Mask-pose Cascaded CNN for 2D Hand Pose Estimation from Single Color Image

Yangang Wang, *Member IEEE*, Cong Peng and Yebin Liu

Abstract—We present a cascaded convolutional neural network for 2D hand pose estimation from single in-the-wild RGB images. Inspired by the commonly used silhouette information in the generative pose estimation approaches, we build the cascaded network with two stages, including mask prediction stage as well as pose estimation stage. We find that the two stages network architecture for end-to-end training could benefit with each other for detecting the hand mask and 2D pose. To further improve the hand pose detection accuracy, we contribute a new RGB hand dataset named **OneHand10K**, which contains 10K RGB images. Each image contains one single hand. We manually obtained the segmented mask and labeled keypoints for guided learning. We hope that this dataset will give a benchmark and encourage more people to perform research on this challenging topic. Experiments on the validation dataset have demonstrated the superior performance of the proposed cascaded convolutional neural network.

Index Terms—hand pose estimation, cascaded CNN, mask prediction

1 Introduction

Human hand pose estimation is vital for many computer vision applications, such as sign language, human-machine interaction, learning from demonstration, and *etc*. Solving the problem with only visual cues [1] (without using markers [2], [3]) is extremely hard because of the pose and appearance ambiguities, strong articulation, and heavy self-occlusion. This leads to the fact that, although the problem has been extensively investigated in the last decades, nearly all of available methods have bounded to depth sensors [4], [5], [6], [7], [8]. In recently, a few pioneer works [9], [10] consider using RGB image for hand pose estimation. However, these methods still suffer from unconventional lighting, pose ambiguities, similar hand and background colors, *etc*., and as shown in Fig.8. The problem is far from being well investigated.

In this paper, we investigate the problem of hand pose estimation from a single in-the-wild RGB image. Our key idea is to take advantage of the silhouette information of the hand. As has been observed and validated in available literature, silhouette is one of the most important cues for 3D pose estimation in human hand tracking [11], human body shape tracking and reconstruction [12]. However, hand segmentation itself is also a very challenging problem. Essentially, segmentation and pose estimation is inherently a chicken-and-egg problem.

To solve the above puzzle and improve the performance of hand pose detection from in-the-wild color images, we propose a cascaded mask-pose Convolutional Neural Network(CNN) architecture to jointly learn the hand segmentation and pose estimation. Our network consists of two sequential stages, including a mask prediction stage and a pose detection stage. The hand mask in the first stage as well as its corresponding feature maps make contributions to the hand pose detection. With the back-propagation training strategy in deep learning based techniques, the followed up 2D hand pose detection would give a soft constraint for the earlier supervised mask prediction. Thus, the pose estimation can improve the accuracy of hand segmentation. Finally, the sequential two tasks can be beneficial to each other and the performance of pose estimation is substantially improved.

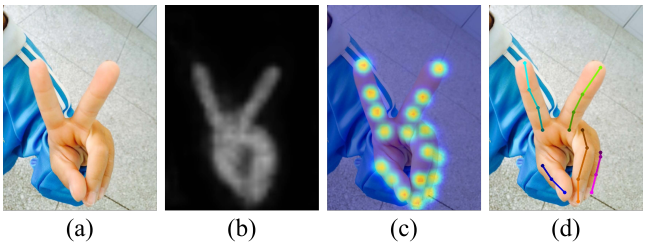


Fig. 1. Given a single color image, our cascaded convolutional neural network can detect the 2D hand pose. We explicitly encode the hand mask into an end-to-end network training for 2D pose estimation. The hand mask can dramatically improve the accuracy of joint heatmap detection. On the other hand, the 2D hand pose estimation can also improve the hand mask prediction.

Furthermore, to fully take advantage of our cascaded deep leaning network, we contribute a new hand dataset ***OneHand10K***. Existing RGB hand datasets are either generated synthetically [10], or captured by depth sensors [13], or captured with multiview in the dome setting [9], or in the lab environment [14]. All of these exhibit a certain level of appearance differences from in-the-wild RGB images, and most importantly, are not well annotated. Our ***OneHand10K***

- *Yangang Wang is now with the School of Automation, Southeast University, Nanjing, China, 210096. Before that, he was an associate researcher at Microsoft Research. E-mail: ygwangthu@gmail.com. Personal website: http://www.yangangwang.com. Corresponding author: Yangang Wang.*
- *Cong Peng is with the School of Automation, Nanjing University of Aeronautics and Astronautics, Nanjing, China, 211106. E-mail: pengcong.2006@163.com.*
- *Yebin Liu is with the Department of Automation, Tsinghua University, Beijing, China, 100083. E-mail: liuyebin@mail.tsinghua.edu.cn.*

Manuscript received XXX.

dataset contains more than 10 thousand RGB single hand images and each one of them is carefully annotated with 21 joints. As stated above, the silhouette information is crucial, we also manually segment the mask of the hand for each image. The dataset will be public in the future.

With our proposed cascaded mask-pose CNN and well annotated hand dataset ***OneHand10K***, we show remarkable performance improvement over the two state-of-the-art hand pose detection methods [9], [10], both qualitatively and quantitatively, as shown in Fig. 8 and Fig. 9. Although our method only produces 2D pose but not 3D, the recent parallel research on 3D human body pose detection [15], [16] has demonstrated that, the accuracy of 2D estimation is very crucial to the performance of 3D detection. And more importantly, the 3D pose estimation problem can be greatly beneficial and directly lift from the 2D detection results by a 2D-to-3D regression algorithms [10]. We believe that 3D hand pose estimation can be improved from our proposed dataset and method.

The main contribution of this work include:

- We build a new hand dataset, named ***OneHand10K***, to facilitate the research of hand gesture estimation. Our dataset contains 11703 original color images with one hand per image, labeled hand joints for each image and manual segmented hand mask images. For each type of hand gesture, occlusion, light, shadow and background are all considered in the dataset. We hope that the novel dataset would promote the development of distinctive hand gesture estimation from images.
- We propose a novel mask-pose cascaded convolutional neural network for simultaneous mask and 2D pose estimation from single color images. Different from previous neural network, we explicitly encode the hand mask information into the network architecture. The explicitly encoded hand mask can give soft constraints for the 2D hand keypoint detection. Such encoded hand mask can also make the convolutional neural network be more accurate.
- We demonstrate that mask and pose prediction in an end-to-end pose estimation framework could benefit with each other. The proposed network includes two stages: the mask prediction stage and pose prediction stage. The hand mask in the mask prediction stage as well as the corresponding feature maps to produce the hand mask both contribute to the generation of joint heatmaps. With the back-propagation of the end-to-end training, hand pose prediction can also improve the accuracy of hand mask prediction.

As a reminder, the data and source code of our work are made public on the project website [1].

2 Related Work

Hand pose estimation is originally considered by the applications of human-computer interaction (HCI) [17] with RGB data. Various approaches are proposed to solve the problem of hand pose estimation in decades, which have been reviewed in the survey [1]. Typically, the approaches of hand pose estimation can be split into three categories, including generative approaches [18], [19], [20], discriminative approaches [21], [22], [23] and hybrid approaches [20], [24], [25], [26], [27], [28]. The generative approaches use 3D hand models to estimate the hand pose by maximizing the consistency between the projected hand model and visual cues extracted from the images. Silhouettes, shading, skin color and optical flow are the commonly used visual features [24], but these methods are almost restricted in controlled environments.

