network) as the input of estimator network. On the other hand, Feedback loop and Deep prior use the finely processed image, *i.e.*, they use a fixed 3D cube around the hand to precisely extract the hand and subtract the background. Fig. 6 illustrates two different processed images as an example. One can easily notice that there will be some body parts involved in our input which may cause noise to our estimator network. On the contrary, the finely processed image will only contain the information of hand. Though we can process our input image to avoid this problem as well, it will take much more time to eliminate the background pixel and re-compute the convolutional feature map again. As a result, it becomes a trade-off between the efficiency and the accuracy for this task in our experiment. However, in spite of this drawback, our system still achieves state-of-the-art which has average error distance of 8.92 pixels. Moreover, with the reliable detection and computational efficiency, our system shows the robustness and applicability. As you can see in Fig. 8, it illustrates some samples of our result in comparison with the state-of-the-art method.

D. Running time:

We measure the running time of our system on a computer equipped with Intel I5 CPU, 16GB of RAM and a GPU of NVIDA GeForce GTX 980. The operating system is Windows 10 and our system is

implemented in Matlab with Caffe [18]. The running time in which the system does not share convolutional layers between three networks is 329 milliseconds for one single frame. On the other hand, the running time of sharing convolutional weights is 74 milliseconds per frame, which is 4.5 times faster than the previous one. This result clearly shows the computational efficiency of our system.

5. CONCLUSION

In this paper, we propose a novel framework which integrates both human hand detection and pose estimation as one single pipeline based on deeply learned networks. Unlike most of the previous works which only focus on the pose estimation part subject to some strong assumptions or relying on a weak detector to detect human hands only, we design a deep learning architecture to complete these two tasks at the same time. By letting the convolutional layers be shared by the three different neural networks, this deeply learned architecture can efficiently and accurately detect human hands and compute their pose configuration. Moreover, we also propose a new energy function in order to optimize the predicted result of CNN. The experimental result is able to show that the not only the average precision of our detection system almost achieves 90%, but also the average error distance of pose estimation is 8.92 in pixels. Therefore, it is confirmed that our proposed method is highly reliable and suitable for real-world applications.

For the future work, we want to further improve the accuracy of our estimator network. As the problem we discussed in section 4, the background of hand may cause noise that deteriorate the performance of pose estimation. Consequently, it is necessary to enhance the quality of input data while keeping efficient computation as well.

REFERENCES

[1] Zhang, Zhengyou. "Microsoft kinect sensor and its effect." MultiMedia, IEEE 19.2 (2012): 4-10.

[2] Girshick, R., Donahue, J., Darrell, T., Malik, J. “Rich feature hierarchies for accurate object detection and semantic segmentation.” In Proceedings of the IEEE conference on computer vision and pattern recognition. 2014 (pp. 580-587).

[3] Uijlings, J. R., van de Sande, K. E., Gevers, T., Smeulders, A. W. “Selective search for object recognition.” International journal of computer vision, 104.2 (2013): 154-171.

[4] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., Zisserman, A. “The pascal visual object classes (voc) challenge.” International journal of computer vision, 88.2 (2010): 303-338.

[5] He, K., Zhang, X., Ren, S., Sun, J. “Spatial pyramid pooling in deep convolutional networks for visual recognition.” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9), 1904-1916.

[6] Girshick, R. “Fast r-cnn.” In Proceedings of the IEEE International Conference on Computer Vision. 2015 (pp. 1440-1448).

[7] Ren, S., He, K., Girshick, R., Sun, J. “Faster R-CNN: Towards real-time object detection with region proposal networks.” In Advances in Neural Information Processing Systems. 2015 (pp. 91-99).

[8] Oberweger, M., Wohlhart, P., Lepetit, V. “Hands deep in deep learning for hand pose estimation. In Proceedings of 20th Computer Vision Winter Workshop (CVWW) 2015, pp. 21-30.

[9] Oberweger, M., Wohlhart, P., Lepetit, V. “Training a Feedback Loop for Hand Pose Estimation.” In Proceedings of the IEEE International Conference on Computer Vision. 2015 (pp. 3316-3324).

[10] Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Moore, R. “Real-time human pose recognition in parts from single depth images.” Communications of the ACM, 56.1 (2013): 116-124.

[11] Qian, C., Sun, X., Wei, Y., Tang, X., Sun, J. “Realtime and robust hand tracking from depth.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014 (pp. 1106-1113).

[12] Kondori, F. A., Yousefit, S., Ostovar, A., Liu, L., & Li, H. “A Direct Method for 3D Hand Pose Recovery.” In 2014 22nd International Conference on Pattern Recognition (ICPR). (pp. 345-350).

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

[14] Supancic, J. S., Rogez, G., Yang, Y., Shotton, J., Ramanan, D. “Depthbased hand pose estimation: data, methods, and challenges.” InProceedings of the IEEE International Conference on Computer Vision. 2015 (pp. 1868-1876).

[15] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." Computer vision–ECCV 2014. Springer International Publishing. 818-833.

[16] Toshev, Alexander, and Christian Szegedy. "Deeppose: Human pose estimation via deep neural networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.

[17] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Berg, A. C. “Imagenet large scale visual recognition challenge.” International Journal of Computer Vision, 115.3 (2015): 211-252.

[18] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T. “Caffe:

Convolutional architecture for fast feature embedding.” In Proceedings of the ACM International Conference on Multimedia. 2014 (pp. 675-678).

[19] Danhang Tang, H.J. Chang, A. Tejani, T-K. Kim. “Latent Regression Forest: Structured Estimation of 3D Hand Posture” Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2014.

[20] Danhang Tang, Tsz-Ho Yu, and Tae-Kyun Kim. "Real-time articulated hand pose estimation using semi-supervised transductive regression forests."Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2013.