Discriminative approaches, especially methods that are benefit from deep learning, have a much closer step to the practical applications. In general, discriminative approaches learn a mapping from image features to hand poses from a training set. They do not require an explicit hand model, but more importantly, are free to the restrictions of settings. Although discriminative methods have a more wider application scope, their accuracy and the type of poses they can handle are largely depend on the training data. Recently, depth data has been easier to obtained with the introduction of commodity depth sensors. Single-view depth-based hand pose estimation thus became the major focus of research [4], [5], [6], [7], [8]. Meanwhile, it is relatively easy to synthesize for depthmaps via computer graphics techniques, which also promotes a large number of depth-based hand pose estimation approaches. Nevertheless, obtaining the hand dataset for in-the-wild color images has significant challenges.

In order to solve the lack of in-the-wild hand color images, [9] proposed a multiview bootstrapping method to train the hand keypoint detectors for single RGB image. Their approach allows the generation of large annotated datasets using a weak initial detector. However, the augmented dataset is limited by the multiview situation, which can not cover the majority of hand images in real life including the variations of color, background and *etc*.In this paper, we try to improve the performance of hand detection for in-the-wild RGB images. Inspired by the utilized features in the generative methods [12], we integrate the silhouette information into the convolutional neural network. Our 2D hand pose estimation network architecture is based on [29], which is also used in [9]. They cast the 2D pose detection as estimating the confidence map or score map of human skeleton joint positions. The most likely location is selected as the maximum confidence of the corresponding position in the confidence maps. Differently, we address that the detection accuracy would be dramatically improved by explicitly explaining the silhouette information in the network structure. Our network can simultaneously output the hand segmentation mask and 2D pose in a unified framework.

There are two recent works that also use the silhouette information to obtain the 2D hand pose. The work in [10] detect the 2D hand keypoints and then lift to estimate the 3D hand pose. They use three separate networks to perform the task, including hand segmentation network, 2D pose estimation network and 3D pose prior network. In their approach, the hand mask is obtained from the single RGB image by a hand segmentation network at the first step . The 2D hand keypoints are localized by a 2D pose estimation network with the hand mask and cropped single

1. http://www.yangangwang.com/papers/wang-mcc-2018-10

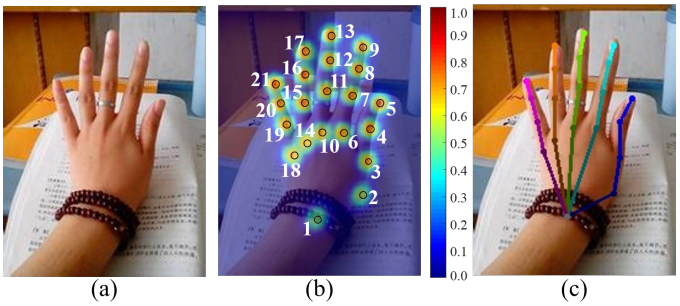


Fig. 2. Illustration of the 2D hand keypoints (better viewed in color). (a) is the input RGB image, (b) visualizes the detected heatmaps and the index of keypoints, (c) is the connected 2D hand pose.

RGB image as input. Different from their several individual steps to obtain the hand pose, we train the hand keypoints detectors end-to-end. We find that it is more accurate to simultaneously train the network for segmentation mask and pose estimation in a uniformed framework.

It is noted that Mask R-CNN [30] also can estimate the segmentation mask and pose at the same time. In their framework, the pose is treated as soft segmentation masks. Thus, the network can output the joint soft segmentation masks as well as the hand segmentation silhouette in parallel. In contrast, we sequentially estimate the segmentation mask and 2D pose information. Our basic idea is originally that the silhouette information serves as the important feature for estimating the hand pose in the framework of generative approaches. We demonstrate that the cascaded network architecture can both improve the detection accuracy of hand segmentation silhouette as well as 2D hand pose.

Recently, there are some works perform the 3D hand pose estimation from monocular RGB images [10], [31], [32]. Based on the state-of-the-art deep learning based Convolutional neural networks, these methods can estimate the 3D hand poses in an end-to-end manner. Our work is focused on 2D hand pose estimation. The proposed mask-pose cascaded neural network, especially the mask prediction sub-network, can be integrated into existing 3D hand pose estimation neural network, which may promote the 3D hand pose estimation in the future.

3 METHOD

Given a color image $\mathbf{I} \in \mathbb{R}^{w \times h \times 3}$, we want to detect the 2D hand pose by convolutional neural network (CNN). We represent the 2D hand pose as K keypoint locations $\mathbf{x}_k \in \mathbb{R}^2$ with $k \in [1, K]$ (as illustrated in Fig. 2). Each keypoint k corresponds to a landmark of the 2D hand pose. In our case, we use $K = 21$ landmarks for describing the 2D hand pose.

Beyond the 2D hand pose, we can also implicitly detect the hand mask $\mathbf{M} \in \mathbb{R}^{w \times h \times 1}$ as we have addressed that silhouette is very important for hand detection. In such case, our objective is to train the hand keypoint detectors and hand segmentation mask from $\mathbf{I}^t$ and $(\mathbf{M}^t, \{\mathbf{x}_k^t\})$, where t denotes the t-th RGB color image, hand mask and keypoint annotations from the training examples.

3.1 Data Representation

The size of input images in our proposed network is $m \times m \times 3$ and we need to transform the original image $\mathbf{I}$. Specifically, we first resize the color image $\mathbf{I}$ with a ratio τ, which is denoted as

$$\tau = \min(\frac{m}{w}, \frac{m}{h}). \quad (1)$$

The resized color image fills the output $m \times m \times 3$ image from the left-top corner and the other empty region is filled with the gray value $(128, 128, 128)$. Similar procedures are done for the hand mask $\mathbf{M}$, but filled with zeros.

The 2D hand keypoint locations $\{\mathbf{x}_k\}$ are represented by heatmaps [33], which is denoted as $\mathbf{H} \in \mathbb{R}^{m \times m \times (K+1)}$. Each heatmap encodes the keypoint location via a 2D Gaussian distribution, where the mean is $\tau \cdot \mathbf{x}_k$ and the variance $\boldsymbol{\sigma}_k \in \mathbb{R}^2$ controls the influence range. The additional one heatmap represents the background information, which is computed by subtracting the summary of K heatmaps with 1. It is noted that if the 2D keypoint is missing (*e.g.*, the keypoint is out of range or occluded), the corresponding heatmap is set as zero. The benefits of regressing a heatmap rather than 2D coordinates are discussed in [33]. In short, the keypoint heatmap can be regarded as the confidence map. It is easier to aggregate the final keypoint location $\mathbf{x}_k$ from mutlimodes, such as the pyramid multiple spatial locations.

3.2 Network Architecture

We use the convolutional neural network to predict the the 2D hand joint locations. The proposed network structure can be divided into two stages: mask prediction stage and pose prediction stage as illustrated in Fig. 3.

Hand mask encodes the spatial layout of hand in the input color image. Such spatial structure can be naturally addressed by the pixel-to-pixel correspondence provided by convolutions. Similar as previous semantic segmentation work [30], [34], we follow the fully convolutional networks (FCN) [35] to predict the hand mask. Specifically, we use the convolutional stages of the VGG-19 network [36] up to $conv4\_4$ to produce 128-channel feature $\mathbf{F}$. Then the feature $\mathbf{F}$ is convoluted to produce 2-channel hand mask. For an input image with the size $m \times m$, the resulting size of the mask prediction stage is $m' \times m'$, with $m' = \frac{m}{8}$. It is because that there are 3 pooling operators in the proposed network, which are marked with red color in Fig. 3.

The mask prediction stage is followed by a pose prediction stage, which generates $K + 1$ joint heatmaps. Similar to [9], we also perform the pose prediction stage with several sequential sub-stages, denoted as $(S^1, \cdots, S^T)$. We found that 5 sequential prediction stages are balanced for the efficiency of training and testing. In each sub-stage S^t, we predict the 2D human poses with the information from the mask prediction stage and the predicted human poses from the previous sub-stage S^{t-1}. The first sub-stage S^1 only uses the information from previous mask prediction stage. Specifically, we generate the input for pose prediction stage by concatenating the 128-channel feature $\mathbf{F}$ and the 2-channel hand mask $\mathbf{M}$, which are finalized into new 130-channel maps, denoted as $\mathbf{F}^+$. The new maps $\mathbf{F}^+$ serve as the input of the pose prediction stage. Each sub-stage S^t can produce $K + 1$ human pose 2D heatmaps, which takes

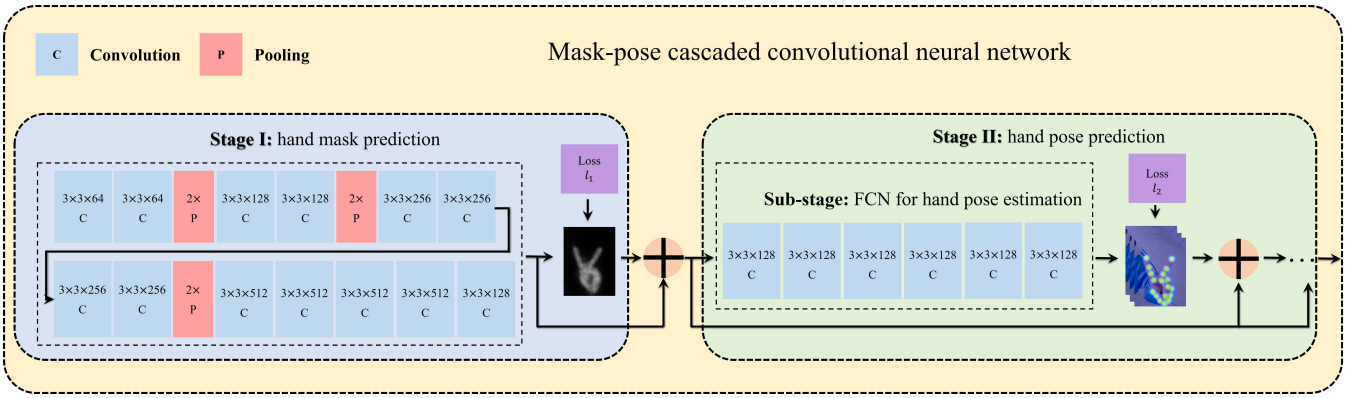


Fig. 3. Network architecture. The proposed network structure includes two stages: mask prediction stage and pose prediction stage.

the $\mathbf{F}^+$ as well as $K+1$ heatmaps from the previous sub-stage S^{t-1} as input. In total, the input for each human pose prediction sub-stage S^t has a channel of $130+(K+1)$. It is noted that the first sub-stage S^1 only use the 130-channel feature map $\mathbf{F}^+$ as input. Since there are no additional pooling or down-sampling operators in the pose prediction stage, the final stride of our convolutional neural network is similar as the mask prediction stage, i.e., 8. It also means that we need to resize the heatmaps $\mathbf{H}$ with a ratio of $\frac{1}{8}$ to evaluate the training loss in our network architecture.

The produced heatmaps in the final sub-stage S^T are treated as the output of the proposed network. We then resize the heatmaps into the original size $m \times m$. The predicted heatmaps are up-sampled into the original size using bicubic resampling, with the ratio of 8. The 2D hand keypoint location can be extracted as the pixel with the maximum confidence in its respective map. As for the loss function of our network structure, we use softmax loss for hand mask prediction and L_2 loss for the hand pose prediction. Support the number of training examples in a batch is N, the cross-entropy loss L_m for mask prediction is computed as

$$L_m = -\frac{1}{N}\sum \log(\frac{e^{s_y}}{\sum_j e^{s_j}}), \tag{2}$$

where y is the ground-truth label. s_j is the output score for the j-th label in our mask prediction stage and $j \in \{0, 1\}$.

As for the L_2 loss for the hand pose prediction L_p, we compute it as

$$L_p = \frac{1}{N}\sum ||f - g||, \tag{3}$$

where g is the ground-truth joint heatmaps, f is the output heatmap of the human pose prediction stage in the proposed network. Note that the loss for all the T sub-stages are computed independently.

Discussion: We sequentially predict the hand mask and 2D pose in an end-to-end manner. Although the similar network structure is utilized compared against the network in [9], there are two main differences. The first one is that we explicitly encode the hand mask information into the proposed network architecture. Our intuition to encode the hand mask into the convolutional neural network comes from the commonly used object silhouette in the approaches of generative methods [12]. They claim that the silhouette or mask is a very strong feature for pose estimation, which is more robust compared against other visual features such as color, lighting or shadows. The explicitly encoded hand mask can give soft constraints for the 2D hand keypoint detection. Such encoded hand mask can also make the convolutional neural network [9], [29] be more accurate. The second one is that our network can be interpreted into two stages, *i.e.*, the mask prediction stage and pose prediction stage. Different from the feature extractors with original convolutional neural network, we use the fully convolutional network to predict the hand mask. The hand mask in the first mask prediction stage as well as the corresponding feature maps to produce the hand mask both contribute to the generation of joint heatmaps. With the back-propagation of the end-to-end training, hand pose prediction can also improve the accuracy of hand mask prediction. The experiments have confirmed what we think. Please refer to Sec. 4.2 for more details.

3.3 Implementation Details

We implemented our experiments by modifying the network given in [9] and using Caffe [37] for CNN computation. We describe the most important implementation details in this section. Source code for the complete system is available, thus providing documentation of the remaining implementation details.

Dataset: We focus on the 2D hand pose estimation from a single RGB color image. Public datasets for hand pose estimation can be divided into two groups and pioneer works have been investigated on these two groups of datasets: One group of datasets focuses on the hand pose estimation from depth data, *e.g.*, [5], [38], [39]. The other one majors in the hand action recognition from color images, *e.g.*, [40], [41], [42], [43]. Nevertheless, these datasets are not sufficient for training a deep network. It is not only because that the variation of hand gestures is limited, but also the environment of performing such actions is synthesized or restricted in the lab settings. Besides, the 2D hand pose are always have incomplete annotations.

In order to train the 2D hand pose predictors from in-the-wild color images, we try to build a new 2D hand pose estimation dataset, which is named ***OneHand10K***. The hand images in the newly built dataset are required to

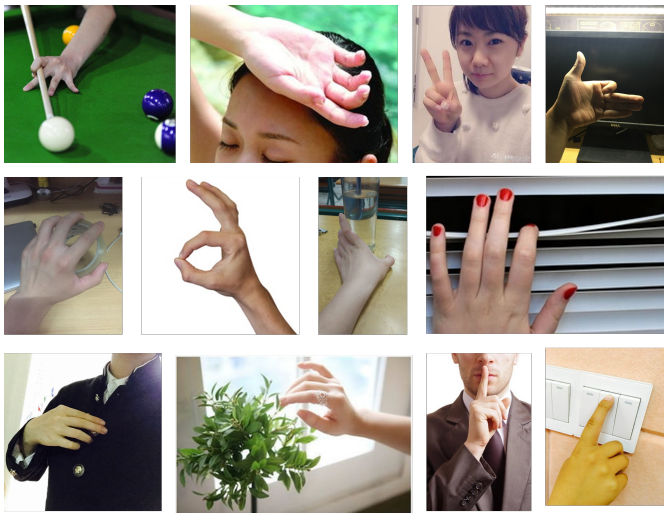


Fig. 4. Selected images from the dataset ***OneHand10K***. Our dataset contains various hand gestures in the wild environment including hand object interactions, cluttered background, severe self-occlusions and variant illuminations.

have cluttered background and cover as many as possible types of hand poses. We collected the images from public online sources, where only one single hand exists per image. Even though, the gestures of the collected hand images are still limited in a small range. We then asked a majority of volunteers to perform the lacked typical hand gestures and capture the hand images under several different environments, such as lighting changes, different background and *etc*.. In total, the ***OneHand10K*** includes 11703 hand images. Some of the selected hand images are shown in Fig. 4. The dataset will be available online in the future.

After that, we annotate the 21 hand keypoints for the collected hand images. Each keypoint $\mathbf{x}_k$ is annotated by its image coordinate. The order of the keypoints is indicated in Fig. 2. It is noted that not all keypoints are visible in the collected hand color images. Such keypoints are annotated to be the locations $(-1, -1)$, which produce zero maps in the heatmap generation as described in Sec. 3.1. We also perform the hand mask segmentation for all the hand images. The hand mask is segmented for all visible hand regions of the images, including the palm and fingers. There are some hand images contain the wrist pixels, which is hard to determinee whether it belongs to hand or not. For those pixels, we enroll all of them into our hand segmentation mask.

The main concern about the annotation is the quality. In order to control the annotation quality, we apply the cross-validation method. We first divide the whole dataset into several packages. Each package includes about 100 images. All the packages are randomly distributed into different users. We then ask the users to randomly select 5 annotated yet not completed packages (excluding their own annotated ones) for checking the annotation results. Here, the package is marked as completed when it is validated by more than 3 persons. The users can correct the annotation results by themselves. It is noted that existing tool, *e.g.*, LabelMe [44] can also be used for our task.

Training: We use the method described in Sec. 3.1 to reshape all the input frames to the size of 368×368. The corresponding hand mask of the input frame is simultaneously reshaped into the same size of 368×368. We also rescale the joint positions with the scaling ratio τ, described in Eqn. (1). After that, we generate the joint heatmaps with the scaled joint positions. All the reshaped frames, masks and joint heatmaps are fed into the proposed convolutional neural network.

To guarantee the performance of the training and validation, all the images in the dataset are shuffled. We selected 10000 images for training and the rest 1703 images are used for validation. The validation images are fixed for further evaluation and all the training images are provided in the project website. Beyond the image selection, we perform the data augmentation for each hand image as follows: At first, each image is resized with a random scale factor, which is linearly interpolated between 0.7 and 1.2. The interpolation ratio is drawn from an uniform distribution range at $[0, 1]$. We then randomly rotate the resized image in the 2D plane with the angle between $-40°$ and $40°$. The same strategy as resizing the image is utilized for generating the random rotation angle. After that, the output image is randomly cropped into the original size, *i.e.*, 368×368. Since the images for the left and right hands are not distinguished in the proposed dataset ***OneHand10K***, the augmented image is finally horizontally flipped with the probability of 0.5. The empty value in the procedure of data augmentation is all set as $(128, 128, 128)$. We perform the data augmentation along with the whole training procedures. We find that this step is critical for successful training.

All of our method are trained by Caffe [37] on a server with a double GPUs of Nvidia GTX1080Ti. We set the variance of Gaussian for producing the heatmaps $\mathbf{H}$ in Sec. 3.1 as $\boldsymbol{\sigma} = [1.5, 1.5]^T$. The mini-batch is set to 13. Momentum is set to 0.9. The initial learning rate is set to 5×10^{-5}, and decreased per $50K$ iterations with the ratio of 0.3, and stopped at $200K$ iterations. The total training loss are the weighted sum of the mask prediction stage and 5 sub-stages of the pose prediction stage. Since the evaluation loss for hand mask and joint heatmaps has large quantity differences, we assess such differences and set the weight for mask and heatmaps to be 0.05 and 1.0, respectively. In our current implementation, we train the neural network weights from random initialization. The training loss reduces from about the original 6K to the final 0.1K.

Inference: At the test time, we build a image pyramid for the input frame. For generating the different input for the image pyramid, we randomly crop the image pyramid with the maximum pixels of 0.1 times image width. Each cropped image pyramid is then reshaped into the same size $m \times m$ and fed into the convolutional neural network. The number of pyramid is determined by the image size of the smallest one, which equals 50 pixels in our current implementation. We then aggregate all the heatmaps of the image pyramid with maximum-winning strategy. It means the final confidence for each joint is computed as the maximum value of the whole pyramid image network output. We set a threshold for the joint confidence. If the joint confidence is larger than the threshold, the detected joint position is regarded as the corresponding image coordinate

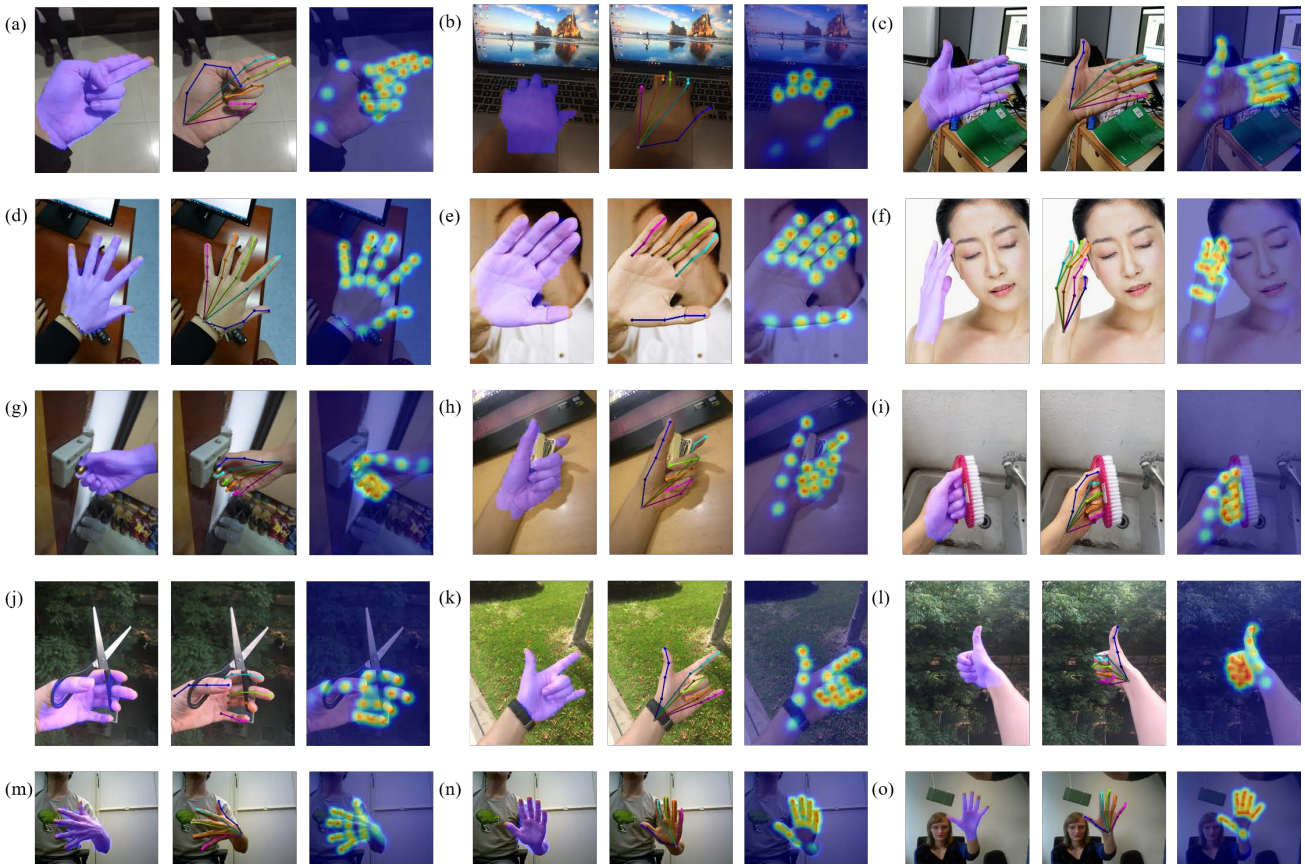


Fig. 5. Typical results. We show some typical results including the examples with different hand gestures, cluttered background, variant illumination, occlusions and hand object interactions. Note that (m) and (n) are the test images from [10], (o) is the test image from [14]. Our method correctly detects the 2D hand poses from in-the-wild single color images. From left to right, we show the hand mask (indicated with the color of blue), the detected 2D hand pose and the joint heatmaps for each example, respectively.

of the maximum confidence. Such threshold equals 0.2 in our current implementation.

Timing: We report the training and inference time of our neural network in the following paragraphs.

- **Training**: Our network is easy to train. Training with the proposed cascaded mask-pose covolutional neural network on our dataset takes 60 hours in our synchronized 2-GPU implementation (0.49s per 13-image mini-batch). It can be accelerated via enlarging the number of mini-batches or with a pre-trained network weights by reducing the training iterations.
- **Inference**: Our network runs at about 30ms per image with the same input image size on an Nvidia GTX1080Ti GPU. The cost time includes the CPU time for resizing the images to feed into our neural network, as well as the joint determination from the network output heatmaps.

It is noted that our design is not optimized for speed, and better speed/accuracy trade-offs could be achieved, *e.g.*, by varying image sizes, which is beyond the scope of this paper.

4 Experiments

We conducted several experiments to test the performance of the proposed method. The components of the overall approach, including the network architecture of mask prediction stage and the pose prediction stage, are both evaluated on our validation dataset. The experiments have demonstrated the superior performance of our proposed cascaded mask pose CNN for 2D hand pose estimation.

4.1 Main Results

We evaluate our method on the validation dataset, which contains 1703 in-the-wild RGB images with a single hand per image. Different types of hand gestures, as well as the hand object interactions performed under cluttered background exist in the validation dataset. The typical results of detected hand mask (covered in the original image), detected joint heatmaps and 2D hand poses are shown in Fig. 5. For all the examples, we show the results with hand mask, 2D hand pose as well as the detected joint heatmaps covered in the original color image. Specifically, (a) and (b) show the example of hand gestures under bad lighting. (c) and (d) show the estimated hand pose in the indoor environment with cluttered background. (e) and (f) demonstrate that the proposed method can also estimate the 2D hand pose when the hand is near the face region, which is hard to distinguish with only color feature. (g) and (h) show the cases that our method can detect the hand when there exist occlusions and similar color variations in the background . (g), (h), (i) and (j) are the examples from real-life hand object interactions, where the hands are partially occluded

Fig. 6. Selected image frames from the real-case desktop environment. Our proposed method can detect the hand in real-time.

by objects. (k), (l), (i) and (j) show the hand poses under the outdoor environment, which has different shadow and lighting compared with the lab settings. (m) and (n) are the test images from [10]. (o) is the test image from [14]. Our method can all work for these cases. It demonstrates the strength of our technique that detecting the hands to in-the-wild RGB images. We also developed a web camera system for desktop hand detection applications. In our current implementation, we did not perform the keypoint temporal smoothing. Our 2D hand pose estimation system can run at 30fps on a single GTX1080Ti. Higher frame rate can be obtained by performing the detection per 2-4 frames. Fig. 6 shows the real-case desktop result. Our proposed method can detect the hand in real-time.

4.2 Evaluations

Our covolutional neural network is composed by two stages, including mask prediction stage and 2D hand pose estimation stage. The effects of these two components are both evaluated. We also compare our method with the state-of-the-art model described in [9]. We will report the evaluation results in the following paragraphs.

Hand mask prediction: To evaluate the mask prediction performance of our network, we perform the experiments that we dropped out the stage II of our cascaded convolutional neural network architecture, *i.e.*, the FCN for pose prediction, to train the hand mask prediction. Some of the selected results are visualized in Fig. 7. From this figure, we can see that our cascaded convolutional neural network can obtain better hand mask predictions. We also use *Averaged Precision*, *i.e.*, AP, to perform the quantitative evaluations. The AP is averaged over the mask IoU thresholds. We report the AP values with AP_{50}, AP_{75} and AP_{80}, which indicates the threshold as $0.5, 0.75$ and 0.8, respectively). The quantitative AP values are shown in Tab. 1. The evaluation results demonstrate that pose prediction stage can improve the performance of mask prediction in the same network structure.

Hand pose prediction: We also evaluate the pose prediction result with/without predicting the mask in the first stage

	AP_{50}	AP_{75}	AP_{80}
Without pose prediction stage	67.6	33.3	21.9
Proposed method	**76.9**	**41.3**	**30.0**

TABLE 1
Hand mask prediction with / without the convolutional neural network of pose prediction stage, *i.e.*, stage II in our network architecture. The AP is computed on the whole validation dataset with different IoU thresholds.

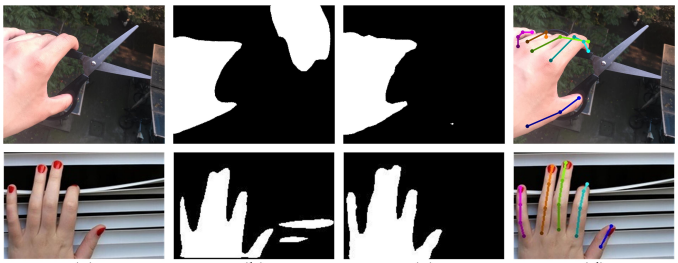


Fig. 7. Evaluation on mask prediction. (a) is the original image, (b) is the predicted mask without pose prediction stage, (c) is the predicted mask of the proposed method and (d) is the predicted 2D hand pose.

of the proposed network. It is noted that the network structure of ***without the mask prediction stage*** is similar to Convolutional Pose Machines used in [9]. Specifically, ***without the mask prediction stage*** means that we drop out the last two layers in the mask prediction stage of our neural network and do not compute the loss l_1 in Fig. 3 to train the whole network. We use the *Probability of Correct Keypoint*, *i.e.*, PCK [45] to quantitatively compare the result. The PCK defines the probability that a predicted keypoint is within a distance threshold σ of its true location. For a particular keypoint p, we denote it by PCK_σ^p and approximate it on a validation dataset $\mathbb{T}$ as

$$\text{PCK}_\sigma^p = \frac{1}{\mathbb{T}} \sum_{\mathbb{T}} \delta(||\mathbf{x}_p - \mathbf{y}_p|| < \sigma), \tag{4}$$

where $\mathbf{x}_p$ is the predicted position of the p-th keypoint, $\mathbf{y}_p$ is

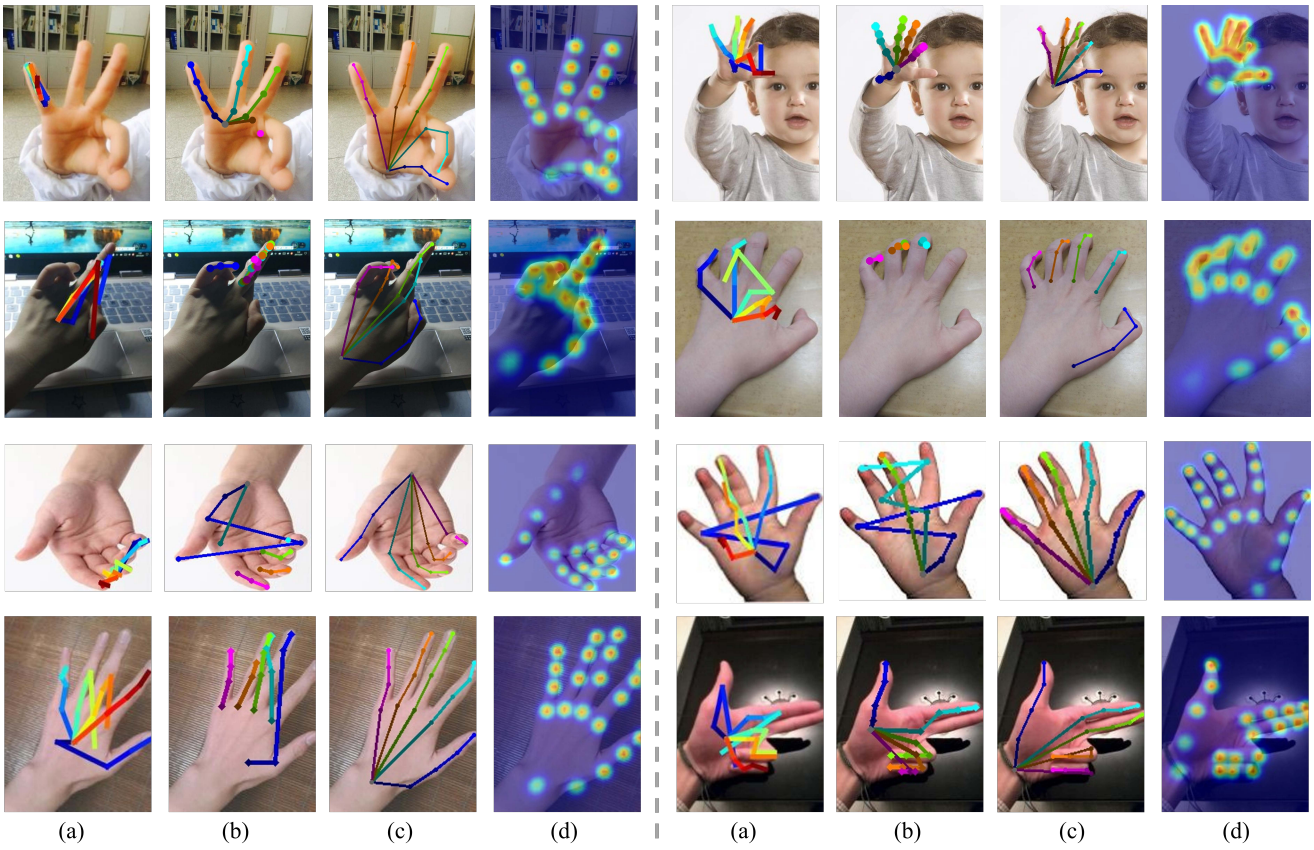


Fig. 8. Comparison results. (a) is the result with the code given by [10], (b) is the estimated hand pose with the model in [9], (c) is the detected hand pose by our method and (d) is the corresponding joint heatmaps.

its groundtruth location, $\delta(\cdot)$ is the indicator function. The PCK curves are plotted by varying the accuracy threshold σ in Eqn. (4), which is shown on the horizontal axis in the Fig. 9. We measure σ as a normalized distance, where pixel distances in each example are normalized by the bounding box size of groundtruth hand keypoints. Fig. 9 shows the evaluation result. We selected a small range of images in the validation dataset, which is named as dataset *EA*. The whole validation dataset is named as *EB*.

From the comparison result, we find that mask prediction stage can improve the accuracy of pose prediction. The reason is that the explicitly encoded hand mask can give soft constraints for the 2D hand keypoint detection. It is noted that the proposed approach is marginally similar or a bit worse than the alternative methods without the mask in Fig. 9. Actually, we find that $\text{PCK}_{0.2}$ is the visible differences in the detected region. The marginal errors in Fig. 9 may come from the annotation variations among different images.

Comparison to state-of-the-art: We compare the results by using the trained model provided by [9] on our validation dataset. We also compare the results with the method in [10] by running their given code. Since the input image of their code is 320×240. For fair comparison, we cropped our test images with the aspect ratio of $4:3$ and resize the images to the size of 320×240. Some of the selected comparison results are shown in Fig. 8.

We then conducted the quantitative comparisons, which are shown in Fig. 9 (a), (b1) and (b2). The dashed lines and solid lines show the evaluation results for the dataset *EA* and *EB*, respectively. It is noted that the proposed mask-pose cascaded neural network is similar to [9] except the mask prediction part. We have trained their network on our training data via removing the mask prediction of the proposed network, where the results are shown as the blue line in Fig. 9. The visualized PCK curves of [9] for all keypoints are under the curve of the proposed network architecture both on dataset *EA* and *EB*, which again demonstrates the superior performance of our proposed cascaded mask pose CNN for 2D hand pose estimation.

Furthermore, we compared our method with the model given by [9] on their dataset [43] and Fig.10 shows the comparison result. The dashed lines visualize the results from the model in [9] and the solid lines are the results of our method. From the curves, we could find that our method is comparable in **MPI+NZSL** and has superior performance in the dataset **Syn2** and **Syn3**. All the comparison code and our trained model for the curves reproduction will be public in the future.

Specifically, compared with the model in [9], our result has a little gap in the dataset **MPI+NZSL**, as shown in Fig. 10 (a) and (b1). Two main issues may cause the detection performance degradation. The first one is that several testing images in **MPI+NZSL** have two hands. However, in this paper, we only consider one hand pose estimation from single color image and the proposed model is trained on the ***OneHand10K*** with only one hand per image. This

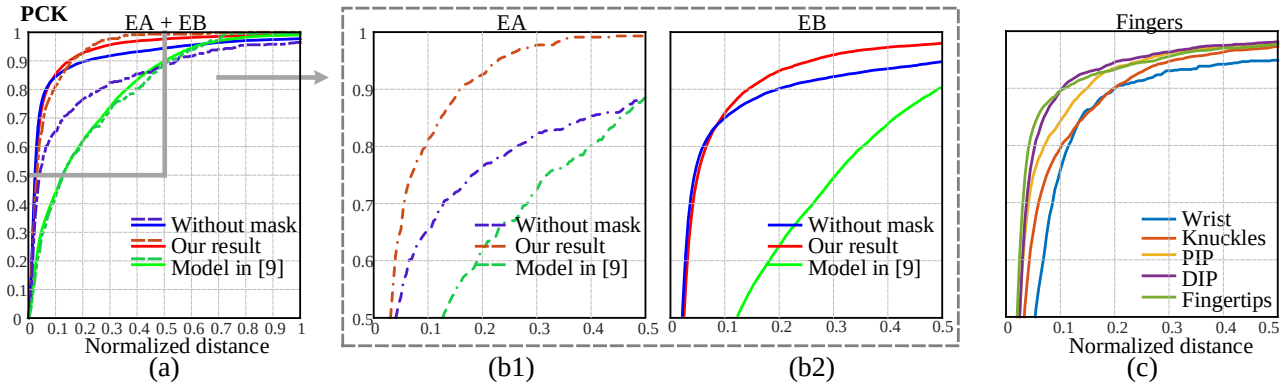


Fig. 9. Evaluation on pose prediction. We evaluate the proposed method on a small range of our validation dataset, named *EA*. The whole validation dataset is named *EB*. The plotted PCK curves demonstrate the superior performance of the proposed method. See texts for more details.

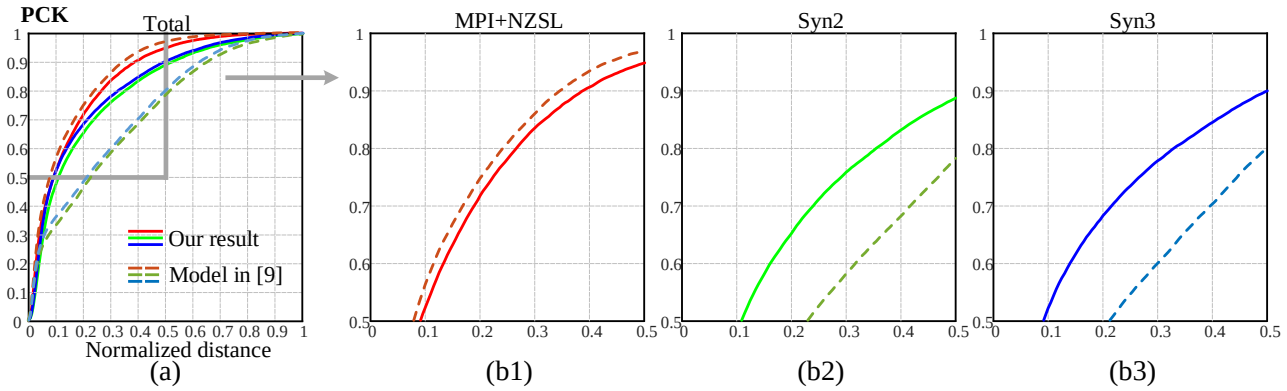


Fig. 10. Evaluation on 3 types of dataset given by [43]. **MPI+NZSL** is the result using the 846 images in the given folder '*manual_test*'. **Syn2** is the result using the 3243 images in the given folder '*synth2*' and **Syn3** is the result using the 2347 images in the given folder '*synth3*'. Our method has superior performance in the latter two dataset and comparable performance in **MPI+NZSL**. More details are addressed in the text.

is also the reason we did not perform the comparison for the two hand images in the given folder '*synth1*' [43]. The second one is that the groundtruth 2D hand joints positions in **MPI+NZSL** do not consider the visibilities. In this work, we train the model with only visible 2D hand joints, thus the 2D hand pose prior is not trained such as multiview bootstrapping [9].

Fig. 10 (a), (b2) and (b3) shows the comparison result in the synthesized dataset **Syn2** and **Syn3**. Our model has consistent performance for different types of dataset and has superior performance compared with the model in [9]. There is one additional interesting finding when performing the hand detection with their given model. That is, the detected hand pose configuration is reasonable, but the fingers may have totally wrong displacements as shown in Fig. 11. In the figure, different color indicates different fingers: blue indicates the thumb, cyan indicates the index finger, green indicates the middle finger, brown indicates the ring finger and purple indicates the pinky. It is mainly because that the multiview bootstrapping model in [9] learns the 2D hand pose prior while our model estimates the hand pose from visible image features. Due to the proposed network and dataset, our model can infer correct visible hand pose configuration.

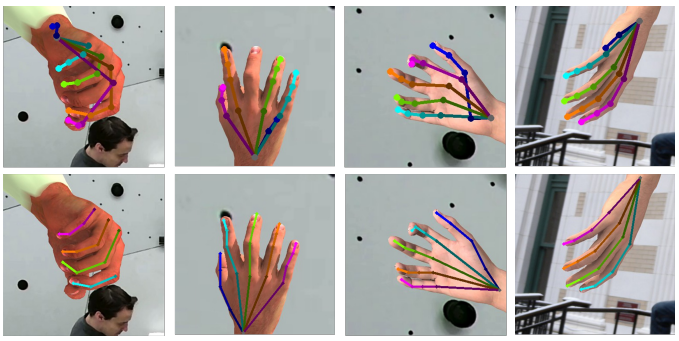


Fig. 11. Selected results. The first row shows the detection results with the model in [9] and the second row shows the results with our model. The detected hand in the first row has reasonable pose configuration but totally wrong finger displacement. See text for more explanations.

4.3 Discussion and Future Work

Our work has a few limitations. One main drawback is that we do not consider the cases of double hands, especially the close-interactions such as hand shaking. Our method may fail with such scenarios. Besides, we perform the mask and pose estimation in only one sequential manner, it may have a better solution to configure the mask and pose estimation, such as several mask and pose prediction components are cascaded for performance boosting. Furthermore, explicitly hand the mis-alignment between the hand mask and joint with some specially layers/operations in the network may also have performance improvement.

5 Conclusion

We have presented a new cascaded convolutional neural network architecture for 2D hand pose estimation from single in-the-wild RGB images. Our network has two stages, including a mask prediction stage and a pose estimation stage. We find that the two stages of mask and pose prediction could benefit with each other in an end-to-end network training. Target to the practical applicability of 2D hand pose estimation, we build a new dataset named ***OneHand10K*** with the single RGB hand images, hand silhouettes and labeled 21 keypoints for each image. Our dataset contains various hand gestures in the wild environment including hand object interactions, cluttered background, severe self-occlusions and variant illuminations. We hope this dataset will remove a major hurdle in this area and encourage more people to perform research on this challenging topic.

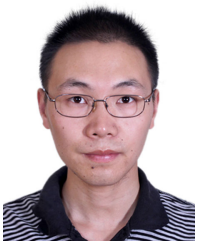
Acknowledgments

The authors would like to thank Biyao Shao, Junjie Zhu and Yining Xie to help us to build the dataset. This work was supported by the National Natural Science Foundation of China (No. 61806054, 61703203), Jiangsu Province Science Foundation for Youths (No. BK20180355 and BK20170812) and Foundation of Southeast University (No. 3208008410 and 1108007121).

References

[1] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *CVIU*, vol. 108, no. 1, pp. 52–73, 2007.
[2] R. Y. Wang and J. Popović, "Real-time hand-tracking with a color glove," *TOG*, vol. 28, no. 3, p. 63, 2009.
[3] A. Wetzler, R. Slossberg, and R. Kimmel, "Rule of thumb: Deep derotation for improved fingertip detection," in *BMVC*, 2015.
[4] D. Tang, H. Jin Chang, A. Tejani, and T.-K. Kim, "Latent regression forest: Structured estimation of 3d articulated hand posture," in *CVPR*, 2014.
[5] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *TOG*, vol. 33, no. 5, p. 169, 2014.
[6] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun, "Cascaded hand pose regression," in *CVPR*, 2015.
[7] C. Wan, A. Yao, and L. Van Gool, "Hand pose estimation from local surface normals," in *ECCV*, 2016.
[8] 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 *CVPR*, 2017.
[9] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *CVPR*, 2017.
[10] C. Zimmermann and T. Brox, "Learning to estimate 3d hand pose from single rgb images," in *ICCV*, 2017.
[11] Y. Wang, J. Min, J. Zhang, Y. Liu, F. Xu, Q. Dai, and J. Chai, "Video-based hand manipulation capture through composite motion control," *TOG*, vol. 32, no. 4, p. 43, 2013.
[12] Y. Wang, Y. Liu, X. Tong, Q. Dai, and P. Tan, "Outdoor markerless motion capture with sparse handheld video cameras," *TVCG*, vol. 24, no. 5, pp. 1856–1866, 2018.
[13] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T.-K. Kim, "Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis," in *CVPR*, 2017.
[14] S. Sridhar, F. Mueller, M. Zollhöfer, D. Casas, A. Oulasvirta, and C. Theobalt, "Real-time joint tracking of a hand manipulating an object from rgb-d input," in *ECCV*, 2016.
[15] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *ICCV*, 2017.
[16] B. Tekin, P. Marquez Neila, M. Salzmann, and P. Fua, "Learning to fuse 2d and 3d image cues for monocular body pose estimation," in *ICCV*, 2017.
[17] J. M. Rehg and T. Kanade, "Digiteyes: Vision-based hand tracking for human-computer interaction," in *Motion of Non-Rigid and Articulated Objects*, 1994.
[18] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Tracking the articulated motion of two strongly interacting hands," in *CVPR*, 2012.
[19] M. de La Gorce, D. J. Fleet, and N. Paragios, "Model-based 3d hand pose estimation from monocular video," *TPMAI*, vol. 33, no. 9, pp. 1793–1805, 2011.
[20] S. Sridhar, A. Oulasvirta, and C. Theobalt, "Interactive markerless articulated hand motion tracking using rgb and depth data," in *ICCV*, 2013.
[21] J. Romero, H. Kjellström, and D. Kragic, "Hands in action: real-time 3d reconstruction of hands in interaction with objects," in *ICRA*, 2010.
[22] M. Salzmann and R. Urtasun, "Combining discriminative and generative methods for 3d deformable surface and articulated pose reconstruction," in *CVPR*, 2010.
[23] C. Xu and L. Cheng, "Efficient hand pose estimation from a single depth image," in *ICCV*, 2013.
[24] L. Ballan, A. Taneja, J. Gall, L. V. Gool, and M. Pollefeys, "Motion capture of hands in action using discriminative salient points," in *ECCV*, 2012.
[25] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei *et al.*, "Accurate, robust, and flexible real-time hand tracking," in *CHI*, 2015.
[26] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt, "Fast and robust hand tracking using detection-guided optimization," in *CVPR*, 2015.
[27] D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall, "Capturing hands in action using discriminative salient points and physics simulation," *IJCV*, vol. 118, no. 2, pp. 172–193, 2016.
[28] Q. Ye, S. Yuan, and T.-K. Kim, "Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation," in *ECCV*, 2016.
[29] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *CVPR*, 2016.
[30] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.
[31] E. Dibra, S. Melchior, A. Balkis, T. Wolf, C. Oztireli, and M. Gross, "Monocular rgb hand pose inference from unsupervised refinable nets," in *CVPR Workshops*, 2018.
[32] P. Panteleris, I. Oikonomidis, and A. Argyros, "Using a single rgb frame for real time 3d hand pose estimation in the wild," in *WACV*, 2018.
[33] T. Pfister, J. Charles, and A. Zisserman, "Flowing convnets for human pose estimation in videos," in *ICCV*, 2015.
[34] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *ECCV*, 2016.
[35] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.
[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.
[37] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM MM*, 2014.
[38] J. Wan, Y. Zhao, S. Zhou, I. Guyon, S. Escalera, and S. Z. Li, "Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition," in *CVPR Workshops*, 2016.
[39] C. Chen, R. Jafari, and N. Kehtarnavaz, "Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," in *ICIP*, 2015.
[40] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," in *CRCV-TR-12-01*, 2012.
[41] I. M. Bullock, T. Feix, and A. M. Dollar, "The yale human grasping dataset: Grasp, object, and task data in household and machine shop environments," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 251–255, 2015.
[42] A. Borji, S. Izadi, and L. Itti, "ilab-20m: A large-scale controlled object dataset to investigate deep learning," in *CVPR*, 2016.
[43] "CMU Panoptic Studio hand dataset." http://domedb.perception.cs.cmu.edu/handdb.html, 2018, [accessed 19-Feb-2018].

[44] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *IJCV*, vol. 77, no. 1-3, pp. 157–173, 2008.
[45] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *PAMI*, vol. 35, no. 12, pp. 2878–2890, 2013.



Yangang Wang received his B.E. degree from Southeast University, Nanjing, China, in 2009 and his Ph.D. degree in control theory and technology from Tsinghua University, Beijing, China, in 2014. He was an associate researcher at Microsoft Research Asia from 2014 to 2017. He is currently an associate professor at Southeast University. His research interests include image processing, computer vision, computer graphics, motion capture and animation.

Cong Peng received her B.S. degree from Southeast University, Nanjing, China, in 2010, and the Ph.D. degree in electrical engineering from Beihang University, Beijing, China, in 2016. She is currently an Associate Professor with the College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China. Her research interests include the active vibration control, the vibration measurement and the computer vision.

Yebin Liu received the B.E. degree from the Beijing University of Posts and Telecommunications, China, in 2002, and the PhD degree from the Automation Department, Tsinghua University, Beijing, China, in 2009. He was a research fellow in the Computer Graphics Group of the Max Planck Institute for Informatik, Germany, in 2010. He is currently an associate professor at Tsinghua University. His research areas include computer vision, computer graphics and computational photography.